The World of Fourier and Wavelets: Theory, Algorithms and Applications

Martin Vetterli
École Polytechnique Fédérale de Lausanne and University of California, Berkeley

Jelena Kovačević
Carnegie Mellon University

Vivek K Goyal
Massachusetts Institute of Technology

April 15, 2008

Copyright (c) 2008 Martin Vetterli, Jelena Kovačević, and Vivek K Goyal. These materials are protected by copyright. They have been made available for your personal use. You may not otherwise copy, reproduce, publish, post, transmit, display, store, sublicense, transfer, distribute, and/or make derivative works without the express written permission of the authors.
The photograph captures an experiment first described by Isaac Newton in “Opticks” in 1730. Newton indicates how white light can be split into its color components and then resynthesized. It is a physical implementation of a decomposition into Fourier components, followed by a synthesis to recover the original, where the components are the colors of the rainbow. This experiment graphically summarizes the major theme of the book—many signals or functions can be split into essential components, from which the original can be recovered.
## Contents

Image Attribution xi  
Abbreviations xiii  
Quick Reference xv  
Preface xix  
Preface to Release 3.0 xxi  
Acknowledgments xxiii  
Reading Guide xxv  
From Rainbows to Spectra 1  

### I Tools of the Trade 7

1 From Euclid to Hilbert 9  
1.1 Introduction ........................................ 10  
1.2 Vector Spaces ...................................... 17  
  1.2.1 Definition and Properties .................... 17  
  1.2.2 Inner Products and Norms .................... 20  
  1.2.3 Some Standard Normed Vector Spaces .......... 23  
1.3 Hilbert Spaces ...................................... 27  
  1.3.1 Completeness and Separability* ............... 27  
  1.3.2 Orthogonality ................................ 29  
  1.3.3 Linear Operators ............................... 30  
  1.3.4 Projection Operators ......................... 34  
1.4 Bases and Frames .................................. 38  
  1.4.1 Orthonormal Bases ............................ 38  
  1.4.2 General Bases ................................. 42  
  1.4.3 Frames ......................................... 43  
  1.4.4 Matrix View of Bases and Frames ............ 44
Contents

2.7.2 Convolution .......................... 147
2.A Elements of Complex Analysis ........... 148
2.B Elements of Algebra: Discrete Polynomials 150
Chapter at a Glance .......................... 151
Historical Remarks .......................... 153
Further Reading ............................ 153
Exercises with Solutions ...................... 154
Exercises .................................. 157

3 Continuous-Time Signals and Systems 159
  3.1 Introduction ........................... 160
  3.2 Continuous-Time Fourier Transform—CTFT ....................... 160
     3.2.1 Properties of the CTFT ........................... 160
  3.3 Continuous-Time Fourier Series—CTFS .......................... 163
     3.3.1 Properties of the CTFS ........................... 164
  3.4 Shannon Sampling .......................... 165
     3.4.1 Sampling Theorem ........................... 166
Chapter at a Glance .......................... 170
Historical Remarks .......................... 170
Further Reading ............................ 170
Exercises with Solutions ...................... 170
Exercises .................................. 170

4 Sampling, Interpolation, and Approximation 173
  4.1 Introduction ........................... 174
  4.2 Functions on a Closed, Bounded Interval ....................... 175
     4.2.1 Approximation by Polynomials ....................... 175
     4.2.2 Approximation by Trigonometric Polynomials .......... 181
  4.3 Functions on the Real Line ....................... 181
     4.3.1 Basic Idea of Sampling ........................... 181
     4.3.2 Best Approximation in Shift-Invariant Spaces .......... 181
     4.3.3 Interpolation ........................... 181
  4.4 Functions on the Real Line Revisited: Splines ................... 181
  4.5 Approximations with Finite Precision ....................... 181
  4.6 Approximating Random Processes ....................... 181
  4.7 Algorithms ............................. 181
Chapter at a Glance .......................... 181
Historical Remarks .......................... 181
Further Reading ............................ 181
Exercises with Solutions ...................... 181
Exercises .................................. 182

II Fourier and Wavelet Representations 183

5 Time, Frequency, Scale and Resolution 185
## Contents

5.1 Introduction ......................................................... 186
5.2 Time and Frequency Localization .............................. 187
5.3 Heisenberg Boxes and the Uncertainty Principle .......... 189
5.4 Scale and Scaling .................................................. 194
5.5 Resolution, Bandwidth and Degrees of Freedom ........... 196
5.6 An Efficient Orthonormal Basis with Time-Frequency Structure 197
5.7 Case Studies on Real Signals ..................................... 201
5.7.1 Music and Time-Frequency Analysis ...................... 201
5.7.2 Images and Pyramids .......................................... 201
5.7.3 Singularities, Denoising and Superresolution .......... 201
5.7.4 Channel Equalization and OFDM .......................... 201

Chapter at a Glance ......................................................... 202

6 Filter Banks: Building Blocks of Time-Frequency Expansions 209
6.1 Introduction ......................................................... 210
6.2 Theory of Orthogonal Two-Channel Filter Banks ........... 215
   6.2.1 A Single Channel and Its Properties .................... 215
   6.2.2 Complementary Channels and Their Properties ....... 217
   6.2.3 Orthogonal Two-Channel Filter Bank .................. 218
   6.2.4 Polyphase View of Orthogonal Filter Banks .......... 221
   6.2.5 Polynomial Approximation by Filter Banks .......... 224
6.3 Design of Orthogonal Two-Channel Filter Banks ........... 225
   6.3.1 Lowpass Approximation Design ......................... 226
   6.3.2 Polynomial Approximation Design ...................... 227
   6.3.3 Lattice Factorization Design ............................ 229
6.4 Theory of Biorthogonal Two-Channel Filter Banks ........ 231
   6.4.1 A Single Channel and Its Properties .................... 234
   6.4.2 Complementary Channels and Their Properties ....... 236
   6.4.3 Biorthogonal Two-Channel Filter Bank ................ 236
   6.4.4 Polyphase View of Biorthogonal Filter Banks ....... 238
   6.4.5 Linear-Phase Two-Channel Filter Banks ............. 239
6.5 Design of Biorthogonal Two-Channel Filter Banks .......... 241
   6.5.1 Factorization Design ..................................... 241
   6.5.2 Complementary Filter Design ............................ 243
   6.5.3 Lifting Design ............................................. 244
6.6 Theory of Stochastic Filter Banks .............................. 245
6.7 Algorithms .......................................................... 245

Chapter at a Glance ......................................................... 248

Historical Remarks ..................................................... 252
Further Reading .......................................................... 252
Exercises with Solutions ............................................... 253
Exercises ................................................................. 255
7 Wavelet Series on Sequences 257
  7.1 Introduction ........................................... 259
  7.2 Iterated Filter Banks and Equivalent Filters .......... 259
    7.2.1 The Iterated Lowpass Filter ...................... 260
    7.2.2 Bandpass Equivalent Filters ..................... 264
  7.3 The DWT as an Orthonormal Expansion .................. 268
  7.4 Properties of the DWT .................................. 269
  7.5 Discrete Wavelets as a Basis for $\ell^2(\mathbb{Z})$ ... 275
  7.6 Variations on the DWT .................................. 278
    7.6.1 Biorthogonal DWT ................................ 278
    7.6.2 Discrete Wavelet Packets ......................... 278
  7.7 Processing Stochastic Signals ........................... 281
  7.8 Algorithms ............................................. 281
Chapter at a Glance ......................................... 282
Historical Remarks ......................................... 282
Further Reading ............................................. 282
Exercises with Solutions .................................... 282
Exercises ..................................................... 282

8 Wavelet Series on Functions 283
  8.1 Introduction ............................................. 283
  8.2 Wavelet Bases from Filter Banks ....................... 286
    8.2.1 Limits of Iterated Filters ....................... 287
    8.2.2 The Scaling Function and its Properties ......... 294
    8.2.3 The Wavelet and its Properties ................... 300
    8.2.4 Orthonormal Bases of Wavelets ................... 303
  8.3 The Wavelet Series and its Properties .................. 306
    8.3.1 Time-frequency Localization ....................... 306
    8.3.2 Zero Moments and Decay of Wavelet Coefficients ... 307
    8.3.3 Characterization of Singularities ................. 308
  8.4 Multiresolution Analysis ................................ 310
    8.4.1 Introduction ....................................... 310
    8.4.2 The Haar Case Revisited ........................... 310
    8.4.3 Axioms of Multiresolution Analysis ............... 315
    8.4.4 The Sinc and Meyer Wavelets ..................... 322
  8.5 Algorithms .............................................. 325
    8.5.1 Mallat’s Algorithm ................................ 325
    8.5.2 Initialization ...................................... 328
    8.5.3 The Synthesis Problem ............................. 329
  8.A Proof of Proposition 8.2 ................................ 332
  8.B Biorthogonal Wavelets from Filter Banks .............. 333
Chapter at a Glance ......................................... 338
Historical Remarks ......................................... 338
Further Reading ............................................. 338
Exercises with Solutions .................................... 338
Exercises ..................................................... 339
<table>
<thead>
<tr>
<th>Chapter</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>9 Localized Fourier Series on Sequences and Functions</strong></td>
</tr>
<tr>
<td>9.1 Introduction</td>
</tr>
<tr>
<td>9.2 N-channel filter banks</td>
</tr>
<tr>
<td>9.3 Fourier-based N-channel filter banks</td>
</tr>
<tr>
<td>9.4 Cosine Modulated N-channel Filter Banks</td>
</tr>
<tr>
<td>9.4.1 Uniform modulation</td>
</tr>
<tr>
<td>9.4.2 Modulated lapped orthogonal transforms</td>
</tr>
<tr>
<td>9.4.3 Block Matrix Notation</td>
</tr>
<tr>
<td>9.4.4 Windowing</td>
</tr>
<tr>
<td>9.5 Examples</td>
</tr>
<tr>
<td>9.5.1 Spectrograms</td>
</tr>
<tr>
<td>9.5.2 Filter banks for audio coding</td>
</tr>
<tr>
<td>9.5.3 Orthogonal frequency division multiplexing</td>
</tr>
<tr>
<td>9.6 What have we accomplished in this chapter?</td>
</tr>
</tbody>
</table>

**Chapter at a Glance**

- Historical Remarks | 364
- Further Reading | 364
- Exercises with Solutions | 364
- Exercises | 365

<table>
<thead>
<tr>
<th>Chapter</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>10 Frames on Sequences</strong></td>
</tr>
<tr>
<td>10.1 Introduction</td>
</tr>
<tr>
<td>10.2 Frame Definitions and Properties</td>
</tr>
<tr>
<td>10.2.1 Frame Operators</td>
</tr>
<tr>
<td>10.2.2 Useful Frame Facts</td>
</tr>
<tr>
<td>10.2.3 Dual Frame Operators</td>
</tr>
<tr>
<td>10.3 Finite-Dimensional Frames</td>
</tr>
<tr>
<td>10.3.1 Harmonic Tight Frames</td>
</tr>
<tr>
<td>10.3.2 Other Finite-Dimensional Frame Families</td>
</tr>
<tr>
<td>10.4 Infinite-Dimensional Frames Via Filter Banks</td>
</tr>
<tr>
<td>10.4.1 Gabor-Like Frames</td>
</tr>
<tr>
<td>10.4.2 Oversampled Discrete Wavelet Transform</td>
</tr>
<tr>
<td>10.4.3 Other Infinite-Dimensional Frame Families</td>
</tr>
<tr>
<td>10.5 Algorithms</td>
</tr>
<tr>
<td>10.5.1 The Algorithme à Trous</td>
</tr>
</tbody>
</table>

**Chapter at a Glance**

- Historical Remarks | 387
- Further Reading | 388
- Exercises with Solutions | 390
- Exercises | 392

<table>
<thead>
<tr>
<th>Chapter</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>11 Continuous Wavelet and Windowed Fourier Transforms</strong></td>
</tr>
<tr>
<td>11.1 The Continuous Wavelet Transform</td>
</tr>
<tr>
<td>11.1.1 Definition</td>
</tr>
<tr>
<td>11.1.2 Invertibility</td>
</tr>
<tr>
<td>11.1.3 Energy Conservation</td>
</tr>
</tbody>
</table>
# Contents

11.1.4 Shift and Scale Invariance ............................... 397
11.1.5 Singularity Characterization ............................... 399
11.1.6 Decay in Smooth Regions ............................... 402
11.1.7 Redundancy of the Continuous Wavelet Transform .................. 404
11.1.8 Examples .................................................. 406
11.1.9 Scalograms ................................................. 407
11.1.10 Remarks .................................................. 408

11.2 The Windowed Fourier Transform ............................... 408

11.2.1 Introduction and Definition ............................... 408
11.2.2 Inversion ................................................. 410
11.2.3 Energy Conservation ....................................... 411
11.2.4 Reproducing Kernel and Redundancy ....................... 412
11.2.5 Shift and Modulation Properties ........................... 413
11.2.6 Characterization of Singularities and Smoothness ............... 413
11.2.7 Spectrograms .............................................. 415
11.2.8 Remarks .................................................. 416

11.3 Sampling and Computing the CWT and WFT ......................... 416

11.3.1 Sampling Grids .............................................. 416
11.3.2 Frames from Sampled CWT and WFT .......................... 416
11.3.3 Fast computation of Sampled CWT ........................... 416
11.3.4 Fast computation of Sampled WFT ........................... 416

11.4 Discussion and Summary ....................................... 416

Chapter at a Glance .............................................. 416
Historical Remarks .............................................. 416
Further Reading ................................................. 417
Exercises with Solutions .............................................. 417
Exercises .................................................. 417

12 Approximation, Estimation, and Compression 419

12.1 Introduction ................................................. 419
12.2 Modeling .................................................. 419
12.3 Approximation .............................................. 420
12.4 Estimation ................................................. 420
12.5 Denoising .................................................. 420
12.6 Compression .............................................. 420
12.7 Algorithms ................................................. 420

Chapter at a Glance .............................................. 420
Historical Remarks .............................................. 421
Further Reading ................................................. 421
Exercises with Solutions .............................................. 421
Exercises .................................................. 421

Bibliography 423

Index 435
Image Attribution

This list summarizes the sources for various images used in the book. Most of the text in the description is taken verbatim from the source (most often Wikipedia (W), Wikimedia Commons WC) or The MacTutor History of Mathematics archive (MT).

**Newton’s treatise on Opticks** The first, 1704, edition of “Opticks or A Treatise of the Reflections, Refractions, Inflections and Colours of Light”. The image is in the public domain because its copyright has expired.

**Rainbow explained** Permission is granted to copy, distribute and/or modify this document under the terms of the GNU Free Documentation license, Version 1.2 or any later version published by the Free Software Foundation; with no Invariant Sections, no Front-Cover Texts, and no Back-Cover Texts.

**Haar** According to the MT website: ”We believe that most of the images are in the public domain and that provided you use them on a website you are unlikely to encounter any difficulty.”
Source: MT [http://www-history.mcs.st-andrews.ac.uk/PictDisplay/Haar.html].

**Euclid** Euclid of Megara (lat: Evklidi Megaren), Panel from the Series “Famous Men”, Justus of Ghent, about 1474. Panel, 102 x 80 cm, Urbino, Galleria Nazionale delle Marche. The image is in the public domain because its copyright has expired.

**Hilbert** This photograph was taken in 1912 for postcards of faculty members at the University of Göttingen which were sold to students at the time (see “Hilbert”, Constance Reid, Springer 1970). It was therefore published before 1923. Since it is not written in a foreign language, it is public domain in the US.

**FFT Butterfly** Will recreate or ask for permission to use.

Heisenberg  Werner Karl Heisenberg (1901 - 1976). According to the MT website: "We believe that most of the images are in the public domain and that provided you use them on a website you are unlikely to encounter any difficulty." This image (or other media file) is in the public domain because its copyright has expired. Source: WC [http://commons.wikimedia.org/wiki/Image:Heisenberg_10.jpg]


Rainbow  Classic diagram of a dispersion prism from English Wikipedia. Permission is granted to copy, distribute and/or modify this document under the terms of the GNU Free Documentation license. Source: WC [http://commons.wikimedia.org/wiki/Image:Prism_rainbow_sche-ma.png]
## Abbreviations

<table>
<thead>
<tr>
<th>Abbreviation</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>CTFT</td>
<td>Continuous-time Fourier transform</td>
</tr>
<tr>
<td>DFT</td>
<td>Discrete Fourier transform</td>
</tr>
<tr>
<td>DTFT</td>
<td>Discrete-time Fourier transform</td>
</tr>
<tr>
<td>DWT</td>
<td>Discrete wavelet transform</td>
</tr>
<tr>
<td>FB</td>
<td>Filter bank</td>
</tr>
<tr>
<td>FFT</td>
<td>Fast Fourier transform</td>
</tr>
<tr>
<td>FIR</td>
<td>Finite impulse response</td>
</tr>
<tr>
<td>FS</td>
<td>Fourier series</td>
</tr>
<tr>
<td>FT</td>
<td>Fourier transform</td>
</tr>
<tr>
<td>IIR</td>
<td>Infinite impulse response</td>
</tr>
<tr>
<td>LSI</td>
<td>Linear shift invariant</td>
</tr>
<tr>
<td>MB</td>
<td>Mercedes-Benz</td>
</tr>
<tr>
<td>ONB</td>
<td>Orthonormal basis</td>
</tr>
<tr>
<td>PTF</td>
<td>Parseval tight frame</td>
</tr>
<tr>
<td>SVD</td>
<td>Singular value decomposition</td>
</tr>
<tr>
<td>TF</td>
<td>Tight frame</td>
</tr>
<tr>
<td>WP</td>
<td>Wavelet packets</td>
</tr>
<tr>
<td>WT</td>
<td>Wavelet transform</td>
</tr>
</tbody>
</table>
Abbreviations
### Quick Reference

<table>
<thead>
<tr>
<th>Concept</th>
<th>Notation</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Natural numbers</td>
<td>( \mathbb{N} )</td>
<td>0, 1, ...</td>
</tr>
<tr>
<td>Integer numbers</td>
<td>( \mathbb{Z} )</td>
<td>( \ldots, -1, 0, 1, \ldots )</td>
</tr>
<tr>
<td>Real numbers</td>
<td>( \mathbb{R} )</td>
<td>( (\infty, \infty) )</td>
</tr>
<tr>
<td>Complex numbers</td>
<td>( \mathbb{C} )</td>
<td>( a + jb, re^{j\theta} )</td>
</tr>
<tr>
<td>A generic vector space</td>
<td>( V )</td>
<td>§1.2</td>
</tr>
<tr>
<td>A generic Hilbert space</td>
<td>( H )</td>
<td>§1.3</td>
</tr>
<tr>
<td>Real part of set ( S )</td>
<td>( \Re(\cdot) )</td>
<td></td>
</tr>
<tr>
<td>Imaginary part of set ( S )</td>
<td>( \Im(\cdot) )</td>
<td></td>
</tr>
<tr>
<td>Functions</td>
<td>( x(t) )</td>
<td>Argument ( t ) is continuous valued, ( t \in \mathbb{R} )</td>
</tr>
<tr>
<td>Sequences</td>
<td>( x_n )</td>
<td>Argument ( n ) is an integer, ( n \in \mathbb{Z} )</td>
</tr>
<tr>
<td>Ordered sequence ( (x_n)_n )</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Set containing ( x_n )</td>
<td>( {x_n}_n )</td>
<td></td>
</tr>
<tr>
<td>Vector ( x ) with ( x_n ) as elements</td>
<td>( [x_n] )</td>
<td></td>
</tr>
<tr>
<td>Dirac delta “function”</td>
<td>( \delta(t) )</td>
<td>( \int x(t)\delta(t), dt = x(0) )</td>
</tr>
<tr>
<td>Kronecker/Dirac/discrete impulse sequence</td>
<td>( \delta_n )</td>
<td>( \delta_n = 1 ) for ( n = 0 ); ( \delta_n = 0 ) otherwise</td>
</tr>
</tbody>
</table>

#### Elements of Real Analysis

- Integration by parts:
  \[ \int u\, dv = uv - \int v\, du \]

#### Elements of Complex Analysis

- Complex number:
  \( z = a + jb, \, re^{j\theta}, \, a, b \in \mathbb{R}, \, r \in \mathbb{R}^+, \, \theta \in [0, 2\pi] \)
- Conjugation:
  \( z^* = a - jb, \, re^{-j\theta} \)
- Principal root of unity:
  \( W_N = e^{-j\frac{2\pi}{N}} \)
- Conjugation of coefficients but not of \( z \):
  \( X_\ast(z) \)
### Quick Reference

#### Standard Vector Spaces

<table>
<thead>
<tr>
<th>Space Description</th>
<th>Notation</th>
<th>Condition</th>
<th>norm</th>
</tr>
</thead>
<tbody>
<tr>
<td>Banach space of sequences with finite p norm</td>
<td>$\ell^p(\mathbb{Z})$</td>
<td>$x : \mathbb{Z} \rightarrow \mathbb{C} \mid \sum_n</td>
<td>x_n</td>
</tr>
<tr>
<td>Banach space of bounded sequences with supremum norm</td>
<td>$\ell^\infty(\mathbb{Z})$</td>
<td>$x : \mathbb{Z} \rightarrow \mathbb{C} \mid \sup_n</td>
<td>x_n</td>
</tr>
<tr>
<td>Banach space of functions with finite p norm</td>
<td>$\mathcal{L}^p(\mathbb{R})$</td>
<td>$x : \mathbb{R} \rightarrow \mathbb{C} \mid \int</td>
<td>x(t)</td>
</tr>
<tr>
<td>Hilbert space of square-summable sequences</td>
<td>$\ell^2(\mathbb{Z})$</td>
<td>$x : \mathbb{Z} \rightarrow \mathbb{C} \mid \sum_n</td>
<td>x_n</td>
</tr>
<tr>
<td>Hilbert space of square-integrable functions</td>
<td>$\mathcal{L}^2(\mathbb{R})$</td>
<td>$x : \mathbb{R} \rightarrow \mathbb{C} \mid \int</td>
<td>x(t)</td>
</tr>
</tbody>
</table>

#### Bases and Frames

<table>
<thead>
<tr>
<th>Basis/Frame</th>
<th>Description</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>standard Euclidean basis</td>
<td>${e_n}$</td>
<td>$e_{nk} = 1$, for $k = n$, and 0 otherwise</td>
</tr>
<tr>
<td>vector, element of basis or frame</td>
<td>$\varphi$</td>
<td>when applicable, a column vector</td>
</tr>
<tr>
<td>basis or frame</td>
<td>$\Phi$</td>
<td>set of vectors ${\varphi_n}$</td>
</tr>
<tr>
<td>operator</td>
<td>$\Phi$</td>
<td>concatenation of $\varphi_n$s in a linear operator: $[\varphi_0 \varphi_1 \ldots \varphi_{N-1}]$</td>
</tr>
<tr>
<td>vector, element of dual basis or frame</td>
<td>$\tilde{\varphi}$</td>
<td>when applicable, a column vector</td>
</tr>
<tr>
<td>operator</td>
<td>$\tilde{\Phi}$</td>
<td>concatenation of $\tilde{\varphi}_n$s in a linear operator: $[\tilde{\varphi}_0 \tilde{\varphi}<em>1 \ldots \tilde{\varphi}</em>{N-1}]$</td>
</tr>
</tbody>
</table>

#### Transforms

<table>
<thead>
<tr>
<th>Transform</th>
<th>Description</th>
<th>Formula</th>
</tr>
</thead>
<tbody>
<tr>
<td>DFT: discrete Fourier trans.</td>
<td>$x_n \xrightarrow{\text{DFT}} X_k$</td>
<td>$X_k = \sum_{n=0}^{N-1} x_n W_N^{kn}$</td>
</tr>
<tr>
<td>DTFT: discrete-time Fourier trans.</td>
<td>$x_n \xrightarrow{\text{DTFT}} X(\omega)$</td>
<td>$X(\omega) = \sum_{n=-\infty}^{\infty} x_n e^{-j\omega n}$</td>
</tr>
<tr>
<td>FS: Fourier series</td>
<td>$x(t) \xrightarrow{\text{FS}} X_k$</td>
<td>$X_k = \frac{1}{T} \int_{-T/2}^{T/2} x(t) e^{-j2\pi knT} dt$</td>
</tr>
<tr>
<td>FT: continuous-time Fourier trans.</td>
<td>$x(t) \xrightarrow{\text{CTFT}} X(\omega)$</td>
<td>$X(\omega) = \int_{-\infty}^{\infty} x(t) e^{-j\omega t} dt$</td>
</tr>
<tr>
<td>ZT: $z$-trans.</td>
<td>$x_n \xrightarrow{\text{ZT}} X(z)$</td>
<td>$X(z) = \sum_{n=-\infty}^{\infty} x_n z^{-n}$</td>
</tr>
</tbody>
</table>

#### Discrete-Time Nomenclature

<table>
<thead>
<tr>
<th>Sequence</th>
<th>Description</th>
<th>Formula</th>
</tr>
</thead>
<tbody>
<tr>
<td>discrete-time system</td>
<td>$T_0$</td>
<td>filter, operator</td>
</tr>
<tr>
<td>linear</td>
<td>$T_0$</td>
<td>filter, operator, matrix</td>
</tr>
<tr>
<td>linear shift-invariant convolution</td>
<td>$h \ast x$</td>
<td>$\sum_k x_k h_{n-k}$</td>
</tr>
<tr>
<td>eigensequence</td>
<td>$v_n$</td>
<td>eigenfunction, eigenvector</td>
</tr>
<tr>
<td>infinite time</td>
<td>$v_n = e^{j\omega n}$</td>
<td>$h \ast v = H(e^{j\omega}) v$</td>
</tr>
<tr>
<td>finite time</td>
<td>$v_n = e^{j2\pi n}$</td>
<td>$h \ast v = H_k v$</td>
</tr>
<tr>
<td>frequency response</td>
<td>$H(e^{j\omega})$</td>
<td>$\sum_{n=-\infty}^{\infty} h_n e^{-j\omega n}$</td>
</tr>
<tr>
<td>infinite time</td>
<td>$H_k$</td>
<td>$\sum_{n=0}^{\infty} h_n e^{-j\omega n} = \sum_{n=0}^{N-1} h_n W_N^{kn}$</td>
</tr>
</tbody>
</table>
Quick Reference

**Filters**
- synthesis lowpass: $g_n$
- synthesis highpass: $h_n$
- analysis lowpass: $\tilde{g}_n$
- analysis highpass: $\tilde{h}_n$

**Two-Channel Filter Banks**
- lowpass sequence: $\alpha_k = \langle \tilde{g}_{2k-n}, x_n \rangle$
- highpass sequence: $\beta_k = \langle \tilde{h}_{2k-n}, x_n \rangle$
- synthesis basis: even elements: $\varphi_{2k,n} = g_{n-2k}$
- synthesis basis: odd elements: $\varphi_{2k+1,n} = h_{n-2k}$
- analysis basis: even elements: $\tilde{\varphi}_{2k,n} = \tilde{g}_{n-2k}$
- analysis basis: odd elements: $\tilde{\varphi}_{2k+1,n} = \tilde{h}_{n-2k}$
- synthesis filter length: $L$
Preface

The aim of these notes is to present, in a comprehensive way, a number of results, techniques, and algorithms for signal representation that have had a deep impact on the theory and practice of signal processing and communications. While rooted in classic Fourier techniques for signal representation, many results appeared during the flurry of activity of the 1980’s and 1990’s, when new constructions were found for local Fourier transforms and for wavelet orthonormal bases. These constructions 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.

The presentation consists of two main parts, corresponding to background material and the central theme of signal representations. A companion book on applications is in the works.

Part I, Tools of the Trade, reviews all the necessary mathematical material to make the notes self-contained. For many readers, this material might be well known, for others, it might be welcome. It is a refresher of the basic mathematics used in signal processing and communications, and it develops the point of view used throughout the book. Thus, in Chapter 1, From Euclid to Hilbert, the basic geometric intuition central to Hilbert spaces is reviewed, together with all the necessary tools underlying the construction of bases. Chapter 2, Sequences and Signal Processing, is a crash course on processing signals in discrete time or discrete space. In Chapter 3, Fourier’s World, the mathematics of Fourier transforms and Fourier series is reviewed. The final chapter in Part I, Chapter 4, Sampling, Interpolation, and Approximation, talks about the critical link between discrete and continuous domains as given by the sampling theorem. It also veers from the exact world to the approximate one.

Part II, Fourier and Wavelet Representations, is the heart of the book. It aims at presenting a consistent view of signal representations that include Fourier, local Fourier, and wavelet bases, as well as related constructions, frames, and continuous transforms. It starts in Chapter 5, Time, Frequency, Scale and Resolution, with time-frequency analysis and related concepts, showing the intuitions central to the signal representations constructed in the sequel. Chapter 6, Filter Banks: Building Blocks of Time-Frequency Expansions, presents a thor-
ough treatment of the most elementary block—the two-channel filter bank, a signal processing device that splits a signal into a coarse, lowpass approximation, and a highpass difference. This block is then used to derive the discrete wavelet transform in Chapter 7, Wavelet Series on Sequences. It is also used to construct wavelets for the real line in Chapter 8, Wavelet Series on Functions, where other wavelet constructions are also given, in particular those based on multiresolution analysis. We then return to a more Fourier-like view of signal representations in Chapter 9, Localized Fourier Series on Sequences and Functions based on modulated filter banks. Relaxing the condition of completeness inherent in bases to allow for overcompleteness leads to frames, studied in Chapter 10, Frames on Sequences. Chapter 11, Continuous Wavelet and Windowed Fourier Transforms, develops continuous time-frequency transforms, where time, frequency, or scale indices are now continuous (and thus “infinitely” overcomplete!). The final Chapter 12, Approximation, Estimation, and Compression ends with three classical tasks, making a step towards the real world and modeling of that world; no small task. Fourier and wavelet representations are natural models for at least some objects of interest, and are thus shown in action.

As can be seen from the outline, we try to present a synthetic view from basic mathematical principles to actual construction of bases, always with an eye on concrete applications. While the benefit is a self-contained presentation, the cost is a rather sizable manuscript. We provide a reading guide with numerous routes through the material. The level spans from elementary to advanced material, but in a gradual fashion and with indications of levels of difficulty. In particular, starred sections can be skipped without breaking the flow of the material.

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, Prentice Hall, 1995, which they and others used to teach graduate courses at various US and European institutions. With a decade of experience, the maturing of the field, and the broader interest arising from and for these topics, the time was right for a 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 are used at Carnegie Mellon University in an undergraduate class on bioimage informatics, where some of the students are biology 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 which have arisen 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 decides which one is the best!
Preface to Release 3.0

This release has a number of chapters in an “almost” finished form. Below, we summarize the differences from Release 2.1, and what is left to be done. All the front material, as well as Chapters 1, 2, 5 and 6 have been completely revised. For figures, we have strived to provide any version of available, even if handwritten and then scanned. While this can sometimes lead to confusion as notation in the figure might not correspond to the notation in the text, we felt it was still beneficial to have some version of the figure available. Such figures have a note in the caption and should be used with caution.

Front Matter Front matter consists of the following items:

- Image Attribution V3.0: This is a new addition, giving credit to all the images not produced by the authors. Currently, most of those have been obtained from Wikimedia. For some, we put a placeholder image with a comment.
- Abbreviations V3.0: This is a new addition, listing all the abbreviations used throughout the book. Currently these abbreviations are those in Chapters 1, 2, 5, 6, and 7.
- Quick Reference V3.0: Revised from V2.1, and renamed from Notation.
- Preface: Revised from V2.1.
- Reading Guide V3.0: This is a new addition, giving several “roadmaps” of how the book could be used. It is not clear whether the same format will be used in the final version.
- From Rainbows to Spectra V3.0: Revised from V2.1.

Chapter 1 V3.0 Revised from V2.1. This chapter has undergone major revisions.

Chapter 2 V3.0 Revised from V2.1. This chapter has undergone major revisions.

Chapter 3 V2.1 Notation might not agree with V3.0 Chapters 1,2,5,6.

Chapter 4 V2.1 Notation might not agree with V3.0 Chapters 1,2,5,6.

Chapter 5 V3.0 Revised from V2.1. This chapter has undergone major revisions.

Chapter 6 V2.1 Revised from V2.1. This chapter has undergone major revisions.
Chapter 7 V2.1 Notation might not agree with V3.0 Chapters 1,2,5,6.

Chapter 8 V2.1 Notation might not agree with V3.0 Chapters 1,2,5,6.

Chapter 9 V2.1 Notation might not agree with V3.0 Chapters 1,2,5,6.

Chapter 10 V2.1 Notation might not agree with V3.0 Chapters 1,2,5,6.

Chapter 11 V2.1 Notation might not agree with V3.0 Chapters 1,2,5,6.

Chapter 12 V2.1 Notation might not agree with V3.0 Chapters 1,2,5,6.

Back Matter  Back matter consists of the following items:

- *Bibliography V3.0*: Bibliography is still not consistent in style.
- *Index V3.0*: Currently, as a test, only Quick Reference, From Rainbows to Spectra and Chapter 1 have been indexed.
Acknowledgments

A project like this exists only thanks to the collaboration and help of many people, most of whom we list and thank below.

Françoise Behn and Jocelyne Plantefol typed and organized parts of the manuscript, and Eric Strattman assisted with many of the figures. We thank them for their expertise and patience.

A special thanks to Patrick Vandewalle, who designed, wrote and implemented large parts of the Matlab companion to the book. Similarly special thanks go to Grace Chang, who helped organize and created the problem companion.

We are very grateful to Prof. Libero Zuppiroli of EPFL who proposed and designed the experiment from Newton’s treatise on Opticks [119], and to Christiane Grimm whose photograph thereof graces the cover.

We thank our extended “family”, many of whom proofread parts of the book, taught from it, sent comments and suggestions, and provided several exercises. In particular, Zoran Cvetković, Minh Do, Pier Luigi Dragotti, Michael Gastpar, Thao Nguyen, Antonio Ortega and Kannan Ramchandran have done more than their share.

We thank Yue M. Lu, Hossein Rouhani and Christophe Tourney for careful reading of the manuscript and many useful comments.

Martin Vetterli thanks many graduate students from EPFL who helped mature the material, solve problems, catch typos, suggest improvements, among other things. They include Amina Chebira, Minh Do, Pier Luigi Dragotti, Ali Hormati, Ivana Jovanović, Jérôme Lebrun, Pina Marziliano, Fritz Menzer, Paolo Prandoni, Olivier Roy, Rahul Shukla, Patrick Vandewalle, Vladan Velisavljević.

Jelena Kovačević thanks her graduate students Ramu Bhagavatula, Amina Chebira, Charles Jackson, Tad Merryman, Aliaksei Sandryhaila and Gowri Srinivasu, 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: 42-403, 42-431, 42-503, 42-703, 42-731, 18-486, 18-795, 18-799, taught from 2003-2008.

Vivek K Goyal thanks students and TAs including in particular Baris Erkmen, Zahi Karam, Vincent Tan, and Serhii Zhak.
Reading Guide

Below we give suggestions on how to material could be covered in a standard, one-semester, course. Most of these scenarios have been taught already by one of the authors (where appropriate, we will note that). We will also note levels and audience whenever possible.

<table>
<thead>
<tr>
<th>Chapter</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
<th>11</th>
<th>12</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Route 1</strong></td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Level</td>
<td>Graduate</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Taught</td>
<td>Martin Vetterli at EPFL, S’07, S’08.</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Course Name</td>
<td>Advanced Signal Processing: Wavelets and Applications</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Comments</td>
<td>The course has Signal Processing for Communications as a prerequisite.</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Route 2</strong></td>
<td>✓</td>
<td>✓</td>
<td>·</td>
<td>·</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>·</td>
<td>·</td>
<td>·</td>
<td>·</td>
</tr>
<tr>
<td>Level</td>
<td>Graduate</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Taught</td>
<td>Jelena Kovačević at CMU, S’08.</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Course Name</td>
<td>Wavelets and Multiresolution Techniques.</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Comments</td>
<td>This course covers only sequence representations. It has only Signals and Systems as a prerequisite.</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Route 3</strong></td>
<td>✓</td>
<td>✓</td>
<td>·</td>
<td>·</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>·</td>
<td>·</td>
<td>✓</td>
</tr>
<tr>
<td>Level</td>
<td>Graduate</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Taught</td>
<td>Vivek Goyal at MIT, S’07.</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Course Name</td>
<td>Wavelets, Approximation, and Compression</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Comments</td>
<td>This course does not have DTSP as a prerequisite, but most students are well prepared.</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Chapter</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>5</td>
<td>6</td>
<td>7</td>
<td>8</td>
<td>9</td>
<td>10</td>
<td>11</td>
<td>12</td>
</tr>
<tr>
<td>---------</td>
<td>---</td>
<td>---</td>
<td>---</td>
<td>---</td>
<td>---</td>
<td>---</td>
<td>---</td>
<td>---</td>
<td>---</td>
<td>----</td>
<td>----</td>
<td>----</td>
</tr>
<tr>
<td>Route 4</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Level</td>
<td>Graduate</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Taught</td>
<td>Jelena Kovačević at CMU, S’04, F’04, F’05, F’06.</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Course Name</td>
<td>Advanced Bioimage Informatics</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Comments</td>
<td>This material was taught as part of the course, covering modern tools to be used in bioimage informatics. The course has only Signals and Systems as a prerequisite.</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Route 5</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Level</td>
<td>Undergraduate</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Taught</td>
<td>Jelena Kovačević at CMU, S’05, S’06, F’06, F’07.</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Course Name</td>
<td>Bioimage informatics</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Comments</td>
<td>This material was taught as part of the course, covering modern tools to be used in bioimage informatics. The course has only Signals and Systems as a prerequisite.</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
In the early part of the 14th century, the Dominican monk Dietrich von Freiberg performed a simple experiment: he held a spherical bottle filled with water in the sunlight. The bottle played the role of water drops, and the effect of dispersion—breaking the light into colors—together with reflection, gave a scientific explanation of the rainbow effect, including the secondary rainbow with weaker reversed colors. This is often considered the first scientific experiment performed and documented in Western civilization.

Von Freiberg fell short of complete understanding of the phenomenon because he, like many of his contemporaries, believed that colors were simply intensities between black and white. A full understanding emerged three hundred years later when Descartes and Newton gave a complete explanation based on the fact that dispersion separates the white light into spectral components of different wavelengths, the colors of the rainbow!

This brings us to the central theme of this book: complex phenomena can be broken into simpler constituent components, simplifying explanations—a “divide and conquer” approach. In the case of the rainbow, sunlight contains a combination of all wavelengths within the visible spectrum.

A French mathematician, Joseph Fourier (1768-1830), formalized the notion of the spectrum in the early 19th century. He was interested in the heat equation—the differential equation governing the diffusion of heat. Fourier’s key insight was to decompose a periodic function \( x(t) = x(t + T) \) into an infinite sum of sines and cosines of periods \( T/k, k \in \mathbb{Z}^+ \). Since these sine and cosine components are eigenfunctions of the heat equation, the solution of the problem is simplified: one can analyze the differential equation for each component separately, and combine the intermediate results, thanks to the superposition principle of linear systems. Fourier’s decomposition earned him a coveted prize from the French Academy of Sciences, but with a mention that his work lacked rigor. Indeed, the question of which functions admit a Fourier decomposition is a deep one, and it took another 150 years to settle.
Orthonormal Bases  Fourier’s work is at the heart of the present book. We aim to study possible representations for various functions (signals) of interest. Such representations are in the form of expansion formulas for the function in terms of basis vectors. Call the basis vectors \( \varphi_k, k \in \mathbb{Z} \). Then

\[
x = \sum_k X_k \varphi_k,
\]

(0.1)

where the coefficients \( X_k \) are obtained from the function \( x \) and the basis vectors \( \varphi_k \), through an inner product,

\[
X_k = \langle x, \varphi_k \rangle,
\]

(0.2)

and we have assumed that the basis vectors form an orthonormal set, that is,

\[
\langle \varphi_k, \varphi_i \rangle = \delta_{k-i}.
\]

To make matters specific, let us look at Fourier’s groundbreaking construction of a series representation for periodic functions with the period \( T = 1 \). Such functions can be written as

\[
x(t) = \sum_{k=-\infty}^{\infty} X_k e^{j2\pi kt},
\]

(0.3)

where

\[
X_k = \int_0^1 x(t) e^{-j2\pi kt} dt.
\]

(0.4)

We can define basis vectors \( \varphi_k, k \in \mathbb{Z} \), on the interval \([0,1)\), as

\[
\varphi_k(t) = e^{j2\pi kt}, \quad 0 \leq t < 1,
\]

(0.5)

and the Fourier series coefficients as

\[
X_k = \langle x, \varphi_k \rangle = \int_0^1 x(t) \varphi_k^*(t) dt.
\]

The basis vectors form an orthonormal set (the first few are shown in Figure 0.1):

\[
\langle \varphi_k, \varphi_i \rangle = \int_0^1 e^{j2\pi kt} e^{-j2\pi it} dt = \delta_{k-i}.
\]

(0.6)

The Fourier series is certainly a key orthonormal basis with many outstanding properties. What other bases exist, and what are their properties? Early in the 20th century, Alfred Haar proposed a basis which looks very different from Fourier’s. It is based on a function \( \psi(t) \) defined as

\[
\psi(t) = \begin{cases} 
1, & 0 \leq t < 1/2; \\
-1, & 1/2 \leq t < 1; \\
0, & \text{otherwise.}
\end{cases}
\]

(0.7)
For the interval \([0, 1)\), an orthonormal system can be built by scaling \(\psi(t)\) by powers of 2, and then shifting the scaled versions appropriately. The basis functions are given by

\[
\psi_{m,n}(t) = 2^{-m/2} \psi\left(\frac{t - n2^m}{2^m}\right),
\]

with \(m \in \{0, -1, -2, \ldots\}\) and \(n \in \{0, 1, \ldots, 2^m - 1\}\). Rather than dwell on this formula, which will be justified in detail later, we show the resulting function system in Figure 0.2, where visual inspection will convince the reader that the various basis functions are indeed orthogonal to each other.

By adding to the \(\psi_{m,n}\) s the function

\[
\varphi_0(t) = \begin{cases} 
1, & 0 \leq t < 1; \\
0, & \text{otherwise},
\end{cases}
\]

an orthonormal basis for the interval \([0, 1)\) is obtained. This is a very different basis than the Fourier basis; for example, instead of being infinitely differentiable, none of the \(\psi_{m,n}\) s is even continuous.

It is natural to ask: Which basis is better? Such a question does not have a simple answer, and that answer will depend on the classes of functions we wish to represent as well as our goals in the representation. Furthermore, we will have to
be careful in describing what we mean by equality in an expansion such as (0.3); otherwise we would be repeating Fourier’s mistake.

**Approximation** One way to assess the “quality” of a basis is to see how well it can approximate a given function with a finite number of terms. In that sense, history is again enlightening. Fourier series became such a useful tool during the 19th century, that researchers built elaborate mechanical devices that would compute a function based on Fourier series coefficients. Clearly, as they had no luxury of access to digital computers, they built analog computers, based on harmonically related rotating wheels, where amplitudes of Fourier coefficients could be set and the sum computed. One such machine, the Harmonic Integrator, designed by the physicists Michelson and Stratton, could compute a series with 80 terms. To the designers’ dismay, the synthesis of a square wave from its Fourier series led to oscillations around the discontinuity that would not go away as the number of terms was increased. They were convinced of a mechanical “bug” in their machine, until Gibbs proved in 1899 that Fourier series of discontinuous functions cannot converge uniformly: any finite series would be plagued by this **Gibbs phenomenon** around the discontinuity (see Figure 0.3 for an example).

So what would the Haar basis provide in this case? Surely, it seems more appropriate for a square wave! Unfortunately, taking the first $2^{-m}$ coefficients in the natural ordering (for $m = 0, -1, -2, \ldots$) leads to similarly poor performance, shown in Figure 0.4.

A slight change in the approximation procedure makes a big difference; By retaining the largest coefficients (in absolute value) instead of simply keeping a fixed set of coefficients, the approximation quality changes drastically, as seen in Figure 0.5. Referring back to Figures 0.3 and 0.4, the Fourier approximations have 5, 33, and 255 nonzero terms while the similar-quality Haar approximations have only 4, 7, and 10 nonzero terms.
Figure 0.4: Approximation of a square wave with 8, 64 and 512 coefficients in a Haar basis, using the first 8, 64 and 512 scale coefficients $m$. The discontinuity was chosen at the irrational point $1/\sqrt{2}$. On the right is a detail of the left in the neighborhood of the discontinuity along with the target function.

Figure 0.5: Approximation of a square wave by retaining the largest coefficients (in magnitude) of a Haar expansion. Approximations with 8 and 64 terms are shown, zoomed in near the point of discontinuity; the 64-term approximation is visually indistinguishable from the target function.

Through this comparison, we have discovered that the “quality” of a basis for approximation depends on the method of approximation. Retaining a predefined set of coefficients, as in the Fourier series case (Figure 0.3) or the first Haar case (Figure 0.4) gives a linear, subspace approximation. In the second Haar example (Figure 0.5), the selection process leads to a nonlinear approximation procedure, and a superior approximation quality.

The central theme of the book is the design of bases with certain features. While not the only criterion used to compare bases, approximation quality arises repeatedly. We will see that approximation quality is closely related to the central signal processing tasks of sampling, filtering, estimation and compression.
From Rainbows to Spectra
Part I

Tools of the Trade
Chapter 1

From Euclid to Hilbert

Contents

1.1 Introduction ............................................. 10
1.2 Vector Spaces .......................................... 17
1.3 Hilbert Spaces .......................................... 27
1.4 Bases and Frames ......................................... 38
1.5 Random Variables as Hilbert Space Vectors ...... 46
1.6 Algorithms ............................................... 46
1.7 Elements of Real Analysis .............................. 49
1.8 Elements of Algebra: Polynomials ................. 52
1.9 Elements of Linear Algebra ......................... 53
Chapter at a Glance ......................................... 63
Historical Remarks .......................................... 65
Further Reading ............................................. 65
Exercises with Solutions .................................... 66
Exercises ...................................................... 73

We all start our journeys into the world of Fourier and wavelets with different backgrounds and perspectives. This chapter aims to establish a common language, develop the foundations for our study, and begin to draw out key themes.

You will see many more formal definitions in this chapter than in any other, to approach the ideal of a self-contained treatment. However, we must assume some background in common: We expect the reader to be familiar with signals and systems at the level of [121, Ch. 1–7,10], linear algebra at the level of [144, Ch. 1–5], and probability at the level of [17, Ch. 1–4]. (The textbooks we have cited are just examples; nothing unique to those books is necessary.) On the other hand, we are not assuming prior knowledge of general vector space abstractions or mathematical analysis beyond basic calculus; these are the topics that we develop in this chapter so that you can extend your geometric intuition from ordinary Euclidean space to spaces of sequences and functions. If you would like more details on abstract vector spaces, we recommend books by Kreyszig [104], Luenberger [111], and Young [170].
Chapter 1. From Euclid to Hilbert

1.1 Introduction

We start our journey by introducing many topics of this chapter through the familiar setting of the real plane. In the more general treatment in the subsequent sections, the intuition we developed through years of dealing with the Euclidean spaces around us (\( \mathbb{R}^2 \) and \( \mathbb{R}^3 \)), will generalize to some not-so-familiar spaces. If you are comfortable with notions of vector space, inner product and norm, projections and bases, skip this section; otherwise, this will be a “gentle” introduction into the Euclid world.

Real Plane as a Vector Space

Let us start with a look at the familiar setting of \( \mathbb{R}^2 \), that is, real vectors with two coordinates. We adopt the convention of writing vectors as columns, such as \( x = \begin{bmatrix} x_1 & x_2 \end{bmatrix}^T \). Denote by \( e_1 \) and \( e_2 \) the standard horizontal and vertical unit vectors \( \begin{bmatrix} 1 & 0 \end{bmatrix}^T \) and \( \begin{bmatrix} 0 & 1 \end{bmatrix}^T \), respectively.

Adding two vectors in the plane produces a third one also in the plane, and multiplying a vector by a real scalar also yields a vector in the plane. These two ingrained facts make the real plane be a vector space.

Inner Product and Norm

The inner product of vectors \( x = \begin{bmatrix} x_1 & x_2 \end{bmatrix}^T \) and \( y = \begin{bmatrix} y_1 & y_2 \end{bmatrix}^T \) in the real plane is defined as

\[
\langle x, y \rangle = x_1y_1 + x_2y_2. \tag{1.1}
\]

While other names for inner product are scalar product and dot product, we use “inner product” exclusively. The length or norm of a vector \( x \) is

\[
\|x\| = \sqrt{\langle x, x \rangle}. \tag{1.2}
\]

In (1.1), the inner product computation depends on the choice of coordinate axes. Let us now derive an expression in which the coordinates disappear. Consider \( x \) and \( y \) as shown in Figure 1.1. Define the angle between \( x \) and the positive horizontal axis as \( \theta_x \) (measured counterclockwise), and define \( \theta_y \) similarly. Using a little algebra and trigonometry, we get

\[
\langle x, y \rangle = x_1y_1 + x_2y_2 \\
= (\|x\| \cos \theta_x)(\|y\| \cos \theta_y) + (\|x\| \sin \theta_x)(\|y\| \sin \theta_y) \\
= \|x\| \|y\| (\cos \theta_x \cos \theta_y + \sin \theta_x \sin \theta_y) \\
= \|x\| \|y\| \cos(\theta_x - \theta_y). \tag{1.3}
\]

Thus, the inner product of the two vectors is the product of their lengths and the cosine of the angle \( \theta = \theta_x - \theta_y \) between them.

The inner product measures similarity of orientation. For fixed vector lengths, the greater the inner product, the closer the vectors are in orientation. They are the closest when they are colinear and pointing in the same direction, that is, when
1.1. Introduction

\[ \theta = 0 \text{ or } \cos \theta = 1; \text{ they are the farthest when they are antiparallel, with } \theta = \pi \text{ or } \cos \theta = -1. \]

When \( \langle x, y \rangle = 0 \), the vectors are called orthogonal or perpendicular. From (1.3), we see that \( \langle x, y \rangle \) is zero only when the length of one vector is zero (meaning one of the vectors is the vector \([0 \ 0]^T\)) or the cosine of the angle between them is zero (\( \theta = \pm \pi/2 \), vectors are orthogonal). So at least in the latter case, this is consistent with the conventional concept of perpendicularity.

The distance between two vectors is defined as the length of their difference:

\[ d(x, y) = \| x - y \| = \sqrt{\langle x - y, x - y \rangle} = \sqrt{(x_1 - y_1)^2 + (x_2 - y_2)^2}. \quad (1.4) \]

Subspaces and Projections

Any line through the origin is described completely by one of its nonzero points \( y \). This is because that line is the set of all scalar multiples of \( y \). A line is the simplest case of a subspace, and projection to a subspace is intimately related to inner products.

Starting with a vector \( x \) and applying an orthogonal projection to some subspace, gives the vector \( \hat{x} \) in the subspace that is closest to \( x \). The connection to orthogonality is that \( (x - \hat{x}) \) is orthogonal to every vector in the subspace. When the subspace is specified by element \( y \) and \( \| y \| = 1 \), the projection is \( \hat{x} = \langle x, y \rangle y \).

Two examples of orthogonal projections are illustrated in Figure 1.2. Figure 1.2(a) depicts projection to the subspace (line) defined by \( y = e_1 \). Since \( y \) has unit length, the projection can be written as

\[ \hat{x} = \langle x, y \rangle y = \langle x, e_1 \rangle e_1 = x_1 e_1. \quad (1.5) \]

So, orthogonal projection of \( x \) onto a line specified by a coordinate axis picks off the corresponding component of \( x \). The case with \( y \) of unit length but not aligned
Figure 1.2: Examples of an orthogonal projection onto the subspace specified by the unit vector $y$, assuming the standard basis $e_1 = [1 \ 0]^T, e_2 = [0 \ 1]^T$.
(a) Projection onto a basis axis.
(b) Projection onto a nonbasis axis.

with either coordinate axis is depicted in Figure 1.2(b), where we obtain

$$\hat{x} = \langle x, y \rangle y = (\|x\| \|y\| \cos \theta) y = (\|x\| \cos \theta) y.$$  \hfill (1.6)

If $y$ is not of unit length, then

$$\hat{x} = (\|x\| \cos \theta) \frac{y}{\|y\|} = (\|x\| \|y\| \cos \theta) \frac{y}{\|y\|^2} = \frac{1}{\|y\|^2} \langle x, y \rangle y,$$  \hfill (1.7)

where $(\|x\| \cos \theta)$ is the length of the projection, and $(y/\|y\|)$ is the unit vector giving the direction to the projection.

Bases and Coordinates

We defined the real plane as a vector space using coordinates, and we made the natural association of the first coordinate as signed distance as measured from left to right, and the second coordinate as signed distance as measured from bottom to top. In doing so, we implicitly used the standard basis $e_1 = [1 \ 0]^T, e_2 = [0 \ 1]^T$, which is a particular orthonormal basis of $\mathbb{R}^2$. Expressing vectors in a variety of bases is central to our study, and their coordinates will differ depending on the underlying basis.

Orthonormal Bases  Vectors $e_1 = [1 \ 0]^T$ and $e_2 = [0 \ 1]^T$ constituting the standard basis are depicted in Figure 1.3(a). They are orthogonal and of unit length, and are thus called orthonormal. We have been using this basis implicitly in that

$$x = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = x_1 \begin{bmatrix} 1 \\ 0 \end{bmatrix} + x_2 \begin{bmatrix} 0 \\ 1 \end{bmatrix} = x_1 e_1 + x_2 e_2$$ \hfill (1.8)

is an expansion of $x$ with respect to the basis $\{e_1, e_2\}$. For this basis, it is obvious that an expansion exists for any $x$ because the coefficients of the expansion $x_1$ and $x_2$ are simply “read off” of $x$. 

The World of Fourier and Wavelets

Copyright 2008 M. Vetterli, J. Kovacevic, and V. K. Goyal
### 1.1. Introduction

In general, the existence of an expansion like (1.8) for any vector $x$ depends on the set of vectors spanning the whole space. The uniqueness of the expansion depends on the set of vectors being linearly independent. Having both of these properties makes the set a basis.

Expansions with orthonormal bases have several convenient characteristics. When \( \{ \varphi_1, \varphi_2 \} \) is any orthonormal basis (not necessarily \( \{ e_1, e_2 \} \)), one can find the coefficients of \( x = \alpha_1 \varphi_1 + \alpha_2 \varphi_2 \) simply through the inner products

\[
\alpha_1 = \langle x, \varphi_1 \rangle, \quad \text{and} \quad \alpha_2 = \langle x, \varphi_2 \rangle.
\]

The resulting coefficients satisfy

\[
|\alpha_1|^2 + |\alpha_2|^2 = \|x\|^2
\]

by the Pythagorean theorem, because \( \alpha_1 \varphi_1 \) and \( \alpha_2 \varphi_2 \) form sides of a right triangle with hypotenuse of length \( \|x\| \). The equality (1.9) is an example of Parseval’s equality and is related to Bessel’s inequality (these will be formally introduced in Section 1.4.1).

**General Bases** Referring to Figure 1.3(b), consider the problem of representing an arbitrary vector \( x = \begin{bmatrix} x_1 & x_2 \end{bmatrix}^T \) as an expansion \( \alpha_1 \varphi_1 + \alpha_2 \varphi_2 \) with respect to \( \varphi_1 = \begin{bmatrix} 1 & 0 \end{bmatrix}^T \) and \( \varphi_2 = \begin{bmatrix} \frac{1}{2} & 1 \end{bmatrix}^T \). This is not a trivial exercise such as the one of expanding with the standard basis, but we can still come up with an intuitive procedure.

Since \( \varphi_1 \) has no vertical component, we should use \( \varphi_2 \) to match the vertical component of \( x \), yielding \( \alpha_2 = x_2 \). (This is illustrated with the diagonal dashed line in Figure 1.3(b).) Then, we need \( \alpha_1 = x_1 - \frac{1}{2}x_2 \) for the horizontal component to be correct. We can express what we have just done with inner products as

\[
\alpha_1 = \langle x, \bar{\varphi}_1 \rangle, \quad \alpha_2 = \langle x, \bar{\varphi}_2 \rangle, \quad \text{where} \quad \bar{\varphi}_1 = \begin{bmatrix} 1 & \frac{1}{2} \end{bmatrix}^T \quad \text{and} \quad \bar{\varphi}_2 = \begin{bmatrix} 0 & 1 \end{bmatrix}^T,
\]

\( \bar{\varphi}_1 \) and \( \bar{\varphi}_2 \) are the components of \( \bar{\varphi}_1 \) and \( \bar{\varphi}_2 \) with respect to the basis \( \{ \varphi_1, \varphi_2 \} \).
with vectors $\tilde{\varphi}_1$ and $\tilde{\varphi}_2$ as shown in Figure 1.3(c).

We have just derived an instance of the expansion formula

$$x = \langle x, \tilde{\varphi}_1 \rangle \varphi_1 + \langle x, \tilde{\varphi}_2 \rangle \varphi_2,$$

(1.10)

where $\{\tilde{\varphi}_1, \tilde{\varphi}_2\}$ is the basis dual to the basis $\{\varphi_1, \varphi_2\}$, and the two bases are a biorthogonal pair. For any basis, the dual basis is unique. The defining characteristic for a biorthogonal pair is

$$\langle \varphi_k, \tilde{\varphi}_i \rangle = \delta_{k-i} = \begin{cases} 1, & \text{if } k = i; \\ 0, & \text{if } k \neq i. \end{cases}$$

You can check that this is satisfied in our example and that any orthonormal basis is its own dual. Clearly, designing a biorthogonal basis pair has more degrees of freedom than designing an orthogonal basis. The disadvantage is that (1.9) does not hold, and furthermore, computations can become numerically unstable if $\varphi_1$ and $\varphi_2$ are too close to colinear.

**Frames**  The signal expansion (1.10) has the minimum possible number of terms to work for every $x \in \mathbb{R}^2$—two terms because the dimension of the space is two. It can also be useful to have an expansion of the form

$$x = \langle x, \tilde{\varphi}_1 \rangle \varphi_1 + \langle x, \tilde{\varphi}_2 \rangle \varphi_2 + \langle x, \tilde{\varphi}_3 \rangle \varphi_3,$$

(1.11)

Here, an expansion will exist as long as $\{\varphi_1, \varphi_2, \varphi_3\}$ spans $\mathbb{R}^2$, which is even easier with three vectors than with two. Then, even after the set $\{\varphi_1, \varphi_2, \varphi_3\}$ is fixed, there are infinitely many dual sets $\{\tilde{\varphi}_1, \tilde{\varphi}_2, \tilde{\varphi}_3\}$ such that (1.11) holds for all $x \in \mathbb{R}^2$. Such redundant sets are called *frames* and their dual sets *dual frames*. This flexibility

---

**Figure 1.4:** Illustration of overcomplete expansions (frames). (a) A frame obtained by adding a vector to an ONB. (b) A tight frame called Mercedes-Benz (MB) frame.
can be used in various ways. For example, setting a component of \( \tilde{\varphi}_1 \) to zero could save a multiplication and an addition in computing an expansion, or the dual could be chosen to make the coefficients as small as possible.

As an example, let us take the standard ONB, \( \{ \varphi_1 = [1 \ 0]^T, \varphi_2 = [0 \ 1]^T \} \), add a vector to it, \( \varphi_3 = [-1 \ -1]^T \), and see what happens (see Figure 1.4). As there are now three vectors in \( \mathbb{R}^2 \), they are linearly dependent; indeed, \( \varphi_3 = -\varphi_1 - \varphi_2 \). Moreover, these three vectors must be able to represent every vector in \( \mathbb{R}^2 \) since their subset is able to do so (which also means that we could have added any other vector \( \varphi_3 \) to our ONB with the same result). To show that, we use the ONB expansion, \( x = \langle x, \varphi_1 \rangle \varphi_1 + \langle x, \varphi_2 \rangle \varphi_2 \), and add a zero to it to read:

\[
x = \langle x, \varphi_1 \rangle \varphi_1 + \langle x, \varphi_2 \rangle \varphi_2 + \sum_3 (\langle x, \varphi_2 \rangle - \langle x, \varphi_2 \rangle) \varphi_1 + (\langle x, \varphi_2 \rangle - \langle x, \varphi_2 \rangle) \varphi_2.
\]

We now rearrange it slightly:

\[
x = \langle x, (\varphi_1 + \varphi_2) \rangle \varphi_1 + \langle x, 2\varphi_2 \rangle \varphi_2 + \langle x, \varphi_2 \rangle (-\varphi_1 - \varphi_2) = \sum_{k=1}^3 \langle x, \varphi_k \rangle \varphi_k,
\]

with \( \tilde{\varphi}_1 = \varphi_1 + \varphi_2, \tilde{\varphi}_2 = 2\varphi_2, \tilde{\varphi}_3 = \varphi_2 \). This expansion is reminiscent of the one for general, biorthogonal bases, we have seen earlier, except that the vectors involved in the expansion are now linearly dependent, and shows that indeed, we can expand any \( x \in \mathbb{R}^2 \) in terms of the frame \( \Phi = \{ \varphi_k \}_{k=1}^3 \) and its dual frame \( \tilde{\Phi} = \{ \tilde{\varphi}_k \}_{k=1}^3 \).

Can we now get a frame that somehow mimics ONBs? Consider the system given in Figure 1.4(b) with

\[
\varphi_1 = [0 \ 1]^T, \quad \varphi_2 = \left[-\frac{\sqrt{2}}{2} \ -\frac{1}{2}\right]^T, \quad \varphi_3 = \left[\frac{\sqrt{2}}{2} \ -\frac{1}{2}\right]^T.
\]

By expanding for arbitrary \( x = [x_1 \ x_2]^T \), one can verify that \( x = (2/3) \sum_{k=1}^3 \langle x, \varphi_k \rangle \varphi_k \) holds for any \( x \). We see that this frame is self dual, and we can think of the expansion as a generalization of an ONB except that the vectors are not linearly independent anymore and the factor 2/3 in front. The norm is preserved similarly to what happens with ONBs (\( \sum_{k=1}^3 |\langle x, \varphi_k \rangle|^2 = (3/2)||x||^2 \)), except for the factor 3/2 indicating the redundancy of the system. A frame with this property is called a **tight frame**, and this particular one we termed the **Mercedes-Benz (MB) frame**.

### Matrix View of Bases and Frames

We now look at how we can work with bases and frames using a more intuitive and “visual” tool—matrices. Suppose we are given two vectors \( \varphi_1 = (1/\sqrt{2}) [1 \ -1]^T \) and \( \varphi_2 = (1/\sqrt{2}) [1 \ 1]^T \) as in Figure 1.5. You can easily check that \( \{ \varphi_1, \varphi_2 \} \) is an ONB. Given this basis and an arbitrary vector \( x \) in the plane, what is \( x \) in this new basis? We answer this question by projecting \( x \) onto the new basis. Take a specific example: Suppose that \( x = [1 \ 0]^T \), then, \( x_1 = \langle x, \varphi_1 \rangle = 1/\sqrt{2}, x_2 = \langle x, \varphi_2 \rangle = 1/\sqrt{2} \). Thus, in this new coordinate system, our point \( [1 \ 0]^T \) becomes \( x_{\varphi} = (1/\sqrt{2}) [1 \ 1]^T \). It is still the same point in the
plane, we only read its coordinates depending on which basis we are considering. With the standard basis \{\begin{bmatrix} 1 \\ 0 \end{bmatrix}, \begin{bmatrix} 0 \\ 1 \end{bmatrix}\}, the point has coordinates 1 and 0, while with \{\varphi_1, \varphi_2\}, it has coordinates \(\frac{1}{\sqrt{2}}\) and \(\frac{1}{\sqrt{2}}\). Let us express the above process a bit more elegantly:

\[
x_\varphi = \begin{bmatrix} x_{\varphi_1} \\ x_{\varphi_2} \end{bmatrix} = \begin{bmatrix} \langle x, \varphi_1 \rangle \\ \langle x, \varphi_2 \rangle \end{bmatrix} = \begin{bmatrix} \varphi_{11}x_1 + \varphi_{12}x_2 \\ \varphi_{21}x_1 + \varphi_{22}x_2 \end{bmatrix} = \begin{bmatrix} \varphi_{11} \\ \varphi_{21} \\ \varphi_{12} \\ \varphi_{22} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \Phi^T x. \tag{1.12}
\]

We see that the matrix \(\Phi\) describes the ONB in the real plane, with its columns being the basis vectors. Thus, the process of finding coordinates of a vector in a different coordinate system can be conveniently represented using a matrix whose columns are the new basis vectors, \(x_\varphi = \Phi^T x\). If the matrix is singular, we do not have a basis.

We can interpret the above process differently; namely \(\Phi^T\) acting as a rotation. To see that, rewrite \(\Phi^T\) as

\[
\Phi^T = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & -1 \\ 1 & 1 \end{bmatrix} = \begin{bmatrix} 1/\sqrt{2} & -1/\sqrt{2} \\ 1/\sqrt{2} & 1/\sqrt{2} \end{bmatrix} = \begin{bmatrix} \cos \pi/4 & -\sin \pi/4 \\ \sin \pi/4 & \cos \pi/4 \end{bmatrix},
\]

showing that \(\Phi^T\) acts as a rotation by \(\pi/4\) (counterclockwise) on points in the plane. The general form of a rotation in \(\mathbb{R}^2\) (also called planar of Givens rotation) is

\[
R = \begin{bmatrix} \cos \theta & -\sin \theta \\ \sin \theta & \cos \theta \end{bmatrix}, \tag{1.13}
\]
1.2. Vector Spaces

and defines an ONB. (Figure 1.5 depicts a general rotation by an angle \( \theta \).) We will touch upon the rotation matrix in Section 1.C.4. Note that if we swapped \( \varphi_1 \) and \( \varphi_2 \), we would get a reflection instead of a rotation. Both are represented by unitary matrices (see Section 1.C.4); the difference is in their determinants (rotation matrices have \( \det = 1 \), while reflection matrices have \( \det = -1 \)).

Chapter Outline

The next several sections follow the progression of topics in this brief introduction: In Section 1.2, we formally introduce vector spaces and equip them with inner products and norms. We also give several common examples of vector spaces. In Section 1.3, we discuss the issue of completeness that makes an inner product space a Hilbert space. More importantly, we define the central concept of orthogonality and then introduce linear operators and projection operators. In Section 1.4, we define bases and frames. This step gives us the tools to analyze signals and to create approximate representations. The final sections go beyond the scope of this introduction. In Section 1.5, we discuss random variables and random vectors, which are critical in many applications. This is both to see how randomness affects the earlier concepts as well as to demonstrate geometric interpretations of many basic operations with random variables. Section 1.6 discusses a few algorithms pertaining to the material covered. Appendices review some elements of real analysis such as convergence of scalar series and functions, as well as concepts from algebra (polynomials) and linear algebra, both basic and less familiar ones (such as polynomial matrices), needed for the rest of the book.

1.2 Vector Spaces

As sets of mathematical objects can be highly abstract, imposing the axioms of a normed vector space is amongst the simplest ways to induce useful structure. Furthermore, we will see that images, audio signals and many other types of signals can be modeled and manipulated well using vector space models.

This section introduces the vector spaces formally, including inner products, norms and distances. We give pointers to reference texts in Further Reading.

1.2.1 Definition and Properties

A vector space is a set that has objects (vectors) that can be combined and manipulated in certain ways. Following our Euclidean-space intuition, we expect to be able to add vectors and to multiply a vector by a scalar. The formal definition of a vector space needs to specify the set of scalars and properties that are required of the addition and multiplication operations. Throughout the section we merge our definitions for real and complex scalars.

**Definition 1.1 (Vector space)**. A vector space over the set of real numbers \( \mathbb{R} \) (or the set of complex numbers \( \mathbb{C} \)) is a set of vectors, \( V \), together with *vector addition* and *scalar multiplication* operations. For any \( x, y, z \) in \( V \) and \( \alpha, \beta \) in \( \mathbb{R} \) (or \( \mathbb{C} \)), these operations must satisfy the following properties:
(i) **Commutativity:** \( x + y = y + x \).

(ii) **Associativity:** \((x + y) + z = x + (y + z)\) and \((\alpha \beta)x = \alpha(\beta x)\).

(iii) **Distributivity:** \(\alpha(x + y) = \alpha x + \alpha y\) and \((\alpha + \beta)x = \alpha x + \beta x\).

Furthermore, the following hold:

(iv) **Additive identity:** There exists an element \( 0 \) in \( V \), such that \( x + 0 = 0 + x = x \), for every \( x \) in \( V \).

(v) **Additive inverse:** For every \( x \) in \( V \), there exists a unique element \( -x \) in \( V \), such that \( x + (-x) = (-x) + x = 0 \).

(vi) **Multiplicative identity:** For every \( x \) in \( V \), \( 1 \cdot x = x \).

We have used the bold \( 0 \) to emphasize that the zero vector is different than the zero scalar. In later chapters we will drop this distinction.

In the standard finite-dimensional Euclidean spaces \( \mathbb{R}^N \) and \( \mathbb{C}^N \), the elements of \( V \) are \( N \)-tuples, and the addition and scalar multiplication operations are defined as

\[
x + y = [x_1 \ldots x_N]^T + [y_1 \ldots y_N]^T = [x_1 + y_1 \ldots x_N + y_N]^T,
\]

and

\[
\alpha x = \alpha [x_1 \ldots x_N]^T = [\alpha x_1 \ldots \alpha x_N]^T.
\]

It is easy to verify that the six properties in the definition above hold so these are vector spaces (see also the solved Exercise 1.1). Componentwise addition and multiplication can be used similarly to define vector spaces of one-sided sequences, two-sided sequences, matrices, etc. We will see many further examples later in the chapter.

**Example 1.1.** Fix any positive integer \( N \) and consider the real-valued polynomials of degree at most \((N - 1)\), \( x(t) = \sum_{i=0}^{N-1} t^i \). These form a vector space over the reals under the natural addition and multiplication operations:

\[
(x_1 + x_2)(t) = x_1(t) + x_2(t) \quad \text{and} \quad (\alpha x)(t) = \alpha x(t).
\]

Since each polynomial is specified by its coefficients, the polynomials combine exactly like vectors in \( \mathbb{R}^N \). However, interpreting them as polynomials leads to different concepts of closeness, size, etc. ■

**Definition 1.2 (Subspace).** A subset \( W \) of a vector space \( V \) is a subspace if:

(i) For all \( x \) and \( y \) in \( W \), \( x + y \) is in \( W \).

(ii) For all \( x \) in \( W \) and \( \alpha \) in \( \mathbb{R} \) (or \( \mathbb{C} \)), \( \alpha x \) is in \( W \).

One can show that a subspace \( W \) is itself a vector space over the same set of scalars as \( V \) and with the same vector addition and scalar multiplication operations as \( V \).

The definition of a subspace is suggestive of one way in which subspaces arise—by combining a finite number of vectors in \( V \). A set of all finite linear combinations of elements in that set is a span.
DEFINITION 1.3 (Span). The span of a set of vectors $S \subset V$ is a set of all finite linear combinations of vectors in $S$:

$$\operatorname{span}(S) = \left\{ \sum_{k=1}^{N} \alpha_k s_k \mid \alpha_k \in \mathbb{R} \text{ (or } \mathbb{C} \text{), } s_k \in S \text{ and } N \in \mathbb{N} \right\}.$$ 

Note that a span is always a subspace and that the sum has a finite number of terms even if the set $S$ is infinite.

EXAMPLE 1.2. The following two examples illustrate some of the concepts above:

(i) Proper subspaces (subspaces smaller than the entire space) arise in linear algebra when one looks at all matrix-vector products with a tall matrix. Specifically, consider $\{y = Ax \mid x \in \mathbb{R}^N\}$ with $A \in \mathbb{R}^{M \times N}$ and $M > N$. The conditions in the definition of subspace follow easily from properties of matrix multiplication. Note also that the subspace is the span of the columns of $A$.

(ii) In a vector space of real-valued functions of $\mathbb{R}$, the functions that are constant on intervals $[k - \frac{1}{2}, k + \frac{1}{2})$, $k \in \mathbb{Z}$, form a subspace. This is verified simply by noting that a sum of two functions each of which is constant on $[k - \frac{1}{2}, k + \frac{1}{2})$ is also constant on $[k - \frac{1}{2}, k + \frac{1}{2})$ and similarly, that multiplying by a scalar does not disturb this property.

Many different sets can have the same span, and it is of fundamental interest to find the smallest set with a particular span. This leads to the dimension of a vector space, which depends on concepts of linear independence and bases.

DEFINITION 1.4 (Linear Independence). The set $\{s_1, s_2, \ldots, s_N\} \subset V$ is called linearly independent when $\sum_{k=1}^{N} \alpha_k s_k = 0$ is true only if $\alpha_k = 0$ for all $k$. Otherwise, these vectors are linearly dependent. An infinite set of vectors is called linearly independent when every finite subset is linearly independent.

DEFINITION 1.5 (Basis). The set $S \subset V$ is a basis for $V$ when $V = \operatorname{span}(S)$ and $S$ is linearly independent. The number of elements in $S$ is the dimension of $V$ and may be infinite.

The definition of a basis requires the closure of the span for the cases where $S$ is infinite, because a span involves only a finite number of terms. We will frequently see closures of spans, and you can think of it as a span in which an infinite number of terms is allowed.

As an example, consider the vector space of doubly-infinite sequences $x = [\ldots, x_{-1}, x_0, x_1, \ldots]$ over $\mathbb{R}$, with componentwise addition and multiplication operations. For every $k \in \mathbb{Z}$, let $e_k$ be the sequence that is nonzero only at position $k$, $e_{kk} = 1$. It is clear that $\{e_k\}_{k \in \mathbb{Z}}$ is a linearly independent set. Intuitively, it would seem that this set is also a basis for our space; however, we need more machinery—specifically a way to define convergence—before $\sum_{k=-\infty}^{\infty} \alpha_k e_k$ can make sense. The intuition is not wrong: The set $\{e_k\}_{k \in \mathbb{Z}}$ is indeed a basis in the scenarios of most interest to us.
1.2.2 Inner Products and Norms

Our intuition from Euclidean space goes farther than just adding and multiplying. It has geometric notions of orientation and orthogonality as well as metric notions of length and distance. We now introduce these to our abstract spaces.

**Inner Products**

As visualized in Figure 1.2, an inner product is like a “signed length” of a projection of one vector onto another. It thus measures length along with relative orientation.

**Definition 1.6 (Inner product).** An inner product on a vector space \( V \) over \( \mathbb{R} \) (or \( \mathbb{C} \)) is a real-valued (or complex-valued) function \( \langle \cdot, \cdot \rangle \) defined on \( V \times V \) with the following properties for any \( x, y, z \in V \) and \( \alpha \in \mathbb{R} \) (or \( \mathbb{C} \)):

(i) \( \langle x + y, z \rangle = \langle x, z \rangle + \langle y, z \rangle \).

(ii) \( \langle \alpha x, y \rangle = \alpha \langle x, y \rangle \).

(iii) \( \langle x, y \rangle^* = \langle y, x \rangle \).

(iv) \( \langle x, x \rangle \geq 0 \), and \( \langle x, x \rangle = 0 \) if and only if \( x = 0 \).

Note that (ii) and (iii) imply \( \langle x, \alpha y \rangle = \alpha^* \langle x, y \rangle \). Putting this together with (i)-(iii), the inner product is said to be linear in the first argument and conjugate-linear in the second argument.\(^1\) The definition of an inner product in a vector space turns it into an **inner product space**, sometimes also called a pre-Hilbert space. When nonzero \( x \) and \( y \) in \( V \),

\[
\langle x, y \rangle = 0,
\]

\( x \) and \( y \) are called **orthogonal**.

The standard inner product for \( \mathbb{R}^N \) or \( \mathbb{C}^N \) is

\[
\langle x, y \rangle = \sum_{k=1}^{N} x_k y_k^*.
\]

We will use this frequently and without special mention, but note that this is not the only valid inner product for these spaces (see Exercise 1.1). Other examples of standard inner products are

\[
\langle x, y \rangle = \int_{-\infty}^{\infty} x(t)y^*(t) \, dt \quad (1.15)
\]

for the vector space of complex-valued functions over \( \mathbb{R} \), and

\[
\langle x, y \rangle = \sum_{k=-\infty}^{\infty} x_k y_k^* \quad (1.16)
\]

\(^1\)We have used the convention that is dominant in mathematics. Some authors, especially in physics, define inner products to be linear in the second argument and conjugate-linear in the first argument.
1.2. Vector Spaces

for sequences over \( \mathbb{Z} \).

**Example 1.3.** Consider vectors in \( \mathbb{C}^2 \) over scalars in \( \mathbb{C} \) with standard component-wise addition and multiplication.

(i) \( \langle x, y \rangle = x_1^*y_1 + x_2^*y_2 \) is not a valid inner product because it violates requirement (ii). For example, if \( x = y = [0 \ 1]^T \) and \( \alpha = j \), \( \langle \alpha x, y \rangle = -1 \) and thus, \( \langle \alpha x, y \rangle \neq \alpha \langle x, y \rangle \).

(ii) \( \langle x, y \rangle = x_1^*y_1 \) is not a valid inner product. It violates requirement (iv) because \( x = [0 \ 1]^T \) is nonzero yet yields \( \langle x, x \rangle = 0 \).

(iii) \( \langle x, y \rangle = x_1^*y_1 + 5x_2^*y_2 \) is a valid inner product.

In \( \mathbb{R}^N \), there is an expression analogous to the one given in (1.3) for \( \mathbb{R}^2 \):

\[
\langle x, y \rangle = \sum_{k=1}^{N} x_k y_k = \|x\| \|y\| \cos \theta,
\]

and is used to define the angle \( \theta \) between \( x \) and \( y \) in \( n \) dimensions.

**Norms**

A norm is a function that assigns a length to a vector. A vector space on which a norm is defined is called a *normed vector space*.

**Definition 1.7 (Norm).** Given a vector space \( V \) over \( \mathbb{R} \) (or \( \mathbb{C} \)), a norm \( \| \cdot \| \) is a function mapping \( V \) into \( \mathbb{R} \) with the following properties:

(i) \( \|x\| \geq 0 \) for any \( x \) in \( V \), and \( \|x\| = 0 \) if and only if \( x = 0 \).

(ii) \( \|\alpha x\| = |\alpha| \|x\| \) for any \( x \) in \( V \) and \( \alpha \) in \( \mathbb{R} \) (or \( \mathbb{C} \)).

(iii) \( \|x + y\| \leq \|x\| + \|y\| \) for any \( x, y \) in \( V \), with equality if and only if \( y = \alpha x \) or at least one of \( x, y \) is zero.

The inequality in Definition 1.7.(iii) is called the *triangle inequality* because it has the following geometric interpretation: the length of any side of a triangle is smaller than or equal to the sum of the lengths of the other two sides; equality occurs only when two sides are colinear, that is, when the triangle degenerates into a line segment. For example, if \( V = \mathbb{R} \) or \( V = \mathbb{C} \), then for any \( x, y \in \mathbb{R} \) or \( \mathbb{C} \), the triangle inequality becomes:

\[
|x + y| \leq |x| + |y|.
\]

An inner product can be used to define a norm, which we say is the norm *induced* by the inner product. As an example, for \( \mathbb{R}^N \) or \( \mathbb{C}^N \), the norm induced by the standard inner product is called the *Euclidean norm* and yields the conventional notion of length:

\[
\|x\| = \sqrt{\langle x, x \rangle} = \left( \sum_{k=1}^{N} |x_k|^2 \right)^{\frac{1}{2}}.
\]
In any inner product space we have the Cauchy-Schwarz inequality,

$$|\langle x, y \rangle| \leq \|x\| \|y\|,$$  \hspace{1cm} (1.19)

with equality if and only if $x = \alpha y$, or at least one of $x$, $y$ is zero. For example, for $L^2[a,b]$:

$$\left| \int_a^b x(t)y(t)\,dt \right| \leq \sqrt{\int_a^b |x(t)|^2\,dt \sqrt{\int_a^b |y(t)|^2\,dt}},$$  \hspace{1cm} (1.20)

and, for $y(t) = 1$, and squaring both sides, we get a common form:

$$\left| \int_a^b x(t)\,dt \right|^2 \leq \int_a^b |x(t)|^2\,dt.$$  \hspace{1cm} (1.21)

In any inner product space, we also have the parallelogram law:

$$\|x + y\|^2 + \|x - y\|^2 = 2(\|x\|^2 + \|y\|^2).$$  \hspace{1cm} (1.22)

Be careful that even though no inner product appears in the parallelogram law, it necessarily holds only in an inner product space. In fact, (1.22) is a necessary and sufficient condition for a norm to be induced by an inner product (see Exercise 1.2). Many useful norms are not compatible with any inner product.

For $\mathbb{R}^N$ or $\mathbb{C}^N$, any $p \in [1,\infty)$ yields a $p$-norm defined as

$$\|x\|_p = \left( \sum_{k=1}^N |x_k|^p \right)^{1/p}.$$  \hspace{1cm} (1.23)

For $p = 1$, this norm is called the taxicab or Manhattan norm because $\|x\|_1$ represents the driving distance from the origin to $x$ following a rectilinear street grid. For $p = 2$, we get our usual Euclidean square norm (induced by the standard inner product).

The natural extension of (1.23) to $p = \infty$ (see Exercise 1.3) is defined as

$$\|x\|_\infty = \max(|x_1|, |x_2|, \ldots, |x_N|).$$  \hspace{1cm} (1.24)

Using (1.23) for $p \in (0,1)$ does not give a norm, but it can still be a useful quantity. The failure to satisfy the requirements of a norm and an interpretation of (1.23) with $p \to 0$ are explored in Exercise 1.4.

All norms on finite-dimensional spaces are equivalent in the sense that any two norms bound each other within constant factors (see Exercise 1.5). This is a crude equivalence that leaves significant differences in which vectors are considered larger than others, and does not extend to infinite-dimensional spaces. Figure 1.6 shows this pictorially by showing the sets of vectors of norm 1 for different $p$-norms. All vectors ending on lines have a unit norm in the corresponding $p$-norm. For example, with the usual Euclidean norm, vectors of norm 1 fall on a circle; on the other hand, vectors with norm 1 in 1-norm fall on the diamond-shaped polygon.
Distance

Intuitively, the length of a vector can be thought of as the vector’s distance from the origin. This extends naturally to a distance induced by a norm, or a metric.

**Definition 1.8 (Distance).** In a normed vector space, the distance between vectors \(x\) and \(y\) is the norm of their difference:

\[
d(x, y) = \|x - y\|.
\]

Much as norms induced by inner products are a small fraction of all possible norms, distances induced by norms are a small fraction of all possible distances. However, we will have no need for more general concepts of distance. For the interested reader, Exercise 1.6 gives the axioms that a distance must satisfy and explores distances that are not induced by norms.

### 1.2.3 Some Standard Normed Vector Spaces

We have now already seen a few examples of norms. When the definition of a norm includes a series (an infinite sum) or an integral, it is important to limit our attention to vectors for which the sum or integral converges. This is essentially what defines many of the standard vector spaces that we will encounter throughout the book. In addition, with a norm defined through a series or integral, closure with respect to vector addition is not immediately obvious.
Chapter 1. From Euclid to Hilbert

For the examples presented in this section, remember that all summands and integrands are scalar; convergence of scalar series is reviewed in Appendix 1.A. Convergence in normed vector spaces is developed in Section 1.3.1.

Finite-Dimensional Spaces

The simplest vector spaces of interest in this book are the finite-dimensional spaces \( \mathbb{R}^N \) and \( \mathbb{C}^N \). As stated earlier, the vectors are \( n \)-tuples and addition and multiplication are componentwise. The (standard) inner product is defined as

\[
\langle x, y \rangle = \sum_{k=1}^{N} x_k y_k^*,
\]

and it induces the norm

\[
\|x\| = \sqrt{\langle x, x \rangle} = \left( \sum_{k=1}^{N} |x_k|^2 \right)^{1/2}.
\]

There are also other valid inner products (which induce norms) and norms that are not induced by any inner product, such as the one given by (1.23) for \( p \neq 2 \) as well as (1.24).

Since each sum in (1.25) has a finite number of terms, there is no doubt that the sums converge. Thus, we take as a vector space of interest “all real (or complex) \( n \)-tuples (with standard addition, multiplication, inner product, . . . ).” Note how this contrasts with some of the examples below.

\( \ell^p \) Spaces

For sequences with domain \( \mathbb{Z} \), define the \( \ell^p \)-norm, for \( p \in [1, \infty) \), as

\[
\|x\|_p = \left( \sum_{k \in \mathbb{Z}} |x_k|^p \right)^{1/p},
\]

and the \( \ell^\infty \)-norm as

\[
\|x\|_\infty = \sup_{k \in \mathbb{Z}} |x_k|.
\]

As these norms are not necessarily finite; to use them we must define our vector space to include only those sequences for which the norm is well defined:

**Definition 1.9 (\( \ell^p(\mathbb{Z}) \)).** For any \( p \in [1, \infty] \), the space \( \ell^p(\mathbb{Z}) \) is the vector space of all complex-valued sequences \( (x_i)_{i \in \mathbb{Z}} \) with finite \( \ell^p \)-norm.

Implicit in this definition is that the addition and multiplication operations are componentwise and that the norm used is the \( \ell^p \)-norm. Note also that the notation includes the domain \( \mathbb{Z} \). One can similarly write \( \ell^p(\mathbb{N}) \) for sequences starting at time 0 or \( \ell^p(\{1, 2, \ldots, N\}) \) for \( \mathbb{C}^N \) equipped with the \( p \)-norm, etc.
1.2. Vector Spaces

Example 1.4. Consider the sequence \( x \) given by

\[
x_n = \begin{cases} 
0, & n \leq 0; \\
1/n^a, & n > 0,
\end{cases}
\]

for some real number \( a \geq 0 \). Let us determine which of the spaces \( \ell^p(\mathbb{Z}) \) contain \( x \).

To check whether \( x \) is in \( \ell^p(\mathbb{Z}) \) for \( p \in [1, \infty) \), we need to determine whether

\[
\|x\|_p^p = \sum_{n=1}^{\infty} \left| \frac{1}{n^a} \right|^p = \sum_{n=1}^{\infty} \frac{1}{n^{pa}}
\]

converges. The necessary and sufficient condition for convergence is \( pa > 1 \), so we conclude that \( x \in \ell^p(\mathbb{Z}) \) for \( p > 1/a \) and \( a > 0 \). For \( a = 0 \), the above does not converge. To have \( x \in \ell^\infty(\mathbb{Z}) \), we must have \( (x_n) \) bounded, which occurs for all \( a \geq 0 \).

This example illustrates a simple inclusion property that is proven as Exercise 1.7:

\[
p < q \quad \text{implies} \quad \ell^p(\mathbb{Z}) \subset \ell^q(\mathbb{Z}). \tag{1.27}
\]

This can loosely be visualized with Figure 1.6: the larger the value of \( p \), the larger the set of vectors with a given norm. In particular, \( \ell^1(\mathbb{Z}) \subset \ell^2(\mathbb{Z}) \). In other words, if a sequence has a finite \( \ell^1 \)-norm, then it has a finite \( \ell^2 \)-norm. Beware that the opposite is not true; if a sequence has a finite \( \ell^2 \)-norm, it does not follow that is has a finite \( \ell^1 \)-norm. An example is the sequence \( x_n = (1/n) \), for \( n \in \mathbb{N}^+ \):

\[
\|x\|_2^2 = \sum_{n=1}^{\infty} \left| \frac{1}{n} \right|^2 = \frac{1}{6} \pi^2 \quad \text{converges, while} \quad \|x\|_1 = \sum_{n=1}^{\infty} \left| \frac{1}{n} \right| \quad \text{diverges.}
\]

Only in the case of \( p = 2 \) is an \( \ell^p \)-norm induced by an inner product. In that case, the inner product is

\[
\langle x, y \rangle = \sum_{n \in \mathbb{Z}} x_n y_n^*,
\]

and induces the norm \( \|x\| = \sqrt{\langle x, x \rangle} \). The space \( \ell^2(\mathbb{Z}) \) is by far the one most commonly used in this book, and is often referred to as the space of square-summable of finite-energy sequences.

\[ \mathcal{L}^p \] Spaces

For functions, define the \( \mathcal{L}^p \)-norm, for \( p \in [1, \infty) \), as\(^2\)

\[
\|x\|_p = \left( \int_{t \in \mathbb{R}} |x(t)|^p \, dt \right)^{1/p}, \tag{1.28a}
\]

\(^2\)Technically, we must require functions to be Lebesgue measurable, and all integrals should be seen as Lebesgue integrals; this is the historical reason for the letter \( \mathcal{L} \). We try to avoid unnecessary technicalities as much as possible and follow the creed [78]: “...if whether an airplane would fly or not depended on whether some function... was Lebesgue but not Riemann integrable, then I would not fly in it.”
and the $L^\infty$-norm as
\[ \|x\|_\infty = \text{ess sup}_{t \in \mathbb{R}} |x(t)|. \] (1.28b)

As with the $\ell^p$-norms, finiteness of the norm is an issue, so we define vector spaces with the requirement of finite $L^p$-norm:

**Definition 1.10 ($L^p(\mathbb{R})$).** For any $p \in [1, \infty]$, the space $L^p(\mathbb{R})$ is the vector space of all complex-valued functions $x(t)$, $t \in \mathbb{R}$, with finite $L^p$-norm.

Again, it is implicit that the vector addition and scalar multiplication are standard and that the norm used within the space is the $L^p$-norm. One can also use the same norms on different domains; for example, we can define the domain to be $[a, b]$ and use a finite $L^p$-norm on it to yield the space $L^p([a, b])$.

Only in the case of $p = 2$ is an $L^p$-norm induced by an inner product. The inner product on $L^2(\mathbb{R})$ is given by
\[ \langle x, y \rangle = \int_{t \in \mathbb{R}} x(t)y^*(t) \, dt, \] (1.29a)
and induces the norm
\[ \|x\| = \sqrt{\langle x, x \rangle} = \left( \int_{t \in \mathbb{R}} |x(t)|^2 \, dt \right)^{\frac{1}{2}}. \] (1.29b)

This space is infinite dimensional; for example, $e^{-t^2}$, $te^{-t^2}$, $t^2e^{-t^2}$, ... are linearly independent. The space $L^2(\mathbb{R})$ is often referred to as the space of square-integrable or finite-energy functions.

**$C^p([a, b])$ Spaces**

For any finite $a$ and $b$, the space $C([a, b])$ is defined as the inner product space of all complex-valued continuous functions on $[a, b]$, over the complex numbers, with pointwise addition and scalar multiplication. The inner product is
\[ \langle x, y \rangle = \int_{t \in [a, b]} x(t)y^*(t) \, dt, \] (1.30)
which induces the norm
\[ \|x\| = \left( \int_{t \in [a, b]} |x(t)|^2 \, dt \right)^{\frac{1}{2}}. \] (1.31)

$C([a, b])$ is also called $C^0([a, b])$ because $C^p([a, b])$ is defined similarly but with the additional requirement that the functions have $p$ continuous derivatives.

As an example, the set of polynomial functions forms a subspace of $C^p([a, b])$ for any $a, b \in \mathbb{R}$ and $p \in \mathbb{N}$. This is because the set is closed under the vector space operations and polynomials are infinitely differentiable.

A $C^p([a, b])$ space is very similar to $L^2([a, b])$. The distinction is completeness (which $C^p([a, b])$ can lack), which is a defining characteristic of Hilbert spaces.
1.3 Hilbert Spaces

We are going to do most of our work in Hilbert spaces. These are inner product spaces as we have seen in the previous section, with the additional requirement of completeness. Completeness is somewhat technical, and for a basic understanding it will suffice to have faith that we work in vector spaces of sequences and functions in which convergence makes sense. We will furthermore be mostly concerned with separable Hilbert spaces because these spaces have bases.

Section 1.3.1 explains the meaning of completeness in detail and briefly describes separability; it can reasonably be skipped. Sections 1.3.2–1.3.4 develop the critical concepts of orthogonality, linear operators and projections that make Hilbert spaces so powerful.

1.3.1 Completeness and Separability

Completeness

It is awkward to do analysis in $\mathbb{Q}$, the set of rational numbers, instead of in $\mathbb{R}$ because $\mathbb{Q}$ has infinitesimal gaps. For example, a sequence of rational numbers can converge to an irrational number. If we want the limit of the sequence to make sense, we need to work in $\mathbb{R}$, which is the completion of $\mathbb{Q}$. Working only in $\mathbb{Q}$, it would be harder to distinguish between sequences that converge to an irrational number and ones that do not converge at all—neither would have a limit point in the space.

Complete vector spaces are those in which sequences that intuitively ought to converge (the Cauchy sequences) have a limit in the space. We now define the relevant terms. Note that having a metric is necessary for defining convergence, and we limit our attention to metrics induced by norms.

**Definition 1.11 (Convergence in normed vector spaces).** Let $x_1, x_2, \ldots$ be a sequence in a normed vector space $V$. The sequence is said to converge to $x \in V$ if $\lim_{n \to \infty} \| x - x_n \| = 0$. In other words: Given any $\epsilon > 0$, there exists $N$ such that

$$\| x - x_n \| < \epsilon \quad \text{for all } n \geq N.$$ 

**Definition 1.12 (Cauchy sequence).** The sequence $x_1, x_2, \ldots$ in a normed vector space is called a Cauchy sequence if: Given any $\epsilon > 0$, there exists $N$ such that

$$\| x_n - x_m \| < \epsilon \quad \text{for all } n, m \geq N.$$ 

In these definitions, the elements of a convergent sequence eventually stay arbitrarily close to $x$. Similarly, the elements of a Cauchy sequence eventually stay arbitrarily close to each other. Thus it may be intuitive that a Cauchy sequence must converge; this is in fact true for real-valued sequences. However this is not true in all vector spaces, and it gives us important terminology:

**Definition 1.13 (Completeness, Banach space and Hilbert space).** In a normed vector space $V$, if every Cauchy sequence converges to a vector in $V$, then
V is said to be complete and is called a **Banach space**. A complete inner product space is called a **Hilbert space**.

**Example 1.5.** We consider the completeness of various vector spaces.

(i) Consider \( \mathbb{Q} \) as a normed vector space over the scalars \( \mathbb{Q} \), with ordinary addition and multiplication and with norm \( \|x\| = |x| \).\(^3\) This vector space is not complete because there exist rational sequences with irrational limits.

(ii) \( \mathbb{C} \) as a normed vector space over \( \mathbb{C} \) with ordinary addition and multiplication and with norm \( \|x\| = |x| \) is complete. This can be used to show that \( \mathbb{C}^N \) is complete under ordinary addition and multiplication and with any \( p \)-norm; see Exercise 1.8. Furthermore, the \( \ell^p(\mathbb{Z}) \) spaces are complete.

(iii) \( C([0, 1]) \) is not complete. It is easy to construct a Cauchy sequence of continuous functions whose limit is discontinuous (and hence not in \( C([0, 1]) \)). See Exercise 1.9.

We see that \( C^p([a, b]) \) spaces are not complete, but we have not addressed whether \( \mathcal{L}^p \) spaces are complete. They are in fact complete “by construction.” An \( \mathcal{L}^p \) space can either be understood to be complete because of Lebesgue measurability and the use of Lebesgue integration, or it can be taken as the completion of the space of continuous functions with finite \( \mathcal{L}^p \)-norm.

Hilbert spaces are the inner product spaces that are also complete, and hence Banach spaces. Completeness only makes sense in a normed vector space. These facts are shown as a Venn diagram in Figure 1.7 as well as Table 1.1. The figure shows the properties of several of the vector spaces that we have discussed or are introduced in exercises.

<table>
<thead>
<tr>
<th>Vector space (VS)</th>
<th>Properties</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Norm</td>
</tr>
<tr>
<td>Normed VS</td>
<td>✓</td>
</tr>
<tr>
<td>Inner product space</td>
<td>✓</td>
</tr>
<tr>
<td>Banach space</td>
<td>✓</td>
</tr>
<tr>
<td>Hilbert space</td>
<td>✓</td>
</tr>
</tbody>
</table>

**Table 1.1:** Relationships and properties of various types of vector spaces.

**Separability**

Separability is, for our purposes, even more technical than completeness. A space is separable if it contains a dense countable subset. For example, \( \mathbb{R} \) is separable.

\(^3\)Definition 1.1 restricted the set of scalars to \( \mathbb{R} \) or \( \mathbb{C} \) because those are the only sets of scalars of use later in the book. But the scalars can be any field.
1.3. Hilbert Spaces

Vector spaces

- Normed vector spaces
- Inner product spaces
- Banach spaces
- Hilbert spaces

- $\mathbb{Q}^N$
- $\mathbb{R}^N$
- $\mathbb{C}^N$
- $C([a,b])$
- $\ell^2(\mathbb{Z})$
- $\ell^1(\mathbb{Z})$
- $\ell^\infty(\mathbb{Z})$
- $L^\infty(\mathbb{R})$
- $L^1(\mathbb{R})$
- $\ell^0(\mathbb{Z})$
- $(C([a,b]), \| \cdot \|_\infty)$
- $(V,d)$

Figure 1.7: Relationships between types of vector spaces. Several examples of vector spaces are marked. (For $\mathbb{Q}^N$, $\mathbb{R}^N$ and $\mathbb{C}^N$ we assume the standard inner product. $(V,d)$ represents any vector space with the discrete metric as described in Exercise 1.6. $(C([a,b]), \| \cdot \|_\infty)$ represents $C([a,b])$ with the $L^2$-norm replaced by the $L^\infty$-norm. $\ell^0(\mathbb{Z})$ is described in Exercise 1.10.) These concepts are also summarized in Table 1.1.

since $\mathbb{Q}$ is dense in $\mathbb{R}$ and is countable. These topological properties are not of much interest here.

We are interested in separable Hilbert spaces because a Hilbert space contains a countable basis if and only if it is separable. The Hilbert spaces that we will use frequently (as marked in Figure 1.7) are all separable. Also, a closed subspace of a separable Hilbert space is separable, so it too contains a countable basis. Since the existence of bases is so important to our study, let us be explicit about the meaning of closed subspace:

**Definition 1.14 (Closed space).** A subspace $W$ of a Hilbert space $H$ is closed if it contains all limits of sequences of vectors in $W$.

Subspaces of finite-dimensional Hilbert spaces are always closed. Exercise 1.10 gives an example of a subspace of an infinite-dimensional Hilbert space that is not closed.

1.3.2 Orthogonality

Having dispensed with technicalities, we are now ready to develop operational Hilbert space machinery. An inner product endows a space with the geometric
properties of orientation, for example perpendicularity and angles. This is in contrast to having merely topological properties of closeness induced by a norm. In particular, an inner product being zero has special significance.

**Definition 1.15 (Orthogonality).** Vectors \( x \) and \( y \) in \( V \) are said to be **orthogonal** when \( \langle x, y \rangle = 0 \), written as \( x \perp y \).

Orthogonality between vectors extends to give several additional usages.

(i) \( x \perp S \): A vector \( x \) is said to be **orthogonal to a set of vectors** \( S \), when \( x \perp s \), for all \( s \in S \).

(ii) \( S_1 \perp S_2 \): Two sets \( S_1 \) and \( S_2 \) are **orthogonal** when every vector \( s_1 \in S_1 \) is orthogonal to \( S_2 \), \( s_1 \perp S_2 \).

(iii) A set of vectors \( S \) is called **orthogonal** when \( s_1 \perp s_2 \) for every \( s_1 \) and \( s_2 \) in \( S \), such that \( s_1 \neq s_2 \).

(iv) A set of vectors \( S \) is called **orthonormal** when it is orthogonal and \( \langle s, s \rangle = 1 \) for every \( s \) in \( S \).

(v) Given a subspace \( W \) of a vector space \( V \), the **orthogonal complement** of \( W \), denoted \( W^\perp \), is the set \( \{ x \in V \mid x \perp W \} \).

Vectors in an orthonormal set \( \{ s_k \}_{k \in \mathbb{Z}} \) are linearly independent, since \( 0 = \sum \alpha_k s_k \) implies that \( 0 = \langle \sum \alpha_k s_k, s_i \rangle = \sum \alpha_k \langle s_k, s_i \rangle = \alpha_i \) for any \( i \).

A fundamental fact that carries over from Euclidean geometry to general inner product spaces is the **Pythagorean theorem**:

\[
x \perp y \implies \| x + y \|^2 = \| x \|^2 + \| y \|^2.
\]

This follows from expanding \( \langle x + y, x + y \rangle \) into four terms and noting that \( \langle x, y \rangle = \langle y, x \rangle = 0 \).

### 1.3.3 Linear Operators

Linear operators generalize finite-dimensional matrix multiplications.

**Definition 1.16 (Linear operator).** A function \( A : H_1 \to H_2 \) is called a linear operator from \( H_1 \) to \( H_2 \), if for all \( x, y \) in \( H_1 \) and \( \alpha \) in \( \mathbb{R} \) (or \( \mathbb{C} \)):

(i) \( A(x + y) = Ax + Ay \).

(ii) \( A(\alpha x) = \alpha(Ax) \).

When \( H_1 = H_2 \), \( A \) is also called a **linear operator on** \( H_1 \).

Note the convention of writing \( Ax \) instead of \( A(x) \), just as is done for matrix multiplication. In fact, linear operators from \( \mathbb{C}^N \) to \( \mathbb{C}^M \) and matrices in \( \mathbb{C}^{M \times N} \) are exactly the same thing.

Many concepts from finite-dimensional linear algebra extend in rather obvious ways (see also Appendix 1.C). The **kernel** of a linear operator \( A : H_1 \to H_2 \) is the subspace of \( H_1 \) that maps to \( 0 \):

\[
\text{ker}(A) = \{ x \in H_1 \mid Ax = 0 \}.
\]
The range of a linear operator $A : H_1 \to H_2$ is

$$R(A) = \{ Ax \in H_2 \mid x \in H_1 \}.$$  

The concept of inverse becomes somewhat more subtle and is typically only defined for bounded operators.

As in linear algebra, we can define an eigenvector (also called eigenfunction, eigensequence) of a linear operator $A$:

**Definition 1.17 (Eigenvector of a linear operator).** An eigenvector $x$ of a linear operator $A$, is a nonzero vector $x \in H_1$ such that

$$Ax = \alpha x, \quad (1.32)$$

for some $\alpha \in \mathbb{C}$. The constant $\alpha$ is the corresponding eigenvalue.

**Definition 1.18 (Operator norm, bounded linear operator).** The operator norm of $A$, denoted by $\| A \|$, is defined as

$$\| A \| = \sup_{\| x \| = 1} \| Ax \|.$$ 

A linear operator is called bounded when its operator norm is finite.

It is implicit in the definition that $\| x \|$ uses the norm of $H_1$ and $\| Ax \|$ uses the norm of $H_2$. An important property of bounded linear operators is that they are continuous, that is, if $x_n \to x$ then $Ax_n \to Ax$.

A bounded linear operator $A : H_1 \to H_2$ is called invertible if there exists a bounded linear operator $B : H_2 \to H_1$ such that

$$BAx = x, \quad \text{for every } x \in H_1, \quad (1.33a)$$

$$ABy = y, \quad \text{for every } y \in H_2. \quad (1.33b)$$

When such a $B$ exists, it is unique, denoted by $A^{-1}$, and is called the inverse of $A$. As in finite-dimensional linear algebra, $B$ is called a left inverse of $A$ when (1.33a) holds and $B$ is called a right inverse of $A$ when (1.33b) holds.

**Example 1.6.** We now give some examples of linear operators:

(i) Ordinary matrix multiplication by the matrix

$$A = \begin{bmatrix} 3 & 1 \\ 1 & 3 \end{bmatrix}$$

defines a linear operator on $\mathbb{R}^2$, $A : \mathbb{R}^2 \to \mathbb{R}^2$. It is bounded, and its operator norm is 4. We show here how to obtain the norm of $A$ by direct computation.
(we could also have used the relationship between eigenvalues, singular values, and the operator norm, explored in Exercise 1.11):

\[ \|A\| = \sup_{\|x\| = 1} \|Ax\| = \sup \left\| \begin{bmatrix} 3 & 1 \\ 1 & 3 \end{bmatrix} \begin{bmatrix} \cos \theta \\ \sin \theta \end{bmatrix} \right\| = \sup \left\| \begin{bmatrix} 3 \cos \theta + \sin \theta \\ \cos \theta + 3 \sin \theta \end{bmatrix} \right\| = \sup \sqrt{(3 \cos \theta + \sin \theta)^2 + (\cos \theta + 3 \sin \theta)^2} = \sup \sqrt{10 \cos^2 \theta + 10 \sin^2 \theta + 12 \sin \theta \cos \theta} = \sup \sqrt{10 + 6 \sin 2\theta} = 4. \]

Its kernel is only the vector 0, its range is all of \( \mathbb{R}^2 \), and

\[ A^{-1} = \begin{bmatrix} 3/8 & -1/8 \\ -1/8 & 3/8 \end{bmatrix}. \]

(ii) Ordinary matrix multiplication by the matrix

\[ A = \begin{bmatrix} 1 & j & 0 \\ 1 & 0 & j \end{bmatrix} \]

defines a linear operator from \( \mathbb{C}^3 \) to \( \mathbb{C}^2 \). It is bounded, and its operator norm is \( \sqrt{3} \). Its kernel is \( \{ [x_1 \ j x_1 \ j x_1]^T \} \), its range is all of \( \mathbb{C}^2 \), and it is not invertible. (There exists \( B \) satisfying (1.33b), but no \( B \) can satisfy (1.33a).)

(iii) For some fixed complex-valued sequence \( (s_k)_{k \in \mathbb{Z}} \), consider the componentwise multiplication

\[ (Ax)_k = s_k x_k \tag{1.34} \]

as a linear operator on \( \ell^2(\mathbb{Z}) \), It is easy to check that the conditions (i) and (ii) in Definition 1.16 are satisfied, but we must constrain \( s \) to ensure that the result is in \( \ell^2(\mathbb{Z}) \). For example, \( \|s\|_\infty = M < \infty \) ensures that \( Ax \) is in \( \ell^2(\mathbb{Z}) \) for any \( x \) in \( \ell^2(\mathbb{Z}) \). Furthermore, the operator is bounded and \( \|A\| = M \). The operator is invertible when \( \inf_k |s_k| > 0 \).

(iv) The operator

\[ (Ax)_k = \int_{k-1/2}^{k+1/2} x(t) \, dt \tag{1.35} \]

takes local averages of the function \( x(t) \) to yield a sequence \( (Ax)_k \). (This operation is depicted in Figure 1.9, and is a form of sampling, covered in detail in Chapter 4.) Let us check that \( A \) is a linear operator from \( \mathcal{L}^2(\mathbb{R}) \) to \( \ell^2(\mathbb{Z}) \). It clearly satisfies conditions (i) and (ii) in Definition 1.16; we just need to be sure that the result is in \( \ell^2(\mathbb{Z}) \). Suppose \( x \) is in \( \mathcal{L}^2(\mathbb{R}) \). Then

\[
\sum_{k=-\infty}^{\infty} |(Ax)_k|^2 = \sum_{k=-\infty}^{\infty} \left| \int_{k-1/2}^{k+1/2} x(t) \, dt \right|^2 \\
\leq \sum_{k=-\infty}^{\infty} \int_{k-1/2}^{k+1/2} |x(t)|^2 \, dt = \int_{-\infty}^{\infty} |x(t)|^2 \, dt = \|x\|_{\mathcal{L}^2}^2,
\]
where the inequality follows from the version of Cauchy-Schwarz inequality in (1.21). Thus, $Ax$ is indeed in $\ell^2(\mathbb{Z})$.

Finite-dimensional linear algebra has many uses for transposes and conjugate transposes. The conjugate transpose (or Hermitian transpose) is generalized by the adjoint of an operator.

**Definition 1.19 (Adjoint and Self-Adjoint Operators).** The linear operator $A^* : H_2 \to H_1$ is called the adjoint of the linear operator $A : H_1 \to H_2$ when

$$\langle Ax, y \rangle_{H_2} = \langle x, A^* y \rangle_{H_1}, \quad \text{for every } x \in H_1 \text{ and } y \in H_2. \quad (1.36)$$

If $A = A^*$, then $A$ is called a self-adjoint or Hermitian operator.

An important and nontrivial result is that every bounded linear operator has a unique adjoint. Also, as we would expect from the notation being common with linear algebra, the operators $AA^*$ and $A^*A$ are always self-adjoint. The adjoint satisfies $\|A\| = \|A^*\|$. While all these facts can be found in Exercise 1.12, the proof of the last one is rather advanced and not included.

**Example 1.7.** We now give some examples of adjoint operators:

(i) We derive the adjoint of $A : \mathbb{C}^N \to \mathbb{C}^M$. The adjoint operator is a mapping from $\mathbb{C}^M$ to $\mathbb{C}^N$ and hence can be represented by $B \in \mathbb{C}^{N \times M}$. The $\mathbb{C}^N$ and $\mathbb{C}^M$ inner products can both be written as $\langle x, y \rangle = y^* x$, where $^*$ represents the Hermitian transpose. Thus (1.36) is equivalent to

$$\langle Ax, y \rangle_{\mathbb{C}^M} = y^*(Ax) = (By)^*x = \langle x, By \rangle = \langle x, A^*y \rangle_{\mathbb{C}^N},$$

for every $x \in \mathbb{C}^N$ and $y \in \mathbb{C}^M$, and thus, $B = A^*$.

(ii) We derive the adjoint of the operator (1.34). Writing out the defining relation for the adjoint (1.36) using the $\ell^2(\mathbb{Z})$ inner product gives

$$\langle Ax, y \rangle_{\ell^2} = \sum_{k \in \mathbb{Z}} (s_k x_k y^*_k) = \sum_{k \in \mathbb{Z}} x_k (s_k y_k^*) = \sum_{k \in \mathbb{Z}} x_k (A^* y^*_k) = \langle x, A^* y \rangle_{\ell^2}.$$  

Thus, the componentwise multiplication $(A^* y)_k = s_k y_k$ defines the adjoint.

(iii) We derive the adjoint of the operator (1.35). Let us keep sight of the goal of finding an operator $A^*$ such that $\langle Ax, y \rangle_{\ell^2} = \langle x, A^* y \rangle_{\ell^2}$ while we manipulate $\langle Ax, y \rangle_{\ell^2}$:

$$\langle Ax, y \rangle_{\ell^2} = \sum_{k = -\infty}^{\infty} \left( \int_{k-1/2}^{k+1/2} x(t) \, dt \right) y^*_k$$

$$= \sum_{k = -\infty}^{\infty} \int_{k-1/2}^{k+1/2} x(t) y^*_k \, dt \overset{(a)}{=} \int_{-\infty}^{\infty} x(t) (A^* y)^*(t) \, dt = \langle x, A^* y \rangle_{\ell^2},$$
where (a) uses the fact that $y_k$ does not depend on $t$. The key step is (b), where we define a piecewise-constant function $\tilde{x}(t) = A^*y(t)$ such that $\tilde{x}(t) = y_k$ for $t \in [k-\frac{1}{2}, k+\frac{1}{2})$. This step is inspired by wanting to reach an expression using the $L^2$ inner product. We have thus shown that the adjoint maps $y \in \ell^2(\mathbb{Z})$ to $\tilde{x} = A^*y \in L^2(\mathbb{R})$ as defined above (see Figure 1.9).

1.3.4 Projection Operators

Many of the linear operators that we define in later chapters are projection operators. In fact, because of the projection theorem which we will see shortly, projection operators do the important job of finding best approximations.

**Definition 1.20 (Projection operator).** An operator $P$ is called idempotent if $P^2 = P$. A linear operator that is idempotent is a projection operator.

**Definition 1.21 (Orthogonal projection operator).** A projection operator that is self-adjoint is an orthogonal projection operator.

One common practice is to drop “orthogonal” from “orthogonal projection operator,” in which case an idempotent linear operator that is not self-adjoint is an oblique projection operator. We will use the terminology in Definitions 1.20 and 1.21, to emphasize the importance of orthogonality and to avoid having to write “possibly-oblique projection.”

**Example 1.8.** Let us consider explicitly the orthogonal projection operators with one-dimensional ranges. This was discussed informally in Section 1.1.

Given a vector $\varphi$ of unit norm, $\|\varphi\| = 1$,

$$P_{x} = \langle x, \varphi \rangle \varphi,$$

defines an orthogonal projection operator with range equal to all the scalar multiples of $\varphi$. The scalar $\langle x, \varphi \rangle$ is the length of the projected vector, while $\varphi$ gives its direction. We can easily check $P$ is a projection operator by verifying idempotency:

$$P^2_{x} = \langle (\langle x, \varphi \rangle \varphi, \varphi \rangle \varphi = \langle x, \varphi \rangle \langle \varphi, \varphi \rangle \varphi = \langle x, \varphi \rangle \varphi = P_{x},$$

where (a) follows from linearity of the inner product in the first argument; and (b) follows from $\langle \varphi, \varphi \rangle = 1$. For $P$ to be an orthogonal projection operator it should also be self adjoint. Given $x, y$ in $H$,

$$\langle P_{x}, y \rangle = \langle (x, \varphi) \varphi, y \rangle = \langle x, \varphi \rangle \langle \varphi, y \rangle = \langle \varphi, y \rangle \langle x, \varphi \rangle = \langle y, \varphi \rangle (x, \varphi) = (x, P_{y}) h,$$

where (a) follows from linearity of the inner product in the first argument; (b) from conjugate symmetry of the inner product; and (c) from conjugate linearity of the
1.3. Hilbert Spaces

inner product in the second argument. Comparing the above to the definition of the adjoint, we have shown that \( P = P^* \).

From linear algebra, we are familiar with \( P = P^* \) as a condition for \( P \) being orthogonal (or unitary). The projection theorem establishes a geometric reason for referring to such operators as orthogonal.

**Theorem 1.1 (Projection theorem [111]).** Let \( W \) be a closed subspace of the Hilbert space \( H \). Corresponding to any vector \( x \in H \), there exists a unique vector \( w_0 \in W \) such that \( \|x - w_0\| \leq \|x - w\| \) for all \( w \in W \). A necessary and sufficient condition for \( w_0 \) to be the unique minimizing vector is that \( x - w_0 \perp W \). The operator \( P_W \) mapping \( x \) into \( w_0 \) is an orthogonal projection operator.

**Proof.** The theorem has three parts: (a) **existence** of \( w_0 \), (b) **uniqueness** of \( w_0 \), and (c) **orthogonality** \( x - w_0 \perp W \). We prove existence last since it is the most technical and is the only part that requires completeness of the space. (Uniqueness and orthogonality hold with \( H \) replaced by any inner product space and \( W \) replaced by any subspace.)

(c) **Orthogonality:** To prove the desired orthogonality, suppose that \( w_0 \) minimizes \( \|x - w\| \) but that \( x - w_0 \not\perp W \). Then there exists a unit vector \( w' \in W \) such that \( \langle x - w_0, w' \rangle = \delta \neq 0 \). Let \( w_1 = w_0 + \delta w' \) and note that \( w_1 \) is in \( W \) since \( W \) is a subspace. The calculation

\[
\|x - w_1\|^2 = \|x - w_0 - \delta w'\|^2 = \|x - w_0\|^2 - \langle x - w_0, \delta w \rangle - \langle \delta w, x - w_0 \rangle + \|\delta w\|^2
\]

then shows that \( w_0 \) is not the minimizing vector. This contradiction implies that \( x - w_0 \perp W \) must hold.

(b) **Uniqueness:** Suppose \( x - w_0 \perp W \). For any \( w \in W \),

\[
\|x - w\|^2 = \|x - w_0 + w_0 - w\|^2 \leq \|x - w_0\|^2 + \|w_0 - w\|^2,
\]

where \( x - w_0 \perp w_0 - w \in W \) allows an application of the Pythagorean theorem in (a). This shows \( \|x - w\| > \|x - w_0\| \) for any \( w \neq w_0 \).

(a) **Existence:** We finally show existence of a minimizing \( w_0 \). If \( x \) is in \( W \), then \( w_0 = x \) achieves the minimum so there is no question of existence. We thus restrict our attention to \( x \not\in W \). Let \( \delta = \inf_{w \in W} \|x - w\| \). Then there exists a sequence of vectors \( w_1, w_2, \ldots \) in \( W \) such that \( \|x - w_i\| \to \delta \); the challenge is to show that the infimum is achieved by some \( w_0 \in W \). We do this by showing that \( \{w_i\}_{i \geq 1} \) is a Cauchy sequence and thus converges, within the closed subspace \( W \), to the desired \( w_0 \).

By applying the parallelogram law to \( x - w_j \) and \( w_i - x \),

\[
\|(x - w_j) + (w_i - x)\|^2 + \|(x - w_j) - (w_i - x)\|^2 = 2\|x - w_j\|^2 + 2\|w_i - x\|^2.
\]
Figure 1.8: Illustration of the projection theorem. The vector in subspace $W$ closest to vector $x$ is obtained via an orthogonal projection $P_W$. The difference vector $(x - w_0)$ is orthogonal to $W$.

Canceling $x$ in the first term and moving the second term to the right yields

$$\|w_i - w_j\|^2 = 2\|x - w_j\|^2 + 2\|w_i - x\|^2 - 4\|x - \frac{1}{2}(w_i + w_j)\|^2. \quad (1.38)$$

Now since $W$ is a subspace, $\frac{1}{2}(w_i + w_j)$ is in $W$. Thus by the definition of $\delta$ we have $\|x - \frac{1}{2}(w_i + w_j)\| \geq \delta$. Substituting in (1.38) and using the nonnegativity of the norm,

$$0 \leq \|w_i - w_j\|^2 \leq 2\|x - w_j\|^2 + 2\|w_i - x\|^2 - 4\delta^2.$$

With the convergence of $\|x - w_j\|^2$ and $\|w_i - x\|^2$ to $\delta^2$, we conclude that $\{w_i\}_{i \geq 1}$ is a Cauchy sequence. Now since $W$ is a closed subspace of a complete space, $\{w_i\}_{i \geq 1}$ converges to $w_0 \in W$. The continuity of the norm implies $\|x - w_0\| = \delta$.

The projection theorem shows that to get from a vector to the nearest point in a subspace, the direction of travel should be orthogonal to the subspace. This is shown schematically in Figure 1.8. A related result shows that one can demonstrate the geometric properties of an orthogonal projection operator using Definitions 1.20, 1.21. Moreover, one can start from the geometric definition of the orthogonal projection operator and prove it is then idempotent and self-adjoint.

**Theorem 1.2.** Given is an operator $P$ and $x, y$ in a Hilbert space $H$. The operator $P$ satisfies

$$\langle x - Px, Py \rangle = 0,$$  \hspace{1cm} (1.39)

if and only if it is idempotent and self-adjoint.

**Proof.** We first prove sufficiency by assuming (1.39) and proving that an operator $P$ satisfying it must necessarily be idempotent and self adjoint. Take an arbitrary $z \in H$ and let $x = Pz$ and $y = z - Pz$. Substituting these two into (1.39) we get

$$\langle Pz - P(Pz), P(z - Pz) \rangle = \|(P - P^2)z\|^2 = 0,$$

which implies $Pz = P^2z$. Since $z$ is arbitrary, we have $P = P^2$. 

\[\]
1.3. Hilbert Spaces

Given any \( x, y \in H \), we can use (1.39) to get

\[
\langle x, Py \rangle = \langle Px, y \rangle = \langle Px, y \rangle.
\]

It then follows that

\[
\langle P^* x, y \rangle = \langle x, Py \rangle = \langle Px, y \rangle \Rightarrow \langle P^* x - Px, y \rangle = 0.
\]

By choosing \( y = P^* x - Px \), we have that \( \| P^* x - Px \| = 0 \) for all \( x \), and thus \( P^* = P \).

Necessity, and thus the desired orthogonality, is established via

\[
\langle x - Px, Py \rangle = \langle x, Py \rangle - \langle Px, Py \rangle \overset{(a)}{=} \langle x, Py \rangle - \langle x, P^* Py \rangle \\
\overset{(b)}{=} \langle x, Py \rangle - \langle x, P^2 y \rangle \overset{(c)}{=} \langle x, Py \rangle - \langle x, Py \rangle = 0,
\]

where (a) uses the definition of the adjoint operator; (b) uses self-adjointness of the orthogonal projection operator, \( P^* = P \); and (c) uses idempotency of the projection operator, \( P^2 = P \).

This theorem shows that an orthogonal projection operator always creates an orthogonal decomposition between its range and a difference space.

The final theorem of the section establishes important connections between inverses, adjoints and projections. In the case of finite-dimensional spaces, the matrix manipulations needed to prove the theorem are quite simple.

**Theorem 1.3.** Let \( A : H_1 \to H_2 \) and \( B : H_2 \to H_1 \) be linear operators. If \( A \) is a left inverse of \( B \), then \( BA \) is a projection operator. Furthermore, if \( B = A^* \), then \( BA = A^* A \) is an orthogonal projection operator.

**Example 1.9.** Let us draw together some earlier results to give an example of an orthogonal projection operator on \( L^2(\mathbb{R}) \). Let \( A : L^2(\mathbb{R}) \to \ell^2(\mathbb{Z}) \) be the local averaging operator (1.35) and let \( A^* : \ell^2(\mathbb{Z}) \to L^2(\mathbb{R}) \) be its adjoint, as derived in Example 1.7. If we verify \( A \) is a left inverse of \( A^* \), we will have that \( A^* A \) is an orthogonal projection operator.

Remembering to compose from right-to-left, \( AA^* \) starts with a sequence, creates a piecewise-constant function, and then recovers the original sequence. So \( AA^* \) is indeed an identity operator. One conclusion to draw by combining the projection theorem with \( A^* A \) being an orthogonal projection operator is the following: Given a function \( x(t) \in L^2(\mathbb{R}) \), the function in the subspace of piecewise constant functions \( A^* \ell^2(\mathbb{Z}) \) that is closest in \( L^2 \)-norm, is the one obtained by replacing \( x(t) \), \( t \in [k - \frac{1}{2}, k + \frac{1}{2}] \), by its local average \( \int_{k - \frac{1}{2}}^{k + \frac{1}{2}} x(t) \, dt \). The process of going between a continuous-time function to a sequence of samples to a piecewise-constant function is depicted in Figure 1.9. ■
Figure 1.9: Illustration of an orthogonal projection operator on $L^2(\mathbb{R})$. We start with a function $x(t) \in L^2(\mathbb{R})$, and map it to a sequence $y_k \in l^2(\mathbb{Z})$ using a linear operator $A$ given in Example 1.7. It is clear that we cannot invert this operation; the best we can do is to use its adjoint operator and reconstruct a piecewise-constant function $\tilde{x}(t) \in L^2(\mathbb{R})$—an orthogonal projection of $x(t)$. This projection is given by $\tilde{x} = A^*Ax$. (TBD: Create figure.)

1.4 Bases and Frames

Our initial flurry of definitions in Section 1.2.1 included the definition of a basis, but we have not yet made much use of bases. A basis is used to represent vectors as linear combinations of basis elements, and the choice of a basis has dramatic consequences. In fact, the variety of bases for sequences $l^2(\mathbb{Z})$ and functions $L^2(\mathbb{R})$ is at the heart of this book. We will see how the computational complexity, stability, and approximation accuracy of signal expansions depend on the properties of the bases that are employed.

This section develops general properties of bases with an emphasis on representing vectors in Hilbert spaces using bases. This is to bridge the gap from simple finite-dimensional concepts to the spaces of sequences and functions that we will encounter in later chapters. We start with the important special case of orthonormal bases and conclude with representations that use sets that are actually not bases because they are not linearly independent.

1.4.1 Orthonormal Bases

Recall from Section 1.2.1 that a set of vectors $S$ in a vector space $V$ is a basis if it is linearly independent and $\text{span}(S) = V$. A set of orthogonal vectors is always linearly independent, so a set of orthogonal vectors in a Hilbert space is a basis when the closure of its span is the whole space. An orthonormal basis (ONB) is a basis of orthogonal, unit-norm vectors.
In this section, we consider finite, $\Phi = \{\varphi_1, \varphi_2, \ldots, \varphi_N\}$, or countable sets of vectors, $\Phi = \{\varphi_1, \varphi_2, \ldots\}$. Orthonormality of these sets can be expressed as

$$\langle \varphi_k, \varphi_i \rangle = \delta_{k-i}. \quad (1.40)$$

The completeness condition for $\Phi$ to be an ONB is that every vector in the space can be expressed as a linear combination of the vectors from $\Phi$:

$$x = \sum_k \alpha_k \varphi_k. \quad (1.41)$$

The coefficients $\alpha_k$ are called the expansion coefficients\(^4\) of $x$ with respect to the basis $\{\varphi_k\}$, and are given by

$$\alpha_k = \langle x, \varphi_k \rangle. \quad (1.42)$$

The expansion coefficient $\alpha_k$ is the (signed) length of the projection of $x$ onto the subspace spanned by $\varphi_k$. The projection itself is given by $\alpha_k \varphi_k = \langle x, \varphi_k \rangle \varphi_k$.

Since $\{\varphi_k\}_{k \in I}$ is an orthonormal set, where $I$ is any index set, then an orthogonal projection onto the subspace spanned by $\{\varphi_k\}_{k \in I}$ is

$$Px = \sum_{k \in I} \langle x, \varphi_k \rangle \varphi_k, \quad (1.43)$$

the sum of projections onto individual subspaces. Beware that this is not true when $\{\varphi_k\}_{k \in I}$ is not orthonormal. We will discuss this in more detail later.

**Bessel’s Inequality and Parseval’s Equality**

**Proposition 1.4.** Given an orthonormal set $\{\varphi_k\}$ in $H$, Bessel’s inequality holds:

$$\|x\|^2 \geq \sum_k |\langle x, \varphi_k \rangle|^2, \quad \text{for every } x \in H. \quad (1.44)$$

Equality is achieved if and only if $\{\varphi_k\}$ is complete in $H$, yielding Parseval’s equality:

$$\|x\|^2 = \sum_k |\langle x, \varphi_k \rangle|^2, \quad \text{for every } x \in H. \quad (1.45)$$

Figure 1.10 visualizes the above result through a simple example in $\mathbb{R}^3$. If we consider an orthonormal set of two vectors (not a basis) given by $\varphi_1 = e_1 = [1 \ 0 \ 0]^T$ and $\varphi_2 = e_2 = [0 \ 1 \ 0]^T$, then the length of our vector $x$ is at least as large as the projection onto the $(\varphi_1, \varphi_2)$-plane, $x_{12}$:

$$\|x\|^2 \geq \|x_{12}\|^2 = |\langle x, \varphi_1 \rangle|^2 + |\langle x, \varphi_2 \rangle|^2.$$
If, however, we now add \( \varphi_3 = e_3 = [0 \ 0 \ 1]^T \) to the mix, the length of the vector is exactly the sum of all three projections. This is the Parseval’s equality:

\[
\|x\|^2 = \sum_{k=1}^{3} |\langle x, \varphi_k \rangle|^2.
\]

When Is an Orthonormal Set a Basis?

In finite dimensions (\( \mathbb{R}^N \) or \( \mathbb{C}^N \)), having an orthonormal set of size \( n \) is sufficient to have an ONB. In infinite dimensions this is more delicate; it is not sufficient to have an infinite orthonormal set. The following theorem gives several equivalent statements which permit us to check if an orthonormal system is also a basis (for a proof, see Exercise 1.13):

**Theorem 1.5.** Given an orthonormal set \( \{\varphi_k\}_{k \in I} \) in \( H \), the following are equivalent:

(i) The set of vectors \( \{\varphi_k\}_{k \in I} \) is an ONB for \( H \).

(ii) If \( \langle x, \varphi_k \rangle = 0 \) for all \( k \), then \( x = 0 \).

(iii) The span(\( \{\varphi_k\}_{k \in I} \)) is dense in \( H \), that is, every vector in \( H \) is a limit of a sequence of vectors in span(\( \{\varphi_k\}_{k \in I} \)).

(iv) For every \( x \) in \( H \), the *Parseval’s equality* holds:

\[
\|x\|^2 = \sum_{k \in I} |\langle x, \varphi_k \rangle|^2.
\]  \hspace{1cm} (1.46a)

(v) For every \( x_1 \) and \( x_2 \) in \( H \), the *generalized Parseval’s equality* holds:

\[
\langle x_1, x_2 \rangle = \sum_{k \in I} \langle x_1, \varphi_k \rangle \langle x_2, \varphi_k \rangle^*.
\]  \hspace{1cm} (1.46b)
**Orthogonal Projection and Least Squares Approximation**

We have touched upon orthogonal projections already and will discuss now their power in approximating vectors. (We have already seen how to do this in the real plane in Section 1.1.)

Our task is to approximate a vector \(x\) from a Hilbert space \(H\), by a vector \(\hat{x}\) lying in a closed subspace \(W\). We assume that \(H\) is separable, thus, \(W\) contains an ONB \(\{\varphi_k\}_{k \in I}\).

**Theorem 1.6.** Given a separable Hilbert space \(H\), a closed subspace \(W\) with an ONB \(\{\varphi_k\}_{k \in I}\) and a vector \(x \in H\), the orthogonal projection of \(x\) onto \(W\) is
\[
\hat{x} = \sum_k \langle x, \varphi_k \rangle \varphi_k.
\]
(1.47)

The difference vector \(d = x - \hat{x}\) satisfies
\[
d \perp W,
\]
(1.48)
and, in particular, \(d \perp \hat{x}\), as well as
\[
\|x\|^2 = \|\hat{x}\|^2 + \|d\|^2.
\]

This approximation is best in the least squares sense, that is,
\[
\hat{x} = \arg \min_{y \in W} \|x - y\|,
\]
(1.49)
is attained for \(y = \sum_k \alpha_k \varphi_k\) with
\[
\alpha_k = \langle x, \varphi_k \rangle
\]
the expansion coefficients.

In other words, the best approximation is our \(\hat{x}\) previously defined in (1.47). An immediate consequence of this result is the successive approximation property of orthogonal expansions.

**Theorem 1.7 (Successive Approximation Property).** Given \(\hat{x}^{(k-1)}\), the best approximation of \(x\) on the subspace spanned by \(\{\varphi_1, \varphi_2, \ldots, \varphi_{k-1}\}\) with \(\{\alpha_1, \alpha_2, \ldots, \alpha_{k-1}\}\), \(\alpha_i = \langle x, \varphi_i \rangle\), the approximation \(\hat{x}^{(k)}\) is given by
\[
\hat{x}^{(k)} = \hat{x}^{(k-1)} + \underbrace{\langle x, \varphi_k \rangle}_{\alpha_k} \varphi_k,
\]
that is, the new approximation is the sum of the previous best approximation plus the projection along the added vector \(\varphi_k\).

Figure 1.11(a) gives an example in \(\mathbb{R}^2\) for an ONB: \(\hat{x}^{(2)} = \hat{x}^{(1)} + \langle x, \varphi_2 \rangle \varphi_2\). For nonorthogonal bases, the successive approximation property does not hold; When calculating the approximation \(\hat{x}^{(k)}\), one cannot simply add one term to the previous approximation, but has to recalculate the whole approximation. This is illustrated in Figure 1.11(b): The first approximation is \(\hat{x}^{(1)} = \langle x, \varphi_1 \rangle \varphi_1\). However, if we wish to obtain \(\hat{x}^{(2)}\), we cannot simply add a term to \(\hat{x}^{(1)}\); instead, we must recalculate the entire approximation \(\hat{x}^{(2)} = \langle x, \varphi_1 \rangle \varphi_1 + \langle x, \varphi_2 \rangle \varphi_2\).
Figure 1.11: Expansion in orthogonal and biorthogonal bases. (a) Orthogonal case: The successive approximation property holds. (b) Biorthogonal case: The first approximation cannot be used in the full expansion.

1.4.2 General Bases

We are now going to relax the constraint of orthogonality and see what happens. The reasons for doing that are numerous, and the most obvious one is that we have more freedom in choosing our basis vectors. We saw an example of that in Section 1.1, which we now generalize.

Definition 1.22 (Biorthogonal bases). A system \( \{ \varphi_k, \tilde{\varphi}_k \} \) constitutes a pair of biorthogonal bases of a Hilbert space \( H \) if:

(i) For all \( k, i \) in \( \mathbb{Z} \),
\[
\langle \varphi_k, \tilde{\varphi}_i \rangle = \delta_{k-i}. \tag{1.50a}
\]

(ii) There exist strictly positive constants \( A, B \), such that, for all \( x \) in \( H \),
\[
A \| x \|^2 \leq \sum_k |\langle x, \varphi_k \rangle|^2 \leq B \| x \|^2, \tag{1.50b}
\]
\[
\frac{1}{B} \| x \|^2 \leq \sum_k |\langle x, \tilde{\varphi}_k \rangle|^2 \leq \frac{1}{A} \| x \|^2. \tag{1.50c}
\]

Compare these inequalities with equalities in (1.46a) in the orthonormal case. Bases which satisfy (1.50b) or (1.50c) are called Riesz bases, and are interchangeable. Then, the signal expansion formula becomes
\[
x = \sum_k \langle x, \tilde{\varphi}_k \rangle \varphi_k = \sum_k \langle x, \varphi_k \rangle \tilde{\varphi}_k. \tag{1.51}
\]

The term biorthogonal is used since to the (nonorthogonal) basis \( \{ \varphi_k \} \) corresponds a dual basis \( \{ \tilde{\varphi}_k \} \) that satisfies the biorthogonality constraint (1.50a). If the basis
1.4. Bases and Frames

{φ_k} is orthogonal, then it is its own dual and the expansion formula (1.51) becomes the usual orthogonal expansion given by (1.41)-(1.42).

Equivalences similar to Theorem 1.5 hold in the biorthogonal case as well, and we give the generalized Parseval’s relations which become

\[ \|x\|^2 = \sum_k \langle x, \varphi_k \rangle \langle x, \bar{\varphi}_k \rangle^*, \] (1.52a)

and

\[ \langle x_1, x_2 \rangle = \sum_k \langle x_1, \varphi_k \rangle \langle x_2, \bar{\varphi}_k \rangle^* = \sum_k \langle x_1, \bar{\varphi}_k \rangle \langle x_2, \varphi_k \rangle^*. \] (1.52b)

For a proof, see Exercise 1.14.

1.4.3 Frames

In all our discussions until now about representations and bases, the vectors forming those bases were always linearly independent. In finite dimensions, that means that there were always as many vectors as the dimension of the space.

But what stops us from adding vectors to those sets of linearly independent vectors? Of course, such extended sets will no longer be linearly independent but they will still provide a representation for our space. Furthermore, what stops us with coming up with completely new sets of linearly dependent vectors which are able to represent our space?

The main question is: Why would we want to do that? If a linearly independent set is already a representation of our space, what do we gain by having more vectors? As you may imagine, such sets are less constrained than their linearly independent counterparts and could thus sometimes provide better solutions. We gave examples in Section 1.1 for \( \mathbb{R}^2 \). We now generalize this discussion.

A family of functions \{φ_k\} in a Hilbert space \( H \) is called a frame if there exist two constants \( A > 0, B < \infty \), such that, for all \( x \) in \( H \),

\[ A \|x\|^2 \leq \sum_k |\langle x, \varphi_k \rangle|^2 \leq B \|x\|^2. \]

A, B are called frame bounds, and when they are equal, we call the frame tight (TF). In a TF we have

\[ \sum_k |\langle x, \varphi_k \rangle|^2 = A \|x\|^2, \]

which, by pulling \( 1/\sqrt{A} \) into the sum can be written as:

\[ \sum_k |\langle x, \frac{1}{\sqrt{A}} \varphi_k \rangle|^2 = \|x\|^2. \] (1.53)

In other words, any TF can be rescaled to be a TF with frame bound 1, yielding a Parseval TF. In a Parseval TF, the signal can be expanded as follows:

\[ x = \sum_k \langle x, \varphi_k \rangle \varphi_k. \] (1.54)
While this last equation resembles the expansion formula in the case of an ONB, a frame does not constitute an ONB in general. In particular, the vectors are linearly dependent and thus do not form a basis. If all the vectors in a TF have unit norm, then the constant $A$ gives the redundancy ratio (for example, $A = 2$ means there are twice as many vectors as needed to cover the space). Note that if $A = B = 1$, and $\|\varphi_k\| = 1$ for all $k$, then $\{\varphi_k\}$ is an ONB.

Because of the linear dependence that exists among the vectors used in the expansion, the expansion is not unique anymore. Consider the set $\{\varphi_k\}$ where $\sum_k \beta_k \varphi_k = 0$ (and not all $\beta_k$’s are zero) because of linear dependence. If $x$ can be written as

$$x = \sum_k \alpha_k \varphi_k,$$

then one can add $\beta_k$ to each $\alpha_k$ without changing the validity of the expansion (1.55). The expansion (1.54) is unique in the sense that it minimizes the norm of the expansion among all valid expansions. Similarly, for general frames, there exists a unique dual frame which is discussed later in the book (in the TF case, the frame and its dual are equal).

### 1.4.4 Matrix View of Bases and Frames

Although our discussion on bases and frames seems complete, we now look into how we can use the power of linear operators (matrices) to efficiently and compactly study those concepts. We developed intuition in Section 1.1; we now summarize what we learned there in a more general case: The following are true in a Hilbert space:

(i) Any Hilbert space basis (orthonormal or biorthogonal) can be represented as a square matrix $\Phi$ having basis vectors $\{\varphi_k\}$ as its columns. If the basis is orthonormal, the matrix is unitary. Here is an example of $N$ basis vectors (of dimension $N$) forming the $N \times N$ basis matrix $\Phi$:

$$\Phi = \begin{bmatrix} \varphi_{11} & \varphi_{21} & \cdots & \varphi_{N1} \\ \varphi_{12} & \varphi_{22} & \cdots & \varphi_{N2} \\ \vdots & \vdots & \ddots & \vdots \\ \varphi_{1N} & \varphi_{2N} & \cdots & \varphi_{NN} \end{bmatrix}_{N \times N}.$$  

(ii) Any Hilbert space frame (tight or general) can be represented as a rectangular matrix $\Phi$ having frame vectors $\{\varphi_k\}$ as its columns. If the frame is Parseval tight, the matrix is unitary. Here is an example of $M$ frame vectors.
1.4. Bases and Frames

(of dimension $N, M > N$) forming the $N \times M$ frame matrix $\Phi$:

$$\Phi = \begin{bmatrix}
\varphi_{11} & \varphi_{21} & \cdots & \varphi_{N1} & \cdots & \varphi_{M1} \\
\varphi_{12} & \varphi_{22} & \cdots & \varphi_{N2} & \cdots & \varphi_{M2} \\
\vdots & \vdots & \ddots & \vdots & \ddots & \vdots \\
\varphi_{1N} & \varphi_{2N} & \cdots & \varphi_{NN} & \cdots & \varphi_{MN} \\
\varphi_1 & \varphi_2 & \cdots & \varphi_N & \cdots & \varphi_M 
\end{bmatrix}_{N \times M} \tag{1.57}$$

(iii) If the matrix is not of full rank, it does not represent a basis/frame.

(iv) An equation of the form $y = \Phi^*x$, either means that (a) $y$ are the coordinates of the vector $x$ in the new coordinate system specified by the columns of $\Phi$, or (b) $\Phi$ is a transformation of the vector $x$ into a vector $y$ in the same coordinate system.

Given $y = \Phi^*x$ and $\Phi$ square (basis), we can go back to $x$ by inverting $\Phi^*$ (that is why we require $\Phi$ to be nonsingular). Therefore,

$$x = (\Phi^*)^{-1}y.$$ 

If the original basis is orthonormal, $\Phi$ is unitary and $(\Phi^*)^{-1} = \Phi$ (see (1.97)). The representation formula can then be written as

$$x = \sum_k \langle x, \varphi_k \rangle \varphi_k = \Phi \Phi^* x = \Phi^* \Phi x, \tag{1.58}$$

where $\Phi^* x$ ($\Phi^T x$ in the real case) computes the expansion coefficients (lengths of projections). If, on the other hand, the original basis is biorthogonal, there is not much more we can say about $\Phi$. The representation formula is

$$x = \sum_k \langle x, \varphi_k \rangle \varphi_k = \Phi \Phi^* x = \Phi^* \Phi x = \sum_k \langle x, \varphi_k \rangle \varphi_k. \tag{1.59}$$

For TFs, a relation similar to (1.58) holds:

$$x = \frac{1}{A} \sum_k \langle x, \varphi_k \rangle \varphi_k = \bar{\Phi} \Phi^* x, \tag{1.60}$$

where $\Phi$ is now a rectangular matrix of full rank. Note that one could scale $\varphi_k$ by $1/\sqrt{A}$, resulting in a tight frame with norm 1—a Parseval tight frame. If the frame is not tight, then a more general relation holds:

$$x = \sum_k \langle x, \varphi_k \rangle \varphi_k = \bar{\Phi} \Phi^* x = \bar{\Phi}^* x = \sum_k \langle x, \varphi_k \rangle \varphi_k. \tag{1.61}$$

Here, the dual frame matrix $\bar{\Phi}$ contains the vectors needed to project (as seen in Section 1.4.3), and can be seen as the right inverse of $\Phi^*$. Some examples of bases and frames are explored in the solved Exercise 1.3.

What we have done in this section is to connect the concepts we learned while talking about Hilbert spaces with the more familiar notions from linear algebra. Implicitly, we have dealt here with linear operators defined in Section 1.3.3.
Discussion
The material in this section forms the basic toolbox for the latter developments in
the book. Basic properties of bases and frames for finite-dimensional spaces are
given in the table in Chapter at a Glance. If you are learning this for the first time,
we recommend you go through it several times, as particular instances of these
abstract objects and notions will show time and again in the book. The basic idea
behind signal representations is “divide and conquer”; decompose the signal into
its building blocks and try to understand its characteristics based on those building
blocks. As you will see, we will make conclusions and inferences on a signal based on
the set of lengths of projections on various subspaces (building blocks). Choosing
those subspaces will be the most important task given an application.

1.5 Random Variables as Hilbert Space Vectors

1.6 Algorithms

We now present two standard algorithms from linear algebra: The first, Gaussian
elimination, is used to determine the inverse of a matrix and solve a system of
linear equations, while the second, Gram-Schmidt orthogonalization, orthogonalizes
a nonorthogonal basis.

1.6.1 Gaussian Elimination

Gaussian elimination is an algorithm used to, among other tasks, determine the
inverse of a matrix and solve a system of linear equations of the form $Ax = y$
(see Section 1.C.2). It consists of two parts: (1) Eliminating variables one by one
resulting in either a triangular or echelon form, and (2) back substituting variables
to find the solution. The first step can result in a form indicating that there is no
solution. The triangle form is achieved using elementary row operations (operations
preserving equivalence of matrices):

- Exchange two rows of a matrix.
- Multiply a row of a matrix by a nonzero constant.
- Add to one row of a matrix a multiple of another row.

The solved Exercise 1.4 defines most of the above terms and gives two examples.

1.6.2 Gram-Schmidt Orthogonalization

We start with a standard linear algebra procedure which can be used to orthog-
onalize an arbitrary basis, that is, turn that general basis into an ONB. We give
the algorithm in the form of pseudo code in Algorithm 1.1. The initial ordering of
the vectors to be orthogonalized determines which ONB will be formed; a different
ordering yields a different ONB (see Example 1.11).
In the pseudo code, the vector $v_k$ is the orthogonal projection of $x_k$ onto the subspace spanned by the previous orthogonalized vectors and this is subtracted from $x_k$, followed by normalization.

Algorithm 1.1 [Gram-Schmidt Orthogonalization]
Input: Set of linearly independent vectors $\{x_k\}$ in $H$ with a given ordering.
Output: Set of orthonormal vectors $\{\varphi_k\}$ in $H$, with the same span as $\{x_k\}$.  

\[
\text{GramSchmidt}(\{x_k\}) \\
\varphi_1 = \frac{x_1}{\|x_1\|} \\
k = 2 \\
\text{while } \varphi_k \text{ exists do} \\
v_k = \sum_{i=1}^{k-1} \langle x_k, \varphi_i \rangle \varphi_i \\
\varphi_k = \frac{x_k - v_k}{\|x_k - v_k\|} \\
k++ \\
\text{end while} \\
\text{return } \{\varphi_k\}
\]

We illustrate the algorithm with a few examples. In the first, the algorithm is used to orthogonalize functions over a closed interval. In the second, the algorithm orthogonalizes the same set of two vectors in $\mathbb{R}^2$ with different orderings, to illustrate the different ONBs obtained. Then, we look at what happens if we have more vectors than the dimension of the space. The final example is a bit more involved and steps through the process of orthogonalizing three vectors in $\mathbb{R}^3$. In Further Reading, we comment on various aspects of Gram-Schmidt orthogonalization.

**Example 1.10.** Consider Legendre polynomials over the interval $[-1,1]$. Start with $x_k(t) = t^k$, $k \in \mathbb{N}$, define an appropriate inner product, and apply the Gram-Schmidt procedure to get $\varphi_k(t)$, of degree $k$, norm 1 and orthogonal to $\varphi_i(t)$, $i < k$ (see Exercise 1.18).

**Example 1.11.** Take vectors $x_1 = [2, 0]^T$ and $x_2 = [2, 2]^T$ in the real plane. We proceed in steps:

(i) Normalize first the first vector:

\[
\varphi_1 = \frac{x_1}{\|x_1\|} = \begin{bmatrix} 1 \\ 0 \end{bmatrix}.
\]

(ii) Project the second vector onto the subspace spanned by the first vector:

\[
v_2 = \langle x_2, \varphi_1 \rangle \varphi_1 = 2\varphi_1 = \begin{bmatrix} 2 \\ 0 \end{bmatrix}.
\]
(iii) Find the difference between the vector \( x_2 \) and its projection \( v_2 \). This difference should be orthogonal to the subspace spanned by \( \varphi_1 \):

\[
d_2 = x_2 - v_2 = \begin{bmatrix} 0 \\ 2 \end{bmatrix}.
\]

(iv) Normalize that difference to get the second vector \( \varphi_2 \):

\[
\varphi_2 = \frac{d_2}{\|d_2\|} = \begin{bmatrix} 0 \\ 1 \end{bmatrix}.
\]

(v) The result can be put into a final ONB matrix as we have two linearly independent vectors in \( \mathbb{R}^2 \):

\[
\Phi = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}.
\] (1.62)

Take now the same vectors as in the previous case, but suppose instead we start with \( x_2 \) as the first vector:

(i) Normalize that vector:

\[
\varphi_1 = \frac{x_2}{\|x_2\|} = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 \\ 1 \end{bmatrix}.
\]

(ii) Project the second vector (now \( x_1 \)) onto the subspace spanned by \( x_2 \):

\[
v_2 = (x_1, \varphi_1) \varphi_1 = \begin{bmatrix} 1 \\ 1 \end{bmatrix}.
\]

(iii) Find the difference between the vector \( x_1 \) and its projection \( v_2 \). This difference is orthogonal to the subspace spanned by \( \varphi_1 \):

\[
d_2 = x_1 - v_2 = \begin{bmatrix} 1 \\ -1 \end{bmatrix}.
\]

(iv) Normalize that difference to get the second vector \( \varphi_2 \):

\[
\varphi_2 = \frac{d_2}{\|d_2\|} = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 \\ -1 \end{bmatrix}.
\]

(v) The final ONB matrix is given as (compare this to (1.62))

\[
\Phi = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}.
\] (1.63)

The resulting ONBs are different in the above two cases, although we started with the same set of vectors.

**Example 1.12.** Consider now three vectors in \( \mathbb{R}^2 \): \( x_1 = [2, 1]^T, x_2 = [-1, 1]^T, \) and \( x_3 = [1, -2]^T \). Apply the Gram-Schmidt procedure, and explain your result.

**Example 1.13.** Consider now a more complicated example of three vectors in \( \mathbb{R}^4 \):

\( x_1 = [1, -1, 0, 0]^T, x_2 = [0, 2, 1, 1]^T, x_3 = [1, -1, 1, 1]^T \). The orthogonalization in this case is shown as the solved Exercise 1.5.
Appendix

1.A Elements of Real Analysis

This appendix briefly reviews some basic elements of real analysis such as maps, convergence of scalar series and sequences of functions. Our aim is not to be exhaustive here; rather, we just review concepts needed throughout the book.

The closure of a set \( S \), denoted by \( \bar{S} \), is the intersection of all closed sets containing that set, while a closed set is a set containing limits of all sequences contained within the set.

The supremum of a set (denoted by \( \sup \)) is the least upper bound on that set, while the infimum of a set (denoted by \( \inf \)) is the greatest lower bound.

The essential supremum (denoted by \( \esssup \)), has the same definition as the supremum, except that it is allowed not to hold over a set of measure 0.

A map is most often a synonym for a function. A function \( x \) takes an argument (input) \( t \) and produces a value (output) \( x(t) \). The acceptable values of the argument form the domain, while the possible function values form the range, also called the image. If the range is a subset of a larger set, that set is termed codomain. Often, a map is denoted as \( x : D \to C \), where \( D \) is the domain and \( C \) is the codomain.

For example, the function \( x(t) = \cos t \), maps \( x : \mathbb{R} \to \mathbb{R} \). Often the rule is specified together with the map as, for example, \( t \mapsto \cos t \). The graph of the function is the set of ordered pairs \( \{(t, x(t)) | t \in D, x \in C\} \).

A function is an injection if \( x(t_1) = x(t_2) \) implies that \( t_1 = t_2 \). In other words, different values of the function must have been produced by different arguments. A function is a surjection if the range equals the codomain, that is, if for every \( y \in C \), there exists a \( t \in D \), such that \( x(t) = y \). A function is a bijection if it is both an injection and a surjection. A composition of functions uses the output of one function as the input to another. For example, \( y(t) = x_2(x_1(t)) \) will be denoted as \( y : D \xrightarrow{x_1} C_1 \xrightarrow{x_2} C_2 \).

Convergence

We now move on to the convergence of scalar series and sequences of functions. Convergence in a normed vector space is in the body, Section 1.3.1.

Definition 1.23 (Convergence of Scalar Series). Let \( a_1, a_2, \ldots \) be scalars. The infinite series \( \sum_{n=1}^{\infty} a_n \) is said to converge when the limit of the \( N \)th partial sum \( S_N = \sum_{n=1}^{N} a_n \) exists and is finite, that is, when \( \lim_{N \to \infty} S_N = S < \infty \).

Definition 1.24 (Absolute/Conditional Convergence). The series \( \sum_{n=1}^{\infty} a_n \) is said to converge absolutely when \( \sum_{n=1}^{\infty} |a_n| \) converges. A series that converges but not absolutely, is said to converge conditionally.

Note that the definition of convergence takes the terms of a series in a particular order. When a series is absolutely convergent, it can be rearranged without changing its value (and thus also the fact that it converges), which allows us to change the order of summations, for example, interchanging \( \sum_{n} \) with \( \sum_{k} \sum_{n} \). (A strange fact known as the Riemann series theorem is that a conditionally convergent series can be rearranged to converge to any desired value or to diverge!)
Determining whether a series converges or computing the value of a series can be challenging. Tests for convergence are reviewed in Exercise 1.19 and a few useful series are explored in Exercise 1.20.

It will at times be important to distinguish between pointwise and uniform convergence, so we define them here and state a few consequences of uniform convergence.

**Definition 1.25 (Pointwise convergence).** Consider a sequence \( x_1, x_2, \ldots \) of real-valued functions on some domain \( W \). We say the sequence converges pointwise to \( x \) if for every \( \epsilon > 0 \) and \( t \in W \), there exists a natural number \( N \) (depending on \( \epsilon \) and \( t \)) such that

\[
|x_n(t) - x(t)| < \epsilon \quad \text{for all } n \geq N.
\]

**Definition 1.26 (Uniform convergence).** Consider a sequence \( x_1, x_2, \ldots \) of real-valued functions on some domain \( W \). We say the sequence converges uniformly to \( x \) if for every \( \epsilon > 0 \), there exists a natural number \( N \) (depending on \( \epsilon \)) such that

\[
|x_n(t) - x(t)| < \epsilon \quad \text{for all } t \in W \text{ and all } n \geq N.
\]

Uniform convergence implies pointwise convergence. Furthermore:

- If a sequence of continuous functions is uniformly convergent, the limit function is necessarily continuous.
- Suppose a sequence of partial sums \( x_N(t) = \sum_{n=1}^{N} s_n(t) \) is uniformly convergent to \( x(t) \) on \([a, b] \). Then the series may be integrated term by term:

\[
\int_{a}^{b} x(t) \, dt = \sum_{i=n}^{\infty} \int_{a}^{b} s_i(t) \, dt.
\]

### 1.A.1 Functions of Interest

**Dirac Delta Function** The *Dirac delta function*, which is a generalized function or distribution, is defined as a limit of rectangular functions. For example,

\[
\delta(t) = \lim_{\epsilon \to 0} \delta_{\epsilon}(t) = \begin{cases} 
\frac{1}{\epsilon}, & 0 \leq t < \epsilon; \\
0, & \text{otherwise.}
\end{cases}
\]  

(One could use any smooth function \( \psi(t) \) with integral 1 and define the Dirac.) Any operation involving a Dirac function requires a limiting operation. Since we are reviewing standard results, and for notational convenience, we will skip the limiting process. However, we caution that Dirac functions have to be handled with care to get meaningful results. When in doubt, it is best to go back to the definition and the limiting process. A *Dirac impulse sequence* is defined in Chapter 2, (2.7).
We list some useful properties of the Dirac function:

\[
\int_{-\infty}^{\infty} \delta(t) \, dt = 1, \quad (1.65)
\]

\[
\int_{-\infty}^{\infty} x(t_0 - t) \, \delta(t) \, dt = \int_{-\infty}^{\infty} x(t) \, \delta(t_0 - t) \, dt = x(t_0), \quad (1.66)
\]

\[
(x(t) * \delta(t-t_0))_{t} = x(t+t_0), \quad (1.67)
\]

\[
x(t) \, \delta(t) = x(0) \, \delta(t), \quad (1.68)
\]

\[
\delta(t-t_0) \xrightarrow{\text{FT}} e^{-j\omega t_0}, \quad (1.69)
\]

\[
e^{j\omega t_0} \xrightarrow{\text{FT}} 2\pi\delta(\omega - \omega_0), \quad (1.70)
\]

\[
x(t)e^{j\omega t} \xrightarrow{\text{FT}} X(\omega - \omega_0). \quad (1.71)
\]

**Sinc Function**  A sinc function and a sinc sequence appear frequently in signal processing and approximation. We discuss here the sinc function; the sinc sequence is defined in Chapter 2, (2.11), and shown in in Figure 1.12(b).

A **sinc function** is defined as:

\[
sinc_T(t) = \frac{\sin \frac{t\pi}{T}}{\frac{t\pi}{T}} = \frac{e^{j\frac{t\pi}{T}} - e^{-j\frac{t\pi}{T}}}{2j(\frac{t\pi}{T})}. \quad (1.72)
\]

The zeros \( t_k \) of the sinc function are at:

\[
t_k \pi = k\pi \quad \Rightarrow \quad t_k = kT, \quad k \in \mathbb{Z}, \quad (1.73)
\]

as seen in Figure 1.12(a). The value of sinc at \( t = 0 \) is obtained through a limiting operation since sinc\(_T(0) = 0/0\). Applying the l’Hôpital rule to (1.72) we get that

\[
sinc_T(0) = 1. \quad (1.74)
\]

The above two equations can be expressed as

\[
sinc_T(kT) = \frac{\sin k\pi}{k\pi} = \delta_k, \quad k \in \mathbb{Z}. \quad (1.75)
\]

---

**Figure 1.12:** (a) Sinc function. (b) Sinc sequence. (TBD: Figure to be redone.)
Gaussian Function

A Gaussian function is a function of the form:

$$g(t) = \gamma e^{-\alpha(t-\mu)^2},$$

(1.76)

where $\mu$ just shifts the center of the function to $t = \mu$.

When $\alpha = 1/2\sigma^2$ and $\gamma = 1/\sigma\sqrt{2\pi}$, $\|g\|_1 = 1$, and can thus be seen as a probability density function, with $\mu$ and $\sigma$ interpreted as the mean and standard deviation, respectively:

$$\|g(t)\|_1 = \int_{-\infty}^{\infty} |g(t)| \, dt = \int_{-\infty}^{\infty} \gamma e^{-\alpha(t-\mu)^2} \, dt$$

$$= \int_{-\infty}^{\infty} \frac{1}{\sigma\sqrt{2\pi}} e^{-(t-\mu)^2/2\sigma^2} \, dt = 1. \quad (1.77)$$

When $\gamma = (2\alpha/\pi)^{1/4}$, $\|g\|_2 = 1$, that is, it is of unit energy:

$$\|g(t)\|_2^2 = \int_{-\infty}^{\infty} |g(t)|^2 \, dt = \int_{-\infty}^{\infty} \gamma^2 e^{-2\alpha(t-\mu)^2} \, dt$$

$$= \int_{-\infty}^{\infty} \frac{2\alpha}{\pi} e^{-2\alpha(t-\mu)^2} \, dt = 1. \quad (1.78)$$

The Fourier transform of a Gaussian function is again a Gaussian function:

$$\gamma e^{-\alpha t^2} \xrightarrow{\text{FT}} \gamma \sqrt{\frac{\pi}{\alpha}} e^{-\pi/\alpha \omega^2}. \quad (1.79)$$

1.B Elements of Algebra: Polynomials

A polynomial is a finite sum of the following form:

$$p(x) = \sum_{n=0}^{N} a_n x^n,$$

(1.80)

where $N$ denotes the degree of the polynomial as well as the number of roots of the polynomial. For example,

$$p(x) = a_2 x^2 + a_1 x + a_0,$$

(1.81)

is a quadratic polynomial and has two, in general complex, roots:

$$x_{1,2} = \frac{-a_1 \pm \sqrt{a_1^2 - 4a_2a_0}}{2a_2}. \quad (1.82)$$

An irreducible polynomial is a polynomial with no real roots. For example, $p(x) = x^2 + 2$ has no real roots; rather its roots are complex $x_{1,2} = \pm j\sqrt{2}$.
Fundamental Theorem of Algebra  The following theorem, formulated by Gauss, is a useful tool in algebra:

**Theorem 1.8 (Fundamental theorem of algebra).** Every polynomial of order \( N \) possesses exactly \( N \) complex roots.

In other words, we can factor any polynomial into a product of linear factors and irreducible quadratic factors as:

\[
p(x) = \sum_{n=0}^{N} a_n x^n = a_N \prod_{n=0}^{N-2k-1} (x - x_{1n}) \prod_{n=0}^{k-1} (x - x_{2n})(x - x_{2n}^*),
\]

where \((x - x_{1n})\) terms are called linear, while \((x - x_{2n})(x - x_{2n}^*)\) are quadratic terms. The roots \( x_{1n} \) are real, while \( x_{2n} \) are complex and appear in pairs.

**Polynomial Interpolation**  We all know that if given two points, a unique line can be drawn through them; if three points are given, a unique quadratic curve can be drawn through. Extending this argument, there is a single interpolating polynomial of degree \((N - 1)\) through given \( N \) points, \( x_i, i \in \{1, 2, \ldots, N\} \). This polynomial is given by the **Lagrange interpolating formula**:

\[
p(x) = \frac{\prod_{n \neq 1}(x - x_n)}{\prod_{n \neq 1}(x_1 - x_n)} y_1 + \frac{\prod_{n \neq 2}(x - x_{2n})}{\prod_{n \neq 2}(x_2 - x_n)} y_2 + \ldots + \frac{\prod_{n \neq N}(x - x_N)}{\prod_{n \neq N}(x_N - x_n)} y_N,
\]

typically implemented using the so-called **Neville’s algorithm**.

1.C  Elements of Linear Algebra

The finite-dimensional cases of Hilbert spaces, namely \( \mathbb{R}^N \) and \( \mathbb{C}^N \), are very important, and linear operators on such spaces are studied in linear algebra. We give pointers to reference texts in Further Reading, and present only a brief account here, focusing on basic concepts and topics which are needed later, such as polynomial matrices.

1.C.1  Basic Definitions and Properties

We first show how we can associate a possibly infinite matrix with a given linear operator \( A \) on a Hilbert space. Given is the ONB \( \Phi = \{ \varphi_k \} \). Then any \( x \) from \( H \) can be written as \( x = \sum_k \langle x, \varphi_k \rangle \varphi_k \). As this is valid for any \( x \) from \( H \), it is valid for \( x = A \varphi_i \) as well, and thus:

\[
A \varphi_i = \sum_k \langle A \varphi_i, \varphi_k \rangle \varphi_k.
\]
We can use the above expression and expand \( x \) in terms of its ONB and, as \( A \) is linear, compute \( Ax \) by pulling \( A \) through the summation to yield:

\[
Ax = A \left( \sum_i \langle x, \varphi_i \rangle \varphi_i \right) = \sum_i \langle x, \varphi_i \rangle A\varphi_i
\]

\[
= \sum_i \langle x, \varphi_i \rangle \sum_k \langle A\varphi_i, \varphi_k \rangle \varphi_k = \sum_k \sum_i \langle A\varphi_i, \varphi_k \rangle \langle x, \varphi_i \rangle \varphi_k. \tag{1.85}
\]

To relate a linear operator to a matrix, we will try to express a standard system of equations, \( Ax = y \) in terms of what we computed above, and to do that, we need to express \( y \) in terms of the given ONB: \( y = \sum_k \langle y, \varphi_k \rangle \varphi_k \). Using this and (1.85), we can write \( Ax = y \) as

\[
Ax = \sum_k \sum_i \langle A\varphi_i, \varphi_k \rangle \langle x, \varphi_i \rangle \varphi_k = \sum_k \langle y, \varphi_k \rangle \varphi_k = y.
\]

Taking the term inside the summation for a given \( k \), canceling \( \varphi_k \) and writing it as a row vector/column vector product, we get:

\[
\begin{bmatrix}
\langle A\varphi_1, \varphi_k \rangle \\
\langle A\varphi_2, \varphi_k \rangle \\
\vdots
\end{bmatrix}
\begin{bmatrix}
\langle x, \varphi_1 \rangle \\
\langle x, \varphi_2 \rangle \\
\vdots
\end{bmatrix}
= \langle y, \varphi_k \rangle.
\]

If we now collect the above for all \( k \) and stack them as rows of a matrix, we get:

\[
A = \begin{bmatrix}
\langle A\varphi_1, \varphi_1 \rangle & \langle A\varphi_2, \varphi_1 \rangle & \cdots \\
\langle A\varphi_1, \varphi_2 \rangle & \langle A\varphi_2, \varphi_2 \rangle & \cdots \\
\vdots & \vdots & \ddots
\end{bmatrix}
\begin{bmatrix}
\langle x, \varphi_1 \rangle \\
\langle x, \varphi_2 \rangle \\
\vdots
\end{bmatrix}
= \begin{bmatrix}
\langle y, \varphi_1 \rangle \\
\langle y, \varphi_2 \rangle \\
\vdots
\end{bmatrix},
\]

or, in other words, the matrix \( A = \{ A_{ki} \} \) corresponding to the operator \( A \) expressed with respect to the basis \( \{ \varphi_k \} \), is defined by \( A_{ki} = \langle A\varphi_i, \varphi_k \rangle \).

An \( N \times 1 \) matrix is called a column vector; a \( 1 \times N \) matrix a row vector. A row vector would then be written as \( v^T \), where \( ^T \) denotes transposition (interchange of rows and columns, that is, if \( A \) has elements \( A_{ki} \), \( A^T \) has elements \( A_{ik} \)). If the entries are complex, one often uses Hermitian transposition, which is complex conjugation followed by usual transposition, and is denoted by a superscript *.

When \( M = N \), an \( M \times N \) matrix is called square, otherwise it is called rectangular. A \( 1 \times 1 \) matrix is called scalar. We denote by 0 the null matrix (all elements are zero) and by \( I \) the identity (\( A_{kk} = 1 \), and 0 otherwise). The identity matrix is a special case of a diagonal matrix, which has nonzero values only for elements \( A_{kk} \) (and thus is valid for both rectangular as well as square matrices).

The antidiagonal matrix \( J \) has all the elements on the other diagonal equal to 1, while the rest are 0, that is, \( A_{ki} = 1 \), for \( i = N + 1 - k \), and \( A_{ki} = 0 \) otherwise. A
lower (or upper) triangular matrix is a square matrix with all of its elements above (or below) the main diagonal equal to zero.

Apart from addition/subtraction of same-size matrices (by adding/subtracting the corresponding elements), one can multiply matrices \(A\) and \(B\) with sizes \(M \times N\) and \(n \times p\) respectively, yielding a matrix \(C\) whose elements are given by

\[
C_{ij} = \sum_{k=1}^{N} A_{ik} B_{kj}.
\]

Note that the matrix product is not commutative in general, that is,\(^6\) It can be shown that \((AB)^T = B^T A^T\).

In matrix notation, the inner product of two column vectors from \(\mathbb{R}^N\) is \(\langle x_1, x_2 \rangle = x_2^T x_1\), and if the vectors are from \(\mathbb{C}^N\), then \(\langle x_1, x_2 \rangle = x_2^* x_1\). The outer product of two vectors from \(\mathbb{R}^N\) and \(\mathbb{R}^M\) is an \(N \times M\) matrix given by \(x_1 x_2^T\).

To define the notion of a determinant, we first need to define a minor. A minor \(M_{ki}\) is a submatrix of the matrix \(A\) obtained by deleting its \(k\)th row and \(i\)th column. More generally, a minor can be any submatrix of the matrix \(A\) obtained by deleting some of its rows and columns. Then the determinant of an \(N \times N\) matrix can be defined recursively as

\[
\det(A) = \sum_{k=1}^{N} A_{ki} (-1)^{k+i} \det(M_{ki}),
\]

where \(i\) is fixed and belongs to \(\{1, 2, \ldots, N\}\). The cofactor \(C_{ki}\) is \((-1)^{k+i} \det(M_{ki})\).

A square matrix is said to be singular if \(\det(A) = 0\). The product of two matrices is nonsingular only if both matrices are nonsingular. Here are some properties of determinants:

(i) If \(C = AB\), then \(\det(C) = \det(A) \det(B)\).

(ii) If \(B\) is obtained by interchanging two rows/columns of \(A\), then \(\det(B) = -\det(A)\).

(iii) \(\det(A^T) = \det(A)\).

(iv) For an \(N \times N\) matrix \(A\), \(\det(\alpha A) = \alpha^N \det(A)\).

(v) The determinant of a triangular, and in particular, of a diagonal matrix, is the product of the elements on the main diagonal.

An important interpretation of the determinant is that it corresponds to the volume of the parallelepiped obtained when taking the column vectors of the matrix as its edges (one can take the row vectors as well, leading to a different parallelepiped, but the volume remains the same). Thus, a zero determinant indicates linear dependence of the row and column vectors of the matrix, since the parallelepiped is not of full dimension.

\(^6\)When there is possible confusion, we will denote a matrix product by \(A \cdot B\); otherwise we will simply write \(AB\).
The rank of a matrix is the size of its largest nonsingular minor (possibly the matrix itself). In a rectangular $M \times N$ matrix, the column rank equals the row rank, that is, the number of linearly independent rows equals the number of linearly independent columns. In other words, the dimension of $\text{span}\{\text{columns}\}$ is equal to the dimension of $\text{span}\{\text{rows}\}$. For an $N \times N$ matrix to be nonsingular, its rank must equal $N$. Also, $\text{rank}(AB) \leq \min(\text{rank}(A), \text{rank}(B))$.

For a square nonsingular matrix $A$, the inverse matrix $A^{-1}$ can be computed using Cramer’s formula

$$A^{-1} = \frac{\text{adjugate}(A)}{\text{det}(A)},$$

where the elements of adjugate($A$) are $(\text{adjugate}(A))_{ji} = \text{cofactor of } A_{ki} = C_{ki}$. For a square matrix, $AA^{-1} = A^{-1}A = I$. Also, $(AB)^{-1} = B^{-1}A^{-1}$. Note that Cramer’s formula is not actually used to compute the inverse in practice; rather, it serves as a tool in proofs.

For an $M \times N$ rectangular matrix $A$, an $N \times M$ matrix $L$ is its left inverse if $LA = I$. Similarly, an $N \times M$ matrix $R$ is a right inverse of $A$ if $AR = I$. These inverses are not unique and may not even exist. However, if the matrix $A$ is square and of full rank, then its right inverse equals its left inverse, and we can apply Cramer’s formula to find that inverse.

While we have defined the kernel and range in Section 1.3.3 on linear operators, we repeat them here for completeness. The column space of $A$ denotes the linear span of the columns of $A$ and is called the range of $A$, $R(A)$, while the row space is the linear span of the rows of $A$. The kernel of $A$, $\ker(A)$ (also called the null space), is spanned by the vectors orthogonal to the row space, or $Av = 0$. If $A$ is of size $M \times N$ (the system of equations has $M$ equations in $N$ unknowns), then the dimension of the range (which equals the rank $\rho$) plus the dimension of the kernel is equal to $N$, that is

$$\text{dim}(R(A)) + \text{dim}(\ker(A)) = N. \quad (1.86)$$

### 1.C.2 Linear Systems of Equations and Least Squares Solutions

Going back to the equation $Ax = y$, with $A$ square, one can say that the system has a unique solution provided $A$ is nonsingular, and this solution is given by $x = A^{-1}y$. Note that one would rarely compute the inverse matrix in order to solve a linear system of equations; rather Gaussian elimination would be used, since it is much more efficient (the algorithm is given in Section 1.6.1, while the solved Exercise 1.4 gives examples).

Let us give an interpretation of solving the problem $Ax = y$. The product $Ax$ constitutes a linear combination of the columns of $A$ weighted by the entries of $x$. We now have four possibilities:

(i) If $y$ belongs to the column space of $A$, $R(A)$, a solution exists.

(i) If the columns of $A$ are linearly independent, the solution is unique.

(ii) If the columns of $A$ are not linearly independent, there are infinitely many solutions.
If \( y \) does not belong to the column space of \( A \), there is no exact solution and only approximations are possible, denoted by \( \hat{x} \). The approximate solution is the solution to another equation, where the approximation to \( y \) is used, \( \hat{y} \), which, as we know from before, is the orthogonal projection of \( y \) onto the span of the columns of \( A \). The approximation to \( x \) is \( \hat{x} \). So now we are solving \( A\hat{x} = \hat{y} \), so that the error \( \|y - \hat{y}\| \) is minimized.

The error between \( y \) and its projection \( \hat{y} \) is orthogonal to the column space of \( A \). That is, any linear combination of the columns of \( A \), for example \( A\alpha \), is orthogonal to \( y - \hat{y} = y - A\hat{x} \). Thus

\[
\langle y - A\hat{x}, A\alpha \rangle = (A\alpha)^T(y - A\hat{x}) = 0 \implies A^TA\hat{x} = A^Ty,
\]

which are called the normal equations of the least squares problem. The normal equations always have a solution, and the uniqueness depends on the rank of \( A \).

(i) If the columns of \( A \) are linearly independent, then \( A^TA \) is invertible and the solution \( \hat{x} \) is unique:

\[
\hat{x} = (A^TA)^{-1}A^Ty.
\]

The orthogonal projection \( \hat{y} \) is then equal to

\[
\hat{y} = A\hat{x} = A(A^TA)^{-1}A^Ty.
\]

Note that the matrix \( P = A(A^TA)^{-1}A^T \) satisfies \( P^2 = P \) and is symmetric \( P = P^T \), thus satisfying the conditions for being an orthogonal projection. Also, it can be verified that the partial derivatives of the squared error with respect to the components of \( \hat{x} \) are zero for the above choice (see Exercise 1.24).

(ii) If the columns of \( A \) are not linearly independent, there are infinitely many solutions to (1.87).

We now give an example to illustrate our discussion.

**Example 1.14.** We start in \( \mathbb{R}^3 \) and choose \( y = [1 \quad 1 \quad 1]^T \).

(i) As an example of \( y \) belonging to the column space of \( A \), \( R(A) \), and \( A \) of full rank, we choose

\[
A = \begin{bmatrix}
1 & 0 & 0 \\
1 & 1 & 1 \\
0 & 0 & 1
\end{bmatrix}.
\]

We can easily solve this system to get its unique solution \( x = [1 \quad -1 \quad 1]^T \).

This is illustrated in Figure 1.13(a).

(ii) As an example of \( y \in R(A) \), but \( A \) not of full rank, we choose

\[
A = \begin{bmatrix}
1 & 0 & 1 \\
1 & 0 & 1 \\
0 & 1 & 1
\end{bmatrix}.
\]
This system has infinitely many solutions $x = [1 - 2a 1 - a]^T$. This is illustrated in Figure 1.13(b).

(iii) We now look at what happens if $y$ is not in $R(A)$ and $A$ is of full rank. We choose

$$A = \begin{bmatrix} 1 & 0 \\ 1 & 1 \\ 0 & 0 \end{bmatrix}.$$ 

Obviously, since the columns of $A$ span only the $(e_1, e_2)$-plane, there can be no solution since $y$ does not lie in that plane. The best we can do now is to find an approximate solution that minimizes the error. Such an approximate solution is unique, though. We can now compute $\hat{x}$ as in (1.88a) to get $\hat{x} = [1 0]^T$. The orthogonal projection of $y$ is $\hat{y} = [1 1 0]^T$. This is illustrated in Figure 1.13(c).

(iv) Finally, for $y$ not in $R(A)$ and $A$ not of full rank, we choose

$$A = \begin{bmatrix} 1 & 0 & 1 \\ 0 & 1 & 1 \\ 0 & 0 & 0 \end{bmatrix}.$$ 

Again, the columns of $A$ span only the $(e_1, e_2)$-plane. There is no exact solution in this case either and there are infinitely many approximate ones. We compute them now as $\hat{x} = [1 - a 1 - a a]^T$. The orthogonal projection of $y$ is $\hat{y} = [1 1 0]^T$. This is illustrated in Figure 1.13(d).

1.C.3 Eigenvectors and Eigenvalues

The characteristic polynomial for a matrix $A$ is $D(x) = \det(xI - A)$. Its roots are called eigenvalues $\lambda_i$. In particular, a vector $p \neq 0$ for which

$$Ap = \lambda p,$$ (1.89)

is an eigenvector associated with the eigenvalue $\lambda$. If a matrix of size $N \times N$ has $N$ linearly independent eigenvectors, then it can be diagonalized, that is, it can be written as

$$A = T \Lambda T^{-1},$$ (1.90)

where $\Lambda$ is a diagonal matrix containing the eigenvalues of $A$ along the diagonal, and $T$ contains the eigenvectors of $A$ as its columns. Using the properties of the determinant we have seen earlier, (1.90) implies that

$$\det A = \det(T \Lambda T^{-1}) = \det(T T^{-1}) \det(\Lambda) = \prod_{n=1}^{N} \lambda_n.$$ (1.91)

An important case is when $A$ is symmetric or, in the complex case, Hermitian (or, self-adjoint), $A^* = A$. Then, the eigenvalues are real, and a full set of orthogonal
eigenvectors exists. Taking them as columns of a matrix $U$ after normalization so that $U^*U = I$, we can write a Hermitian matrix as

$$A = U A U^*.$$ 

This result is the spectral theorem for Hermitian matrices. Hermitian matrices commute with their Hermitian transpose. More generally, a matrix $A$ that commutes with its Hermitian transpose is called normal, that is, it satisfies $A^*A = AA^*$. Normal matrices are exactly those that have a complete set of orthogonal eigenvectors.

The importance of eigenvectors in the study of linear operators comes from
the following fact: Assuming a full set of eigenvectors, a vector \( x \) can be written as a linear combination of eigenvectors \( x = \sum_k \alpha_k v_k \). Then,

\[
Ax = A \left( \sum_k \alpha_k v_k \right) = \sum_k \alpha_k (Av_k) = \sum_k \alpha_k \lambda_k v_k.
\]

**Positive Definite Matrices**

A real symmetric matrix \( A \) is called *positive definite (semi-definite)* if all its eigenvalues are greater than (or equal to) 0. Equivalently, for all nonzero vectors \( x \), the following is satisfied:

\[
x^T Ax > 0.
\]

Finally, for a positive definite matrix \( A \), there exists a nonsingular matrix \( W \) such that

\[
A = W^T W,
\]

where \( W \) is intuitively a “square root” of \( A \). One possible way to choose such a square root is to diagonalize \( A \) as \( A = Q\Lambda Q^T \) and then, since all the eigenvalues are positive, choose \( W^T = Q\sqrt{\Lambda} \) (the square root is applied on each eigenvalue in the diagonal matrix \( \Lambda \)). The above discussion carries over to Hermitian matrices by using Hermitian transposes.

**Singular Value Decomposition**

*Singular value decomposition (SVD)* is a general method for factoring real or complex matrices (rectangular or square). A given \( M \times N \) real or complex matrix \( A \) can be factored as follows:

\[
A = U\Sigma V^*,
\]

where \( U \) is an \( M \times M \) unitary matrix, \( V \) is an \( N \times N \) unitary matrix, and \( \Sigma \) is an \( M \times N \) diagonal matrix with nonnegative value on the diagonal, \( \sigma_k \), called *singular values*. The following are the facts related to SVD (see also Exercise 1.11):

(i) If the matrix \( A \) is Hermitian, for every nonnegative eigenvalue \( \lambda_k \), there is an identical singular value \( \sigma_i = \lambda_k \). For every negative eigenvalue \( \lambda_k \), there is a corresponding singular value \( \sigma_i = |\lambda_k| \).

(ii) If the matrix \( A \) is symmetric with eigenvalues \( \lambda_k \),

\[
\|A\|_2 = \max\{\lambda_k\}.
\]

(iii) Call \( \mu_k \) the eigenvalues of \( A^*A \); then

\[
\|A\|_2 = \sqrt{\max\{\mu_k\}} = \max\{\sigma_k\}.
\]
1.C.4 Special Matrices

Circulant Matrices

A (right) circulant matrix is a matrix where each row is obtained by a (right) circular shift of the previous row:

\[
C = \begin{bmatrix}
    c_0 & c_1 & \cdots & c_{N-1} \\
    c_{N-1} & c_0 & \cdots & c_{N-2} \\
    \vdots & \vdots & \ddots & \vdots \\
    c_1 & c_2 & \cdots & c_0
\end{bmatrix}.
\] (1.94)

Toeplitz Matrices

A Toeplitz matrix is a matrix whose entry \(T_{ki}\) depends only on the value of \(k - i\) and thus it is constant along the diagonals:

\[
T = \begin{bmatrix}
    t_0 & t_1 & t_2 & \cdots & t_{N-1} \\
    t_{N-1} & t_0 & t_1 & \cdots & t_{N-2} \\
    t_{N-2} & t_{N-1} & t_0 & \cdots & t_{N-3} \\
    \vdots & \vdots & \vdots & \ddots & \vdots \\
    t_{-N+1} & t_{-N+2} & t_{-N+3} & \cdots & t_0
\end{bmatrix}.
\] (1.95)

Sometimes, the elements \(t_k\) are matrices themselves, in which case the matrix is called block Toeplitz.

Band Matrices

Band matrices of size \(N \times N\) are those matrices which are nonzero only on a “band” around the diagonal. That band need not be symmetric, one may have \(N_r\) diagonals on the right side and \(N_\ell\) on the left side. This definition means that the following types of matrices are all special cases of band matrices: diagonal \((N_r = N_\ell = 0)\), tridiagonal \((N_r = N_\ell = 1)\), upper- \((N_r = N - 1, N_\ell = 0)\) or lower-triangular \((N_\ell = N - 1, N_r = 0)\), and many others. We will see these matrices when we discuss multirate operators in Chapter 2. An example with \(N = 5\), \(N_r = 2\) and \(N_\ell = 1\) is given below:

\[
B = \begin{bmatrix}
    b_{11} & b_{12} & b_{13} & 0 & 0 \\
    b_{21} & b_{22} & b_{23} & b_{24} & 0 \\
    0 & b_{32} & b_{33} & b_{34} & b_{35} \\
    0 & 0 & b_{43} & b_{44} & b_{45} \\
    0 & 0 & 0 & b_{54} & b_{55}
\end{bmatrix}.
\] (1.96)

Polynomial Matrices

Since several of results given in Chapter 6 will make use of polynomial matrices, we now give a brief overview. We give pointers to reference texts in Further Reading.
A polynomial matrix (or a matrix polynomial) is a matrix whose entries are polynomials. The fact that the above two names can be used interchangeably is due to the following forms of a polynomial matrix \( H(x) \):

\[
H(x) = \begin{bmatrix}
\sum a_k x^k & \cdots & \sum b_k x^k \\
\vdots & \ddots & \vdots \\
\sum c_k x^k & \cdots & \sum d_k x^k
\end{bmatrix} = \sum_k H_k x^k,
\]

that is, it can be written either as a matrix containing polynomials as its entries, or a polynomial having matrices as its coefficients.

The question of the rank in polynomial matrices is more subtle than for scalar ones. For example, the matrix

\[
\begin{bmatrix}
a + bx & 3(a + bx) \\
c + dx & \lambda(c + dx)
\end{bmatrix},
\]

with \( \lambda = 3 \), always has rank less than 2, since the two columns are proportional to each other. On the other hand, if \( \lambda = 2 \), then the matrix would have the rank less than 2 only if \( x = -a/b \) or \( x = -c/d \). This leads to the notion of normal rank. First, note that \( H(x) \) is nonsingular only if \( \det(H(x)) \) is different from 0 for some \( x \). Then, the normal rank of \( H(x) \) is the largest of the orders of minors that have a determinant not identically zero. In the above example, for \( \lambda = 3 \), the normal rank is 1, while for \( \lambda = 2 \), the normal rank is 2.

A polynomial matrix is called unimodular if its \( |\det| = 1 \). An example is the following matrix:

\[
H(x) = \begin{bmatrix}
1 + x & x \\
2 + x & 1 + x
\end{bmatrix}.
\]

There are several useful properties pertaining to unimodular matrices: For example, the product of two unimodular matrices is again unimodular. The inverse of a unimodular matrix is unimodular as well. A polynomial matrix \( H(x) \) is unimodular if and only if its inverse is a polynomial matrix. All these facts can be proven using properties of determinants.

A generalization of polynomial matrices are rational matrices, with entries ratios of two polynomials. As will be shown in Chapter 6, polynomial matrices in \( z \) correspond to finite impulse response (FIR) discrete-time filters, while rational matrices can be associated with infinite impulse response (IIR) filters. Unimodular and unitary matrices can be defined for the rational matrices, in a similar fashion as for polynomial matrices.

**Unitary Matrices**

An \( N \times N \) matrix is called a unitary matrix if it satisfies

\[
U^*U = UU^* = I.
\]

(1.97)

Its inverse \( U^{-1} \) is its (Hermitian) transpose \( U^* \) (\( U^T \) for real \( U \)). When the matrix has real entries, it is often called orthogonal or orthonormal, and sometimes, a
scale factor is allowed on the right of (1.97). Rectangular unitary matrices are also possible, that is, an \( M \times N \) matrix \( U \) with \( M < N \) is unitary if
\[
\|UX\| = \|X\|
\]
for all \( X \in \mathbb{C}^N \), as well as
\[
\langle UX, UY \rangle = \langle X, Y \rangle
\]
for all \( X, Y \in \mathbb{C}^N \), which are the usual Parseval’s relations. Then it follows that
\[
UU^* = I,
\]
where \( I \) is of size \( M \times M \) (beware that the product does not commute). Unitary matrices have eigenvalues of unit modulus and a complete set of orthogonal eigenvectors. Note that a square unitary matrix performs a rotation (we touched upon it in Section 1.1 when we were speaking about the matrix view of bases and frames).

When a square \( N \times N \) matrix \( A \) has full rank, its columns (or rows) form a basis for \( \mathbb{R}^N \), and the Gram-Schmidt orthogonalization procedure can be used to turn it into an ONB. Gathering the steps of the Gram-Schmidt procedure into a matrix form, we can write \( A \) as
\[
A = QR,
\]
where the columns of \( Q \) form an ONB and \( R \) is upper triangular.

An \( N \times N \) real unitary matrix has \( N(N - 1)/2 \) degrees of freedom (up to a permutation of its rows or columns and a sign change in each vector). These degrees of freedom are used to parametrize unitary matrices in various forms (see Further Reading for more details).

Some standard unitary matrices are the rotation matrix from (1.13) and the reflection matrix. Their geometric interpretation is intuitive, given by their names. The mathematical distinction is in the value of the determinant; while both are 1 in absolute value, the rotation matrix has \( \det = 1 \), while the reflection matrix has \( \det = -1 \).

### Chapter at a Glance

It seems that what we have done until now is to solve the following equation:
\[
x = x.
\]
Is that really our aim? Well, sort of. What we are really doing is finding representations given by matrices such that:
\[
x = Ix = \Phi \Phi^* x.
\]
(We have taken the Hermitian transpose of \( \Phi \) as opposed to the ordinary transpose to allow for complex vectors.) After finding \( \Phi \) and \( \Phi^* \) such that \( \Phi \Phi^* = I \), we call
\[
X = \Phi^* x
\]
a decomposition and
\[ x = \Phi X = \Phi \Phi^* x \]  
(1.100b)
a reconstruction. Also, \( \Phi \Phi^* x \) is often called a representation of a signal. The elements of \( X \) are called expansion or transform coefficients and include Fourier, wavelet, Gabor coefficients, as well as many others. We decompose signals to have a look at their properties in the transform domain. After analysis or manipulations such as compression, transmission, etc., we reconstruct the signal from its expansion coefficients.

Furthermore, we studied special cases distinguished by the properties of \( \Phi \):
- If \( \Phi \) is square and nonsingular, then \( \Phi \) is a basis and \( \Phi \) is its dual basis.
- If \( \Phi \) is unitary, that is, \( \Phi \Phi^* = I \), then \( \Phi \) is an ONB and \( \Phi^* = \Phi \).
- If \( \Phi \) is rectangular and full rank, then \( \Phi \) is a frame and \( \Phi^* \) is its dual frame.
- If \( \Phi \) is rectangular and \( \Phi \Phi^* = I \), then \( \Phi \) is a TF and \( \Phi^* = \Phi \).

Which of these options we will choose depends entirely on our application and the criteria for designing such matrices (representations). In this book, these criteria are based on time-frequency considerations and will be explored in more detail in Chapter 5.

<table>
<thead>
<tr>
<th>Property</th>
<th>Orthonormal Basis</th>
<th>Biorthogonal Basis</th>
<th>Tight Frame</th>
<th>General Frame</th>
</tr>
</thead>
<tbody>
<tr>
<td>Expansion Set</td>
<td>( \Phi = {\varphi_k}_{k=1}^N )</td>
<td>( \Phi = {\varphi_k}_{k=1}^N )</td>
<td>( \Phi = {\varphi_k}_{k=1}^M )</td>
<td>( \Phi = {\varphi_k}_{k=1}^M )</td>
</tr>
<tr>
<td></td>
<td>( \varphi_k \in \mathbb{C}^N )</td>
<td>( \varphi_k, \tilde{\varphi}_k \in \mathbb{C}^N )</td>
<td>( \varphi_k \in \mathbb{C}^N )</td>
<td>( \varphi_k, \tilde{\varphi}_k \in \mathbb{C}^N )</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>( M \geq N )</td>
<td>( M \geq N )</td>
</tr>
<tr>
<td>Self-dual</td>
<td>Yes</td>
<td>No</td>
<td>Can be</td>
<td>No</td>
</tr>
<tr>
<td>Linearly indep.</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>Orthogonal relations</td>
<td>( \langle \varphi_k, \varphi_i \rangle = \delta_{k,i} )</td>
<td>( \langle \varphi_k, \tilde{\varphi}<em>i \rangle = \delta</em>{k-i} )</td>
<td>None</td>
<td>None</td>
</tr>
<tr>
<td>Matrix view</td>
<td>( \sum_{k=1}^N \langle x, \varphi_k \rangle \varphi_k )</td>
<td>( \sum_{k=1}^N \langle x, \tilde{\varphi}_k \rangle \varphi_k )</td>
<td>( \sum_{k=1}^M \langle x, \varphi_k \rangle \varphi_k )</td>
<td>( \sum_{k=1}^M \langle x, \tilde{\varphi}_k \rangle \varphi_k )</td>
</tr>
<tr>
<td></td>
<td>( \Phi ) full rank</td>
<td>( \Phi ) full rank</td>
<td>( \Phi ) full rank</td>
<td>( \Phi ) full rank</td>
</tr>
<tr>
<td></td>
<td>( \Phi \Phi^* = \Phi^* \Phi = I )</td>
<td>( \Phi \Phi^* = I, \tilde{\Phi} = (\Phi^*)^{-1} )</td>
<td>( \Phi \Phi^* = I )</td>
<td>( \Phi \Phi^* = I )</td>
</tr>
<tr>
<td>Norm</td>
<td>Yes, ( |x|^2 = \sum_{k=1}^N</td>
<td>\langle x, \varphi_k \rangle</td>
<td>^2 )</td>
<td>No</td>
</tr>
<tr>
<td>Preserv.</td>
<td>( \sum_{k=1}^N</td>
<td>\langle x, \varphi_k \rangle</td>
<td>^2 )</td>
<td>No</td>
</tr>
<tr>
<td>Successive approx.</td>
<td>Yes, ( \hat{x}^{(k)} = \hat{x}^{(k-1)} + \langle x, \varphi_k \rangle \varphi_k )</td>
<td>No</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>Redundant</td>
<td>No</td>
<td>No</td>
<td>Yes</td>
<td>Yes</td>
</tr>
</tbody>
</table>
Historical Remarks

The choice of the title for this chapter requires at least some acknowledgment of the two mathematical giants figuring in it: Euclid and Hilbert.

Little is known about Euclid (around 300) apart from his writings. He was a Greek mathematician, who lived and worked in Alexandria, Egypt. His book *Elements* [82] (in fact, 13 books), remains the most successful textbook in the history of mathematics. In it, he discusses and introduces many topics, most of which have taken hold in our consciousness as immutable truths, such as the principles of Euclidean geometry (2D and 3D), developed from a set of axioms. He also has numerous results on number theory, among which the procedure for finding the greatest common divisor of two numbers. Moreover, it was he who introduced the formal method of proving theorems, on which all mathematical knowledge today is based. The development of non-Euclidean geometry, usually attributed to Lobachevsky, although developed in parallel by Bolyai and to a certain degree Gauss, led to a “parallel universe” where “There exist two lines parallel to a given line through a given point not on the line.”.

David Hilbert (1862–1943) was a German mathematician, to the diligent readers of this chapter known as the creator of the Hilbert space (infinite-dimensional Euclidean space). He is known for the axiomatization of geometry, supplanting the original five Euclid’s axioms. He is one of the most universal mathematicians of all time, contributing towards knowledge in functional analysis, number theory and physics, among others. His efforts towards banishing theoretical uncertainties in mathematics ended in failure: “Gödel demonstrated that any non-contradictory formal system, which was comprehensive enough to include at least arithmetic, cannot demonstrate its completeness by way of its own axioms.” [167]. He is famous for the list of 23 unsolved problems, generally thought to be the most thoughtful and comprehensive such list ever. He worked closely with another famous mathematician, Minkowsky, and had as students or assistants such illustrious names as Weyl, Von Neumann, Courant and many others. He taught all his life, first at the University of Königsberg and then at the University of Göttingen, where he died in 1943. On his tombstone, one of his famous sayings is inscribed:

*Wir müssen wissen.* (We must know.)

*Wir werden wissen.* (We will know.)
Books by Kreyszig [104], Luenberger [111], Gohberg and Goldberg citeGohbergG:81, and Young [170] provide details on abstract vector spaces.

More on the Dirac function can be found in [122]. Details on polynomial matrices are covered in [72], while self-contained presentations on polynomial matrices can be found in [92, 154]. Parameterizations of unitary matrices in various forms, such as using Givens rotations or Householder building blocks are given in [154].

Daubechies in [52] discusses Riesz bases in some detail. More on frames can be found in [36, 98].

**Gram-Schmidt Orthogonalization Procedure for Frames**

While we said that the Gram-Schmidt orthogonalization procedure orthogonalizes a nonorthogonal basis, it can actually be used to orthogonalize any set of \( M \) vectors \( \{ x_k \}_{k=1}^{M} \) in a space of dimension \( N \). When \( M < N \), the result is a set of \( N \) orthonormal vectors \( \{ \varphi_k \}_{k=1}^{N} \); when \( M = N \), the result is an ONB \( \{ \varphi_k \}_{k=1}^{N} \); while when \( M > N \), the result is an ONB \( \{ \varphi_k \}_{k=1}^{N} \); while when \( M > N \), plus \( (M-N) \) zero vectors. The question of how in this last case one can obtain a Parseval TF instead, is answered by Casazza and Kutyniok in [32].

**Exercises with Solutions**

1.1. **Vector Spaces**

   (i) Check that \( C^N \) is a vector space.

   (ii) Show that \( \ell^p(\mathbb{Z}) \) is a vector space.

   (*Hint:* Think of \( \ell^p(\mathbb{Z}) \) as a subspace of the vector space of all sequences of complex numbers and use Minkowski inequality:
   \[
   \left( \sum_{k=1}^{\infty} |x_k + y_k|^p \right)^{1/p} \leq \left( \sum_{k=1}^{\infty} |x_k|^p \right)^{1/p} + \left( \sum_{k=1}^{\infty} |y_k|^p \right)^{1/p}.
   \]

   (iii) Now, consider the vector space \( C^N \) of finite sequences \( x = [x_1 \ x_2 \ \cdots \ x_N]^T \). Prove that \( v_1 \) and \( v_2 \) are norms on \( C^N \) where
   \[
   v_1(x) = \sum_{k=1}^{N} |x_k|, \quad v_2(x) = \left( \sum_{k=1}^{N} |x_k|^2 \right)^{1/2}.
   \]

   (*Hint:* For \( v_2 \), use Minkowski inequality.)

**Solution:**

(i) To prove that \( C^N \) is a vector space, we need to check that the conditions stated in Definition 1.1 are satisfied. We prove the following for any \( x, y \) and \( z \) in \( C^N \). While these are rather trivial, we go through the details once.

   **Commutativity:**
   \[
   x + y = [x_1 \ x_2 \ \cdots \ x_N]^T + [y_1 \ y_2 \ \cdots \ y_N]^T
   = [x_1 + y_1 \ x_2 + y_2 \ \cdots \ x_N + y_N]^T
   = [y_1 + x_1 \ y_2 + x_2 \ \cdots \ y_N + x_N]^T
   = [y_1 \ y_2 \ \cdots \ y_N]^T + [x_1 \ x_2 \ \cdots \ x_N]^T = y + x.
   \]
Exercises with Solutions

**Associativity:**

\[(x + y) + z = \left[ x_1 + y_1 \quad x_2 + y_2 \quad \cdots \quad x_N + y_N \right]^T + \left[ z_1 \quad z_2 \quad \cdots \quad z_N \right]^T = \left[ x_1 + z_1 \quad x_2 + z_2 \quad \cdots \quad x_N + z_N \right]^T = x + z + z\]

\[\left[ x_1 \quad x_2 \quad \cdots \quad x_N \right]^T + \left[ y_1 + z_1 \quad y_2 + z_2 \quad \cdots \quad y_N + z_N \right]^T\]

\[= x + (y + z),\]

and

\[\langle \alpha \beta \rangle x = \left[ \langle \alpha \beta \rangle x_1 \quad \langle \alpha \beta \rangle x_2 \quad \cdots \quad \langle \alpha \beta \rangle x_N \right]^T = \langle \alpha \beta x_1 \rangle \quad \langle \alpha \beta x_2 \rangle \quad \cdots \quad \langle \alpha \beta x_N \rangle) = \langle \alpha \beta x \rangle.\]

**Distributivity:** Follows similarly to the above two.

**Additive identity:** The element \(\mathbf{0} = [0 \quad 0 \quad \cdots \quad 0]^T \in \mathbb{C}^N\) is unique, since all its components \((0 \in \mathbb{C})\) are unique, and

\[x + \mathbf{0} = \left[ x_1 + 0 \quad x_2 + 0 \quad \cdots \quad x_N + 0 \right]^T = \left[ 0 + x_1 \quad 0 + x_2 \quad \cdots \quad 0 + x_N \right]^T = \mathbf{0} + x = \left[ x_1 \quad x_2 \quad \cdots \quad x_N \right]^T = x.\]

**Additive inverse:** The element \((-x) = [-x_1 \quad -x_2 \quad \cdots \quad -x_N]^T \in \mathbb{C}^N\) is unique, since \((-x_i)\) for \(i = 1, 2, \ldots, N\) are unique in \(\mathbb{C}\), and

\[x + (-x) = \left[ x_1 + (-x_1) \quad x_2 + (-x_2) \quad \cdots \quad x_N + (-x_N) \right]^T = \left[ (-x_1) + x_1 \quad (-x_2) + x_2 \quad \cdots \quad (-x_N) + x_N \right]^T = \left[ 0 \quad 0 \quad \cdots \quad 0 \right]^T = \mathbf{0}.\]

**Multiplicative identity:** Follows similarly to additive identity.

Thus \(\mathbb{C}^N\) is a vector space. Note that all the arguments above rely on the fact that \(\mathbb{C}\) itself is a vector space, and thus addition and scalar multiplication satisfy all the necessary properties. The finite cross product of vector spaces is a vector space. That is, let \(V_1, V_2, \ldots, V_N\) be \(N\) vector spaces. We denote by \(V = V_1 \times V_2 \times \cdots \times V_N\) the set of sequences \(x = [x_1 \quad x_2 \quad \cdots \quad x_N]^T\) where \(x_1 \in V_1, \ldots, x_N \in V_N\). Then \(V\) is a vector space with the following operations:

\[x + y = \left[ x_1 \oplus V_1 \quad y_1 \quad \cdots \quad x_N \oplus V_N \quad y_N \right]^T\]

where “\(\oplus V_i\)” is the addition in \(V_i\) for \(i = 1, \ldots, N\), and

\[\alpha x = \left[ \alpha \circ V_1 \quad x_1 \quad \cdots \quad \alpha \circ V_N \quad x_N \right]^T\]

where “\(\circ V_i\)” is the scalar multiplication in \(V_i\) for \(i = 1, \ldots, N\).

(ii) Since \(\ell^p(\mathbb{Z})\) is a subset of all sequences of complex numbers, to prove that it is a vector space, it suffices to show that, if \((x_k), (y_k) \in \ell^p(\mathbb{Z})\) and \(\lambda \in \mathbb{C}\), then \((\lambda x_k) \in \ell^p(\mathbb{Z})\) and \((x_k + y_k) \in \ell^p(\mathbb{Z})\). To check the first property, note that

\[\sum_{k=1}^{\infty} |\lambda x_k|^p = |\lambda|^p \sum_{k=1}^{\infty} |x_k|^p < \infty.\]

The second property \(\sum_{k=1}^{\infty} |x_k + y_k|^p < \infty\) follows immediately from Minkowski inequality:

\[\left( \sum_{k=1}^{\infty} |x_k + y_k|^p \right)^{1/p} \leq \left( \sum_{k=1}^{\infty} |x_k|^p \right)^{1/p} + \left( \sum_{k=1}^{\infty} |y_k|^p \right)^{1/p} < \infty.\]
(iii) To prove that \( v \) is a norm on a vector space \( V \), we need to check the conditions from Definition 1.7. For \( v_1 \):

(i) \( v_1(x) = \sum_{k=1}^{N} |x_k| \) is always positive for any \( x \in \mathbb{C}^N \) since it is a finite sum of positive terms (\( |x_k| \geq 0 \), for all \( x_k \in \mathbb{C} \), and

\[
v_1(x) = 0 \iff \sum_{k=1}^{N} |x_k| = 0
\]

\[
\iff |x_k| = 0, \ k \in \{1, 2, \ldots, N\} \text{ (since all terms in the sum are \( \geq 0 \))}
\]

\[
\iff x_k = 0, \ k \in \{1, 2, \ldots, N\}
\]

\[
\iff x = 0.
\]

(ii)

\[
v_1(\lambda x) = \sum_{k=1}^{N} |\lambda x_k| = |\lambda| \sum_{k=1}^{N} |x_k| = |\lambda| v_1(x).
\]

(iii)

\[
v_1(x + y) = \sum_{k=1}^{N} |x_k + y_k| \leq \sum_{k=1}^{N} |x_k| + |y_k| \text{ (triangle inequality in } \mathbb{C})
\]

\[
= \sum_{k=1}^{N} |x_k| + \sum_{k=1}^{N} |y_k| = v_1(x) + v_1(y).
\]

Now, let us check that \( v_2 = \left( \sum_{k=1}^{N} |x_k|^2 \right)^{1/2} \) satisfies the properties of a norm:

(i) \( v_2(x) \) is always positive for any \( x \in \mathbb{C}^N \) since it is a square root of a finite sum of positive terms (\( |x_k|^2 \geq 0 \), for all \( x_k \in \mathbb{C} \), and

\[
v_2(x) = 0 \iff \left( \sum_{k=1}^{N} |x_k|^2 \right)^{1/2} = 0
\]

\[
\iff |x_k|^2 = 0, \ k \in \{1, 2, \ldots, N\} \text{ (since all terms in the sum are } \geq 0)\]

\[
\iff x_k = 0, \ k \in \{1, 2, \ldots, N\}
\]

\[
\iff x = 0.
\]

(ii)

\[
v_2(\lambda x) = \left( \sum_{k=1}^{N} |\lambda x_k|^2 \right)^{1/2} = \left( \sum_{k=1}^{N} (|\lambda|^2 |x_k|^2) \right)^{1/2} = |\lambda| v_2(x).
\]

(iii) Using Minkowski inequality with \( p = 2 \), we have

\[
v_2(x + y) = \left( \sum_{k=1}^{N} |x_k + y_k|^2 \right)^{1/2} \leq \left( \sum_{k=1}^{N} |x_k|^2 \right)^{1/2} + \left( \sum_{k=1}^{N} |y_k|^2 \right)^{1/2}
\]

\[
= v_2(x) + v_2(y).
\]

1.2. Systems of Linearly Independent Vectors

Given a set of \( M \)-dimensional linearly independent vectors \( \{a_1, \ldots, a_N\} \) and a vector \( b \) in \( \mathbb{R}^M \) outside their span.

(i) Is the vector set \( \{a_1, \ldots, a_N, b\} \) a basis? Explain.

(ii) Give an expression for the distance between \( b \) and the projection of \( x \in \mathbb{R}^N \) onto \( A = \{a_1, \ldots, a_N\} \).

(iii) Write the equations for the components of the error vector.

(iv) Find the least-squares solution \( \hat{x} \) and show that the projection of \( b \) onto the space spanned by the columns of \( A \) can be computed with the use of \( P = A(A^T A)^{-1} A^T \).
Exercises with Solutions

(v) Prove that $P$ is an orthogonal projection matrix.

Solution:

(i) Yes. The vectors $a_i$ are given as linearly independent, and $b$ cannot be written as a linear combination of these vectors, $a_i$, because it is outside their span. That is, $b$ is linearly independent to the vector set $\{a_1, \ldots, a_N\}$. Therefore, the vector set $\{a_1, \ldots, a_N, b\}$ is a new set of $(N+1)$ linearly independent vectors. This set spans the space $\mathbb{R}^{N+1}$ because none of its vectors can be zero, given that they are linearly independent. Thus, $\{a_1, \ldots, a_N, b\}$ is a basis of $\mathbb{R}^{N+1}$. Note also that we must have $M \geq N + 1$ for the set to be linearly independent.

(ii) This can be simply stated as $d = \|Ax - b\|$, where $\|\cdot\|$ is the $L^2$-norm.

(iii) If $\hat{x}$ is the least squares solution to the system, we can write the equations as follows:

$$a_i^T(b - \hat{x}) = 0, \quad i \in \{1, 2, \ldots, N\}.$$ 

(iv) The equations in (iii) can be rewritten as

$$A^T(b - \hat{x}) = 0 \quad \Rightarrow \quad A^Tb = A^T\hat{x},$$

which are the normal equations. Their solution can be obtained by taking partial derivatives of the error $E^2 = (Ax - b)^T(Ax - b)$:

$$2A^TAX - 2A^Tb = 0 \quad \Rightarrow \quad A^TAX = A^Tb,$$

and solving for (the best) $\hat{x}$:

$$\hat{x} = (A^TA)^{-1}A^Tb.$$ 

$A^T$A is invertible because $A$ has linearly independent columns.
This $\hat{x}$ is the solution to the system $Ax = b$, and by substitution we find that

$$A\hat{x} = A(A^TA)^{-1}A^Tb,$$

where $A(A^TA)^{-1}A^T = P$, and $A\hat{x} = Pb$ is the projection of $b$ onto the column space of $A$.

(v) We need to prove that $P = P^2$ (idempotency) and $P = P^T$ (self-adjointness):

$$P^2 = A(A^TA)^{-1}A^T A(A^TA)^{-1}A^T = A(A^TA)^{-1}A^T = P$$

$$P^T = (A(A^TA)^{-1}A^T)^T = ((A^TA)^{-1}A^T)^T A^T$$

$$= A((A^TA)^{-1})^T A^T = A(A^TA)^{-1}A^T = P.$$ 

1.3. Bases and Frames

You are given the following matrix:

$$U = \begin{bmatrix} 0 & \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \end{bmatrix} = \begin{bmatrix} u_1 & u_2 & u_3 \end{bmatrix}.$$ 

(i) Does $U$ define a basis? If yes, what kind (ONB or general) and for which space?

(ii) Take

$$\Phi^T = \begin{bmatrix} u_1 & u_2 \end{bmatrix}.$$ 

Is $\Phi$ a basis for $\mathbb{R}^2$? Why? If yes, what kinds of basis, if not, what is $\Phi$?

(iii) Does $\Phi$ preserve the norm in any way?

(iv) Can you say something about the mapping

$$U = \begin{bmatrix} u_1 & u_2 & u_3 \end{bmatrix} \quad \rightarrow \quad \Phi' = \begin{bmatrix} u_1 & u_2 & 0 \end{bmatrix}.$$
Chapter 1. From Euclid to Hilbert

Solution:

(i) Yes, it does define a basis. It is an orthonormal basis for $\mathbb{R}^3$ since $UU^T = U^T U = I$.

(ii) 

$$
\Phi = \begin{bmatrix}
\sqrt{\frac{2}{3}}
& \sqrt{\frac{1}{\sqrt{2}}}
& -\sqrt{\frac{1}{\sqrt{6}}}
\end{bmatrix} = [\varphi_1 \varphi_2 \varphi_3].
$$

$\Phi$ is not a basis since $\{\varphi_i\}_{i=1}^3$ are not linearly independent: $\varphi_1 = -\varphi_2 - \varphi_3$. $\Phi$ is a tight frame since $\Phi\Phi^T = I$.

(iii) Yes, since

$$
\sum_{k=1}^3 |\langle x, \varphi_k \rangle|^2 = \frac{2}{3}(x_2)^2 + \left(\frac{1}{\sqrt{2}}x_1 - \frac{1}{\sqrt{6}}x_2\right)^2 + \left(\frac{1}{\sqrt{2}}x_1 - \frac{1}{\sqrt{6}}x_2\right)^2 = \|x\|^2.
$$

(iv) The operation defines an orthogonal projection $P$, $\Phi' = PU$. We can compute $P$ as

$$
P = \Phi'U^T = \begin{bmatrix}
-\frac{4}{3} & -\frac{1}{3} & -\frac{1}{3}

\end{bmatrix}.
$$

$P$ is an orthogonal projection since $P^2 = P$ and $P^T = P$.

1.4. Gaussian Elimination

Our aim is to solve the system of linear equations $Ax = y$ (general conditions for existence of a solution are given in Section 1.C.2). The algorithm works in two steps: (1) Eliminating variables one by one using elementary row operations, resulting in either a triangular or echelon form (a matrix in an echelon row form is such that all zero rows are below the nonzero ones, and the first nonzero coefficient in each row is strictly to the right of such a coefficient in the row above). (2) Back substituting variables to find the solution.

Comment on whether the solution to each of the following systems of equations exists, and if it does, find it.

(i) 

$$
A = \begin{bmatrix}
1 & 0 & 3
4 & 5 & 2
-1 & -1 & 2
\end{bmatrix}, \quad y = \begin{bmatrix} 10 \\
20 \\
3 
\end{bmatrix}.
$$

Solution:

(i) We start by forming the “augmented” matrix $AU$ (by concatenating $y$ as the last column):

$$
AU = \begin{bmatrix}
1 & 0 & 3 & | & 10
4 & 5 & 2 & | & 20
-1 & -1 & 2 & | & 3
\end{bmatrix}.
$$

We now go through a sequence of row operations: (a) multiply first row by $-4$ and add to second, then add first row to third, (b) divide second row by $5$ and add to
Exercises with Solutions

third, leading to

\[
AU \begin{bmatrix}
1 & 0 & 3 & 10 \\
0 & 5 & -10 & -20 \\
0 & -1 & 5 & 13
\end{bmatrix}
\begin{bmatrix}
1 & 0 & 3 & 10 \\
0 & 5 & -10 & -20 \\
0 & 0 & 3 & 9
\end{bmatrix}
\]

leading to a unique solution \( x = \begin{bmatrix} 1 & 2 & 3 \end{bmatrix}^T \). This is not surprising as the columns of \( A \) are linearly independent and \( y \) belongs to the column space of \( A \).

(ii) Form the new augmented matrix \( AU \) as described above. The sequence of row operations is now as follows: (a) multiply first row by \(-4\) and add to second, then add first row to third, leading to

\[
AU \begin{bmatrix}
1 & 0 & 2 & 7 \\
0 & 5 & 0 & 10 \\
0 & -1 & 0 & -2
\end{bmatrix}
\]

This system has infinitely many solutions given by \( x = \begin{bmatrix} 7 & -2x_3 & 2 \end{bmatrix}^T \). We can see that \( A \) is singular and \( y \) belongs to the column space of \( A \).

(iii) A similar sequence as above leads to

\[
AU \begin{bmatrix}
1 & 0 & 2 & 1 \\
0 & 5 & 0 & -2 \\
0 & -1 & 0 & 4
\end{bmatrix}
\]

which clearly has no solution (\( y \) does not belong to the column space of \( A \)).

1.5. Gram-Schmidt Orthogonalization

Using Gram-Schmidt procedure, orthogonalize the following three vectors in \( \mathbb{R}^4 \): \( x_1 = \begin{bmatrix} 1 & -1 & 0 & 0 \end{bmatrix}^T \), \( x_2 = \begin{bmatrix} 0 & 2 & 1 & 1 \end{bmatrix}^T \), \( x_3 = \begin{bmatrix} 1 & -1 & 1 & 1 \end{bmatrix}^T \).

Solution:

Following a similar sequence of steps as in Example 1.11:

1. Normalize first the first vector:

\[
\varphi_1 = \frac{x_1}{\|x_1\|} = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & -1 & 0 \end{bmatrix}^T.
\]

2. Project the second vector onto the subspace spanned by \( x_1 \):

\[
v_2 = \langle x_2, \varphi_1 \rangle \varphi_1 = \begin{bmatrix} -1 & 1 & 0 \end{bmatrix}^T.
\]

3. Find the difference between the vector \( x_2 \) and its projection \( v_2 \). This difference is orthogonal to the subspace spanned by \( \varphi_1 \). Then, normalize that difference to get the second vector \( \varphi_2 \):

\[
\varphi_2 = \frac{x_2 - v_2}{\|x_2 - v_2\|} = \frac{1}{2} \begin{bmatrix} 1 & 1 & 1 \end{bmatrix}^T.
\]

4. We now project the third vector \( x_3 \) to the subspace spanned by \( \varphi_1 \) and \( \varphi_2 \) as

\[
v_3 = \langle x_3, \varphi_1 \rangle \varphi_1 + \langle x_3, \varphi_2 \rangle \varphi_2 = \begin{bmatrix} \frac{3}{2} & -\frac{3}{2} & \frac{1}{2} \end{bmatrix}^T.
\]

5. We now must find the difference between this vector \( v_3 \) and \( x_3 \), which is, by construction, orthogonal to the subspace spanned by \( \varphi_1 \) and \( \varphi_2 \). That normalized difference will be our third orthonormal vector:

\[
\varphi_3 = \frac{x_3 - v_3}{\|x_3 - v_3\|} = \frac{1}{2} \begin{bmatrix} -1 & -1 & 1 \end{bmatrix}^T.
\]
6. To visualize these vectors in an easy manner, we write them as columns of a matrix \( \Phi \): they obviously do not form a basis for \( \mathbb{R}^4 \), but rather, just an orthonormal set.

\[
\Phi = \begin{bmatrix}
\frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} & 0 & 0 \\
\frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{2}} & 0 & 0 \\
0 & 0 & \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \\
0 & 0 & \frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{2}}
\end{bmatrix}.
\]

You can change the initial ordering of vectors and compute the appropriate orthonormal sets.

### 1.6. Eigenvalues and Eigenvectors

Consider the matrices \( A = \begin{bmatrix} 1 & 2 \\ 2 & 1 \end{bmatrix} \) and \( B = \begin{bmatrix} \alpha & \beta \\ \beta & \alpha \end{bmatrix} \) where \( \alpha \) and \( \beta \) cannot be equal to zero at the same time.

(i) Give the eigenvalues and eigenvectors for \( A \) and \( B \) (make sure that the eigenvectors are of norm 1).

(ii) Show that \( A = V \Phi V^T \) where the columns of \( V \) correspond to the eigenvectors of \( A \) and \( \Phi \) is a diagonal matrix whose main diagonal corresponds to the eigenvalues of \( A \). Is \( V \) a unitary matrix? Why?

(iii) Check your results using the built-in Matlab function \( \text{eig} \).

(iv) Compute the determinants of \( A \) and \( B \). Is \( A \) invertible? If it is, give its inverse; if not, say why.

(v) When is the matrix \( B \) invertible? Compute \( B^{-1} \) when it exists.

**Solution:**

(i) The characteristic polynomial of \( A \) is

\[
\chi(\lambda) = \det(\lambda I - A) = \det \begin{bmatrix} \lambda - 1 & -2 \\ -2 & \lambda - 1 \end{bmatrix} = (\lambda - 1)^2 - 4 = (\lambda - 3)(\lambda + 1).
\]

Thus the eigenvalues of \( A \) are \( \lambda_1 = -1 \) and \( \lambda_2 = 3 \).

For \( \lambda_1 = -1 \), we solve \( Ax = -x \) with \( x = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}^T \). That is,

\[
\begin{cases}
  x_1 + 2x_2 = -x_1 \\
  2x_1 + x_2 = -x_2,
\end{cases}
\]

which leads to \( x_2 = -x_1 \). We take \( x_1 = 1, x_2 = -1 \) and normalize. Thus the eigenvector associated with \( \lambda_1 \) is \( v_1 = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 \\ -1 \end{bmatrix}^T \).

Similarly, for \( \lambda_2 = 3 \), we solve the system

\[
\begin{cases}
  x_1 + 2x_2 = 3x_1 \\
  2x_1 + x_2 = 3x_2,
\end{cases}
\]

which leads to \( x_1 = x_2 \). We take \( x_1 = x_2 = 1 \) and normalize. The second eigenvector is \( v_2 = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 \\ 1 \end{bmatrix}^T \).

The characteristic polynomial of \( B \) is

\[
\psi(\lambda) = \det(\lambda I - B) = (\lambda - \alpha)^2 - \beta^2 = (\lambda - (\alpha + \beta))(\lambda - (\alpha - \beta)).
\]

Thus the eigenvalues of \( B \) are \( \lambda_1 = \alpha - \beta \) and \( \lambda_2 = \alpha + \beta \).

For \( \lambda_1 = \alpha - \beta \), we solve \( Bx = (\alpha - \beta)x \) assuming that \( \beta \neq 0 \).

\[
\begin{cases}
  \alpha x_1 + \beta x_2 = (\alpha - \beta)x_1 \\
  \beta x_1 + \alpha x_2 = (\alpha - \beta)x_2,
\end{cases}
\]

thus \( x_2 = -x_1 \). The eigenvector associated with \( \lambda_1 \) is \( v_1 = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 \\ -1 \end{bmatrix}^T \).
Similarly, for \( \lambda_2 = \alpha + \beta \) assuming \( \beta \neq 0 \), we solve the system
\[
\begin{align*}
\alpha x_1 + \beta x_2 &= (\alpha + \beta)x_1 \\
\beta x_1 + \alpha x_2 &= (\alpha + \beta)x_2,
\end{align*}
\]
which leads to \( x_1 = x_2 \). The second eigenvector of \( B \) is \( v_2 = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 \\ 1 \end{bmatrix}^T \).

If \( \beta = 0 \) (and \( \alpha \neq 0 \)), then \( B \) has one eigenvalue \( \lambda = 1 \) (with multiplicity 2) and two associated eigenvectors: \( \begin{bmatrix} 1 \\ 0 \end{bmatrix}^T, \begin{bmatrix} 0 \\ 1 \end{bmatrix}^T \).

(ii) Let \( V = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 \\ -1 \end{bmatrix} \), then
\[
VDV^T = \frac{1}{2} \begin{bmatrix} 1 & 1 \\ -1 & 1 \end{bmatrix} \begin{bmatrix} -1 & 0 \\ 0 & 3 \end{bmatrix} \begin{bmatrix} 1 & -1 \\ 1 & 1 \end{bmatrix} = \frac{1}{2} \begin{bmatrix} 1 & 1 \\ -1 & 1 \end{bmatrix} \begin{bmatrix} -1 & 3 \\ 1 & 3 \end{bmatrix} = \frac{1}{2} \begin{bmatrix} 2 & 4 \\ 4 & 2 \end{bmatrix} = A. 
\]

One can also verify that \( B = V \begin{bmatrix} \alpha - \beta & 0 \\ 0 & \alpha + \beta \end{bmatrix} V^T. \)

\( V \) is a unitary matrix since \( VV^T = V^TV = I \). Actually, you do not have to compute \( VV^T \) (or \( V^TV \)) to see that \( V \) is unitary. The columns of \( V \) are made of the eigenvectors of \( A \). These vectors form an orthonormal set (in \( \mathbb{R}^2 \)), therefore \( V \) is unitary.

(iii) \( [V, D] = eig(A) \) produces a diagonal matrix \( D \) of eigenvalues and a full matrix \( V \) whose columns are the corresponding eigenvectors so that \( AV =VD^T \) (from the Matlab help).

(iv) Knowing the eigenvalues of \( A \), it is easy to compute its determinant since \( \det(A) = \prod_{i=1}^n \lambda_i = 1 \lambda_1 \lambda_2 = -3. \) Thus \( A \) is an invertible matrix (its determinant is not equal to zero, or equivalently, all its eigenvalues are not equal to zero), and its inverse is
\[
A^{-1} = -\frac{1}{3} \begin{bmatrix} 1 & -2 \\ -2 & 1 \end{bmatrix}. 
\]

Note that one could have computed \( A^{-1} \) using the fact that \( A = VDV^T \) which leads to \( A^{-1} = VD^{-1}V^T \) (\( V \) is unitary) with \( D^{-1} = \begin{bmatrix} \frac{1}{\lambda_1} & 0 \\ 0 & \frac{1}{\lambda_2} \end{bmatrix} \). This way of computing the inverse can be used when the size of the matrices gets bigger than 2 (under the assumption that these matrices can be diagonalized, that is, can be written as \( A = VDV^T \) where the columns of \( V \) correspond to the eigenvectors of \( A \) and \( D \) is a diagonal matrix whose main diagonal corresponds to the eigenvalues of \( A \)).

We have \( \det(B) = (\alpha - \beta)(\alpha + \beta). \)

(v) \( B \) is not invertible if and only if \( \det(B) = 0 \), which is equivalent to \( \alpha = \beta \) or \( \alpha = -\beta. \)

Therefore, to compute \( B^{-1} \), suppose that \( \alpha \neq \beta \) and \( \alpha \neq -\beta. \) Then,
\[
B^{-1} = \frac{1}{(\alpha - \beta)(\alpha + \beta)} \begin{bmatrix} \alpha & -\beta \\ -\beta & \alpha \end{bmatrix}. 
\]

Note that when \( \alpha = 1, \beta = 2 \), we have \( B = A \) and \( B^{-1} = A^{-1} \) (you can use this as a sanity check for your computations).
(i) Cauchy-Schwarz inequality given in (1.19).
(ii) Triangle inequality given in Definition 1.7.
(iii) Parallelogram law given in (1.22). Prove it for \( \mathbb{R}^2 \) only.

1.3. \textit{Definition of \( \ell^\infty \)-norm}

Recall the definition of \( \ell^p \)-norm for \( p \in [1, \infty) \) in (1.23). To show that the definition of the \( \ell^\infty \)-norm in (1.24) is the natural extension of (1.23), prove

\[
\lim_{p \to \infty} \|x\|_p = \max_{i=1,2,\ldots,N} |x_i| \quad \text{for any } x \in \mathbb{R}^N.
\]

(\textit{Hint: Normalize } x \textit{ by dividing by the entry of largest magnitude. Compute the limit for the resulting vector.})

1.4. \( p < 1 \) \textit{Pseudonorms}

The equation (1.23) does not yield a valid norm when \( p < 1 \). It can nevertheless be a useful quantity.

(i) Show that part (iii) of Definition 1.7 fails to hold for (1.23) with \( p = 1/2 \). (It suffices to come up with a single example.)

(ii) Show that for \( x \in \mathbb{R}^N \), \( \lim_{p \to 0} \|x\|_p \) gives the number of nonzero components in \( x \).

(\textit{Inspired by—but abusing—this result, it is now common to use } \|x\|_0 \textit{ to denote the number of nonzero components in } x.)

1.5. \textit{Equivalence of Norms on Finite-Dimensional Spaces}

Two norms \( \| \cdot \|_a \) and \( \| \cdot \|_b \) on a vector space \( V \) are called \textit{equivalent} when there exist finite constants \( c_a \) and \( c_b \) such that

\[
\|v\|_a \leq c_a \|v\|_b \quad \text{and} \quad \|v\|_b \leq c_b \|v\|_a \quad \text{for all } v \in V.
\]

Show that the 1-norm, 2-norm, and \( \ell^\infty \)-norm are equivalent by proving

\[
\|x\|_\infty \leq \|x\|_2 \leq \|x\|_1,
\]

and

\[
\|x\|_1 \leq \sqrt{n} \|x\|_2 \leq n \|x\|_\infty.
\]

for all \( x \in \mathbb{R}^N \). Actually, a stronger result is true: All norms on \( \mathbb{R}^N \) are equivalent.

1.6. \textit{Distances Not Necessarily Induced By Norms}

A distance or metric \( d : V \times V \to \mathbb{R} \) is a function with the following properties:

(i) \textit{Nonnegativity:} \( d(x, y) \geq 0 \) for every \( x, y \) in \( V \).

(ii) \textit{Symmetry:} \( d(x, y) = d(y, x) \) for every \( x, y \) in \( V \).

(iii) \textit{Triangle inequality:} \( d(x, y) + d(y, z) \geq d(x, z) \) for every \( x, y, z \) in \( V \).

(iv) \( d(x, x) = 0 \) and \( d(x, y) = 0 \) implies \( x = y \).

The \textit{discrete metric} is given by

\[
d(x, y) = \begin{cases} 
0, & \text{if } x = y; \\
1, & \text{if } x \neq y.
\end{cases}
\]

Show that the discrete metric is a valid distance and show that it is not induced by any norm.

1.7. \textit{Nesting of } \ell^p \textit{ Spaces}

Prove that if \( x \in \ell^p(\mathbb{Z}) \) and \( p < q \), then \( x \in \ell^q(\mathbb{Z}) \). This proves (1.27).

1.8. \textit{Completeness of } \mathbb{C}^N

Prove that \( \mathbb{C}^N \) equipped with the \( p \)-norm is a complete vector space for any \( p \in [1, \infty) \) or \( p = \infty \). You may assume that \( \mathbb{C} \) itself is complete.

(\textit{Hint: Show that having a Cauchy sequence in } \mathbb{C}^N \textit{ implies each of the } N \textit{ components is a Cauchy sequence.)}
1.9. **Incompleteness of C([0, 1])**

Consider the sequence of functions \( \{ f_n \} \) on \([0, 1]\) defined by

\[
f_n(t) = \begin{cases} 
0, & t \in [0, \frac{1}{2} - \frac{1}{n}]; \\
(\frac{1}{n} - 1) + 1, & t \in [\frac{1}{2} - \frac{1}{n}, \frac{1}{2}]; \\
1, & t \in [\frac{1}{2}, 1].
\end{cases}
\]

(i) Sketch the sequence of functions.
(ii) Show that \( f_n \) is a Cauchy sequence under the \( L^2 \) norm.
(iii) Show that \( f_n \to f \) under the \( L^2 \) norm for a discontinuous function \( f \). Since \( f \not\in C([0, 1]) \), this shows that \( C([0, 1]) \) is not complete.

1.10. **Closed Subspaces and \( \ell^0(\mathbb{Z}) \)**

Let \( \ell^0(\mathbb{Z}) \) denote the set of complex-valued sequences with a finite number of nonzero entries.

(i) Show that \( \ell^0(\mathbb{Z}) \) is a subspace of \( \ell^2(\mathbb{Z}) \).
(ii) Show that \( \ell^0(\mathbb{Z}) \) is not a closed subspace of \( \ell^2(\mathbb{Z}) \).

1.11. **Operator Norm, Singular Values and Eigenvalues**

For matrix \( A \) (bounded linear operator), show the following:

(i) If the matrix \( A \) is Hermitian, for every nonnegative eigenvalue \( \lambda_k \), there is an identical singular value \( \sigma_i = \lambda_k \). For every negative eigenvalue \( \lambda_k \), there is a corresponding singular value \( \sigma_i = |\lambda_k| \).

(ii) If the matrix \( A \) is Hermitian with eigenvalues \( \lambda_k \),

\[
\|A\|_2 = \max \{|\lambda_k|\}.
\]

(iii) Call \( \mu_k \) the eigenvalues of \( A^*A \); then

\[
\|A\|_2 = \sqrt{\max \{\mu_k\}} = \max \{\sigma_k\}.
\]

1.12. **Adjoint Operators**

Prove the following:

(i) Every bounded linear operator has a unique adjoint.
(ii) Both \( AA^* \) and \( A^*A \) are always self-adjoint (such operators are called normal).

1.13. **When Is an Orthonormal System a Basis?**

Prove Theorem 1.5 (i)-(v), for finite-dimensional Hilbert spaces, \( \mathbb{R}^N \) or \( \mathbb{C}^N \).

1.14. **Parseval’s Formulas**

Parseval’s formulas can be proven by using orthogonality and biorthogonality relations of the basis vectors.

(i) Show relations (1.46a)-(1.46b) using the orthogonality of the basis vectors.
(ii) Show relations (1.52a)-(1.52b) using the biorthogonality of the basis vectors.

1.15. **Dual Frame**

Consider the space \( \mathbb{R}^4 \). It is clear that

\[
\phi_{a,k,n} = \delta_{n-k}
\]

for \( k = 1, \ldots, 4, n = 1, \ldots, 4 \), form a basis for this space.

(i) Consider

\[
\phi_{b,k,n} = \delta_{n-k} - \delta_{n-k-1},
\]

for \( k = 1, \ldots, 3, \) and \( \phi_{b,4,n} = -\delta_{n-1} + \delta_{n-4} \). Show that the 4-dimensional set \( \{\phi_{b,k}\}_{k=1}^{4} \) does not form a basis for \( \mathbb{R}^4 \). Which functions, belonging to \( \mathbb{R}^4 \), are not in the span of \( \{\phi_{b,k}\}_{k=1}^{4} \)?

(ii) Show that \( F = \{\phi_{a,k}, \phi_{b,k}\}_{k=1}^{4} \) is a frame. Compute the frame bounds \( A \) and \( B \).
(iii) Find the dual frame to $F$.  
(Hint: Use the Moore-Penrose generalized inverse. You can compute this generalized inverse in Matlab.)

1.16. Tight Frame with Non-Unit-Norm Vectors

Take the following set of vectors:

$$\varphi_1 = \begin{bmatrix} 0 \\ \sqrt{2 \cos 2\alpha} \end{bmatrix}, \quad \varphi_2 = \begin{bmatrix} \sin \alpha \\ \cos \alpha \end{bmatrix}, \quad \varphi_3 = \begin{bmatrix} -\sin \alpha \\ \cos \alpha \end{bmatrix}.$$  

For which values of $\alpha$ is the above set a tight frame? Draw a few representative cases.

1.17. From Hilbert Spaces to Linear Algebra

Refer to Matrix View of Bases and Frames in Section 1.1.

(i) Using the same basis as in the example, what are the coordinates of the following points in the new coordinate system:  
$$\begin{bmatrix} \sqrt{2} \hat{e}_1 - 1 \tilde{T} \\ \sqrt{2} \hat{e}_1 \end{bmatrix}, \begin{bmatrix} \sqrt{2} \hat{e}_1 1 \tilde{T} \\ 0 \end{bmatrix}, \begin{bmatrix} -1 \hat{e}_1 \tilde{T} \\ 0 \end{bmatrix}.$$  

(ii) Take the following basis (is it orthonormal?):  
$$\varphi_1 = \begin{bmatrix} \hat{e}_2 0 \tilde{T} \\ \hat{e}_1 1 \tilde{T} \end{bmatrix}.$$  

Choose a few points in the plane and find out what their coordinates are in the new coordinate system.

(iii) Construct any singular matrix, and extract vectors from this matrix. Draw these vectors and explain why they do not correspond to a basis.

(iv) On the same vectors you used before (that is,  
$$\begin{bmatrix} \sqrt{2} \hat{e}_1 - 1 \tilde{T} \\ \sqrt{2} \hat{e}_1 \end{bmatrix} \text{ and } \begin{bmatrix} \sqrt{2} \hat{e}_1 1 \tilde{T} \\ 0 \end{bmatrix}),$$  

use the rotation matrix $R$ with $\pi/4$ and convince yourself that you get a rotation by $\pi/4$. Then, try the following rotations and apply them to  
$$\begin{bmatrix} -1 \hat{e}_1 \tilde{T} \\ 0 \end{bmatrix}: 0, \pi/3, \pi/2, \pi, 3\pi/2, 7\pi/4.$$

1.18. Legendre Polynomials

Consider the vectors $1, t, t^2, t^3, \ldots$ in the vector space $L^2([-1, 1])$. Using Gram-Schmidt orthogonalization, find an orthonormal set with the same span.

1.19. Convergence Tests

Let $a_k$ and $b_k$ be two series such that $0 \leq a_k \leq b_k$. Then,

- if $\sum_{k=1}^{+\infty} a_k$ converges, $\sum_{k=1}^{+\infty} b_k$ converges as well,
- if $\sum_{k=1}^{+\infty} a_k$ diverges, $\sum_{k=1}^{+\infty} b_k$ diverges as well.

A ratio test for convergence states that, given  
$$\lim_{k \to +\infty} \left| \frac{a_{k+1}}{a_k} \right| = L,$$  

then if

- $0 \leq L < 1$, the series converges absolutely,
- $1 < L$ or $L = \infty$, the series diverges, and
- $L = 1$, the result is inconclusive, the series could converge or diverge. The convergence needs to be determined another way.

Based on the above, for each of the following series, determine whether it converges or not:

(i) $c_k = \frac{k^2}{k - 3}$.
(ii) $c_k = \frac{\log k}{k}.$
(iii) $c_k = \frac{k}{2^k}$.
(iv) $c_k = \frac{n}{2^n}$.

1.20. Useful Series

In this exercise, we explore a few useful series.
Exercises

(i) Geometric Series: Determine whether it converges, as well as when and how:
\[ \sum_{n=1}^{\infty} x^n. \]  
(P1.20-1)

Prove that when it converges its sum is given by
\[ \frac{1}{1-x}. \]  
(P1.20-2)

(ii) Power Series: Determine whether it converges, as well as when and how:
\[ \sum_{n=1}^{\infty} a_n x^n. \]  
(P1.20-3)

(iii) Taylor Series: If a function \( f(x) \) has \( (n+1) \) continuous derivatives, then it can be expanded into a Taylor series around a point \( a \) as follows:
\[ f(x) = \sum_{k=0}^{\infty} \frac{(x-a)^k}{k!} f^{(k)}(a) + R_n, \quad R_n = \frac{(x-a)^{n+1}}{(n+1)!} f^{(n+1)}(\xi). \]  
(P1.20-4)

Find the Taylor series expansion of \( f(x) = \frac{1}{1-x} \) around a point \( a \).

(iv) MacLaurin Series: For \( a = 0 \), the Taylor series is called the MacLaurin series:
\[ f(x) = \sum_{n=0}^{\infty} \frac{x^n}{n!} f^{(n)}(0) + R_n. \]  
(P1.20-5)

Find the MacLaurin series expansion of \( f(x) = \frac{1}{1-x} \).

<table>
<thead>
<tr>
<th>Function</th>
<th>Expansion</th>
<th>Function</th>
<th>Expansion</th>
</tr>
</thead>
<tbody>
<tr>
<td>( \sin x )</td>
<td>( \sum_{n=0}^{\infty} (-1)^n \frac{x^{2n+1}}{(2n+1)!} )</td>
<td>( \cos x )</td>
<td>( \sum_{n=0}^{\infty} (-1)^n \frac{x^{2n}}{(2n)!} )</td>
</tr>
<tr>
<td>( e^x )</td>
<td>( \sum_{n=0}^{\infty} \frac{x^n}{n!} )</td>
<td>( a^x )</td>
<td>( \sum_{n=0}^{\infty} \frac{(x \ln a)^n}{n!} )</td>
</tr>
<tr>
<td>( \sinh x )</td>
<td>( \sum_{n=0}^{\infty} \frac{x^{2n+1}}{(2n+1)!} )</td>
<td>( \cosh x )</td>
<td>( \sum_{n=0}^{\infty} \frac{x^{2n}}{(2n)!} )</td>
</tr>
<tr>
<td>( \ln(1 + x) )</td>
<td>( \sum_{n=1}^{\infty} \frac{(-x)^n}{n} )</td>
<td>( \frac{1 + x}{1 - x} )</td>
<td>( \sum_{n=1}^{\infty} \frac{2x^{2n-1}}{2n-1} )</td>
</tr>
</tbody>
</table>

Table P1.20-1: Useful MacLaurin series expansions.

(v) Finite Sum: Prove the following formula:
\[ \sum_{k=0}^{N-1} q^k = \frac{1 - q^N}{1 - q}. \]  
(P1.20-6)

1.21. Power of a Matrix

Given a square, invertible matrix \( A \), find an expression for \( A^k \) as a function of the eigenvectors and eigenvalues of \( A \).

1.22. Linear Independence

Find the values of the parameter \( a \in \mathbb{C} \) such that the following set \( U \) is linearly independent
\[ U = \left\{ \begin{bmatrix} 0 & a^2 \\ 0 & j \end{bmatrix}, \begin{bmatrix} 0 & 1 \\ 1 & a - 1 \end{bmatrix}, \begin{bmatrix} 0 & 0 \\ ja & 1 \end{bmatrix} \right\}. \]

For \( a = j \), express the matrix \( \begin{bmatrix} 0 & 5 \\ 2 & j - 2 \end{bmatrix} \) as a linear combination of elements of \( U \).
1.23. **Least-Squares Solution to a Linear System of Equations**

The general solution to this problem was given in (1.88a)–(1.88b).

(i) Show that if \( y \) belongs to the column space of \( A \), then \( \tilde{y} = y \).

(ii) Show that if \( y \) is orthogonal to the column space of \( A \), then \( \tilde{y} = 0 \).

1.24. **Least-Squares Solution**

Show that for the least-squares solution obtained in Section 1.C.2, the partial derivatives
\[ \frac{\partial(|y - \tilde{y}|^2)}{\partial x_i} \]
are all zero.
Chapter 2
Discrete-Time Sequences and Systems

Contents

<table>
<thead>
<tr>
<th>Section</th>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.1</td>
<td>Introduction</td>
<td>80</td>
</tr>
<tr>
<td>2.2</td>
<td>Sequences</td>
<td>82</td>
</tr>
<tr>
<td>2.3</td>
<td>Systems</td>
<td>88</td>
</tr>
<tr>
<td>2.4</td>
<td>Analysis of Sequences and Systems</td>
<td>97</td>
</tr>
<tr>
<td>2.5</td>
<td>Multirate Sequences and Systems</td>
<td>130</td>
</tr>
<tr>
<td>2.6</td>
<td>Stochastic Sequences and Systems</td>
<td>145</td>
</tr>
<tr>
<td>2.7</td>
<td>Algorithms</td>
<td>145</td>
</tr>
<tr>
<td>2.8</td>
<td>Elements of Complex Analysis</td>
<td>148</td>
</tr>
<tr>
<td>2.9</td>
<td>Elements of Algebra: Discrete Polynomials</td>
<td>150</td>
</tr>
<tr>
<td>Chapter at a Glance</td>
<td></td>
<td>151</td>
</tr>
<tr>
<td>Historical Remarks</td>
<td></td>
<td>153</td>
</tr>
<tr>
<td>Further Reading</td>
<td></td>
<td>153</td>
</tr>
<tr>
<td>Exercises with Solutions</td>
<td></td>
<td>154</td>
</tr>
<tr>
<td>Exercises</td>
<td></td>
<td>157</td>
</tr>
</tbody>
</table>

The key word in the title of this chapter is *time*, and the key property of time is that it has an inherent direction; it goes from past to future. The word *discrete* brings about yet another key concept—the time points are equidistant. To move from one time point to the next, we use the *shift* operator. If we consider vectors now to be *sequences* (the domain is discrete time), appropriate Hilbert spaces of such sequences might be \( \ell^1(\mathbb{Z}) \), \( \ell^2(\mathbb{Z}) \) and \( \ell^\infty(\mathbb{Z}) \), all corresponding to real physical phenomena (finite power, finite energy, boundedness). The shift operator can be applied repeatedly to such sequences, forming new sequences. Moreover, it generates a whole space of operators we call *discrete-time systems*, which typically belong to \( \ell^1(\mathbb{Z}) \) or \( \ell^2(\mathbb{Z}) \). As we will see later in the chapter, a shift induces *convolution*, the operation of the discrete-time system on a sequence. Once these concepts are available (shift, sequence and system spaces, convolution), spectral theory allows us to construct an appropriate Fourier transform.
All of the above discussion implicitly assumed that the underlying domain—time, was infinite. While this is true, in practice, we do not observe infinite time, and thus, typically, we only glimpse a finite portion of it. While one might think that dealing with finite time is mathematically easier, a host of issues arises. We discuss these throughout the chapter.

The purpose of this chapter is to study sequences in their different guises and review methods for analyzing and processing such sequences.

2.1 Introduction

While sequences in the real world are often one-sided infinite (they start at some initial point and then go on), for mathematical convenience, we look at them as two-sided infinite:

\[ x = \begin{bmatrix} x_{-2} & x_{-1} & x_0 & x_1 & x_2 & \ldots \end{bmatrix}^T. \]  

(2.1)

For example, if you measure some physical quantity every day at some fixed time (for example, the temperature at noon in front of your house), you obtain a sequence starting at time 0 (say January 14th) and continuing to infinity,

\[ x = \begin{bmatrix} \ldots & 0 & 32^\circ & 29^\circ & 30^\circ & \ldots \end{bmatrix}^T. \]  

(2.2)

Implicit in the index is the fact that \( x_n \) corresponds to the temperature (at noon) on the \( n \)th day. A sequence is also known under the names discrete-time signal (in signal processing) or time series (in statistics); mathematically, these are all vectors, most often in an infinite-dimensional Hilbert space.

In real life, most things are finite, that is, we observe only a finite portion of an infinite sequence. Moreover, computations are always done on finite inputs. For our temperature example, serious observations started in the 18th century and necessarily stop at the present time. Thus, we have a finite sequence of length \( N \),

\[ x = \begin{bmatrix} x_0 & x_1 & x_2 & \ldots & x_{N-1} \end{bmatrix}^T. \]  

(2.3)

Once we have a finite sequence, the questions immediately arises: where does it come from? Is it a truly finite sequence, or, is it a piece of an infinite one? In either case, how do we analyze such a sequence? As we will see later in the chapter, this question immediately calls for a decision on what happens at the boundaries. Do we assume everything outside the sequence support is zero? Do we extend it periodically, symmetrically, ...?

While perhaps not a natural extension, a typical one is periodic; We periodize the finite sequence and assume we are dealing with one period of a periodic sequence with period \( N \in \mathbb{N} \):

\[ x = \begin{bmatrix} \ldots & x_0 & x_1 & \ldots & x_{N-1} & x_0 & x_1 & \ldots \end{bmatrix}^T. \]  

(2.4)

Footnote 7: The "boxing" of the origin is intended to serve as a reference point. When we deal with finite vectors/matrices, this is not necessary; however, when these are infinite, knowing where the origin is, is essential.
While finite sequences in (2.3) and infinite periodic ones in (2.4) are clearly different in nature, we will use the same types of tools to analyze them. We will assume that both belong to some space on the domain \(\{0, 1, \ldots, N - 1\}\) (say \(\ell^1\) or \(\ell^2\)), with circular boundary conditions, implicitly making even the first type periodic. When we want to analyze finite sequences as windowed versions of infinite sequences, we will do just that; we will assume such sequence are infinite in nature with only a finite number of nonzero samples.

These considerations allow us to define two broad classes of sequences:

(i) Those with support in \(\mathbb{Z}\) we will call infinite dimensional; and

(ii) Those with support in \(\{0, 1, \ldots, N - 1\}\) we will call finite dimensional.

**Example 2.1 (Sequences).** As an illustration:

- **Infinite-Dimensional Sequences:** The geometric sequence from (P1.20-1) with \(x = \frac{1}{2}\):

  \[x_n = \left(\frac{1}{2}\right)^n, \quad n \in \mathbb{Z}, \quad x = \left[\ldots 4 2 \frac{1}{2} \frac{1}{2} \ldots\right]^T, \quad (2.5)\]

  is infinite dimensional and not convergent (if we made it nonzero only for \(n \geq 0\), it would be).

- **Finite-Dimensional Sequences:** The sequence obtained by tossing a fair coin \(N\) times is finite dimensional:

  \[x = \left[h h t \ldots h\right]^T.\]

A sinusoidal sequence sampled at submultiples of its period is periodic with period \(N\), and is another example of a finite-dimensional sequence:

\[x_n = \sin\left(\frac{2\pi}{N}n + \theta\right),\]

\[x = \left[\ldots \sin(\theta) \sin\left(\frac{2\pi}{N} + \theta\right) \ldots \sin\left(\frac{2\pi}{N}(N - 1) + \theta\right) \sin(\theta) \ldots\right]^T, \quad \text{one period}\]

Given sequences (signals, vectors), one can apply operators (systems, filters). These map input sequences into output ones, and since they involve discrete-time sequences, they are usually called *discrete-time systems* (operators). Among them, linear ones are most common. Even more restricted is the class of *linear shift-invariant* systems (filters, defined later in the chapter), an example of which is the moving average filter:
Chapter 2. Discrete-Time Sequences and Systems

Example 2.2 (Moving average filter). Consider our temperature example, assume we want to detect seasonal trends. The day-by-day variation might be too noisy, and thus, we can compute a local average

\[ y_n = \frac{1}{2N + 1} \sum_{k=-N}^{N} x_{n+k}, \quad n \in \mathbb{Z}, \tag{2.6} \]

where \( N \) is a small integer. The “local” average eliminates daily variations; this simple filter is linear and shift invariant, since at all \( n \), the same local averaging is performed.

Chapter Outline

The next several sections follow the progression of topics in this brief introduction: We start by formally defining the various types of sequences we discussed above. Section 2.3 considers linear discrete-time systems, especially of the shift-invariant kind, which correspond to difference equations, the discrete-time analog of differential equations. Next, in Section 2.4, we develop the tools to analyze discrete-time sequences and systems, in particular the discrete-time Fourier transform, the discrete Fourier transform and the \( z \)-transform. We discuss the fundamental theorem relating filtering to multiplication in Fourier domain—the convolution theorem. Section 2.5 looks into discrete-time sequences and systems that operate with different rates—multirate systems, which are key for filter bank development in later chapters. Then, we present discrete-time stochastic sequences and systems in Section 2.6, while important algorithms for discrete-time processing, such as the fast Fourier transform, are covered in Section 2.7. Appendix 2.A lists basic elements of complex analysis, while Appendix 2.B discusses polynomial sequences.

2.2 Sequences

2.2.1 Infinite-Dimensional Sequences

The set of sequences as in (2.1), where \( x_n \) is either real or complex, together with vector addition and scalar multiplication, forms a vector space (see Definition 1.1). The inner product between two sequences is defined as in (1.16), and induces the standard \( \ell^2 \), Euclidean norm given in (1.26a), with \( p = 2 \). Other norms of interest are the \( \ell^1 \)-norm given in (1.26a) with \( p = 1 \), and the \( \infty \)-norm given in (1.26b).

As opposed to generic infinite-dimensional spaces, where ordering of indices does not matter in general, discrete-time sequences belong to an infinite-dimensional space where the order of indices is important, since it represents time (or space in two dimensions).

Sequence Spaces

Space of Finite-Energy Sequences \( \ell^2(\mathbb{Z}) \) The constraint of a finite square norm is necessary for turning a simple vector space into a Hilbert space of finite-energy sequences \( \ell^2(\mathbb{Z}) \).
2.2. Sequences

**Space of Finite-Power Sequences** $\ell^\infty(\mathbb{Z})$ A looser constraint is *finite power*, which only bounds the magnitude of the samples. This space, which contains all sequences $x_n$ such that $|x_n| < \infty$, is denoted $\ell^\infty(\mathbb{Z})$, since the $\ell^\infty$-norm is finite. While in practice many sequences are of the latter kind, the former is more convenient because of the Euclidean geometry associated with Hilbert spaces.

**Space of Absolutely-Summable Sequences** $\ell^1(\mathbb{Z})$ One more important class of sequences are absolutely-summable sequences, that is, sequences belonging to $\ell^1(\mathbb{Z})$. We briefly touched upon those in Section 1.2.3 (remember that $\ell^1(\mathbb{Z}) \subset \ell^2(\mathbb{Z})$). By definition, sequences in $\ell^1(\mathbb{Z})$ have a finite $\ell^1$-norm.

**Example 2.3 (Sequence spaces).** Revisiting the sequences seen in Example 2.1, we see that $x_n = \alpha^n$, this time zero for $n < 0, \alpha \in \mathbb{R}$, is in the following spaces:

- $|\alpha| < 1$ $\ell^2(\mathbb{Z}), \ell^1(\mathbb{Z}), \ell^\infty(\mathbb{Z})$,
- $|\alpha| = 1$ $\ell^\infty(\mathbb{Z})$,
- $|\alpha| > 1$ none.

**Geometry in $\ell^2(\mathbb{Z})$** We now recall a few “geometrical” facts from Chapter 1, valid for sequences in $\ell^2(\mathbb{Z})$:

(i) As in (1.17),

$$\langle x, y \rangle = \|x\| \|y\| \cos \theta,$$

where $\theta$ is the angle between the two infinite sequences (vectors).

(ii) As in (1.14), if the inner product is zero,

$$\langle x, y \rangle = 0,$$

then either at least one of the sequences is identically zero, or they are said to be orthogonal to each other.

(iii) As in (1.37), given a unit-norm sequence $y$, $\|y\| = 1$,

$$\hat{x} = \langle x, y \rangle y$$

is the orthogonal projection of the sequence $x$ onto the space spanned by the sequence $y$.

**Special Sequences**

**Dirac impulse Sequence** The simplest and one of the most useful sequences is the *Dirac impulse sequence* at location $k$ (it is also defined in the Quick Reference section at the beginning of the book)

$$\delta_{n-k} = \begin{cases} 1, & \text{for } n = k; \\ 0, & \text{otherwise}, \end{cases} \quad \delta_{n-k} = \left[ \cdots 0 \underbrace{1} \cdots \right]^T, \quad (2.7)$$
where \( n, k \in \mathbb{Z} \). The set of Dirac impulses \( \{ \delta_{n-k} \}_{k \in \mathbb{Z}} \) along the discrete timeline forms an ONB for \( \ell^2(\mathbb{Z}) \). We can easily check that as (a) it is clearly an orthonormal set, and (b) any sequence from \( \ell^2(\mathbb{Z}) \) can be written as a linear combination of the basis vectors \( \varphi_{kn} = \delta_{n-k} \):

\[
x_n = \sum_{k \in \mathbb{Z}} \langle x_n, \varphi_{kn} \rangle \varphi_{kn} = \sum_{k \in \mathbb{Z}} \left( \sum_{n \in \mathbb{Z}} x_n \delta_{n-k} \right) \delta_{n-k} = \sum_{k \in \mathbb{Z}} x_k \delta_{n-k}. \tag{2.8}
\]

This identity, while seemingly trivial, might require a moment of thought: for any \( n \), a single term in the infinite sum that is different from zero is the one for \( k = n \), when the sum equals \( \ldots x_n! \) A list of properties of this sequence is given in Chapter at a Glance.

**Sinc Sequence** With \( n \in \mathbb{Z} \), the sinc sequence is given by:

\[
sinc_{T,n} = \frac{\sin \frac{n\pi}{T}}{\frac{n\pi}{T}} = \frac{e^{j\frac{n\pi}{T}} - e^{-j\frac{n\pi}{T}}}{2j(\frac{n\pi}{T})}. \tag{2.9}
\]

The zeros \( n_k \) of a discrete-domain sinc function are at:

\[
n_k = \frac{T}{\pi} n, \quad n \in \mathbb{Z}, \quad k \in \mathbb{Z}, \tag{2.10}
\]

shown in Figure 1.12(b). As opposed to the continuous-time sinc (1.72), the zeros of the discrete-time sinc sequence will not appear at \( kT \) for all \( k \), but just at those \( kT \) which are integers. An example for \( T = 2 \) gives

\[
sinc_n = \frac{\sin \frac{n\pi}{2}}{\frac{n\pi}{2}} \quad \text{sinc} = \begin{bmatrix} \ldots & -\frac{2}{3\pi} & 0 & \frac{2}{\pi} & 1 & \frac{2}{\pi} & 0 & -\frac{2}{3\pi} & \ldots \end{bmatrix}^T. \tag{2.11}
\]

This sequence is in \( \ell^\infty(\mathbb{Z}) \) and in \( \ell^2(\mathbb{Z}) \), but not in \( \ell^1(\mathbb{Z}) \), since the sequence decays as \( 1/n \) (see the illustration of the inclusion property (1.27) of \( \ell^p \) spaces). The value of sinc at \( t = 0 \) is obtained the same way as for the continuous-domain sinc, leading to:

\[
sinc_{T,0} = 1. \tag{2.12}
\]

**Heaviside Sequence** The *Heaviside* sequence is defined as

\[
u_n = \begin{cases} 1, & n \in \mathbb{N}; \\ 0, & \text{otherwise}, \end{cases} \quad u = \begin{bmatrix} \ldots & 0 & 1 & \ldots \end{bmatrix}^T, \tag{2.13}
\]

that is, a unit-step sequence at the origin. As this sequence is bounded, it belongs to \( \ell^\infty(\mathbb{Z}) \), but not to \( \ell^1(\mathbb{Z}) \) or \( \ell^2(\mathbb{Z}) \). The two sequences we introduced so far—Dirac and Heaviside—are related (via “integration” or “differentiation”):

\[
u_n = \sum_{k=-\infty}^{n} \delta_k.
\]
2.2. Sequences

Window Sequences Most often, both because of real-world considerations and for mathematical convenience, a finite-length sequence is seen as a glimpse of an infinite sequence. One way to state this is by introducing a rectangular window sequence as the indicator sequence of \{0, 1, \ldots, N - 1\},

\[
w_n = \begin{cases} 1, & 0 \leq n \leq N - 1; \\ 0, & \text{otherwise}, \end{cases} \quad w = \ldots 0 \overbrace{1 \ldots 1}^N 0 \ldots. \tag{2.14}
\]

Then, we can define an infinite sequence \(y\) with values given by (2.3) for \(n \in \{0, 1, \ldots, N - 1\}\), and arbitrary otherwise,

\[
y_n = \begin{cases} x_n, & 0 \leq n \leq N - 1; \\ \text{arbitrary}, & \text{otherwise}, \end{cases} \quad y = \ldots y_{-1} \overbrace{x_0 x_1 \ldots x_{N-1}}^N y_N \ldots.
\]

Multiplying \(y_n\) with the window \(w_n\), we obtain the projection \(\hat{y}\):

\[
\hat{y}_n = y_n w_n, \quad \hat{y} = \ldots y_{-1} \overbrace{x_0 x_1 \ldots x_{N-1}}^N 0 \ldots, \tag{2.15}
\]

which equals \(x_n\) for \(n \in \{0, 1, \ldots, N - 1\}\) and is zero otherwise. This allows us to study an infinite sequence \(\hat{y}_n\) that coincides with \(x_n\) looking at it through the window of interest.

How good is the window we just used? For example, if \(y\) is smooth, its projection \(\hat{y}\) is not because of the abrupt boundaries of the rectangular window. We might thus decide to use a different window to smooth the boundaries.

**Example 2.4 (Windows).** Consider an infinite sinusoidal sequence of frequency \(\omega_0\) and phase \(\theta_0\),

\[
y_n = \sin(\omega_0 n + \theta_0),
\]

and the following two windows (in both cases \(N = 2K + 1\)):

(i) Rectangular window of length \(N\).

(ii) A raised cosine window:

\[
w_n = \begin{cases} \cos(\frac{\pi}{K}(n - K)) + 1, & 0 \leq n \leq 2K; \\ 0, & \text{otherwise}. \end{cases} \tag{2.16}
\]

The raised cosine window tapers off smoothly at the boundaries, while the rectangular one does not. The trade-off between the two windows is obvious from Figure 2.1; the rectangular window does not modify the sequence inside the window, but leads to abrupt changes at the boundary, while the raised cosine window has smooth transitions, but at the price of modifying the sequence.
2.2.2 Finite-Dimensional Sequences

As we said, finite-dimensional sequences as in (2.3) are those with the domain in \( \{0, 1, \ldots, N - 1\} \), and can be either seen as truly finite or as a period of a periodic sequence (their domain is a “discrete” circle). Once we start applying operators on such sequences, we start worrying about whether the space they belong to is closed. Operations such as convolution would clearly push elements out of the domain of support, and we must consider how to extend such sequences at the boundaries. One possible extension, and the one that is used in discrete-time signal processing, is circular; the sequence is extended periodically, and in effect, the domain becomes the discrete circle (operations mod \( N \)). We can then denote the space of such sequences as \( \mathbb{R}^N \) or \( \mathbb{C}^N \). Note that we could have chosen another extension; however, this is not typically done. More details on this topic can be found in Further Reading.

As we will see in Section 2.4.3, a natural basis for such sequences are the harmonic complex exponential sequences, that is, \( \mathcal{N} \) sequences \( \varphi_k, k \in \{0, 1, \ldots, N - 1\} \), given by

\[
\varphi_{kn} = e^{j \frac{2\pi kn}{N}}, \quad k \in \{0, 1, \ldots, N - 1\}, \quad n \in \mathbb{Z}.
\]

These sequences are periodic with period \( N \), and their linear combinations span the space of all \( N \)-periodic sequences. In Exercise 2.1, we explore a few properties of complex exponential sequences and their periodicity.

2.2.3 Two-Dimensional and Multidimensional Sequences

Two-Dimensional Sequences

Today, one of the most widespread devices is the digital camera, and the size of the acquired image in megapixels\(^{10}\) is the topic of intense marketing battles. In our notation, a digital picture is a two-dimensional sequence, \( x_{nm} \), and can be seen either as an infinite-dimensional sequences with a finite number of nonzero samples (example of a window view of the world, that is, a finite two-dimensional sequence being part of a larger, but invisible sequence):

\[
x_{nm}, \quad n, m \in \mathbb{Z},
\]

or as a sequence with domain \( n \in \{0, 1, \ldots, N - 1\}, \ m \in \{0, 1, \ldots, M - 1\}, \)

\[
x = \begin{bmatrix}
x_{00} & x_{01} & \ldots & x_{0,M-1} \\
x_{10} & x_{11} & \ldots & x_{1,M-1} \\
\vdots & \vdots & \ddots & \vdots \\
x_{N-1,0} & x_{N-1,1} & \ldots & x_{N-1,M-1}
\end{bmatrix},
\]

\(^8\)We could denote such a space as \( \ell(\mathbb{Z}_N) \) to keep the awareness of the boundary conditions; we choose to go with a simpler, \( \mathbb{R}^N / \mathbb{C}^N \) notation.

\(^9\)In the double subscript, the first, \( k \), denotes the sequence, and the second, \( n \), the domain of the sequence.

\(^{10}\)A megapixel is either a true set of 1000 by 1000 pixels, or more often a combination of 500 000 green and 250 000 red and blue pixels.
2.2. Sequences

assumed to be circularly extended at the borders (similarly to the one-dimensional case). Each element $x_{nm}$ is called a pixel, the total image having $NM$ pixels. In reality, for $x_{nm}$ to represent a color image, it must have more than one component; often, red, green and blue components are used (RGB space). While perhaps such an extension is not natural (the left side of the image appears at the right boundary), it is the one that leads to the use of the DFT, as we will see later in the chapter.

<table>
<thead>
<tr>
<th>Sequence Space</th>
<th>Symbol</th>
<th>Finite Norm</th>
</tr>
</thead>
<tbody>
<tr>
<td>Absolutely-summable</td>
<td>$\ell^1(\mathbb{Z}^2)$</td>
<td>$|x|<em>1 = \sum</em>{n,m \in \mathbb{Z}}</td>
</tr>
<tr>
<td>Finite-energy</td>
<td>$\ell^2(\mathbb{Z}^2)$</td>
<td>$|x|<em>2 = \left(\sum</em>{n,m \in \mathbb{Z}}</td>
</tr>
<tr>
<td>Absolutely summable</td>
<td>$\ell^\infty(\mathbb{Z}^2)$</td>
<td>$|x|<em>\infty = \max</em>{n,m \in \mathbb{Z}}</td>
</tr>
</tbody>
</table>

Table 2.1: Norms and two-dimensional sequence spaces.

**Sequence Spaces** The spaces we introduced in one dimension generalize to multiple dimensions; for example in two dimensions, the inner product, given two sequences $x$ and $y$, is

$$\langle x, y \rangle = \sum_{n \in \mathbb{Z}} \sum_{m \in \mathbb{Z}} x_{nm} y_{nm}^*, \quad (2.20)$$

while the $\ell^2$-norm and the appropriate space $\ell^2(\mathbb{Z}^2)$ are given in Table 2.1 (together with other relevant norms and spaces). For example, a digital picture, having finite size and pixel values that are bounded, clearly belongs to all three spaces defined in Table 2.1. Infinite multidimensional sequences, on the other hand, can be trickier to analyze.

**Example 2.5 (NORMS OF TWO-DIMENSIONAL SEQUENCES).** Consider the sequence

$$x_{nm} = \frac{1}{2^n 3^m}, \quad n, m \in \mathbb{N}. \quad (2.21)$$

Its $\ell^2$-norm can be evaluated as

$$\langle x, x \rangle = \sum_{n \in \mathbb{N}} \sum_{m \in \mathbb{N}} \frac{1}{4^n} \frac{1}{9^m} = \frac{4}{3} \sum_{m \in \mathbb{N}} \frac{1}{9^m} = \frac{4}{3} \cdot \frac{9}{8} = \frac{3}{2},$$

yielding $\|x\| = \sqrt{3/2}$. Similarly, $\|x\|_1 = 3$ and $\|x\|_\infty = 1$. 

**Three-Dimensional Sequences**

Higher-dimensional objects are also possible; we briefly consider the three-dimensional case, video, for example. A video sequence is a sequence of digital pictures we represent as:

$$x_{nmk}, \quad n \in \{0, 1, \ldots, N - 1\}, \quad m \in \{0, 1, \ldots, M - 1\}, \quad k \in \mathbb{Z}, \quad (2.22)$$
Figure 2.2: Multidimensional data sets. (a) Two-dimensional sequence (image), with the first quadrant indicated, as well as a finite $N \times M$ digital picture. (b) Three-dimensional sequence (video). (TBD)

Figure 2.3: A discrete-time system. (TBD)

where $N \times M$ is the size of the individual picture in the sequence, while $k$ is a time index (assumed here to be doubly infinite). The size of multidimensional sequences grows as the product of the sizes in each dimension. For example, a video sequence in (2.22) with $k \in \{0, 1, \ldots, K - 1\}$, has $NMK$ pixels, which quickly leads to enormous amounts of data. Figure 2.2 gives examples of two- and three-dimensional sequences.

2.3 Systems

Discrete-time systems are operators having discrete-time sequences as inputs and outputs. Among all discrete-time systems, we will concentrate on the subclass of linear systems, in particular, linear and shift-invariant (LSI) systems. This last class is both important in practice and amenable to easy analysis. The moving average filter in (2.6) is such an LSI system. After an introduction to difference equations, which are natural descriptions of discrete-time systems, we study LSI systems in detail. We then turn our attention to shift-varying systems and an important subclass, linear periodically shift-varying systems.

2.3.1 Discrete-Time Systems and Their Properties

A discrete-time system maps an input sequence $x \in \mathbb{H}$ into an output sequence $y \in \mathbb{H}$ through an operator $T$ (see Figure 2.3)

$$y_n = T_n x_n. \quad (2.23)$$

The subscript $n$ in $T_n$ indicates that the operator may be changing over time. While in the above, $T_n$ operates only on $x$, in general, it can operate both on $x$ and $y$. While this may seem self-referential, it can be done as long as $y$ is computable.

As a discrete-time system $T_n$ is an operator, everything we learned in Chapter 1 about operators (in particular, linear operators) holds here. In general, operators map a sequence from $\mathbb{H}_1$ to $\mathbb{H}_2$, while here, we stay in the same Hilbert space $\mathbb{H}$, space of sequences.
2.3. Systems

Basic Systems

Shift  The shift-by-$k$ operator is defined as:

$$y_n = T x = x_{n-k}, \quad (2.24a)$$

$$\begin{bmatrix} \ldots & x_{-1} & x_0 & x_1 & \ldots \end{bmatrix} \rightarrow \begin{bmatrix} \ldots & x_{k-1} & x_k & x_{k+1} & \ldots \end{bmatrix}. \quad (2.24b)$$

which simply delays $x_n$ by $k$ samples. While this is one of the simplest discrete-time systems, it is also the most important, as the whole concept of time processing is based on this simple operator. The shift-by-one is usually termed a delay.

Modulation  While the shift we just saw is the shift in time, modulation is shift in frequency (as we will see later in this chapter). A modulation by a sinusoid of frequency $\omega_0$, is given by (we use the cosine here)

$$y_n = T x = \cos(\omega_0 n) \ x_n. \quad (2.25)$$

For those already familiar with Fourier analysis, (2.25) shifts the spectrum of $x$ to a position $\omega_0$ in frequency (and to a corresponding negative frequency $-\omega_0$). Imagine $x_n = \alpha$, for all $n$. Assume now $\omega_0 = \pi$. The corresponding $\cos(\omega_0 n)$ becomes $\cos(\pi n) = (-1)^n$. Thus, a constant sequence $x_n$ turns into a fast-varying sequence $x_n = (-1)^n \alpha$ (shifted in frequency to higher frequency):

$$\begin{bmatrix} \ldots & \alpha & \alpha & \alpha & \ldots \end{bmatrix} \rightarrow \begin{bmatrix} \ldots & -\alpha & \alpha & -\alpha & \alpha & \ldots \end{bmatrix}. $$

Hard Limiter  Given a bound $\Delta$, the hard limiter computes the following output:

$$y_n = T x = \begin{cases} \Delta, & x_n \geq \Delta; \\ -\Delta, & x_n \leq -\Delta; \\ x_n, & \text{otherwise}; \end{cases} \quad (2.26)$$

$$\begin{bmatrix} \ldots & x_{-2} & x_{-1} \underbrace{x_0}_{\geq \Delta} & x_1 & x_2 & \ldots \end{bmatrix} \rightarrow \begin{bmatrix} \ldots & x_{-2} & \underbrace{-\Delta}_{\leq \Delta} & x_1 & -\Delta & \ldots \end{bmatrix}^T.$$

Accumulator  The output of the accumulator is the “integral” of the input:

$$y_n = T x = \sum_{k=-\infty}^{n} x_k.$$

Linear Systems

Definition 2.1 (Linear System). A discrete-time system is linear if $T_n$ is linear, that is, if it satisfies Definition 1.16.

For the examples above, it is easy to verify that except for the hard limiter, the systems are linear.
Chapter 2. Discrete-Time Sequences and Systems

Memoryless Systems

**Definition 2.2 (Memoryless System).** A system $y = T_n x$ is memoryless, if its output $y_n$ at time $n$ depends only on the input $x_n$ at time $n$:

$$ y_n = T x_n. $$

(2.27)

In the examples seen previously, the modulation and the hard limiter are memoryless, while the two others are clearly not.

Shift-Invariant Systems

**Definition 2.3 (Shift-invariant system).** Given an input $x_n$ and an output $y_n$, a system is shift invariant if a shifted input produces a shifted output:

$$ y_n = T x_n \Rightarrow y_{n-k} = T x_{n-k}. $$

(2.28)

The above implies that the operator $T_n$ in (2.23) cannot be dependent on time, and thus $T_n = T$. Shift invariance is often a very desirable property. For example, listening to a piece of music on an MP3 player should produce the same music the first as well as the seventh time. Moreover, shift-invariant systems have desirable mathematical properties, especially when they are also linear. Among the examples studied earlier, only modulation is not shift invariant.

Causal Systems

**Definition 2.4 (Causal system).** A causal system is a system whose output at time $n$ depends only on the present and past inputs:

$$ y_n = T_n x_k, \quad k \leq n. $$

(2.29)

We will see once we define the impulse response of a system, that causality implies that the impulse response $h_n$ satisfies $h_n = 0$ for all $n < 0$. If the domain of our function is time, then any real-world system will necessarily be causal. There are no real systems where the output can depend on future inputs, unless the input is completely deterministic and known in advance (for example, it is a constant). Note that causality is a property of discrete-time systems and not of sequences. However, by abuse of language, we often refer to sequences which are nonzero only for positive $n$ as causal sequences, meaning they could be impulse responses of causal systems. Often times this can lead to confusion since we will try to interpret the statement of causality as “there is no future” and thus will think that the past in the sequence is expressed on the right side of the origin. Of course, this is not true, as a sample at $(n+1)$ definitely comes after the sample at $n$.

Stable Systems

While there are various definitions of stability for systems, we consider bounded input bounded output (BIBO) stability exclusively.
2.3 Systems

Definition 2.5 (BIBO stability). A system is BIBO stable if, for every bounded input \( x \in \ell^\infty(\mathbb{Z}) \), the input \( y = T_n x \) is bounded as well:

\[
  x \in \ell^\infty(\mathbb{Z}) \quad \Rightarrow \quad y \in \ell^\infty(\mathbb{Z}).
\]

In the four example systems seen at the start of the section, the first three are BIBO stable, while the accumulator is not, since, for example, a constant input \( x_n = 1 \) will lead to an unbounded output.

2.3.2 Difference Equations

An important class of discrete-time systems can be described by linear difference equations that relate the input sequence and past outputs to the current output,

\[
y_n = \sum_{k \in \mathbb{Z}} a_k^{(n)} x_{n-k} + \sum_{k=1}^\infty b_k^{(n)} y_{n-k}.
\]

If we require shift invariance, then the coefficients \( a_k^{(n)} \) and \( b_k^{(n)} \) are constant, and we get a linear, constant-coefficient difference equation. To make the system causal, we restrict the inputs \( x_n \) to the current and past inputs, leading to

\[
y_n = \sum_{k=0}^\infty a_k x_{n-k} + \sum_{k=1}^\infty b_k y_{n-k}.
\]

Realizable systems will have a finite number of coefficients \( a_k, k \in \{0,1,\ldots,M\} \) and \( b_k, k \in \{1,2,\ldots,N\} \), leading to

\[
y_n = \sum_{k=0}^M a_k x_{n-k} - \sum_{k=1}^N b_k y_{n-k}.
\]

To find the solution to such a difference equation, we follow a similar path as for ordinary differential equations.

- Homogeneous solution: First, we find a solution to the homogeneous equation,

\[
y_n^{(h)} = \sum_{k=1}^N b_k y_{n-k},
\]

by setting the input \( x \) in (2.32) to zero. The solution is of the form

\[
y_n^{(h)} = \sum_{k=1}^N \beta_k \lambda_k^n,
\]

\[11\]In this discussion, we concentrate on sequences where the index \( n \) corresponds to time, and one solves the system forward in time, as is natural in causal systems. A general treatment of difference equations involves also anti-causal solutions, obtained by solving backward in time.
where \( \lambda^n_k \) are obtained by solving the characteristic equation of the system

\[
\sum_{k=1}^{N} b_k \lambda^{N-k} = 0. \tag{2.33c}
\]

- **Particular solution:** Then, any particular solution to (2.32), \( y^{(p)}_n \), is found (independent of \( y^{(h)}_n \)). This is typically done by assuming that \( y^{(h)}_n \) is of the same form as \( x_n \), possibly scaled.

- **Complete solution:** By superposition, the complete solution \( y_n \) is the sum of the homogeneous solution \( y^{(h)}_n \) and the particular solution \( y^{(p)}_n \):

\[
y_n = y^{(h)}_n + y^{(p)}_n = \sum_{k=1}^{N} \beta_k \lambda^n_k + y^{(p)}_n. \tag{2.34}
\]

We determine the coefficients \( \beta_k \) in \( y^{(h)}_n \) by specifying initial conditions for \( y_n \) and then solving the system.

**Example 2.6.** As an example, consider the accumulator seen previously,

\[
y_n = \sum_{k=\infty}^{n} x_k = x_n + \sum_{k=\infty}^{n-1} x_k = x_n + y_{n-1},
\]

which is of the form (2.32), with \( a_0 = 1 \), \( b_1 = 1 \). The infinite sum has been turned into a recursive formula, which now requires initial conditions. For example, assume the system starts at \( n = 0 \), with initial condition \( y_{-1} \) given:

\[
y_n = y_{-1} + \sum_{k=0}^{n} x_k, \quad n \in \mathbb{N}.
\]

From the above example, we see that unless the initial conditions are zero, the system is not linear. Similarly, the system is shift invariant only if the initial conditions are zero. These properties are fundamental and hold beyond the case of the accumulator: difference equations as in (2.32) are linear and shift invariant if and only if initial conditions are zero. This also means that the homogeneous solution is necessarily zero (the proof is left as Exercise 2.2).

### 2.3.3 Linear Shift-Invariant Systems

#### Impulse Response

Consider the linear shift-invariant (LSI) difference equation (2.31) with zero initial conditions. Assume a single nonzero input at the origin of time, \( \delta_n \), solve \( y_n \) for
n ∈ \mathbb{N}, and call this solution \( h_n \):

\[
    h_n = \sum_{k=0}^{\infty} a_k \delta_{n-k} + \sum_{k=1}^{\infty} b_k h_{n-k} = a_n + \sum_{k=1}^{\infty} b_k h_{n-k}, \quad n \in \mathbb{N}. \tag{2.35}
\]

This is called the impulse response of the LSI difference equation and can be seen as the output (response) of the system to the Dirac impulse input.

**Convolution**

If the input is an impulse at time \( \ell \), or \( \delta_{n-\ell} \), then the output is delayed by \( \ell \):

\[
    y_n = \sum_{k=0}^{\infty} a_k \delta_{n-\ell-k} + \sum_{k=1}^{\infty} b_k y_{n-k} = a_{n-\ell} + \sum_{k=1}^{\infty} b_k y_{n-k}, \quad h_{n-\ell},
\]

for \( n => \ell \), since, by shift invariance, it is the solution of the difference equation, but starting at time \( \ell \). We can use the superposition principle, since the system is linear, and use the decomposition of the input sequence seen in (2.8) but now for two-sided sequences:

\[
    x_n = \sum_{k \in \mathbb{Z}} x_k \delta_{n-k},
\]

To write the output of the difference equation as a superposition of impulse responses, properly weighted and delayed, as

\[
    y_n = \sum_{k \in \mathbb{Z}} x_k h_{n-k} = \sum_{k \in \mathbb{Z}} x_{n-k} h_k = (h * x) = T x. \tag{2.36}
\]

This is the convolution\(^{12}\) of the input sequence \( x_n \) with the impulse response \( h_n \) of the system, denoted by \( (h * x) \) (symmetric in \( x_n \) and \( h_n \)). When not obvious, we will use subscript to denote the argument over which we perform the convolution (2.36). For example, the sum \( \sum_k x_k h_{n-k-\ell} \) will be \( (x_k * h_{n-k-\ell})_k \).

**Example 2.7 (Solution of LSI difference equations).** Consider the difference equation

\[
    y_n = a_0 x_n + b_1 y_{n-1}, \quad n \in \mathbb{N},
\]

with zero initial conditions, \( y_{-1} = 0 \). The impulse response, or solution for the input \( x = \delta_n \), is

\[
    h = [\ldots 0 \ a_0 \ b_1 a_0 \ b_1^2 a_0 \ b_1^3 a_0 \ldots]^T.
\]

If the input is delayed by \( k \) samples, \( x = \delta_{n-k} \), the output is

\[
    y = [\ldots 0 0 \ldots 0 a_0 \ b_1 a_0 \ b_1^2 a_0 \ldots],
\]

or, \( h_{n-k} \). Using superposition, we get the convolution formula as in (2.36). \( \blacksquare \)

\(^{12}\)Convolution is sometimes called linear convolution to distinguish it from the circular convolution of finite sequences.
Filters  The impulse response plays a key role (2.35); an LSI difference equation (with zero initial conditions) is entirely specified by its impulse response. The impulse response is often called a filter and the convolution is called filtering. Here are some basic classes of filters:

- **Causal filters**: are such that $h_n = 0, n < 0$. This corresponds to a causal solution of a difference equation.

- **Anticausal filters**: are such that $h_n = 0, n > 0$.

- **Two-sided filters**: are neither causal nor anticausal.

- **Finite impulse response filters (FIR)**: have only a finite number of coefficients $h_n$ different from zero.

- **Infinite impulse response filters (IIR)**: have an infinite number of nonzero terms. For example, the impulse response in Example 2.7 is causal and IIR.

Matrix View of the Convolution Operator  As we have shown in Appendix 1.C, any linear operator can be expressed in matrix form. While this is not often done when dealing with convolution, we can visualize (2.36) as:

\[
\begin{bmatrix}
\vdots \\
\vdots \\
y_{-2} \\
y_{-1} \\
y_0 \\
y_1 \\
y_2 \\
\vdots \\
\end{bmatrix}
= 
\begin{bmatrix}
\vdots \\
\vdots \\
h_0 & h_{-1} & h_{-2} & h_{-3} & h_{-4} & \ldots \\
h_1 & h_0 & h_{-1} & h_{-2} & h_{-3} & \ldots \\
h_2 & h_1 & h_0 & h_{-1} & h_{-2} & \ldots \\
h_3 & h_2 & h_1 & h_0 & h_{-1} & \ldots \\
\vdots & \vdots & \vdots & \vdots & \vdots & \ddots
\end{bmatrix}
\begin{bmatrix}
x_{-2} \\
x_{-1} \\
x_0 \\
x_1 \\
x_2 \\
\vdots \\
\end{bmatrix}
\]  \tag{2.37}

This again shows that linear discrete-time system, linear operator, filter and matrix are all synonyms. The key elements in (2.37) are the time reversal of the impulse response (in each row of the matrix, the impulse response goes from right to left), and the Toeplitz structure of the matrix (each row is a shifted version of the previous row, the matrix is constant along diagonals, see (1.95)). In Figure 2.4, an example convolution is computed graphically, emphasizing the time reversal.

We can easily find the adjoint of the convolution operator, as that $H^*$ satisfying (1.36):

\[
\langle Hx, y \rangle = \langle x, H^*y \rangle.
\]  \tag{2.38}

This is accomplished if $H^*$ is the operator corresponding to $h_{-n}$, the time-reversed
version of \( h_n \). In matrix form:

\[
H^* = \begin{bmatrix}
\ddots & \vdots & \vdots & \vdots & \vdots & \ddots \\
\cdots & h_0 & h_1 & h_2 & h_3 & \cdots \\
\cdots & h_{-1} & h_0 & h_1 & h_2 & \cdots \\
\cdots & h_{-2} & h_{-1} & h_0 & h_1 & \cdots \\
\cdots & h_{-3} & h_{-2} & h_{-1} & h_0 & \cdots \\
\ddots & \vdots & \vdots & \vdots & \vdots & \ddots \\
\end{bmatrix} \tag{2.39}
\]

**Stability** We now discuss stability of LSI systems.

**Proposition 2.1.** An LSI system is BIBO stable if and only if its impulse response is absolutely summable.

**Proof.** To prove sufficiency, consider an absolutely-summable impulse response \( h \in \ell^1(\mathbb{Z}) \), so \( \| h \|_1 < \infty \). Consider a bounded input \( x \in \ell^\infty(\mathbb{Z}) \), so \( \| x \|_\infty < \infty \). The norm of the samples at the output can be bounded as follows:

\[
|y_n| \overset{(a)}{=} \left| \sum_{k \in \mathbb{Z}} h_k x_{n-k} \right| \overset{(b)}{\leq} \sum_{k \in \mathbb{Z}} |h_k| |x_{n-k}| \overset{(c)}{\leq} \| x \|_\infty \sum_{k \in \mathbb{Z}} |h_k| \overset{(d)}{=} \| x \|_\infty \| h \|_1 < \infty,
\]

where (a) follows from (2.36); (b) from the triangle inequality (Definition 1.7.(iii)); (c) from \( x \in \ell^\infty(\mathbb{Z}) \); and (d) from \( h \in \ell^1(\mathbb{Z}) \), proving that \( y \) is bounded.

We prove necessity (BIBO stability implies absolute summability) by contradiction. For any \( h_n \) that is not absolutely summable we choose a particular input \( x_n \) (that depends on \( h_n \)), to create an unbounded output. Consider a real impulse response\(^{13}\) \( h_n \), and define the input sequence to be

\[
x_n = \text{sgn}(h_{-n}), \quad \text{where} \quad \text{sgn}(c) = \begin{cases} 
-1, & c < 0; \\
0, & c = 0; \\
1, & c > 0, \end{cases}
\]

is the sign function. Now, compute the convolution of \( x_n \) with \( h_n \) at time 0:

\[
y_0 = \sum_{k \in \mathbb{Z}} h_k x_{n-k} = \sum_{k \in \mathbb{Z}} |h_k| = \| h \|_1, \tag{2.40}
\]

which is unbounded when \( h_n \) is not in \( \ell^1(\mathbb{Z}) \). 

\(^{13}\) For a complex-valued impulse response, a slight modification, using \( x_n = h_n^*/|h_n| \) for \( |h_n| \neq 0 \), 0, otherwise, leads to the same result.

---

\[\text{v3.0 [April 15, 2008] Do not distribute} \quad \text{Comments to book-errata@FourierAndWavelets.org} \]
Figure 2.4: Example of the convolution between a sequence and a filter. (a) Sequence $x_n$. (b) Impulse response $h_n$. (c)–(e) Various time-reversed or shifted versions of the impulse response. (f) Output $y = (h * x)$.

absolutely-summable geometric series, and thus a BIBO-stable filter. For $|b_1| > 1$, any nonzero input leads to an unstable system. Lastly, for $|b_1| = 1$ (which means $+1$ or $-1$ for $b_1$ real), the system is not BIBO stable either.

Related to this is the fact that if $x \in \ell^p(\mathbb{Z})$ and $h \in \ell^1(\mathbb{Z})$, the result of $(h * x)$ is in $\ell^p(\mathbb{Z})$ as well (the proof is left as Exercise 2.3).

Among spaces we considered for our filters, two are most often used: (a) $\ell^1(\mathbb{Z})$ as it ensures BIBO stability and (b) $\mathbb{C}^N$ (or $\mathbb{R}^N$) when sequences on which we operate are truly finite.
2.4 Analysis of Sequences and Systems

In this section, we introduce various ways to analyze sequences and discrete-time systems. They range from the analytical to the computational, but they all are variations of the Fourier transform. Why this prominent role of Fourier methods? Quite simply because the Fourier sequences are eigensequences of LSI systems (convolution operator), and operating in these eigenspaces is a natural representation for such systems. This leads immediately to the convolution theorem, which states that the convolution operator is diagonalized by the Fourier transform.

The above is a part of a recipe (see Table in Chapter at a Glance):

(i) Start with a given time shift $\delta_{n-1}$;
(ii) Induce an appropriate convolution operator $T x = (h * x)$;
(iii) Find the eigensequences $x_n$ of $T$ ($e^{j\omega n}$ for infinite time and $e^{j\frac{2\pi}{N}n}$ for finite time);
(iv) Identify the frequency response as the eigenvalue corresponding to the above eigensequence ($H(e^{j\omega n})$ for infinite time, $H_k$ for finite time);
(v) Find the appropriate Fourier transform by projecting on the eigenspaces spanned by eigensequences identified in (iii) (discrete-time Fourier transform for infinite time, discrete Fourier transform for finite time).

We start with the discrete-time Fourier transform (DTFT)—the Fourier transform for infinite sequences (discrete-time sequences). The DTFT of a sequence $x_n$ is a $2\pi$-periodic function $X(e^{j\omega})$, with the argument, $e^{j\omega}$, a complex number of magnitude 1. The extension to more general complex arguments is the $z$-transform (ZT) $X(z)$, and we discuss when such a generalization makes sense. We then discuss the discrete Fourier transform (DFT)—the Fourier transform for both infinite periodic sequences as well as finite sequences with integer period $N$ (both of these can be viewed as existing on a “discrete” circle of length $N$). The sequence is thus specified by one of its periods; this also allows us to analyze the system as if it were finite dimensional of size $N$ (with appropriate boundary considerations). The transform is then finite dimensional of the same size, and denoted as $X_k$. Both DTFT as well as DFT will then have a corresponding matrix/vector product (infinite for DTFT and finite for DFT). We end the section by discussing some real-world issues, in particular, when to use which transform.

2.4.1 Fourier Transform of Infinite Sequences—DTFT

We start with and demonstrate a fundamental property of LSI systems: complex exponential sequences $v$ are eigensequences of the convolution operator $T$ (see Definition 1.17), that is

$$Tv_\lambda = (h * v_\lambda) = \lambda v_\lambda,$$

(2.41)

with the convolution operator as in (2.36). To prove this, we must find these $v_\lambda$ and the corresponding eigenvalues $\lambda$. Thus, consider a complex exponential sequence of frequency $\omega$,

$$v_{\omega,n} = e^{j\omega n}.$$

(2.42)
This sequence is bounded since $|x_n| = 1$. If $h \in \ell^1(\mathbb{Z})$, according to Proposition 2.1, $(h * v_\lambda)$ is bounded as well.

This sequence generates an entire space:

$$\mathbb{H}_\omega = \{ \alpha e^{j\omega n} | \alpha \in \mathbb{C}, \omega \in \mathbb{R} \}. \quad (2.43)$$

Let us now check what happens if we convolve $h$ with $v_\omega$:

$$(Tv_\omega)_n = (h * v_\omega)_n = \sum_{k \in \mathbb{Z}} v_{\omega,n-k} h_k = \sum_{k \in \mathbb{Z}} e^{j\omega(n-k)} h_k = \sum_{k \in \mathbb{Z}} h_k e^{-j\omega k} e^{j\omega n}. \quad (2.44)$$

Indeed, applying the convolution operator to the complex exponential sequence $v_\omega$ results in the same sequence, albeit scaled by the corresponding eigenvalue $\lambda_\omega$. We call that eigenvalue the frequency response of the system:

$$H(\omega) = \lambda_\omega = \sum_{k \in \mathbb{Z}} h_k e^{-j\omega k}. \quad (2.45)$$

We can thus rewrite (2.44) as

$$Te^{j\omega n} = (h * e^{j\omega n}) = H(\omega)e^{j\omega n}. \quad (2.46)$$

The above is true for any $v_\omega \in \mathbb{H}_\omega$, and thus, that space does not change under the operation of convolution, making it an eigenspace.

Finding the appropriate Fourier transform of $x$ now amounts to projecting $x$ onto each of the eigenspaces $\mathbb{H}_\omega$:

**Definition 2.6 (Discrete-time Fourier transform).** Given a sequence $x$, its discrete-time Fourier transform $X$ is given by

$$X(e^{j\omega}) = \langle x, v_\omega \rangle = \sum_{n \in \mathbb{Z}} x_n e^{-j\omega n}, \quad \omega \in \mathbb{R}. \quad (2.47a)$$

$X(e^{j\omega})$ is a $2\pi$-periodic function called the spectrum of the sequence.\(^{14}\)

The inverse discrete-time Fourier transform (IDTFT) is given by

$$x_n = \frac{1}{2\pi} \int_{-\pi}^{\pi} \langle x, v_\omega \rangle e^{j\omega n} d\omega = \frac{1}{2\pi} \int_{-\pi}^{\pi} X(e^{j\omega}) e^{j\omega n} d\omega. \quad (2.47b)$$

**DTFT Vs Frequency Response** The DTFT is defined for sequences and we use spectrum to denote their DTFTs. The frequency response is defined for filters (systems), and thus, strictly speaking, “DTFT of a filter” or the “spectrum of a filter” are not correct. However, by abuse of language, as the form of the DTFT of a sequence (2.47a) and the frequency response of a filter (2.45) are the same, we often interchange the two; this is fine as long as we keep in mind the fundamental difference between them.

---

\(^{14}\)We follow a somewhat redundant practice in signal processing of explicitly showing the periodicity of $X(e^{j\omega})$ through the appropriate use of the argument $e^{j\omega}$. 

---

v3.0 [April 15, 2008] Do not distribute Comments to book-errata@FourierAndWavelets.org
### 2.4. Analysis of Sequences and Systems

#### Magnitude and Phase Response

To understand the frequency response of a filter or the DTFT of a sequence, we often write the magnitude and phase separately:

\[
H(e^{j\omega}) = |H(e^{j\omega})| e^{j\arg(H(e^{j\omega}))},
\]

where \(|H(e^{j\omega})|\) is a \(2\pi\)-periodic real positive function—magnitude response, while \(\arg(H(e^{j\omega}))\) is a \(2\pi\)-periodic, real function between 0 and \(2\pi\)—phase response.\(^{15}\) A filter is said to possess linear phase if the phase response is linear in \(\omega\).

#### Existence and Convergence of the DTFT

The existence of the DTFT and its inverse depends on the sequence \(x\). Consider a sequence \(x\) that is nonzero over a finite number of indices around the origin, \(x_n = 0\) for \(|n| > N\). Then the DTFT is a trigonometric polynomial with at most \((2N + 1)\) terms,

\[
X_N(e^{j\omega}) = \sum_{n=-N}^{N} x_n e^{-j\omega n},
\]

where the subscript \(N\) indicates the boundaries of the summation. This can be seen as a partial sum, and the DTFT as the limit of \(X_N\) as \(N \to \infty\), assuming the limit exists.

- If \(x \in \ell^1(\mathbb{Z})\), then \(X(e^{j\omega})\) is bounded, since

\[
|X(e^{j\omega})| = \left| \sum_{n \in \mathbb{Z}} x_n e^{-j\omega n} \right| \leq \sum_{n \in \mathbb{Z}} |x_n| |e^{j\omega n}| = \|x\|_1 < \infty,
\]

where (a) follows from the triangle inequality (Definition 1.7.(iii)). To verify the inversion formula, note that from the definitions (2.47a) and (2.47b), we have

\[
x_n = \frac{1}{2\pi} \int_{-\pi}^{\pi} \sum_{k \in \mathbb{Z}} x_k e^{-j\omega k} e^{j\omega n} d\omega = \sum_{k \in \mathbb{Z}} x_k \frac{1}{2\pi} \int_{-\pi}^{\pi} e^{j\omega(n-k)} d\omega.
\]

When \(n \neq k\), the integral vanishes,

\[
\int_{-\pi}^{\pi} e^{j\omega(n-k)} d\omega = \frac{1}{j2\pi(n-k)} e^{j\omega(n-k)} \bigg|_{-\pi}^{\pi} = 0,
\]

and when \(n = k\), the integral is \(2\pi\), thus proving the inversion.

The partial sum in (2.48) converges uniformly to its limit \(X(e^{j\omega})\), which is bounded as we have just seen.

---

\(^{15}\)While the argument of the complex number \(H(e^{j\omega})\) can be defined to be on \([0,2\pi)\) or \((-\pi,\pi)\), in either case the function \(\arg(H(e^{j\omega}))\) is discontinuous at interval boundaries.
Figure 2.5: Ideal lowpass filter. (a) Impulse response of an ideal lowpass filter. (b) Magnitude response of an ideal lowpass filter. (c) Impulse response of an ideal highpass filter. (d) Magnitude response of an ideal highpass filter. (e) Impulse response of an ideal bandpass filter. (f) Magnitude response of an ideal bandpass filter. (TBD)

- If $x \in l^2(\mathbb{Z})$, the inversion formula holds as well, and follows from $\{v_{\omega,n} = (1/\sqrt{2\pi})\nu_{\omega,n} = (1/\sqrt{2\pi})e^{j\omega n}\}_{n \in \mathbb{Z}}$ being an ONB for the interval $[-\pi, \pi]$ (we will see that when discussing Fourier series in the next chapter).

The partial sum in (2.48) converges to a function $X(e^{j\omega})$ which is square integrable over $[0, 2\pi]$, as seen in Parseval’s equality shortly (2.61a). However, this convergence might be nonuniform as we will show.

- If $x$ belongs to other spaces, the DTFT can be found under certain conditions and using distributions. For example, by allowing a transform with Dirac functions, one can handle sinusoidal sequences. We will treat these cases more formally in the next chapter.

The frequency response of a filter is typically used to design a filter with specific properties. We now explore both the design questions as well as point out some convergence issues.

Example 2.8 (Design of an ideal filter and Gibbs’ phenomenon). Filtering passes/attenuates/blocks certain frequencies. For example, an ideal lowpass filter passes frequencies below some cut-off frequency $\omega_0$ (passband $[-\omega_0, \omega_0]$) and blocks the others (stopband $[\omega_0, 2\pi - \omega_0]$). Figure 2.5(b) illustrates an ideal lowpass filter.

To find the impulse response of such an ideal filter, we start with the desired magnitude response (we assume a real impulse response, $|H(e^{j\omega})| = |H(e^{-j\omega})|)$:

$$|H(e^{j\omega})| = \begin{cases} 1, & |\omega| \leq \omega_0; \\ 0, & \text{otherwise.} \end{cases}$$  \hspace{1cm} (2.49)

For this design example, we set the phase to zero. Applying the inverse DTFT\(^{16}\) given in (2.47b), we obtain the impulse response as

$$h_n = \frac{1}{2\pi} \int_{-\omega_0}^{\omega_0} e^{j\omega n} d\omega = \frac{1}{2\pi jn} \left[ e^{j\omega_0 n} - e^{-j\omega_0 n} \right] = \frac{1}{\pi} \frac{\sin(\omega_0 n)}{jn} = \frac{1}{\pi} \frac{\sinh(\omega_0 n)}{jn}. $$

A particular case of interest is the halfband filter, which passes through half of the spectrum, from $-\pi/2$ to $\pi/2$. We normalize it so $\|h\| = 1$, and thus $|H(e^{j\omega})| = \sqrt{2}$ (see (2.61a)). Then, its impulse response is

$$h_n = \frac{1}{\sqrt{2}} \frac{\sin(\pi/2 n)}{\pi/2 n},$$  \hspace{1cm} (2.50)

as shown in Figure 2.5(a). The impulse responses of these ideal filters have a slow decay of order $1/n$, and are thus not absolutely summable. This leads to convergence problems, because of lack of uniform convergence.

\(^{16}\)The inverse frequency response is equal to the inverse IDTFT.
2.4. Analysis of Sequences and Systems

Figure 2.6: DTFT of the truncated impulse response of the ideal lowpass filter, illustrating the Gibbs phenomenon. Shown are $H_N(e^{j\omega})$ for $N = 10, 100$ and $1000$. (TBD)

This is illustrated in Figure 2.6, where the trigonometric polynomial

$$H_N(e^{j\omega}) = \frac{1}{\sqrt{2}} \sum_{n=-N}^{N} \frac{\sin(\pi/2n)}{\pi/2n} e^{-j\omega n},$$

which corresponds to the frequency response of (2.50), is truncated to $n \in \{-N, -N+1, \ldots, N-1, N\}$, and this for various values of $N$. As is clear from the figure, the convergence is nonuniform; the result oscillates around the points of discontinuity and becomes narrower but does not decrease in size. This over- and undershoot is of the order of 9%, and is called the Gibbs’ phenomenon (see also Figure 0.3).

In the above example, we saw the lack of uniform convergence, and the potentially disturbing oscillations around points of discontinuity due to the Gibbs’ phenomenon. Fortunately, there is mean-square convergence; Given a $2\pi$-periodic function $H(e^{j\omega})$, square integrable on a period, and its coefficients $h_n$ from (2.47b), the truncated DTFT is

$$H_N(e^{j\omega}) = \sum_{n=-N}^{N} h_n e^{-j\omega n}.$$  

The limit of $H_N(e^{j\omega})$ as $N \to \infty$ is equal to $H(e^{j\omega})$ in $L^2$-norm, 

$$\lim_{N \to \infty} \|H(e^{j\omega}) - H_N(e^{j\omega})\|_2 = 0.$$  

While the limit of $H_N(e^{j\omega})$ is not necessarily equal to $H(e^{j\omega})$ everywhere, it is equal almost everywhere.

Properties of the DTFT

Convolution  Given a sequence $x$ and filter $h$, in the DTFT domain, their convolution maps to the product of the spectrum of the sequence and frequency response of the filter, the convolution theorem states that:

$$(h \ast x)_n \quad \text{DTFT} \quad H(e^{j\omega})X(e^{j\omega}).$$

This result is a direct consequence of the eigensequence property of complex exponential sequences $v_\omega$ from (2.42): since each spectral component is the projection of the sequence $x$ onto the appropriate eigenspace, the DTFT diagonalizes the convolution operator. Assume that both $x$ and $h$ are in $\ell^1(\mathbb{Z})$. Then, $(h \ast x)$ is also in $\ell^1(\mathbb{Z})$, since

$$\sum_{n} \left| \sum_{k} x_k h_{n-k} \right| \leq \sum_{n} \sum_{k} |x_k| |h_{n-k}| = \sum_{k} |x_k| \sum_{n} |h_{n-k}| = \sum_{k} |x_k| \|h\|_1 = \|x\|_1 \|h\|_1 < \infty.$$
The spectrum $Y(e^{j\omega})$ of the output sequence $y = (h * x)$ can be written as

$$Y(e^{j\omega}) = \sum_n y_n e^{-j\omega n} = \sum_n \left( \sum_k x_k h_{n-k} \right) e^{-j\omega n}$$

$$= \sum_n \sum_k x_k e^{-j\omega k} h_{n-k} e^{-j\omega(n-k)}$$

$$= \sum_k x_k e^{-j\omega k} \sum_n h_{n-k} e^{-j\omega(n-k)} = X(e^{j\omega})H(e^{j\omega}),$$

where in (a) we interchanged the order of summation, an allowed operation since $x_k h_{n-k}$ is absolutely summable over $k$ and $n$.

While the convolution is clearly commutative, it is associative only under certain conditions. Given sequences $a, b, c \in \ell^1(\mathbb{Z})$,

$$(a * (b * c)) = ((a * b) * c) = (a * b * c),$$

verified by using the DTFT:

$$A(e^{j\omega}) \left[ B(e^{j\omega})C(e^{j\omega}) \right] = \left[ A(e^{j\omega})B(e^{j\omega}) \right] C(e^{j\omega}) = A(e^{j\omega})B(e^{j\omega})C(e^{j\omega}).$$

**Example 2.9.** When sequences are not in $\ell^1(\mathbb{Z})$ or $\ell^2(\mathbb{Z})$, one needs to be careful about associativity, as the following counterexample shows: Take the Heaviside sequence from (2.13) $a_n = u_n$. As the second sequence, choose the first-order differencing sequence $b_n = \delta_n - \delta_{n-1}$. Finally, choose as $c_n = 1$, the constant sequence. Note that neither $a_n$ nor $c_n$ is in $\ell^1(\mathbb{Z})$. Now,

$$(a * (b * c)) = (a * 0) = 0,$$

while

$$((a * b) * c) = (\delta_n * c_n)_n = 1.$$

Thus, the convolution of these particular sequences is not associative.

**Modulation** Given sequences $x$ and $h$, their product maps into the convolution of their spectra in the DTFT domain,

$$x_n h_n \xrightarrow{\text{DTFT}} \frac{1}{2\pi} (X * H)_\omega,$$  \hspace{1cm} (2.53)

where the convolution of the $2\pi$-periodic spectra is defined as

$$(X * H)_\omega = \int_{-\pi}^{\pi} X(e^{j\theta})H(e^{j(\omega-\theta)}) \, d\theta.$$  \hspace{1cm} (2.54)

Modulation is dual to convolution (2.52), (the proof is left as Exercise 2.4).
2.4. Analysis of Sequences and Systems

**Autocorrelation**

Given a sequence \( x \), its autocorrelation \( a_{17} \) and the corresponding DTFT pair are given by

\[
a_n = \sum_{k \in \mathbb{Z}} x_k x_{k+n} \quad \text{DTFT} \quad A(e^{j\omega}) = |X(e^{j\omega})|^2. \tag{2.55}
\]

The autocorrelation measures the similarity of a sequence with respect to shifts of itself. Note that \( a_0 = \sum_k |x_k|^2 = \|x\|_2 \), and \( a_{-n} = a_n \). To show (2.55), introduce the time-reversed version of \( x_n \), \( x_{-n} \), and its DTFT,

\[
x_{-n} \quad \text{DTFT} \quad X^*(e^{j\omega}). \tag{2.56}
\]

Then, another way to write \( a \) is simply

\[
a_n = (x_n \ast x_{-n})_n. \tag{2.57}
\]

Using the convolution theorem (2.52), we obtain (2.55).

We can see autocorrelation in its matrix form as well; using (2.37), we can write \( A \) as a symmetric operator, a positive semi-definite matrix (see (1.92)):

\[
A = X^T X = \begin{bmatrix}
\ddots & \vdots & \vdots & \vdots & \vdots & \vdots & \ddots \\
\vdots & a_0 & a_1 & a_2 & a_3 & a_4 & \ddots \\
\vdots & a_1 & a_0 & a_1 & a_2 & a_3 & \ddots \\
\vdots & a_2 & a_1 & a_0 & a_1 & a_2 & \ddots \\
\vdots & a_3 & a_2 & a_1 & a_0 & a_1 & \ddots \\
\vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \ddots \\
\ddots & \vdots & \vdots & \vdots & \vdots & \vdots & a_0
\end{bmatrix}. \tag{2.58}
\]

**Crosscorrelation**

Given two sequences \( x \) and \( h \), their crosscorrelation \( c \) and the corresponding DTFT are given by

\[
c_n = \sum_{k \in \mathbb{Z}} x_k h_{k+n} = \langle x_k, h_{k+n} \rangle_k \quad \text{DTFT} \quad C(e^{j\omega}) = X^*(e^{j\omega})H(e^{j\omega}). \tag{2.59}
\]

Similarly to the autocorrelation above, crosscorrelation can be seen in its Toeplitz matrix form:

\[
C = H^T X = \begin{bmatrix}
\ddots & \vdots & \vdots & \vdots & \vdots & \vdots & \ddots \\
\vdots & c_0 & c_1 & c_2 & c_3 & c_4 & \ddots \\
\vdots & c_1 & c_0 & c_1 & c_2 & c_3 & \ddots \\
\vdots & c_2 & c_1 & c_0 & c_1 & c_2 & \ddots \\
\vdots & c_3 & c_2 & c_1 & c_0 & c_1 & \ddots \\
\vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \ddots \\
\ddots & \vdots & \vdots & \vdots & \vdots & \vdots & c_0
\end{bmatrix}. \tag{2.60}
\]

Further properties of autocorrelation and crosscorrelation sequences and their transforms are explored in Exercise 2.5.
Parseval’s Equality
Given \( x_n \in \ell^2(\mathbb{Z}) \), its energy is conserved by the DTFT,
\[
\|x\|_2^2 = \sum_{n \in \mathbb{Z}} |x_n|^2 = \frac{1}{2\pi} \int_{-\pi}^{\pi} |X(e^{j\omega})|^2 \, d\omega = \|X\|_2^2.
\] (2.61a)

To verify this formula, recall that the left-hand side is equal to the term \( a_0 \) in the autocorrelation (2.55). Inverting the spectrum and using the inversion formula (2.47b) leads to the desired result. Similarly, generalized Parseval’s relation can be written as
\[
\sum_n x_n h_n^* = \frac{1}{2\pi} \int_{-\pi}^{\pi} X(e^{j\omega}) H^*(e^{j\omega}) \, d\omega,
\] (2.61b)
the proof of which is left as Exercise 2.5.

Both formulas can be rewritten in terms of inner products defined on the appropriate spaces. For example, the following inner product
\[
\langle X, H \rangle_\omega = \frac{1}{2\pi} \int_{-\pi}^{\pi} X(e^{j\omega}) H^*(e^{j\omega}) \, d\omega,
\] (2.62)
is defined on \( L^2([-\pi, \pi]) \), with an appropriate \( L^2 \)-norm:
\[
\|X(e^{j\omega})\|_2 = \langle X, X \rangle_\omega^{\frac{1}{2}}.
\] (2.63)

Then (2.61b) is
\[
\langle x, h \rangle_n = \langle X, H \rangle_\omega,
\]
while (2.61a) is the special case with \( h_n = x_n \). Parseval’s equality (norm conservation) indicates that the DTFT is a unitary map.

Frequency Response of Filters
So far, we have mostly seen ideal filters, which have rather simple frequency responses and are not realizable. Filters that are realizable in practice have more subtle frequency responses. We now explore a few of these.

Example 2.10 (Frequency response of a moving average filter). The impulse response of the moving average filter in (2.6) is:
\[
h_n = \begin{cases} \frac{1}{2N+1}, & |n| \leq N; \\ 0, & \text{otherwise}, \end{cases}
\] (2.64a)
while its frequency response is
\[
H(e^{j\omega}) = \frac{1}{2N+1} \sum_{n=-N}^{N} e^{-j\omega n} = \frac{1}{2N+1} e^{j\omega N} \sum_{n=0}^{2N} e^{-j\omega n}
\]
\[
= \frac{1}{2N+1} \frac{e^{j\omega N} (1 - e^{-j\omega(2N+1)})}{1 - e^{-j\omega}} = \frac{1}{2N+1} \frac{e^{-j\omega/2} (e^{j\omega N + \frac{1}{2}}) - e^{-j\omega N + \frac{1}{2}}}{e^{-j\omega/2} (e^{j\omega/2} - e^{-j\omega/2})}
\]
\[
= \frac{1}{2N+1} \frac{\sin \omega(N + \frac{1}{2})}{\sin \omega/2},
\] (2.64b)
where (a) follows from (P1.20-6). Figure 2.7 shows the impulse response and frequency response of this filter for $N = 7$.

**Example 2.11 (Allpass filters and shift-invariant ONBs).** Consider a simple shift-by-$k$ filter given in (2.24a) with the impulse response $h_n = \delta_{n-k}$. If $x_n$ is a complex exponential sequence as in (2.42), the filter does not influence the magnitude of $x_n$; only the phase is changed:

$$
(h * x) = \sum_{\ell} \delta_{\ell-k} e^{j\omega(n-\ell)} = e^{-j\omega k} e^{j\omega n} = H(e^{j\omega}) x_n,
$$

from which

$$
|H(e^{j\omega})| = 1, \quad \arg(H(e^{j\omega})) = \omega k \mod 2\pi.
$$

Since all frequencies are going through without change of magnitude, this is an example of an **allpass filter**. In addition, it has linear phase with a slope $k$ given by the delay.

An example of both a more sophisticated allpass filter, as well as the fact that while key properties are sometime not plainly visible in the time domain, they become obvious in the frequency domain, is

$$
g_n = a^n u_n, \quad g = \left[ \ldots 0 \begin{array}{c} 1 \end{array} a \quad a^2 \quad a^3 \quad \ldots \right]^T,
$$

where $a$ is real or complex with $|a| < 1$, and $u_n$ is the Heaviside sequence from (2.13). Now, construct

$$
h_n = -a^* g_n + g_{n-1}.
$$

We will show that filtering a sequence $x$ with $h$ will not change its magnitude, and that, moreover, $h$ is of norm 1 and orthogonal to all its shifts

$$
|H(e^{j\omega})| = 1 \quad \text{DTFT} \quad \langle h_n, h_{n-k} \rangle_n = \delta_k.
$$

To show this, find the frequency response of $g_n$,

$$
G(e^{j\omega}) = \sum_{n=0}^{\infty} a^n e^{-j\omega n} = \frac{1}{1 - ae^{-j\omega}},
$$

where we have used (P1.20-1). Then, $H(e^{j\omega})$ follows as

$$
H(e^{j\omega}) = -a^* G(e^{j\omega}) + e^{-j\omega} G(e^{j\omega}) = \frac{e^{j\omega} - a^*}{1 - ae^{-j\omega}}. \quad (2.65)
$$
The magnitude of \( H(e^{j\omega}) \) is
\[
|H(e^{j\omega})|^2 = H(e^{j\omega})H^*(e^{j\omega}) = \frac{(e^{-j\omega} - a^*)(e^{j\omega} - a)}{(1 - ae^{-j\omega})(1 - a^*e^{j\omega})} = 1,
\]
and thus \( |H(e^{j\omega})| = 1 \) for all \( \omega \). By Parseval’s equality, \( \|h\| = 1 \). In frequency domain, the result of \((h \ast x)\) is
\[
|H(e^{j\omega})X(e^{j\omega})| = |H(e^{j\omega})| \cdot |X(e^{j\omega})| = |X(e^{j\omega})|.
\]
The magnitude remains unchanged, from which the name—allpass filter—is derived.

As to be expected, this also corresponds to energy conservation, since using Parseval’s relation \((2.61a)\), we have
\[
\frac{1}{2\pi} \int_{-\pi}^{\pi} |H(e^{j\omega})X(e^{j\omega})|^2 \, d\omega = \frac{1}{2\pi} \int_{-\pi}^{\pi} |H(e^{j\omega})|^2 \, d\omega = \|x\|^2.
\]

Finally, let us show that \( |H(e^{j\omega})| = 1 \) implies that the impulse response \( h_n \) is orthogonal to all its shifts. Using the generalized Parseval’s relation \((2.61b)\) and \( h_{n-k} \xrightarrow{\text{DTFT}} e^{-j\omega k} H(e^{j\omega}) \):
\[
\langle h_n, h_{n-k} \rangle_n = \sum_{n} h_n h_{n-k}^* = \frac{1}{2\pi} \int_{-\pi}^{\pi} H(e^{j\omega}) \left(e^{-j\omega k} H(e^{j\omega})^*\right) \, d\omega
\]
\[
= \frac{1}{2\pi} \int_{-\pi}^{\pi} e^{j\omega k} H(e^{j\omega}) H^*(e^{j\omega}) \, d\omega \overset{(b)}{=} \frac{1}{2\pi} \int_{-\pi}^{\pi} e^{j\omega k} |H(e^{j\omega})|^2 \, d\omega
\]
\[
= \frac{1}{2\pi} \int_{-\pi}^{\pi} e^{j\omega k} \, d\omega = \frac{e^{j\pi k} - e^{-j\pi k}}{2j\pi} = \frac{\sin k\pi}{k\pi} = \delta_k,
\]
where \( (a) \) follows from the generalized Parseval’s relation \((2.61b)\); \( (b) \) from our assumption; \( (c) \) from Euler’s formula \((2.169)\); and \( (d) \) from the properties of the sinc function \((1.75)\). Therefore, the set \( \{\varphi_{kn} = h_{n-k} \}_k \in Z \) forms an orthonormal set in \( \mathbb{L}^2(\mathbb{Z}) \). To check whether it is ONB, we check if we can write any \( x_n \) as
\[
x_n = \sum_{k \in \mathbb{Z}} \alpha_k \varphi_{kn} = \sum_{k \in \mathbb{Z}} \alpha_k h_{n-k},
\]
with \( \alpha_k = \langle x_n, h_{n-k} \rangle_n \), for which, it is sufficient to verify that \( \|\alpha\| = \|x\| \). Write \( \alpha \) as
\[
\alpha_k = \sum_{n \in \mathbb{Z}} x_n h_{n-k} = \sum_{n \in \mathbb{Z}} x_n h_{-(k-n)} = (x_n \ast h_{-n})_n.
\]
We now prove \( \|\alpha\| = \|x\| \):
\[
\|\alpha\|^2 \overset{(a)}{=} \|X(e^{j\omega})H^*(e^{j\omega})\|^2 \overset{(b)}{=} \|X(e^{j\omega})\|^2 \overset{(c)}{=} \|x\|^2,
\]
where \( (a) \) follows from the convolution theorem \((2.52)\), Parseval’s equality \((2.61a)\) and \((2.56)\); \( (b) \) from \( h \) being an allpass filter; and \( (c) \) from Parseval’s equality again. Figure 2.8 shows the phase of \( H(e^{j\omega}) \) given in \((2.65)\).
2.4. Analysis of Sequences and Systems

Figure 2.8: Phase of a first-order allpass filter, with \( a = \frac{1}{2} \). (TBD)

The above example contains a piece of good news—there exist shift-invariant ONBs for \( \ell^2(\mathbb{Z}) \), as well as a piece of bad news—these bases have no frequency selectivity (they are allpass sequences). This is one of the main reasons to search for more general ONBs for \( \ell^2(\mathbb{Z}) \), as will be done in Part II of the book.

2.4.2 The \( z \)-transform

While the DTFT has many nice properties, we discussed it mostly for sequences in \( \ell^1(\mathbb{Z}) \) and \( \ell^2(\mathbb{Z}) \), and extending it further requires care, limiting its applicability. Take the Heaviside sequence from (2.13), which is neither in \( \ell^1(\mathbb{Z}) \) nor \( \ell^2(\mathbb{Z}) \) (nor any other \( \ell^p(\mathbb{Z}) \) space except \( \ell^\infty(\mathbb{Z}) \)). If we were to multiply it by an exponential sequence \( r^n \), \( |r| < 1 \) yielding \( x_n = r^n u_n \), we could take the DTFT of \( x_n \), as it became an absolutely-summable sequence. What would this do to some of the key properties of the DTFT? The way we derived the DTFT relied on the fact that complex exponential sequences are eigensequences of the convolution operator. However, this is true for any exponential sequence \( v_{r,n} = r^n \), since

\[
(h_n * r^n)_n = \sum_{n \in \mathbb{Z}} h_n r^{k-n} = r^k \sum_{n \in \mathbb{Z}} h_n r^{-n} = r^k H(r),
\]

where

\[
H(r) = \sum_{n \in \mathbb{Z}} h_n r^{-n}
\]

(2.66)

is now the counterpart of the frequency response in (2.45), and is well defined for values of \( r \) for which \( h_n r^{-n} \) is absolutely summable.

This sets the stage for an extension of the DTFT to a larger class, those based on exponential sequences of the form

\[
v_{r,\omega,n} = (r e^{j\omega})^n,
\]

(2.67)

for \( r \in \mathbb{R}^+, \omega \in \mathbb{R}, n \in \mathbb{Z} \). Except for \( r = 1 \), these exponential sequences are unbounded, and thus, convergence issues of the resulting transform are even more critical than for the DTFT. For values \( r \) for which (2.66) is well defined, any complex exponential sequence as in (2.67) will lead to a bounded eigenvalue \( H(r) \).

Thus, it is intuitive that we will have access to a convolution property similar to the one for the DTFT, but for more general sequences. Indeed, as we will see shortly, convolution of finite-length sequences becomes polynomial multiplication in the transform domain. This is the essential motivation behind extending the analysis which uses unit-norm complex exponential sequences \( e^{j\omega n} \) as in (2.42), to more general complex exponential sequences \( v_{r,\omega,n} = (r e^{j\omega})^n \) as in (2.67). For historical reasons, the complex number \( r e^{j\omega} \) is denoted by

\[
z = re^{j\omega},
\]
where \( r \geq 0 \) is the magnitude, \( \omega \) the phase, and the associated transform is called the \( z \)-transform:

**Definition 2.7 (\( z \)-transform).** Given a sequence \( x \), the power series

\[
X(z) = \sum_{n \in \mathbb{Z}} x_n z^{-n}, \quad z = re^{j\omega}, r \in \mathbb{R}^+, \omega \in [-\pi, \pi),
\]

is called the \( z \)-transform of \( x_n \).

Associated to this transform is a **region of convergence** (ROC), which is the set of values of \( z \) for which (2.68) converges:

\[
\text{ROC} = \{z \mid |X(z)| < \infty\},
\]

and without specifying the ROC, the \( z \)-transform makes little sense. Let us explore a few features of the ROC.

For the \( z \)-transform to exist, the following must be true:

\[
|X(z)| = \left| \sum_{n \in \mathbb{Z}} x_n z^{-n} \right| \leq \sum_{n \in \mathbb{Z}} |x_n| |z^{-n}| = \sum_{n \in \mathbb{Z}} |x_n| |r^{-n}| |e^{-j\omega n}| = \sum_{n \in \mathbb{Z}} |x_n r^{-n}| < \infty.
\]

Thus, absolute summability of \( x_n r^{-n} \) is a sufficient condition for convergence. If \( z_1 \in \text{ROC} \), than any \( z_2 \) such that \( |z_2| = |z_1| \) also belongs to the ROC, and thus the ROC is a ring of the form

\[
\text{ROC} = \{z \mid 0 \leq \alpha_1 < |z| < \alpha_2 \leq \infty\}.
\]

Exercise 2.9 explores a number of properties of the ROC, summarized in Table 2.2.

<table>
<thead>
<tr>
<th>Sequence</th>
<th>ROC</th>
</tr>
</thead>
<tbody>
<tr>
<td>General</td>
<td>( 0 \leq \alpha_1 &lt;</td>
</tr>
<tr>
<td>Finite length</td>
<td>All ( z ), expect possibly 0 and/or ( \infty )</td>
</tr>
<tr>
<td>Right sided</td>
<td>Outward from largest pole</td>
</tr>
<tr>
<td>Left sided</td>
<td>Inward from smallest pole</td>
</tr>
<tr>
<td>Stable system</td>
<td>Contains unit circle</td>
</tr>
</tbody>
</table>

**Table 2.2:** Some properties of the ROC of the \( z \)-transform. A right-sided sequence is zero for some \( n < N \), and a left-sided sequence is zero for some \( n > N \), with \( N \) arbitrary but finite.

**Example 2.12.** To develop intuition, we look at a few examples:

(i) We start with the shift-by-\( k \) sequence, \( x_n = \delta_{n-k} \). Then

\[
X(z) = z^{-k}, \quad \text{ROC} = \begin{cases} |z| > 0, & k > 0; \\
\text{all } z, & k = 0; \\
|z| < \infty, & k < 0. \end{cases}
\]
Figure 2.9: ROC for Example 2.12. (a) The right-sided sequence, \( \alpha < 1 \). (b) The left-sided sequence, \( \alpha < 1 \). (TBD)

We see that a shift-by-one maps to \( z^{-1} \), which is why \( z^{-1} \) is often called a delay operator.

(ii) The right-sided geometric series sequence has the following \( z \)-transform pair:

\[
x_n = \alpha^n u_n \xrightarrow{ZT} X(z) = \sum_{n \in \mathbb{N}} (\alpha z^{-1})^n = \frac{1}{1 - \alpha z^{-1}},
\]

with the ROC = \( \{ z \mid |z| > |\alpha| \} \). Now, \( z = \alpha \) is a pole (zero of the denominator) of the complex function \( X(z) \), and we see that the ROC is bounded from inside by a circle containing the pole. This is a general property, since the ROC cannot contain a singularity where \( X(z) \) does not exist.

(iii) The left-sided geometric series has the following \( z \)-transform pair:

\[
x_n = -u_{n-1} \alpha^n \xrightarrow{ZT} X(z) = \sum_{n \in \mathbb{N}} (\alpha z^{-1})^n = \frac{1}{1 - \alpha z^{-1}},
\]

with the ROC = \( \{ z \mid |z| < |\alpha| \} \). The expression for \( X(z) \) is exactly as in the previous case; the only difference is in the ROC. Had we been given only the this \( X(z) \) without the associated ROC, we would not be have been able to tell whether it originated from \( x \) in (2.71) or (2.72). This shows why the \( z \)-transform and its ROC form a pair that should not be broken.

A standard way of showing the ROC is a picture of the complex plane, see Figure 2.9. Note the importance of the unit circle \( z = e^{j\omega} \).

A number of useful \( z \)-transforms are given in Table 2.3.

**z-transform and the DTFT** Given a sequence \( x \) and its \( z \)-transform \( X(z) \), with an ROC that includes the unit circle \( |z| = 1 \), the \( z \)-transform evaluated on the unit circle is equal to the DTFT of the same sequence:

\[
X(z)|_{z=e^{j\omega}} = X(e^{j\omega}).
\]

Conversely, given a sequence \( x \) scaled by \( r^n \) and its DTFT \( X(re^{j\omega}) \), for all values of \( r \) for which it is absolutely summable, the DTFT of that scaled sequence is equal to the \( z \)-transform of the same sequence evaluated on the circle \( z = re^{j\omega} \):

\[
X(re^{j\omega}) = X(z)|_{z=re^{j\omega}}.
\]

**Rational \( z \)-transforms**

An important class of \( z \)-transforms are those that are rational functions, since \( z \)-transforms of most realizable systems (systems that can be built and used in prac-
sequence $\delta_n$ transform ROC
$\delta_{n-k}$ $1$ $z^{-k}$ all $z$

all $z$ except possibly $z = 0$ (for $k > 0$)
or $z = \infty$ (for $k < 0$)
a
$\frac{n_0}{z_0}$ & $\frac{1}{(1 - az^{-1})}$ & $|z| > |a|$
$-a^n u_{-n-1}$ & $\frac{1}{(1 - az^{-1})}$ & $|z| < |a|$

$na^n u_n$ & $\frac{(az^{-1})(1 - az^{-1})^2}{(1 - az^{-1})}$ & $|z| > |a|$
$-na^n u_{-n-1}$ & $\frac{(az^{-1})(1 - az^{-1})^2}{(1 - az^{-1})}$ & $|z| < |a|$
cos$(\omega_c n) u_n$ & $\frac{(1 - \cos(\omega_c)z^{-1})(1 - 2 \cos(\omega_c)z^{-1} + z^{-2})}{(1 - \cos(\omega_c)z^{-1} + z^{-2})}$ & $|z| > 1$
sin$(\omega_c n) u_n$ & $\frac{(1 - \sin(\omega_c)z^{-1})(1 - 2 \cos(\omega_c)z^{-1} + z^{-2})}{(1 - \cos(\omega_c)z^{-1} + z^{-2})}$ & $|z| > 1$

\begin{align*}
\begin{cases}
a^n, & 0 \leq n \leq N - 1; \\
0, & \text{otherwise.}
\end{cases} & \frac{1 - az^{-1}}{1 - az^{-1}} & |z| > 0
\end{align*}

Table 2.3: Commonly encountered sequences and their $z$-transforms with the associated ROCs.

\[ H(z) = \frac{A(z)}{B(z)}, \quad (2.75) \]

where $A(z)$ and $B(z)$ are polynomials in $z^{-1}$, of degree $M$ and $N$, respectively. The degrees satisfy $M \leq N$, otherwise, polynomial division would lead to a sum of a polynomial and a rational function satisfying this constraint. It is a standard fact of complex analysis that a rational function has the same number of poles and zeros, $\text{max}(M, N)$.

Consider a finite-length sequence $h = [h_0 \ h_1 \ \ldots \ h_M]^T$. Then $H(z) = \sum_{k=0}^{M} h_k z^{-k}$, which has $M$ poles at $z = 0$ and $M$ zeros at the roots $\{z_k\}_{k=1}^{M}$ of the polynomial $H(z)$.$^{18}$ Therefore, $H(z)$ can be written as

\[ H(z) = h_0 \prod_{k=1}^{M} (1 - z_k z^{-1}) \quad |z| > 0, \quad (2.76) \]

where the form of the factorization shows explicitly both the roots as well as the multiplicative factor $h_0$.

The rational $z$-transform in (2.75) can thus also be written as

\[ H(z) = \frac{a_0 \prod_{k=1}^{M} (1 - z_k z^{-1})}{b_0 \prod_{k=1}^{N} (1 - p_k z^{-1})}, \quad (2.77) \]

$^{18}$The fundamental theorem of algebra (see Theorem 1.8) states that a degree-$M$ polynomial has $M$ complex roots.
2.4. Analysis of Sequences and Systems

Figure 2.10: Simple discrete-time system, where \( z^{-1} \) stands for a unit delay. (TBD)

where \( \{z_k\}_{k=1}^M \) are the zeros and \( \{p_k\}_{k=1}^N \) are the poles of the rational function. (Without loss of generality, we can set \( b_0 = 1 \).) The ROC cannot contain any poles, and is thus, assuming a right-sided sequence, outside of the pole largest in magnitude. If \( M \) is smaller than \( N \), then \( H(z) \) has additional \( (N-M) \) zeros at 0. This can be best seen on our previous example (2.71), which can be written as \( 1/(1 - \alpha z^{-1}) = z/(z - \alpha) \) and has thus a pole at \( z = \alpha \) and a zero at \( z = 0 \).

Difference Equations with Finite Number of Coefficients Where do such rational \( z \)-transforms come from? For the sake of simplicity and unless otherwise specified, we consider right-sided sequences (causal systems). Thus, consider a causal solution of a difference equation with a finite number of terms as in (2.32) with zero initial conditions. Assuming \( x \) and \( y \) have well-defined \( z \)-transforms \( X(z) \) and \( Y(z) \), and using \( x_{n-k} \xrightarrow{\text{ZT}} z^{-k}X(z) \), we can rewrite (2.32) as

\[
Y(z) = \left( \sum_{k=0}^{M} a_k z^{-k} \right) X(z) - \left( \sum_{k=1}^{N} b_k z^{-k} \right) Y(z).
\]

Using (2.77) with \( b_0 = 1 \), the transfer function is given by

\[
Y(z) = \frac{\sum_{k=0}^{M} a_k z^{-k} \cdot X(z)}{1 + \sum_{k=1}^{N} b_k z^{-k}} = H(z)X(z).
\]  

(2.78)

In other words, the impulse response of the difference equation (2.32) (when \( x_n = \delta_n \) or \( X(z) = 1 \), is given by a rational \( H(z) \) in the ZT domain. That is, the impulse response of a linear discrete-time system with a finite number of coefficients has a rational \( z \)-transform.

Example 2.13 (Transfer function). Consider the simple system in Figure 2.10, where the \( z^{-1} \) operators stand for unit delays. The output \( y_n \) and the associated transfer function are:

\[
y_n = x_n + a_1 x_{n-1} - b_1 y_{n-1} - b_2 y_{n-2} \Rightarrow H(z) = \frac{1 + a_1 z^{-1}}{1 + b_1 z^{-1} + b_2 z^{-2}}.
\]

The above discussion leads to the following important result on stability:

Proposition 2.2 (Stability). A causal discrete-time linear system with a finite number of coefficients is BIBO stable, if and only if the poles of its (reduced) \( z \)-transform are inside the unit circle.
Proof. By what we have just seen, the $z$-transform of the transfer function of such a system is a rational function $H(z) = \frac{A(z)}{B(z)}$. We assume that any pole-zero cancellation has been done (reduced system), as well as that $N > N$. Using partial fraction expansion, we can write $H(z)$ as a sum of first-order terms

$$H(z) = \sum_{k=1}^{N} \frac{\alpha_k}{1 - p_k z^{-1}}$$

when all poles are distinct. (In the case of multiple poles, powers of $(1 - p_k z^{-1})$ are involved.) Each pole $p_k$ leads to a geometric series $(p_k)^n$, and thus

$$h_n = \sum_{k=1}^{N} \alpha_k p_k^n.$$  

Each term in the above sum is absolutely summable if $|p_k| < 1$. Since $h_n$ is a finite sum of such terms, the sum is absolutely summable as well, and according to Proposition 2.1, the system is BIBO stable. 

Inverse $z$-transform

Given a $z$-transform and its ROC, what is the time-domain sequence, how do we invert the $z$-transform? The question is all the more important as we have seen that only when the $z$-transform is accompanied by its ROC can we say which sequence it originated from without ambiguity. The general inversion formula for the $z$-transform involves contour integration, a standard topic of complex analysis. However, most $z$-transforms encountered in practice can be inverted using simpler methods which we now discuss; for a more detailed treatment inverse $z$-transform, Further Reading gives pointers.

Inversion by Inspection The first method is just a way of recognizing certain $z$-transform pairs. For example, by looking at Table 2.3, we would see that the following $z$-transform:

$$X(z) = \frac{1}{1 - \frac{1}{4} z^{-1}}.$$  

(2.79a)

has the same form as $1/(1 - az^{-1})$, with $a = 1/4$. From the table, we can then read the appropriate sequence that generated it as one of the following two:

$$\left(\frac{1}{4}\right)^n u_n, \quad |z| > \frac{1}{4}, \quad \text{or} \quad -\left(\frac{1}{4}\right)^n u_{-n-1}, \quad |z| < \frac{1}{4}.$$  

(2.79b)

Inversion Using Partial Fraction Expansion This method works when the $z$-transform is given as ratio of two polynomials as in (2.75), which can also be expressed as in (2.77).

---

While in real systems, pole-zero cancellation can be a delicate issue, we do not discuss it here.
2.4. Analysis of Sequences and Systems

- When \( M < N \), we have \( (N - M) \) zeros. If, moreover, all the poles are first order, we can express \( X(z) \) as

\[
X(z) = \sum_{k=1}^{N} \frac{A_k}{1 - d_k z^{-1}}, \quad A_k = (1 - d_k z^{-1})X(z) \bigg|_{z=d_k}. \tag{2.80}
\]

If we are able to write our \( X(z) \) as above, then

\[
x_n = \sum_{k=1}^{N} x_{k_n}, \tag{2.81}
\]

where, depending on the ROC we are given, each \( x_k \) will be either \((d_k)^n u_n\) or \(- (d_k)^{n-1} u_{n-1}\).

- When \( M \geq N \), then we can write \( X(z) \) as

\[
X(z) = \sum_{k=0}^{M-N} B_k z^{-k} + \sum_{k=1}^{N} \frac{A_k}{1 - d_k z^{-1}}. \tag{2.82}
\]

If there is a pole \( d_i \) of order \( s \), then

\[
X(z) = \sum_{k=0}^{M-N} B_k z^{-k} + \sum_{k=1,k \neq i}^{N} \frac{A_k}{1 - d_k z^{-1}} + \sum_{k=1}^{s} \frac{C_k}{(1 - d_i z^{-1})^k}. \tag{2.83}
\]

**Example 2.14 (Inversion using partial fraction expansion).** Given is

\[
X(z) = \frac{1 - z^{-1}}{1 - 5z^{-1} + 6z^{-2}} = \frac{1 - z^{-1}}{(1 - 2z^{-1})(1 - 3z^{-1})}, \tag{2.84}
\]

with poles at \( z = 2 \) and \( z = 3 \). We compute the coefficients as in (2.80):

\[
A_1 = \left. \frac{1 - z^{-1}}{1 - 2z^{-1}} \right|_{z=3} = 2, \quad A_2 = \left. \frac{1 - z^{-1}}{1 - 3z^{-1}} \right|_{z=2} = -1, \tag{2.85}
\]

yielding

\[
X(z) = \frac{2}{1 - 2z^{-1}} - \frac{1}{1 - 3z^{-1}}. \tag{2.86}
\]

Again, the original sequence will depend on the ROC. (If the ROC is given as the ring \( 2 < |z| < 3 \), what is \( x_n \)?)

**Inversion Using Power-Series Expansion** This method is most useful for finite-length sequences. For example, given \( X(z) = (1 - z^{-1})(1 - 2z^{-1}) \), we can expand it in its power-series form as

\[
X(z) = 1 - 3z^{-1} + 2z^{-2}. \tag{2.87}
\]

Knowing that each of the elements in this power series corresponds to a delayed Dirac impulse, we can read directly that the sequence is

\[
x_n = \delta_n - 3\delta_{n-1} + 2\delta_{n-2}. \tag{2.88}
\]
Example 2.15 (Inversion using power-series expansion). Suppose now

\[ X(z) = \log(1 + 2z^{-1}), \quad |z| > 2. \]  

(2.89)

To invert this \( z \)-transform, we use its power-series expansion from Table P1.20-1. Substituting \( x = 2z^{-1} \), we confirm that \( |x| = |2z^{-1}| < 2(\frac{1}{2}) = 1 \), and thus \( X(z) \) and its \( z \)-transform pair are:

\[ \log(1+2z^{-1}) = \sum_{n=1}^{\infty} (-1)^{n+1} \frac{2^n}{n} z^{-n} \xrightarrow{zT} x_n = \begin{cases} (-1)^{n+1} \frac{2^n}{n} n, & n \geq 1; \\ 0, & \text{otherwise.} \end{cases} \]

Properties of the \( z \)-transform

As expected, the \( z \)-transform has the same properties as the DTFT but for a larger class of sequences. As an example, we will compute the convolution of two sequences that may not have proper DTFTs by moving to a region of the \( z \)-transform plane where the sequences are absolutely summable, and then use the DTFT convolution property. A summary of \( z \)-transform properties can be found in Table 2.4.

<table>
<thead>
<tr>
<th>Sequence</th>
<th>( z )-transform</th>
<th>ROC</th>
</tr>
</thead>
<tbody>
<tr>
<td>( x_n ) real</td>
<td>( X^<em>(z) = X(z^</em>) )</td>
<td>( \text{ROC}_x )</td>
</tr>
<tr>
<td>( x_{-n} )</td>
<td>( X(z^{-1}) )</td>
<td>( 1/\text{ROC}_x )</td>
</tr>
<tr>
<td>( a_1 x_1 n + a_2 x_{2n} )</td>
<td>( a_1 X_1(z) + a_2 X_2(z) )</td>
<td>contains ( \text{ROC}<em>{x_1} \cap \text{ROC}</em>{x_2} )</td>
</tr>
<tr>
<td>( x_{n-k} )</td>
<td>( z^{-k} X(z) )</td>
<td>( \text{ROC}_x )</td>
</tr>
<tr>
<td>( a^n x_n )</td>
<td>( X(z/\alpha) )</td>
<td>( \alpha \text{ROC}_{x_1} )</td>
</tr>
<tr>
<td>( n x_n )</td>
<td>( -z(dX(z)/dz) )</td>
<td>( \text{ROC}_x )</td>
</tr>
<tr>
<td>( x_n^* )</td>
<td>( X^<em>(z^</em>) )</td>
<td>( \text{ROC}_x )</td>
</tr>
<tr>
<td>( \Re(x_n) )</td>
<td>( [X(z) + X^<em>(z^</em>)]/2 )</td>
<td>contains ( \text{ROC}_x )</td>
</tr>
<tr>
<td>( \Im(x_n) )</td>
<td>( [X(z) - X^<em>(z^</em>)]/(2j) )</td>
<td>contains ( \text{ROC}_x )</td>
</tr>
<tr>
<td>( x_n^* )</td>
<td>( X^<em>(1/z^</em>) )</td>
<td>( 1/\text{ROC}_x )</td>
</tr>
<tr>
<td>( x_{1n} \ast x_{2n} )</td>
<td>( X_1(z)X_2(z) )</td>
<td>contains ( \text{ROC}<em>{x_1} \cap \text{ROC}</em>{x_2} )</td>
</tr>
<tr>
<td>( x_{Nn} )</td>
<td>( (1/N) \sum_{k=0}^{N-1} X(W_N^k z^{1/N}) )</td>
<td>( (\text{ROC})^{1/N} )</td>
</tr>
</tbody>
</table>

| \( x_{Nn} \), \( n = kN \) | \( X(z^N) \) | \( (\text{ROC})^N \) |

\[ x_{Nn} \neq kN \]

Table 2.4: Properties of the \( z \)-transform.

Convolution Given a length-\( N \) sequence \( x = [x_0 \ x_1 \ \ldots \ x_{N-1}]^T \) and a length-\( M \) impulse response \( h = [h_0 \ h_1 \ \ldots \ h_{M-1}]^T \), their convolution is given by
2.4. Analysis of Sequences and Systems

(2.36). The $z$-transforms of $x$ and $h$ are\(^{20}\)

\[
H(z) = \sum_{n=0}^{M-1} h_n z^{-n}, \quad X(z) = \sum_{n=0}^{N-1} x_n z^{-n}.
\]

The product polynomial $H(z)X(z)$ has powers of $z^{-1}$ going from 0 to $M + N - 2$, and its $n$th coefficient is obtained from the coefficients in $H(z)$ and $X(z)$ that have powers summing to $n$, that is, the convolution $(h \ast x)$ given in (2.36). Therefore:

\[
(h \ast x) \underset{ZT}{\longleftrightarrow} H(z)X(z),
\]

where $\text{ROC}_{(h \ast x)}$ contains $\text{ROC}_h \cap \text{ROC}_x$.

**Example 2.16 (DTFT does not work, $z$-transform works).** Given are

\[
x_n = u_n, \quad h_n = \alpha^n u_n, \quad 0 < \alpha < \infty,
\]

where $u_n$ is the Heaviside sequence from (2.13). We would like to compute the convolution $y = (h \ast x)$, where $x$ is not absolutely summable, and for $\alpha \geq 1$, neither is $h$. While using the DTFT does not seem to be a viable option, for $|z| > \max\{\alpha, 1\}$, the $z$-transforms of both $x$ and $h$ are well defined:

\[
X(z) = \frac{1}{1 - z^{-1}}, \quad |z| > 1, \quad H(z) = \frac{1}{1 - \alpha z^{-1}}, \quad |z| > \alpha,
\]

and thus

\[
Y(z) = \frac{1}{(1 - \alpha z^{-1})(1 - z^{-1})}, \quad |z| > \max\{\alpha, 1\}.
\]

By partial fraction expansion, we can rewrite $Y(z)$ as

\[
Y(z) = \frac{A}{(1 - \alpha z^{-1})} + \frac{B}{(1 - z^{-1})},
\]

with $A = \alpha / (\alpha - 1)$ and $B = 1 / (1 - \alpha)$. Inverting each term leads to

\[
y_n = A\alpha^n u_n + Bu_n = \frac{1 - \alpha^{n+1}}{1 - \alpha} u_n.
\]

As a sanity check, we can compute the time-domain convolution directly:

\[
y_n = \sum_{k \in \mathbb{Z}} x_k h_{n-k} = \sum_{k=0}^{\infty} h_{n-k} = \sum_{k=0}^{n} \alpha^{n-k} = \frac{1 - \alpha^{n+1}}{1 - \alpha}, \quad n \geq 0,
\]

and is 0 otherwise. When $\alpha > 1$, $y_n$ is unbounded as well, but the $z$-transform $Y(z)$ exists for $|z| > \alpha$. \(\blacksquare\)

\(^{20}\)Again, for historical reasons, the polynomial is in $z^{-1}$, but one can think of $x = z^{-1}$.
Example 2.17 (Neither DTFT, nor $z$-transform are applicable). Here is an example where even the $z$-transform does not help:

$$x_n = 1, \quad n \in \mathbb{Z}, \quad h_n = \alpha^n u_n, \quad 0 < \alpha < 1.$$ 

In sequence domain,

$$y_n = (h * x)_n = \sum_{n \in \mathbb{N}} h_n x_{k-n} = \sum_{n \in \mathbb{N}} \alpha^n = \frac{1}{1 - \alpha}.$$ 

However, because $x_n$ is two-sided and not absolutely summable on either side, there is no $z$ for which the $z$-transform converges, thus leading to an empty ROC. This prohibits the use of the $z$-transform for the computation of this convolution.

**Autocorrelation** Given a real sequence $x$, its autocorrelation is given in (2.55), and by extension of the argument used for the DTFT, we have

$$a_n = \sum_{k \in \mathbb{Z}} x_k x_{k+n} \quad \xrightarrow{ZT} \quad A(z) = X(z)X(z^{-1}),$$

with an ROC that is the intersection of ROC$_x$ and $1$/ROC$_x$.

It is easy to verify that on the unit circle, the autocorrelation is the square magnitude of the spectrum $|X(e^{j\omega})|^2$ as in (2.55). This “quadratic form,” when extended to the $z$-plane, leads to a particular symmetry of poles and zeros when $X(z)$, and thus $A(z)$ as well, are rational functions.

**Proposition 2.3 (Rational autocorrelation of a stable and real sequence).** A rational $z$-transform $A(z)$ is the autocorrelation of a stable real sequence $x_n$, if and only if:

(i) Its complex poles and zeros appear in quadruples:

$$\{z_i, z_i^*, z_i^{-1}, (z_i^{-1})^*\}, \quad \{p_i, p_i^*, p_i^{-1}, (p_i^{-1})^*\}; \quad (2.92a)$$

(ii) Its real poles and zeros appear in pairs:

$$\{z_i, z_i^{-1}\}, \quad \{p_i, p_i^{-1}\}; \quad (2.92b)$$

(iii) Its zeros on the unit circle are double zeros:

$$\{z_i, z_i^*, z_i^{-1}, (z_i^{-1})^*\} = \{e^{j\omega_i}, e^{-j\omega_i}, e^{j\omega_i}, e^{-j\omega_i}\}, \quad (2.92c)$$

with the special case of possibly double zeros at $z = \pm 1$. There are no poles on the unit circle.

**Proof.** The proof follows from the following two facts:

---

21By $1$/ROC$_x$, we denote the set \{\$z \in \text{ROC}_x\}$. 

---
2.4. Analysis of Sequences and Systems

- $a_n$ is real, since $x_n$ is real. From Table 2.4, this means that:

$$A^*(z) = A(z^*) \Rightarrow p_i \text{ pole} \Rightarrow p_i^* \text{ pole}$$
$$z_i \text{ zero} \Rightarrow z_i^* \text{ zero} \quad (2.93a)$$

- $a_n$ is symmetric, since $a_{-n} = a_n$. From Table 2.4, this means that:

$$A(z^{-1}) = A(z) \Rightarrow p_i \text{ pole} \Rightarrow p_i^{-1} \text{ pole}$$
$$z_i \text{ zero} \Rightarrow z_i^{-1} \text{ zero} \quad (2.93b)$$

(i) From (2.93a)-(2.93b), we have that

$$p_i \text{ pole} \Rightarrow p_i^{-1} \text{ pole} \Rightarrow (p_i^*)^{-1} \text{ pole},$$

similarly for zeros, and we obtain the pole/zero quadruples in (2.92a).

(ii) If a zero/pole is real, it is its own conjugate, and thus, quadruples from (2.92a) become pairs from (2.92b).

(iii) $x_n$ stable, real $\Rightarrow$ zeros double on the unit circle: Since $x$ is stable, there are no poles. Since $x$ is real, $X^*(z) = X(z^*)$. Thus, a rational $A(z)$ has only zeros from $X(z)$ and $X(z^{-1})$.

$$z_i \text{ zero of } X(z) \Rightarrow z_i^{-1} \text{ zero of } X(z^{-1}) \Rightarrow (z_i^*)^{-1} \text{ zero of } X(z).$$

Thus, both $X(z)$ and $X(z^{-1})$ have $z_i$ as a zero, leading to double zeros on the unit circle.

zeros double on the unit circle $\Rightarrow x_n$ stable, real: (TBD.)

### Spectral Factorization

The particular pattern of poles and zeros which characterize a rational autocorrelation lead to a key procedure called spectral factorization. It is essentially taking the square root of $A(e^{j\omega})$, and by extension, of $A(z)$, factoring it into rational factors $X(z)$ and $X(z^{-1})$.

**Theorem 2.4 (Spectral factorization).** A rational $z$-transform $A(z)$ is the autocorrelation of a stable real sequence $x_n$ if and only if it can be factored as $A(z) = X(z)X(z^{-1})$.

Instead of proving the theorem, we discuss the actual procedure, based on the results of Proposition 2.3. As $A(z)$ is rational, it is of the form (2.77), with poles and zeros.

---

22Note that since $A(e^{j\omega}) \geq 0$ and real, one could write $X(e^{j\omega}) = \sqrt{A(e^{j\omega})}$. However, such a spectral root will in general not be rational.
being poles and zeros of $X(z)$. Thus, spectral factorization amounts to assigning poles and zeros from quadruples and pairs (2.92a)-(2.92c), to $X(z)$ and $X(z^{-1})$.

For the poles, there is a unique rule: take all poles inside the unit circle and assign them to $X(z)$. This is because $x$ stable requires $X(z)$ to have only poles inside the unit circle (Proposition 2.2), while $x$ real requires the conjugate pairs be kept together.

For the zeros, there is a choice, since we are not forced to only assign zeros inside the unit circle to $X(z)$. Doing so, however, creates a unique solution called the minimum-phase solution.\(^{23}\) It is now clear why it is important that the zeros on the unit circle appear in pairs: it allows for the assignment of one each to $X(z)$ and $X(z^{-1})$.

**Example 2.18 (Spectral factorization).** We now illustrate the procedure (TBD, see Figure 2.11):

(i) A finite symmetric sequence that has spectral factors, showing zero locations.

(ii) An infinite symmetric sequence that has spectral factors, showing pole locations.

(iii) A symmetric, finite-length sequence that is not always positive on the unit circle, and is therefore not an autocorrelation, showing the necessity of positive semi-definitiveness.

\section*{Crosscorrelation}

Given a real sequence $x$, its crosscorrelation is given in (2.59), and by extension of the argument used for the DTFT, we have

$$c_n = \sum_{k \in \mathbb{Z}} x_k h_{k+n} \xrightarrow{ZT} C(z) = X(z^{-1})H(z), \quad (2.94)$$

with ROC containing $1/\text{ROC}_x \cap \text{ROC}_h$. The proof of this is left as an exercise.

\section*{Rational Filters and Filter Design}

A major application of the $z$-transform is in the analysis and design of filters. As pointed out previously, there is a one-to-one relationship between a rational $z$-transform and realizable difference equations (the ones with finite numbers of coefficients). Therefore, designing a desirable filter is essentially the problem of strategically placing poles and zeros in the $z$-plane.

\(^{23}\)The name stems from the fact that among the various solutions, this one will create a minimal delay, or that the sequence is most concentrated towards the origin of time.
As simple as it sounds, this is a rather sophisticated problem, and has led to a vast literature and numerous numerical procedures. For example, a standard way to design FIR filters is to use a numerical optimization procedure such as the Parks-McClellan algorithm, which iteratively modifies coefficients so as to approach a desired frequency response. Rather than embarking on a tour of filter design, we study properties of certain classes of filters; pointers to filter design techniques are given in Further Reading.

**FIR Filters** An FIR filter has a $z$-transform that is a polynomial in $z^{-1}$,

$$H(z) = \sum_{n=0}^{L-1} h_n z^{-n},$$

and is given in its factored form as (2.76).

**Linear-Phase Filters** An important subclass of FIR filters are linear-phase filters, obtained when filters (of length $L$) possess certain symmetries:

- symmetric $h_n = h_{L-1-n}$
- antisymmetric $h_n = -h_{L-1-n}$

$$\text{ZT} \quad H(z) = \pm z^{-L+1}H(z^{-1}), \quad (2.95)$$

where $\pm$ indicates symmetry/antisymmetry, respectively. It is easy to see why the above is true; $H(z^{-1})$ reverses the filter, $z^{-L+1}$ makes it causal again, and $\pm$ then determines the type of symmetry. Figure 2.12 illustrates the above concepts for $L$ even and odd.

We now show an even-length, symmetric filter has linear phase; other cases follow similarly. Given (2.95), the DTFT of the filter is:

$$H(e^{j\omega}) = \sum_{n=0}^{L-1} h_n e^{-j\omega n} = \sum_{n=0}^{L/2-1} h_n \left(e^{-j\omega n} + e^{-j\omega(L-1-n)}\right)$$

$$= \sum_{n=0}^{L/2-1} h_n e^{-j\omega(L-1)/2} \left(e^{j\omega(n-(L-1)/2)} + e^{-j\omega(n-(L-1)/2)}\right)$$

$$= 2 \sum_{n=0}^{L/2-1} h_n \cos \left(\omega(n - \frac{1}{2}(L-1))\right) e^{-j\omega \frac{L-1}{2}} = re^{j\omega},$$

because of symmetry, the sum is real with a single phase factor making it linear.

**Allpass Filters** The basic single-zero/single-pole allpass building block has the $z$-transform was given in (2.65) in Example 2.11:

$$H(z) = \frac{z^{-1} - a^*}{1 - az^{-1}}, \quad (2.96)$$
with the zero \((a^*)^{-1}\) and pole \(a\). For stability, \(|a| < 1\) is required. A more general allpass filter is formed by cascading these elementary building blocks as

\[
H(z) = \prod_{i=1}^{N} \frac{z^{-1} - a_i^*}{1 - a_i z^{-1}} = z^{-N} \frac{A_*(z^{-1})}{A(z)}, \tag{2.97}
\]

(recall \(A_*\) denotes conjugation of coefficients but not of \(z\)). The autocorrelation of such an allpass filter is given by

\[
P(z) = H(z)H_*(z^{-1}) = \prod_{i=1}^{N} \frac{z^{-1} - a_i^*}{1 - a_i z^{-1}} \prod_{i=1}^{N} \frac{z - a_i}{1 - a_i^* z} = 1,
\]

and thus, an allpass filter has an autocorrelation function with \(p_m = \delta_m\) and \(P(z) = P(e^{j\omega}) = 1\). Poles and zeros appear in pairs as \(\{a, (a^*)^{-1}\} = \{r_0 e^{j\omega_0}, 1/r_0 e^{j\omega_0}\}\) for some real \(r_0 < 1\) and angle \(\omega_0\). They appear across the unit circle at reciprocal magnitudes, and is the reason why the magnitude \(|H(e^{j\omega})|\) is not influenced, but phase is, as shown in Figure 2.14.
2.4. Analysis of Sequences and Systems

Figure 2.14: Phase behavior of an allpass filter. (TBD)

2.4.3 Fourier Transform of Finite Sequences—DFT

The DFT is a computational tool; unlike the DTFT, it does not arise naturally from physical systems described by difference equations. This is true for the circular convolution as well, as it is the convolution operator that gives rise to the DFT. As such, it is questionable how to present the DFT. There are a couple of possibilities:

1. We can assume that sequences live on a discrete circle, define the appropriate convolution which will turn out to be circular, and then mimic the approach used for the DTFT by finding eigensequences of the convolution operator and defining the DFT based on that. In this approach, implicitly, we have chosen how our finite sequence behaves at the boundaries; it has been periodized. While one might argue whether this is natural or not, the approach is valid and leads to the DFT. (2) The second view asks the following question: Given that we have a finite sequence of unknown origin, how can we analyze it? One option is to discretize the existing tool—DTFT, by sampling. Thus, we start with the DFT from which we can deduce the circular convolution. The resulting sequence space will be the same as in the first case, discrete circle. As there does not seem to be a consensus whether the first or the second approach are the right ones, we present both. We leave it up to you to decide what makes more sense. The main point to remember is that when one deals with finite sequences (always in practice), one has to keep in mind the boundary issues; a simple cutout of a signal followed by the DFT might produce disastrous results if not handled with care.

Periodic Assumption Leads to the DFT

Periodizing Finite Sequences We thus start with a finite sequence. As discussed before, we must decide what happens at the boundaries, and choose to assume a circular extension, making the sequence $x_n$ periodic with an integer period $N$, that is

$$x_{n+kN} = x_n, \quad k \in \mathbb{Z}.$$  (2.98)

As such a sequence cannot be in $\ell^2(\mathbb{Z})$ (unless it is the zero sequence), we just require it to be bounded $x_n \in \ell^\infty(\mathbb{Z})$. This also means that one period has finite $\ell^2$-norm. Since a period is all we need to specify the sequence, we will define an $N$-dimensional vector $x_n$ describing a period, or

$$x = [x_0 \ x_1 \ \ldots \ x_{N-1}]^T.$$  (2.98)

Circular Convolution Take now an LSI filter with impulse response $h \in \ell^2(\mathbb{Z})$. The convolution of a periodic input $x_n$ with $h_n$ leads to a periodic output of the
same period, since
\[ y_{n+\ell N} = \sum_{k} x_{n+\ell N-k} h_k = \sum_{k} x_{n-k} h_k = y_n, \]
and, according to Proposition 2.1, the output is bounded. Therefore, the convolution \( y = (h * x) \) when \( x_n \) is periodic of period \( N \) is a map from \( \mathbb{R}^N \) (or \( \mathbb{C}^N \)) to \( \mathbb{R}^N \) (or \( \mathbb{C}^N \)).

We want to describe this map based on the impulse response \( h_n \). While \( h_n \) is not periodic, each subsequent piece of length \( N \) sees an identical piece of the input. More precisely, breaking the time index into pieces of length \( N \), or \( k = \ell N + m \) where \( m \in \{0, 1, \ldots, N-1\} \),
\[
y_n = \sum_{k \in \mathbb{Z}} x_{n-k} h_k \overset{(a)}{=} \sum_{\ell \in \mathbb{Z}} \sum_{m=0}^{N-1} x_{n-\ell N-m} h_{\ell N+m}
= \sum_{m=0}^{N-1} x_{n-m} \sum_{\ell \in \mathbb{Z}} h_{\ell N+m} \overset{(b)}{=} \sum_{m=0}^{N-1} x_{n-m} g_m
\]
(2.99)

where in (a) we split the sum over \( k = \ell N + m \) into pieces of length \( N \); in (b) we used periodicity of \( x_n \); and in (c) we introduced
\[
g_m = \sum_{\ell \in \mathbb{Z}} h_{\ell N+m}, \quad m \in \{0, 1, \ldots, N-1\},
\]
(2.100)
\[ g = [g_0 \ g_1 \ \cdots \ g_{N-1}]^T = \left[ \sum_{\ell \in \mathbb{Z}} h_{\ell N} \sum_{\ell \in \mathbb{Z}} h_{\ell N+1} \cdots \sum_{\ell \in \mathbb{Z}} h_{\ell (N+N-1)} \right]^T. \]

As we can see, the elements of this sequence are sums of the impulse response \( h_n \) at multiples of \( N \) shifted by \( m \), and is bounded since \( h_n \) is absolutely summable by assumption. Note that \( g_m \) is itself periodic with period \( N \). Because both \( x_n \) and \( g_n \) are periodic, their indices can be taken mod \( N \), leading to the periodic or circular convolution as
\[
y_n = \sum_{m=0}^{N-1} x_{[(n-m) \mod N]} g_m = \sum_{m=0}^{N-1} x_m g_{[(n-m) \mod N]} = (g * x). \]
(2.101)

We use the same notation for circular convolution as for the convolution we defined before without risking confusion; depending on the domain of the sequence, we will use one or the other.

In matrix format,
\[
\begin{bmatrix}
y_0 \\
y_1 \\
y_2 \\
\vdots \\
y_{N-1}
\end{bmatrix} = \begin{bmatrix}
g_0 & g_{N-1} & g_{N-2} & \cdots & g_1 \\
g_1 & g_0 & g_{N-1} & \cdots & g_2 \\
g_2 & g_1 & g_0 & \cdots & g_3 \\
\vdots & \vdots & \vdots & \ddots & \vdots \\
g_{N-1} & g_{N-2} & g_{N-3} & \cdots & g_0
\end{bmatrix}
\begin{bmatrix}
x_0 \\
x_1 \\
x_2 \\
\vdots \\
x_{N-1}
\end{bmatrix} = Gx,
\]
(2.102)
where $G$ is a circulant matrix containing $g$ (one could say it is a transpose of a circulant matrix defined in (1.94)). Therefore, the convolution of a periodic input with a filter $h_n$ results in a linear operator from $\mathbb{R}^N$ (or $\mathbb{C}^N$) to $\mathbb{R}^N$ (or $\mathbb{C}^N$) which is circulant, and the entries are given by a periodized impulse response $g_n$.

**Eigensequences of the Convolution Operator** Now that we have the appropriate convolution operator defined, we could just mimic what we did for the DTFT, starting with identifying the eigensequences of the convolution operator (eigenvectors of a circulant matrix). We could easily guess what these eigensequences are, but in this case, we have a better method. Since we can identify the space of our sequences with periodic sequences, to find them, take $v_{\omega,n} = e^{j\omega n}$ from (2.42). Since $v_{\omega,n}$ is periodic with period $N$:

$$v_{\omega,n + N} = e^{j\omega (n + N)} = v_{\omega,n} \Rightarrow e^{j\omega N} = 1 \Rightarrow \omega = \frac{2\pi k}{N},$$

(2.103)

for $k \in \mathbb{Z}$. Since $k$ and $k + \ell N$ lead to the same complex exponential sequence, we have $N$ complex exponential sequences of period $N$, indexed by $k \in \{0, 1, \ldots, N-1\}$ instead of $\omega$:

$$v_k = e^{j\frac{2\pi k}{N}n} = W_{N}^{-kn}, \quad v_k = \begin{bmatrix} 1 & W_{N}^{-k} & \ldots & W_{N}^{-(N-1)k} \end{bmatrix}^T.$$  

(2.104)

Thus, instead of infinitely many eigensequences, we now have $N$ of them. Let us check that these are indeed eigensequences of the convolution operator. We can check that easily by applying the circular convolution operator $G$ to $v_k$. We compute the $n$th element of the resulting vector:

$$(Gv_k)_n = \sum_{i=0}^{N-1} g_i W_{N}^{(n-i) \bmod N} k = \sum_{i=0}^{N-1} g_i W_{N}^{(n-i)k} = \sum_{i=0}^{N-1} g_i W_{N}^{-ik} W_{N}^{nk},$$

(2.105)

where (a) follows from the fact that if $(n - i) = \ell N + p$, then $(n - i) \bmod N = p$ and $W_{N}^{(n-i) \bmod N} = W_{N}^{p}$, but also $W_{N}^{(n-i)} = W_{N}^{(N+i)p} = W_{N}^{p}$; and (b) from the fact that $W_{N}^{nk}$ does not depend on $i$ and can be pulled out in front of the sum. Thus indeed, the vector $v_{k,n}$ is the eigenvector of the convolution operator $G$, with the associated eigenvalue

$$G_k = \sum_{i=0}^{N-1} g_i W_{N}^{ik}, \quad k \in \{0, 1, \ldots, N-1\}. $$

(2.106)

the frequency response.

Finding the appropriate Fourier transform of $x$ now amounts to projecting $x$ onto each of the eigenspaces $\mathbb{H}_k$ each spanned by one of the eigenvectors $v_k$, $k \in \{0, 1, \ldots, N-1\}$.
Definition 2.8 (Discrete Fourier Transform). Given a finite sequence \( x \), its discrete Fourier transform (DFT) \( X \) is given by

\[
X_k(e^{j\omega}) = \langle x, v_k \rangle = \sum_{n=0}^{N-1} x_n W_N^{nk}, \tag{2.107a}
\]

\( k \in \{0, 1, \ldots, N-1\} \). \( X_k \) is a length-\( N \) sequence called the spectrum of the sequence.

The inverse discrete Fourier transform (IDFT) is given by

\[
x_n = \frac{1}{N} \sum_{k=0}^{N-1} X_k W_N^{-nk}. \tag{2.107b}
\]

\( n \in \{0, 1, \ldots, N-1\} \).

**Sampling the DTFT to Obtain DFT**

When given a finite sequence of unknown origin to analyze, we might first turn to what we already have—the DTFT. To turn it into a tool for analyzing length-\( N \) sequences, we can simply sample it. As we need only \( N \) points, we can choose them anywhere; let us choose \( N \) uniformly spaced points in frequency, which are actually given by (2.103).

Thus, we evaluate the DTFT at \( k \in \{0, 1, \ldots, N-1\} \) as follows:

\[
X(e^{j\omega}) \bigg|_{\omega = \frac{2\pi k}{N}} = X(e^{j\frac{2\pi k}{N}}) = \sum_{n \in \mathbb{Z}} x_n e^{-j\frac{2\pi}{N}kn} = \sum_{i=0}^{N-1} \sum_{\ell \in \mathbb{Z}} x_{\ell N+i} e^{-j\frac{2\pi}{N}k(iN+i)} = \sum_{i=0}^{N-1} x_i e^{-j\frac{2\pi}{N}ki} = X_K,
\]

where (a) follows from sampling the DTFT uniformly at \( \omega = \frac{2\pi k}{N} \); (b) from the expression for the DTFT (2.47a); (c) from breaking the sum into subsequences modulo \( N \); (d) from \( x \) being finite of length \( N \), and thus \( \ell = 0 \). The final expression is the one for the DFT we have seen in (2.107a). Thus, sampling the DTFT results in the DFT.

We can now easily find the appropriate convolution operator corresponding to the DFT (we know what it is already, but we reverse engineer it here). The way to do it is to ask ourselves which operator will have as its eigensequences the columns of the DFT matrix \( F^* \) (since then, that operator is diagonalized by the DFT). In other words, which operator \( T \) will satisfy the following:

\[
Tv_k = \lambda_k v_k \quad \Rightarrow \quad TF^* = F^* \Lambda \quad \Rightarrow \quad T = \frac{1}{N} F^* \Lambda F.
\]

Even though we do not know \( \Lambda \), we know it is a diagonal matrix of eigenvalues \( \lambda_k \). Using the expressions (2.108)-(2.109) for \( F \) and \( F^* \), we can find the expression for the element \( T_{im} \) as

\[
T_{im} = \sum_{\ell=0}^{N-1} \lambda_\ell W_N^{\ell(m-i)},
\]
which shows $T$ to be a circulant matrix (circular convolution). We leave the details of the derivation as exercise.

What we have seen in this short account is how to reverse engineer the circular convolution given a finite sequence and the DTFT.

Properties of the DFT

Matrix View of the DFT  Introduce the DFT matrix $F$, formed from the $N$ eigensequences $v_k$:

$$
F = \begin{bmatrix}
1 & 1 & 1 & \ldots & 1 \\
1 & W_N & W_N^2 & \ldots & W_N^{N-1} \\
1 & W_N^2 & W_N^4 & \ldots & W_N^{2(N-1)} \\
\vdots & \vdots & \vdots & \ddots & \vdots \\
1 & W_N^{(N-1)} & W_N^{2(N-1)} & \ldots & W_N^{(N-1)^2} \\
v_0 & v_1 & v_2 & \ldots & v_{N-1}
\end{bmatrix},
$$

(2.108)

This matrix has the following inverse:

$$
F^{-1} = \frac{1}{N}F^* = \frac{1}{N} \begin{bmatrix}
1 & 1 & 1 & \ldots & 1 \\
1 & W_N^{-1} & W_N^{-2} & \ldots & W_N^{-(N-1)} \\
1 & W_N^{-2} & W_N^{-4} & \ldots & W_N^{-(2(N-1))} \\
\vdots & \vdots & \vdots & \ddots & \vdots \\
1 & W_N^{-(N-1)} & W_N^{-(2(N-1))} & \ldots & W_N^{-(N-1)^2} \\
v_0 & v_1 & v_2 & \ldots & v_{N-1}
\end{bmatrix},
$$

(2.109)

which shows that $F$ is a unitary matrix up to a scaling factor. To prove this, the product of the $\ell$th row of $F$ with the $k$th column of $F^*$ is of the form

$$
\frac{1}{N} \sum_{n=0}^{N-1} W_N^{-kn}W_N^{\ell n} = \frac{1}{N} \sum_{n=0}^{N-1} W_N^{n(k-\ell)}.
$$

Using the orthogonality of the roots of unity from (2.175c), this is equal to 1 when $k = \ell$, and otherwise it is 0.

The DFT and the Circular Convolution  Equipped with the DFT matrix and its inverse, we are ready to state the various properties seen so far.

**Proposition 2.5 (DFT and Circular Convolution).** Given are a periodic sequence $x_n$, or a length-$N$ sequence periodically extended with period $N$, and a filter $h_n$ with absolutely-summable impulse response $h \in \ell^1(\mathbb{Z})$. The periodized impulse response is $g_n = \sum_{\ell \in \mathbb{Z}} h_{\ell N + n}$. Then,

\[24\] A normalized version uses a $1/\sqrt{N}$ on both $F$ and its inverse.
Chapter 2. Discrete-Time Sequences and Systems

(i) The convolution \((h \ast x)\) is equivalent to the circular convolution \((g \ast x)\), \(y = Gx\), where \(y\) and \(x\) are vectors in \(\mathbb{R}^N\) (or \(\mathbb{C}^N\)) representing one period, and \(G\) is a circulant matrix having \(g\) as its first column.

(ii) The circulant matrix \(G\) has eigenvalues \(G_k\), \(k \in \{0, 1, \ldots, N - 1\}\), called frequency response, given by (2.106) and

\[
\begin{bmatrix}
G_0 \\
G_1 \\
\vdots \\
G_{N-1}
\end{bmatrix} = F \begin{bmatrix}
g_0 \\
g_1 \\
\vdots \\
g_{N-1}
\end{bmatrix},
\tag{2.110}
\]

that is, the vector of these eigenvalues is the DFT of the periodized impulse response.

(iii) The corresponding eigenvectors are the columns of \(F^*\) as in (2.106):

\[
GF^* = F^* \Lambda
\tag{2.111}
\]

where \(\Lambda = \text{diag}(G_0, G_1, \ldots, G_{N-1})\) is the diagonal matrix of eigenvalues (frequency response).

(iv) The circular convolution matrix \(G\) can be diagonalized as

\[
G = \frac{1}{N} F^* \Lambda F,
\tag{2.112}
\]

that is, the DFT diagonalizes the circular convolution.

(v) Circular convolution can be computed in DFT domain as

\[
(g \ast x)_n \xrightarrow{\text{DFT}} G_k X_k,
\tag{2.113}
\]

where \(G_k\) and \(X_k\) are the DFT coefficients of \(g_n\) and \(x_n\), respectively.

Proof. Parts (i)-(iii) follow readily from the developments above, and the details are the subject of Exercise 2.6. Part (iv) follows from Part (iii) by right multiplying with \(F\), using \(F^* F = NI\) and dividing by \(N\). For Part (v) write

\[
Gx \overset{(a)}{=} \frac{1}{N} F^* \Lambda F x,
\tag{2.114}
\]

\[
F(Gx) \overset{(b)}{=} \frac{1}{N} (F F^*) \Lambda F x \overset{(c)}{=} \Lambda F x \overset{(d)}{=} \Lambda X \overset{(e)}{=} \begin{bmatrix}
G_0 X_0 \\
G_1 X_1 \\
\vdots \\
G_{N-1} X_{N-1}
\end{bmatrix},
\tag{2.115}
\]

where the left side of (a) describes circular convolution as in (2.102), the equality in (a) follows from Part (iv); (b) follows by left multiplying (2.114) by \(F\); (c) from \((FF^*) = NI\); (d) from \(X\) being the DFT of \(x\), \(X = Fx\); and (e) from evaluating \(GX\).

The relationships between the convolution of a periodic sequence by \(h_n\) and the circular convolution by \(g_n\) are shown schematically in Figure 2.15 and explored further in the context of circulant matrices in solved Exercise 2.2.
2.4. Analysis of Sequences and Systems

Figure 2.15: Convolution of periodic sequence. (a) The filter $h_n$, convolved with the sequence $x_n$, leads to a periodic output. (b) The equivalent, periodized filter $g_n$, circularly convolved with $x_n$, leads to the same output. (TBD)

The DFT as an ONB That the set of vectors $v_k$, $k \in \{0, 1, \ldots, N - 1\}$ is an orthogonal set was shown when proving that the inverse of $F$ is given by $1/N F^*$ as in (2.109). Normalizing $v_k$ to have norm 1, it is easy to create an ONB for $\mathbb{R}^N$ (or $\mathbb{C}^N$),

$$\varphi_k = \frac{1}{\sqrt{N}} v_k = \frac{1}{\sqrt{N}} \left[ 1 \ W_N^k \ W_N^{2k} \ldots W_N^{(N-1)k} \right]^T,$$

(2.116)

for $k \in \{0, 1, \ldots, N - 1\}$, which satisfies

$$\langle \varphi_k, \varphi_\ell \rangle = \delta_{k-\ell}.$$

(2.117)

By the eigenvalue/eigenvector arguments developed in Proposition 2.5, each of the basis vectors spans an eigenspace of the circular convolution operator. Finally, by periodization, we have a natural representation of periodic sequences in terms of linear combinations of $N$ complex exponential sequences as in (2.104), that is, given either a period of a periodic sequence, or a finite sequence periodically extended, $[x_0 \ x_1 \ldots \ x_{N-1}]^T$, we can write it as

$$x_n = \sum_{k=0}^{N-1} X_k \varphi_{kn},$$

(2.118a)

where

$$X_k = \langle x, \varphi_k \rangle = \sum_{n=0}^{N-1} x_n \varphi_{kn}^*,$$

(2.118b)

which is an ONB representation of periodic sequences. The coefficients $X_k$ are the expansion coefficients, and the representation is the DFT.

Example 2.19. Consider the “square wave” given by

$$x_n = \begin{cases} 
1, & 0 \leq n < N; \\
0, & K \leq n < N,
\end{cases}$$

periodic with period $N$. The expansion coefficients are

$$X_k = \sum_{n=0}^{K-1} W_N^{-nk} = \frac{1 - W_N^{Kk}}{1 - W_N^k}.$$

Note that $X_{N-k}$ is related to $X_k$,

$$X_{N-k} = \sum_{n=0}^{K-1} W_N^{-n(N-k)} = \sum_{n=0}^{K-1} W_N^{nk} = X_k^*.$$
which is true in general for real sequences. Table 2.5 summarizes the important properties of the DFT, while Exercise 2.7 works out some details.

<table>
<thead>
<tr>
<th>Time domain</th>
<th>DFT domain</th>
<th>Property</th>
</tr>
</thead>
<tbody>
<tr>
<td>$x_n = \frac{1}{N} \sum_{k=0}^{N-1} x_k W_N^{-nk}$</td>
<td>$X_k$</td>
<td>Inverse DFT</td>
</tr>
<tr>
<td>$x_n$</td>
<td>$X_k = \sum_{n=0}^{N-1} x_n W_N^{-nk}$</td>
<td>Forward DFT</td>
</tr>
<tr>
<td>$y_n = x_{-n}$</td>
<td>$Y_k = X_k^*$</td>
<td>Time reversal</td>
</tr>
<tr>
<td>$x_n$ real</td>
<td>$X_k = X_{N-k}^*$</td>
<td>Hermitian symmetry</td>
</tr>
<tr>
<td>$x_n = x_{-n}$</td>
<td>$X_k$ real</td>
<td>Real, symmetric sequence</td>
</tr>
<tr>
<td>$x_n = -x_{-n}$</td>
<td>$X_k$ imaginary</td>
<td>Real, antisymmetric sequence</td>
</tr>
<tr>
<td>$(g * x)$</td>
<td>$G_k X_k$</td>
<td>Circular convolution</td>
</tr>
<tr>
<td>$x g$</td>
<td>$(G * X)$</td>
<td>Modulation</td>
</tr>
</tbody>
</table>

Table 2.5: Properties of the DFT.

**Relation between Linear and Circular Convolutions** We have seen how a linear convolution of a periodic or periodized sequence with $h$, can be computed as a circular convolution with the periodized impulse response $g$. When the input $x$ is not periodic and has a finite number of nonzero samples $M$, with the impulse response of $h_n$ with a finite number of nonzero samples $L$, the result of the convolution has also a finite number of nonzero samples, since

$$y_n = (h * x) = \sum_{k=0}^{L-1} h_k x_{n-k}$$

is nonzero when $(n - k)$ is between 0 and $M - 1$, or, equivalently, it is nonzero when $0 \leq n \leq L + M - 2$, since $k$ is between 0 and $L - 1$. Thus, the result of the convolution is of length $L + M - 1$:

$$y = \left[ \begin{array}{cccccc} \cdots & 0 & y_0 & y_1 & \cdots & y_{L+M-1} & 0 & \cdots \end{array} \right]^T.$$
2.4. Analysis of Sequences and Systems

Figure 2.16: Equivalence of circular and linear convolutions: (a) Linear convolution of two sequences of length $M$ and $L$, resulting in a sequence of length $L + M - 1$. (b) Periodic convolution with a period $N \geq L + M - 1$ leads to the same result as in (a). (TBD)

$N$ for both the input and the impulse response. Choosing this common period has implications in computing convolutions, as is shown next.

Observe that if a sequence $x$ has $M$ nonzero coefficients, then its periodized version, with a period $N \geq M$, conserves the $M$ coefficients of the sequence. This, of course, is not the case when $N < M$ (as the copies of the base period will overlap).

Proposition 2.6 (Equivalence of circular and linear convolutions). For two finite-length sequences $x$ and $h$ of lengths $M$ and $L$, respectively, linear and circular convolution lead to the same result, if $N$, the period of the circular convolution, satisfies

$$N \geq M + L - 1. \quad (2.120)$$

Proof. Take $x = [x_0 \ x_1 \ \ldots \ x_{M-1}]^T$ and $h = [h_0 \ h_1 \ \ldots \ h_{L-1}]^T$. Assuming $x$ and $h$ are of infinite support with just a finite number of nonzero samples, the linear convolution $y_{\text{lin}}(n)$ is given by (2.119), while the circular convolution (see (2.101)) is

$$y_{\text{circ}}(n) = \sum_{k=0}^{N-1} h_k x_{((n-k) \mod N)} = \sum_{k=0}^{n} h_k x_{n-k} + \sum_{k=n+1}^{L-1} h_k x_{n+N-k},$$

for $n \in \{0, 1, \ldots, N-1\}$. In the above, we broke the sum into positive indices of $x_n$, while the negative ones, mod $N$, are made positive by adding $N$. Also, $k$ goes from 0 to $(N-1)$, but stops at $(L-1)$ since $h_k$ is zero after that. Now it suffices to show that the second sum is zero when $N \geq L + M - 1$. When the index $k$ goes from $(N+1)$ to $(L-1)$, $x_n$ goes from $x_{N-1}$ to $x_{N-L-1}$. But since $N \geq L + M - 1$, and $x_n$ is zero for $n \geq M$, $x_{n-N-k}$ is zero, and the right sum vanishes, proving that $y_{\text{lin}}(n) = y_{\text{circ}}(n)$. \qed

Figure 2.16 depicts this equivalence, and the following example shows it in matrix notation in a particular case.

Example 2.20 (Equivalence of circular and linear convolutions). We now look at a length-3 filter convolved with a length-4 sequence. The result of the linear convolution is of length 6, and is given by

$$\begin{bmatrix} y_0 \\ y_1 \\ y_2 \\ y_3 \\ y_4 \\ y_5 \end{bmatrix} = \begin{bmatrix} h_0 & 0 & 0 & 0 \\ h_1 & h_0 & 0 & 0 \\ h_2 & h_1 & h_0 & 0 \\ 0 & h_2 & h_1 & h_0 \\ 0 & 0 & h_2 & h_1 \\ 0 & 0 & 0 & h_2 \end{bmatrix} \begin{bmatrix} x_0 \\ x_1 \\ x_2 \\ x_3 \end{bmatrix}.$$
Strictly speaking, we should have used infinite matrices above; however, since \( y_n \) is nonzero only for \( n \in \{0, 1, \ldots, 5\} \), we can use finite matrices only.

To calculate circular convolution, we choose \( N = M + L - 1 = 6 \), and form a \( 6 \times 6 \) circulant matrix \( G \) by using \( h \) as its first column. Then the circular convolution leads to the same result as before:

\[
\begin{bmatrix}
    y_0 \\
y_1 \\
y_2 \\
y_3 \\
y_4 \\
y_5
\end{bmatrix} =
\begin{bmatrix}
    h_0 & 0 & 0 & 0 & h_2 & h_1 \\
    h_1 & h_0 & 0 & 0 & h_2 & 0 \\
    h_2 & h_1 & h_0 & 0 & 0 & 0 \\
    0 & h_2 & h_1 & h_0 & 0 & 0 \\
    0 & 0 & h_2 & h_1 & h_0 & 0 \\
    0 & 0 & 0 & h_2 & h_1 & h_0
\end{bmatrix}
\begin{bmatrix}
x_0 \\
x_1 \\
x_2 \\
x_3 \\
x_4 \\
x_5
\end{bmatrix}.
\]

Note that if the period \( N \) had been chosen smaller (for example, \( N = 5 \)), then the equivalence would not hold anymore.

This example also shows that to compute the linear convolution, we can compute the circular convolution instead by choosing the appropriate period \( N \geq M + L - 1 \). Then, the circular convolution can be computed using the DFT of size \( N \) (see Exercise 2.8).

### 2.5 Multirate Sequences and Systems

So far, we considered sequences in time indexed by integers, and the time index was assumed to be the same for all sequences. Physically, this is as if we had observed a physical process, and each term in the sequence corresponded to a sample of the process taken at regular intervals (for example, every second).

In multirate sequence processing, different sequences may have different time scales. Thus, the index \( n \) of the sequence may refer to different “physical” times for different sequences. The question is both why do that and how to go between these different scales. Let us look at a simple example. Start with a sequence \( x_n \) and derive a downsampled sequence \( y_n \) by dropping every other sample

\[
y_n = x_{2n} \rightarrow [\ldots x_{-2} \ x_0 \ x_2 \ \ldots]^T,
\]

Clearly, if \( x_n \) is the sample of a physical process taken at \( t = n \), then \( y_n \) is a sample taken at time \( t = 2n \). In other words, \( y_n \) has a timeline with intervals of 2 seconds if \( x_n \) has a timeline with intervals of 1 second; the clock of the process is twice as slow. The process above is called downsampling by 2, and while simple, it has a number properties we will study in detail. For example, is is irreversible; once we remove samples, we cannot go back from \( y_n \) to \( x_n \), and is shift-varying, requiring more complicated analysis.

The dual operation to downsampling is upsampling. For example, upsampling a sequence \( x_n \) by 2 results in a new sequence \( y_n \) by inserting zeros between every two samples as

\[
y_n = \begin{cases} 
x_{n/2}, & n \text{ even}; \\
0, & n \text{ odd},
\end{cases} \rightarrow [\ldots x_{-1} \ 0 \ x_0 \ 0 \ x_1 \ 0 \ \ldots]^T
\]

\(^{25}\)The issue of when this sampling can be done, and how, is treated in detail in the next chapter.
2.5. Multirate Sequences and Systems

The index of $y_n$ corresponds to a time that is half of that for $x_n$. For example, if $x_n$ has an interval of 1 second between samples, then $y_n$ has intervals of 0.5 seconds; the clock of the process is twice as fast.

What we just saw for rate changes by 2 can be done for any integer as well as rational rate changes (the latter ones by combining upsampling by $N$ and downsampling by $M$). In addition, to smooth the sequence before dropping samples downsampling is preceded by lowpass filtering, while to fill in the zeros upsampling is followed by interpolation, thus combining filtering with sampling rate changes. The multirate operations are used in any number of today’s physical systems, from MP3 players, to JPEG, MPEG, to name a few.

The purpose of this section is to study these various multirate operations and their consequences on the resulting sequences and their spectra. While the periodically shift-varying nature of multirate systems does complicate analysis, we use a relatively simple and powerful tool called polyphase analysis to mediate the problem. The outline of the section follows naturally the above introduction, moving from down- and upsampling together with filtering to polyphase analysis. Multirate processing, while not standard DSP material, is central to filter bank and wavelet constructions.

2.5.1 Downsampling

Downsampling by 2

Downsampling by 2, as introduced in (2.121) and shown in Figure 2.17(a), is clearly not shift invariant. If the input is $x_n = \delta_n$, the output is $y_n = \delta_n$; however, if the input is $x_n = \delta_{n-1}$, the output is zero! It is instructive to look at (2.121) in matrix notation:

$$
\begin{bmatrix}
\vdots \\
y_{-1} \\
y_0 \\
y_1 \\
y_2 \\
y_3 \\
\vdots \\
\end{bmatrix}
= 
\begin{bmatrix}
\vdots \\
\cdot \\
\cdot \\
\cdot \\
\cdot \\
\cdot \\
\cdot \\
\end{bmatrix}
\begin{bmatrix}
x_{-2} \\
x_{-1} \\
x_0 \\
x_1 \\
x_2 \\
x_3 \\
x_4 \\
\vdots \\
\end{bmatrix}
\begin{bmatrix}
\vdots \\
\cdot \\
\cdot \\
\cdot \\
\cdot \\
\cdot \\
\cdot \\
\end{bmatrix},
$$

where $D_2$ stands for the operator describing downsampling by 2. Inspection of $D_2$ shows that it is similar to an identity matrix, but with the odd rows taken out. Intuitively, it is a "rectangular" operator, with the output space being a subspace of the input space (one would like to say of half the size, but both are infinite dimensional). It is a linear operator and has no inverse.

Other possible terms are subsampling or decimation.

\(^{26}\)Other possible terms are subsampling or decimation.
Figure 2.17: Downsampling by 2. (a) The block diagram of the downsampling operation by 2. (b) The spectrum of a sequence and (c) its downsampled version.

Given an input sequence \( x_n \), the output \( y_n \) keeps only every second sample, given by (2.121). To find the \( z \)-transform \( Y(z) \), the downsampled sequence may be seen as a sequence of even samples with ones set to zero \( x_{0n} \), followed by contraction of the sequence by removing those zeros. Thus,

\[
x_{0n} = \begin{cases} x_n, & n \text{ even;} \\ 0, & n \text{ odd.} \end{cases}
\]

Its \( z \)-transform is

\[
X_0(z) = \frac{1}{2} \left[ X(z) + X(-z) \right] \\
= \frac{1}{2} \left[ (\cdots + x_0 + x_1z^{-1} + x_2z^{-2} + \cdots) + (\cdots x_0 - x_1z^{-1} + x_2z^{-2} + \cdots) \right] \\
= (\cdots + x_{-2}z^2 + x_0 + x_2z^{-2} + \cdots) = \sum_{n \in \mathbb{Z}} x_{2n}z^{-2n},
\]

canceling the odd powers of \( z \) cancel and keeping the even ones. We now get \( Y(z) \) by contracting \( X_0(z) \) as:

\[
Y(z) = \sum_{n \in \mathbb{Z}} x_{2n}z^{-n} = X_0(z^2) = \frac{1}{2} \left[ X(z^2) + X(-z^2) \right]. \quad (2.124)
\]

To find its DTFT, we simply evaluate \( Y(z) \) at \( z = e^{j\omega} \):

\[
Y(e^{j\omega}) = \frac{1}{2} \left[ X(e^{j\omega}) + X(e^{-j\omega}) \right], \quad (2.125)
\]
2.5. Multirate Sequences and Systems

where \(-e^{j\omega/2}\) can be written as \(e^{j(\omega-2\pi)/2}\) since \(e^{-j\pi} = -1\). With the help of Figure 2.17, we now analyze this formula. \(X(e^{j\omega/2})\) is a stretched version of \(X(e^{j\omega})\) (by a factor of 2) and is \(4\pi\) periodic (since downsampling contracts time, it is natural that frequency expands accordingly). This is shown as solid line in Figure 2.17(c). \(X(e^{j(\omega-2\pi)/2})\) is not only a stretched version of \(X(e^{j\omega})\), but also shifted by \(2\pi\), shown as dashed line in Figure 2.17(c). The sum is again \(2\pi\)-periodic, since \(Y(e^{j\omega}) = Y(e^{j(\omega-k2\pi)})\). Both stretching and shifting create “new frequencies”, an unusual occurrence, since in LSI processing, no new frequencies ever appear. The shifted version \(X(e^{j(\omega-2\pi)/2})\) is called the aliased version of the original (a ghost image!).

**Example 2.21 (Downsampling of sequences).** To see how downsampling can change the nature of the sequence, take \(x_n = (-1)^n = \cos \pi n\) (the highest-frequency discrete sequence). The result, its downsampled version, is \(y_n = x_{2n} = \cos 2\pi n = 1\) — the lowest-frequency discrete sequence (a constant):

\[
\begin{bmatrix} 
... & 1 & -1 & 1 & 1 & ... 
\end{bmatrix}^T \xrightarrow{2 \downarrow} \begin{bmatrix} 
... & 1 & 1 & 1 & 1 & ... 
\end{bmatrix}^T.
\]

Consider now \(x_n = \alpha^n u_n, |\alpha| < 1\). From Table 2.3, the z-transform of this right-sided geometric series is:

\[X(z) = \frac{1}{1 - \alpha z^{-1}}.\]

and its downsampling version is

\[y_n = x_{2n} = \alpha^{2n} u_{2n} = \beta^n u_n,\]

with \(\beta = \alpha^2\). Using again Table 2.3, the z-transform of the downsampled sequence is

\[Y(z) = \frac{1}{1 - \alpha^2 z^{-1}}.\]

We can obtain the same expression using (2.124):

\[
\frac{1}{2} \left( \frac{1}{1 - \alpha z^{-1/2}} + \frac{1}{1 + \alpha z^{-1/2}} \right) = \frac{(1 + \alpha z^{1/2}) + (1 - \alpha z^{-1/2})}{(1 - \alpha z^{-1/2})(1 + \alpha z^{-1/2})} = \frac{1}{1 - \alpha^2 z^{-1}}.
\]

**Downsampling by \(N\)** The formulas for downsampling by 2 can be generalized to downsampling by \(N\). Using the usual shorthand \(W_N\) from (2.174), we can write the downsampled-by-\(N\) sequence \(y_n\),

\[y_n = x_{nN}, \tag{2.126}\]

In the ZT domain as

\[Y(z) = \frac{1}{N} \sum_{k=0}^{N-1} X(W_N^k z^{-1/2}). \tag{2.127}\]
The DTFT of $y_n$ is obtained by evaluating the above on the unit circle and using

$$W_N^k z^k \bigg|_{z = e^{j\omega}} = e^{-j\frac{\pi}{N}k}e^{j\omega} = e^{j\frac{(\omega - 2\pi k)}{N}},$$

leading to

$$Y(e^{j\omega}) = \frac{1}{N} \sum_{k=0}^{N-1} X(e^{j\frac{(\omega - 2\pi k)}{N}}). \quad (2.128)$$

The proof is an extension of the $N = 2$ case, and we leave it as an exercise.

### 2.5.2 Upsampling

**Upsampling by 2** Upsampling by 2 as introduced in (2.122) and shown in Figure 2.18(a), also creates new frequencies, as it “stretches” time by a factor of 2. In matrix notation, similarly to (2.123a), we have

$$
\begin{bmatrix}
\vdots \\
y_{-2} \\
y_{-1} \\
y_0 \\
y_1 \\
y_2 \\
y_3 \\
y_4 \\
\vdots
\end{bmatrix} = 
\begin{bmatrix}
\vdots \\
\cdot \cdot \cdot \\
\cdot \cdot \cdot \\
\cdot \cdot \cdot \\
\cdot \cdot \cdot \\
\vdots \\
\cdot \cdot \cdot \\
\cdot \cdot \cdot \\
\cdot \cdot \cdot \\
\cdot \cdot \cdot \\
\end{bmatrix}
\begin{bmatrix}
\begin{bmatrix} x_{-1} \\
0 \\
x_0 \\
x_1 \\
x_2 \\
x_3 \\
0 \\
x_x \\
\vdots
\end{bmatrix}
\end{bmatrix} =
\begin{bmatrix}
\vdots \\
x_{-1} \\
0 \\
x_0 \\
x_1 \\
x_2 \\
0 \\
x_x \\
\vdots
\end{bmatrix},
\quad (2.129a)
$$

where $U_2$ stands for the upsampling-by-2 operator. The matrix $U_2$ looks like an identity matrix with rows of zeros in between every two rows. Another way to look at it is as an identity matrix with every other column removed. It is a “rectangular” matrix in the sense that it has twice as many rows as columns—but when we say that, remember that twice infinity is infinity. The downsampling and upsampling operators are transposes of each other, $D_2$ removes every other row, while $U_2$ adds a row in between every two rows:

$$U_2 = D_2^T. \quad (2.130)$$

As $D_2^* = D_2^T$ is the adjoint of the downsampling operator, (2.130) means that upsampling and downsampling are adjoints of each other.

In the ZT domain, the expression for upsampling by 2 is

$$Y(z) = \sum_{n \in \mathbb{Z}} y_n z^{-n} = \sum_{n \in \mathbb{Z}} x_n z^{-2n} = X(z^2), \quad (2.131)$$

$$y = U_2 x, \quad (2.129b)$$
2.5. Multirate Sequences and Systems

![Diagram of upsampling](image)

**Figure 2.18:** Upsampling by 2. (a) The block diagram of the upsampling operation. (b) The spectrum of a sequence and (c) its upsampled version.

since all odd terms of $y_n$ are zero, while the even ones are $y_{2n} = x_n$ following (2.122). In frequency domain,

$$Y(e^{j\omega}) = X(e^{j2\omega}), \quad (2.132)$$

a contraction by a factor of 2 as shown in Figure 2.18(b) and (c).

**Example 2.22.** Take the constant sequence $x_n = 1$. Its upsampled version is $y_n = [\ldots 1 \ 0 \ [1] \ 0 \ 1 \ \ldots]^T$, and can be written as

$$y_n = \frac{1}{2} [1 + (-1)^n] = \frac{1}{2} (\cos 2\pi n + \cos \pi n),$$

indicating that it contains both the original frequency (constant, DC value at the origin) and a new high frequency (at $\omega = \pi$, since $(-1)^n = e^{j\pi n} = \cos \pi n$).

**Relations between Upsampling and Downsampling** What about combinations of upsampling and downsampling? Clearly, upsampling by 2 followed by downsampling by 2 results in the identity:

$$D_2 U_2 = I, \quad (2.133)$$

since the zeros added by upsampling are at odd-indexed locations and are subsequently eliminated by downsampling. The operations in the reverse order is more
interesting; Downsampling by 2 followed by upsampling by 2 results in a sequence where all odd-indexed samples have been replaced by zeros, or

\[
\begin{bmatrix}
\cdots & x_{-1} & x_0 & x_1 & x_2 & \cdots
\end{bmatrix}^T \rightarrow \downarrow \frac{1}{2} \rightarrow \begin{bmatrix}
\cdots & 0 & x_0 & 0 & x_2 & \cdots
\end{bmatrix}^T
\]

This operator, \( P = U_2D_2 \), is an orthogonal projection operator onto the subspace of all even-indexed samples. To verify this, we check idempotency (for the operator to be a projection, see Definition 1.20),

\[
P^2 = (U_2D_2)(U_2D_2) = U_2(D_2U_2)D_2 = U_2D_2 = P,
\]

using (2.133), as well as self-adjointness (for the operator to be an orthogonal projection, see Definition 1.21),

\[
P^* = (U_2D_2)^T = D_2^TU_2^T = U_2D_2 = P,
\]

using (2.130).

Upsampling by \( N \) Finally, generalize the formulas to upsampling by \( N \). A sequence upsampled by \( N \) is of the form

\[
y_n = \begin{cases}
    x_{n/N}, & n = \ell N; \\
    0, & \text{otherwise},
\end{cases} \quad (2.134)
\]

that is, \((N-1)\) zeros are inserted between every two samples. By the same arguments used for (2.131), the \( z \)-transform of \( y_n \) is simply

\[
Y(z) = X(z^N) \quad (2.135)
\]

or, on the unit circle for the DTFT,

\[
Y(e^{j\omega}) = X(e^{jN\omega}). \quad (2.136)
\]

2.5.3 Filtering and Interpolation

We have seen above that for both downsampling and upsampling, new frequencies appear, and is thus often desirable to filter out such frequencies. While lowpass filtering is usually applied before downsampling to avoid aliasing, interpolation (also form of filtering) is usually applied to fill in the zeros between the samples of the original sequence to smooth the output sequence. We now consider these two cases in more detail.

Filtering Followed by Downsampling Consider filtering followed by downsampling by 2, as illustrated in Figure 2.19. For simplicity, we look at a causal FIR filter of length \( L \), \( h = [\cdots \ 0 \ h_0 \ h_1 \ \cdots \ h_{L-1} \ 0 \ \cdots]^T \). We denote by \( H \) the
2.5. Multirate Sequences and Systems

band Toeplitz convolution operator corresponding to \(h_n\) (see (1.95) and (1.96). An example for \(L = 4\) is

\[
H = \begin{bmatrix}
\vdots & \vdots & \vdots & \vdots & \vdots & \\
\vdots & h_3 & h_2 & h_1 & h_0 & 0 & 0 & \cdots \\
\vdots & 0 & h_3 & h_2 & h_1 & h_0 & 0 & \cdots \\
\vdots & 0 & 0 & h_3 & h_2 & h_1 & h_0 & \cdots \\
\vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \\
\end{bmatrix}
\]

Convolution with the filter \(h_n\) followed by downsampling by 2 can then be written as \(y = D_2Hx\),

\[
\begin{bmatrix}
y_{-1} \\
y_0 \\
y_1 \\
y_2 \\
\vdots
\end{bmatrix} = \begin{bmatrix}
\vdots & \vdots & \vdots & \vdots & \vdots & \\
\vdots & h_1 & h_0 & 0 & 0 & 0 & \cdots \\
\vdots & 0 & h_3 & h_2 & h_1 & h_0 & 0 & \cdots \\
\vdots & 0 & 0 & h_3 & h_2 & h_1 & h_0 & \cdots \\
\vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \\
\end{bmatrix} \begin{bmatrix}
x_{-1} \\
x_0 \\
x_1 \\
x_2 \\
\vdots
\end{bmatrix} = D_2Hx; \quad (2.137)
\]

nothing else but the convolution operator \(H\) with the odd rows removed. From the above matrix-vector product, we can also write the filtered and downsampled sequence using inner products as

\[
y_n = \sum_{k=0}^{L-1} h_k x_{2n-k} = \langle x_{2n-k}, h_k \rangle_k = \sum_{k=0}^{L-1} x_k h_{2n-k} = \langle h_{2n-k}, x_k \rangle_k. \quad (2.138)
\]

Note how the convolution and the inner product operators are related by time reversal:

\[
(h \ast x)_n = \langle h_{2n-k}, x_k \rangle_k. \quad (2.139)
\]

in the ZT domain, apply (2.124) to \(H(z)X(z)\),

\[
Y(z) = \frac{1}{2} \left[ H(z^{1/2})X(z^{1/2}) + H(-z^{1/2})X(-z^{1/2}) \right], \quad (2.140)
\]

and, on the unit circle for the DTFT,

\[
Y(e^{j\omega}) = \frac{1}{2} \left[ H(e^{j\omega/2})X(e^{j\omega/2}) + H(e^{-j\omega/2})X(e^{-j\omega/2}) \right]. \quad (2.141)
\]

Figure 2.19 shows an input spectrum and its downsampled version. Figure 2.17(c) showed the spectrum without filtering, while Figure 2.19(c) shows the spectrum with filtering. As can be seen, when no filtering is used, then aliasing perturbs the spectrum. When ideal lowpass filtering is used (Figure 2.19(b)), the spectrum from \(-\pi/2\) to \(\pi/2\) is conserved, the rest is put to zero, then no aliasing occurs, and the central lowpass part of the spectrum is conserved in the downsampled version (Figure 2.19(c)).
Example 2.23 (Filtering before downsampling). Consider the 2-point average filter $h_n = \frac{1}{2}(\delta_n + \delta_{n-1})$, whose output, downsampled by 2, $y = D_2Hx$, is

$$y_n = \frac{1}{2}(x_{2n} + x_{2n-1}).$$

Because of filtering, all samples of the input influence the output, as opposed to downsampling without filtering, where the odd-indexed samples had no impact.

Upsampling Followed by Filtering Consider now filtering after upsampling, as shown in Figure 2.20. Using the matrix-vector notation, we can write the output as the product $GU_2$, where $G$ is a band Toeplitz operator just like $H$:

$$GU_2 = \begin{bmatrix} x_{-1} \\ \vdots \\ x_0 \\ x_1 \\ \vdots \\ x_{2n-1} \\ \vdots \end{bmatrix} = GU_2x; \quad (2.142)$$
nothing else but the convolution operator $H$ with the odd columns removed. Using inner products, we can express $y_n$ as

$$y_n = \sum_k g_{n-2k}x_k = \langle x_k, g_{n-2k} \rangle_k.$$  

(2.143)

Another way to look at (2.142)–(2.143) is to see that each input sample $x_k$ generates an impulse response $g_n$ delayed by $2k$ samples and weighted by $x_k$. In the ZT domain, the output of filtering followed by downsampling is

$$Y(z) = G(z)X(z^2),$$  

(2.144)

while it is

$$Y(e^{j\omega}) = G(e^{j\omega})X(e^{j2\omega})$$  

(2.145)

in the DTFT domain.

Figure 2.20 shows a spectrum, its upsampled version, and finally, its ideally filtered or interpolated version. As can be seen, the “ghost” spectrum at $\omega\pi$ is removed, and only the base spectrum around the origin remains.

**Example 2.24 (Upsampling and Interpolation).** Consider the piecewise constant interpolator, that is, a filter $g_n$ with impulse response $g_n = \delta_n + \delta_{n-1}$. The sequence $x_n$, upsampled by 2 and interpolated with $g_n$ leads to

$$y_n = \begin{bmatrix} \cdots & x_{-1} & x_{-1} & x_0 & x_0 & x_1 & x_1 & \cdots \end{bmatrix}^T,$$

(2.146)
Figure 2.21: Upsampling and interpolation. (a) Example signal. (b) Piecewise constant interpolation. (c) Linear interpolation.

A “staircase” sequence, with stairs of height \(x_n\) and length 2. A smoother interpolation is obtained with a linear interpolator:

\[ g_n = \frac{1}{2} \delta_{n-1} + \delta_n + \frac{1}{2} \delta_{n+1}. \]

From (2.142) or (2.143), we can see that even-indexed outputs are equal to input samples (at half the index), while odd-indexed outputs are averages of two input samples,

\[ y_n = \begin{cases} x_n/2, & n \text{ even}; \\ \frac{1}{2} x_{n+1/2} + x_{n-1/2}, & n \text{ odd}. \end{cases} \quad (2.147) \]

\[ y = [\ldots, x_{-1}, \frac{1}{2}(x_{-1} + x_0), x_0, \frac{1}{2}(x_0 + x_1), x_1, \ldots]^T. \]

Compare (2.147) with (2.146) to see why (2.147) is a smoother interpolation, and see Figure 2.21 for an example.

**Upsampling, Downsampling and Filtering**  

Earlier, we noted the duality of downsampling and upsampling, made explicit in the transposition relation (2.130). What happens when filtering is involved, when is \(H^T\) equal to \(G\)? By inspection, this holds when \(h_n = g_{-n}\), since

\[ H^T = \begin{bmatrix} \vdots & \vdots & \vdots & \ddots \\ \vdots & h_1 & h_2 & h_3 & \ldots \\ \vdots & h_0 & h_1 & h_2 & \ldots \\ \vdots & h_{-1} & h_0 & h_1 & \ldots \\ \vdots & h_{-2} & h_{-1} & h_0 & \ldots \\ \vdots & \vdots & \vdots & \vdots & \ddots \end{bmatrix} = \begin{bmatrix} \vdots & \vdots & \vdots & \ddots \\ \vdots & g_1 & g_2 & g_3 & \ldots \\ \vdots & g_0 & g_1 & g_2 & \ldots \\ \vdots & g_{-1} & g_0 & g_1 & \ldots \\ \vdots & g_{-2} & g_{-1} & g_0 & \ldots \\ \vdots & \vdots & \vdots & \vdots & \ddots \end{bmatrix} = G. \quad (2.148) \]

Then, it follows that \(h_n = g_{-n}\) followed by downsampling by 2 is the transpose of upsampling by 2 followed by interpolation with \(g_n\):

\[ (D_2H)^T = H^T D_2^T = GU_2, \quad (2.149) \]

where we used (2.148). We could also show the above by trying to find the adjoint operator to \(D_2H\):

\[ \langle D_2H x, y \rangle \overset{(a)}{=} \langle Hx, D_2^* y \rangle \overset{(b)}{=} \langle Hx, U_2 y \rangle \overset{(c)}{=} \langle x, H^* U_2 y \rangle \overset{(d)}{=} \langle x, GU_2 y \rangle, \]

where (a) and (c) follow from conjugate linearity of the inner product in the second argument; (b) follows from (2.130); and (d) follows from (2.148). The above shows that \(D_2H\) and \(GU_2\) are adjoints of each other.
2.5. Multirate Sequences and Systems

The above “transposition equals time-reversal” will appear prominently in our analysis of orthogonal filter banks in Chapter 6. In particular, we will prove that when the impulse response of the filter \( g_n \) is orthogonal to its even shifts as in (2.150), and \( h_n = g_{-n} \), then the operation of filtering, downsampling by 2, upsampling by 2 and interpolation as in (6.21) is an orthogonal projection onto the subspace spanned by \( g_n \) and its even shifts.

2.5.4 Multirate Identities

**Orthogonality of Filter’s Impulse Response to its Even Shifts**

Filters that have impulse responses orthogonal to their even shifts

\[
\langle g_n, g_{n-2k} \rangle = \delta_k, \tag{2.150}
\]

will play an important role in the analysis of filter banks. Geometrically, (2.150) means that the columns of \( GU_2 \) in (2.142) are orthonormal to each other (similarly for the rows of \( D_2H \)), that is,

\[
(GU_2)^T(GU_2) = U_2^T G^T GU_2 = D_2G^T GU_2 = I. \tag{2.151}
\]

Moreover, we can see (2.150) as the autocorrelation of \( g \) sampled by 2. Write the autocorrelation as

\[
a_k = \langle g_n, g_{n-k} \rangle_n,
\]

and note that it has a single nonzero even term, \( g_0 = 1 \),

\[
a_{2k} = \delta_k. \tag{2.152}
\]

In the ZT domain, \( A(z) = G(z)G(z^{-1}) \) using (2.91). Keeping only the even terms can be accomplished by adding \( A(z) \) and \( A(-z) \) and dividing by 2. Therefore, (2.152) can be expressed as

\[
A(z) + A(-z) = G(z)G(z^{-1}) + G(-z)G(-z^{-1}) = 2, \tag{2.153}
\]

which on the unit circle leads to

\[
\left| G(e^{j\omega}) \right|^2 + \left| G(e^{j(\omega+\pi)}) \right|^2 = 2. \tag{2.154}
\]

This *quadrature mirror formula*, also called power complementarity, will be central in the design of orthonormal filter banks in Chapter 6. In the above we have assumed that \( g_n \) is real, and used both

\[
G(z)G(z^{-1}) \big|_{z=e^{j\omega}} = G(e^{j\omega})G(e^{-j\omega}) = G(e^{j\omega})G^*(e^{j\omega}) = \left| G(e^{j\omega}) \right|^2,
\]

as well as

\[
G(-z)G(-z^{-1}) = \left| G(e^{j(\omega+\pi)}) \right|^2.
\]

In summary, a filter satisfying any of the conditions below is called **orthogonal**:

\[
\langle g_n, g_{n-2k} \rangle = \delta_k \quad \text{Matrix View} \quad D_2G^T GU_2 = I \quad \left| G(e^{j\omega}) \right|^2 + \left| G(e^{j(\omega+\pi)}) \right|^2 = 2 \tag{2.155}
\]
Chapter 2. Discrete-Time Sequences and Systems

Noble Identities These identities relate to the interchange of multirate operations and filtering. The first states that downsampling by 2 followed by filtering with $H(z)$ is equivalent to filtering with $H(z^2)$ followed by downsampling by 2, as shown in Figure 2.22(a).

The second states that filtering with $G(z)$ followed by upsampling by 2 is equivalent to upsampling by 2 followed by filtering with $G(z^2)$, shown in Figure 2.22(b). The proof of the Noble identities is left as Exercise 2.10, and both results generalize to sampling rate changes by $N$.

Commutativity of Upsampling and Downsampling Up/downsampling by the same integer does not commute, since, as we have seen, $U_2$ followed by $D_2$ is the identity, while $D_2$ followed by $U_2$ is a projection onto the subspace of even-indexed samples. Interestingly, upsampling by $N$ and downsampling by $M$ commute when $N$ and $M$ are relatively prime (that is, they have no common factor). This is shown in Figure 2.22(c), and the proof is the topic of Exercise 2.11. A couple of illustrative examples are given as the solved Exercise 2.3.

Example 2.25 (Commutativity of upsampling and downsampling). We look at upsampling by 2 and downsampling by 3. If we apply $U_2$ to $x_n$

$$U_2x = \begin{bmatrix} \ldots & x_0 & 0 & x_1 & 0 & x_2 & 0 & x_3 & 0 & \ldots \end{bmatrix}^T,$$

followed by $D_3$, we get

$$D_3U_2x = \begin{bmatrix} \ldots & x_{-3} & 0 & x_0 & 0 & x_3 & 0 & x_6 & \ldots \end{bmatrix}^T.$$

Applying $D_3$ first

$$D_3x = \begin{bmatrix} \ldots & x_{-3} & x_0 & x_3 & x_6 & \ldots \end{bmatrix}^T,$$

followed by $U_2$ leads to the same result, $U_2D_3x = D_3U_2x$. ■

2.5.5 Polyphase Representation

Polyphase Representation of Sequences A major twist multirate processing brings to signal processing is that shift invariance is replaced by periodic shift variance. A
2.5. Multirate Sequences and Systems

A convenient way to express this periodic shift variance is to split the input $x_n$ into its even- and odd-indexed parts. Call $x_0$ and $x_1$ the following sequences

$$x_0 = \left[ \ldots, x_{-2}, x_0, x_2, x_4, \ldots \right]^T \quad \xrightarrow{\text{ZT}} \quad X_0(z) = \sum_{n \in \mathbb{Z}} x_{2n}z^{-n},$$

$$x_1 = \left[ \ldots, x_{-1}, x_1, x_3, x_5, \ldots \right]^T \quad \xrightarrow{\text{ZT}} \quad X_1(z) = \sum_{n \in \mathbb{Z}} x_{2n+1}z^{-n}.$$

Then, we can express the $z$-transform of $x_n$ as

$$X(z) = X_0(z^2) + z^{-1}X_1(z^2), \quad (2.156)$$

which clearly shows the even and odd parts of $X(z)$. This very simple transform that separates $X(z)$ into $X_0(z)$ and $X_1(z)$ is called the polyphase transform, is shown in Figure 2.23(a). The even and odd parts $X_0(z)$ and $X_1(z)$ are called the polyphase components of $X(z)$. The inverse transform simply upsamples the two polyphase components and puts them back together by interleaving, as in (2.156), shown in Figure 2.23(b).

For example, in polyphase transform domain, the effect of the operator $Y = U_2D_2$ is very simple: $X_0(z)$ is kept while $X_1(z)$ is zeroed out:

$$Y(z) = X_0(z^2).$$

**Polyphase Representation of Filters** We now look into how filters can be expressed in the polyphase transform domain.

---

**Figure 2.23:** Forward and inverse polyphase transform for (a)-(b) rate changes by 2, and (c)-(d) rate changes by $N$ (TBD: Figure to be redone.)
Figure 2.24: Polyphase representation of filters. (a) Filtering and downsampling. (b) Upsampling and interpolation.

We start with filtering followed by downsampling by 2, as we discussed previously. We need to find the polyphase components of $H(z)X(z)$, given the polyphase components of $X(z)$:

$$H(z)X(z) = H_0(z^2)X_0(z^2)+H_1(z^2)X_1(z^2)+z^{-1}(H_0(z^2)X_1(z^2)+z^2H_1(z^2)X_0(z^2)).$$

Therefore, $H(z)$ can be decomposed as

$$H(z) = H_0(z^2) + zH_1(z^2) \quad (2.157)$$

(note the $z$ instead of the $z^{-1}$). In particular, filtering followed by downsampling by 2 can now be compactly written as (see Figure 2.24(a)):

$$Y(Z) = \left[ H_0(z) \quad H_1(z) \right] \left[ \begin{array}{c} X_0(z) \\ X_1(z) \end{array} \right]. \quad (2.158)$$

Upsampling by 2 followed by interpolation with $G(z)$ leads to an output $Y(z)$, which when expressed in its polyphase components,

$$Y(z) = Y_0(z^2) + z^{-1}Y_1(z^2),$$

uses the polyphase decomposition of $G(z)$

$$G(z) = G_0(z^2) + z^{-1}G_1(z^2), \quad (2.159)$$

since, in terms of the input $X(z)$, $Y(z)$ equals

$$Y(z) = (G_0(z^2) + z^{-1}G_1(z^2))X(z^2),$$

or, in terms of polyphase components of $Y(z)$,

$$\begin{bmatrix} Y_0(z) \\ Y_1(z) \end{bmatrix} = \begin{bmatrix} G_0(z) \\ G_1(z) \end{bmatrix} X(z), \quad (2.160)$$

shown in Figure 2.24(b). Note that the duality between (filtering, downsampling) and (upsampling, interpolation) we have seen earlier, shows through polyphase decomposition as well, (2.157) and (2.159). This duality, including the change from $z^{-1}$ to $z$, is related to the transposition and time reversal seen in (2.148).

Polyphase Representation with Rate Changes by $N$ The polyphase transform of size $N$ decomposes a sequence into $N$ phases

$$x_k = \left[ \cdots \quad x_{N(n-1)+k} \quad x_{Nn+k} \quad x_{N(n+1)+k} \quad \cdots \right]^T, \quad (2.161)$$
from which we can write $X(z)$ as

$$X(z) = \sum_{k=0}^{N-1} z^{-k} X_k(z^N). \quad (2.162)$$

A filter preceding downsampling by $N$ is decomposed in the reverse order as

$$H(z) = \sum_{k=0}^{N-1} z^k H_k(z^N), \quad (2.163)$$

so that the output of filtering and downsampling by $N$ can be written as

$$Y(z) = \sum_{k=0}^{N-1} H_k(z) X_k(z). \quad (2.164)$$

Conversely, upsampling by $N$ followed by interpolation by $G(z)$ uses a decomposition

$$G(z) = \sum_{k=0}^{N-1} z^{-k} G_k(z^N).$$

To illustrate these, in Figure 2.25 we show a complete system for $N = 3$. Solved Exercise 2.4 illustrates our discussion on polyphase transform.

### 2.6 Stochastic Sequences and Systems

#### 2.6.1 Stationary and Wide-Sense Stationary Processes

#### 2.6.2 Linear Shift-Invariant Processing

#### 2.6.3 Multirate Processing

#### 2.6.4 Wiener Filtering

### 2.7 Algorithms

#### 2.7.1 Fast Fourier Transforms

We have thus far studied the DFT as the natural analysis tool for periodic signals. The use of DFTs extends beyond the scope of periodic signals in part because there are fast algorithms for computing them. These algorithms are called *fast Fourier transform (FFT)* algorithms.
The DFT, as defined in (2.107a), appears as the sum of $N$ complex terms. Each term is the product of a complex constant (a power of $W_N$) and a component of the input signal $x$. Thus, each DFT coefficient can be computed with $N$ multiplications and $(N - 1)$ additions. The full DFT can be computed with $N^2$ multiplications and $N(N - 1)$ additions. Writing the DFT as a matrix multiplication by $F_N$ does not change the operation count—in fact, $N^2$ multiplications and $N(N - 1)$ additions is the cost of a direct multiplication of an $N \times N$ matrix by an $N \times 1$ vector.

Special structure of the DFT allows its computation with far fewer operations. For simplicity, we will consider only the case of $N = 2^\gamma$ with $\gamma \in \mathbb{Z}^+$, though fast algorithms exist for all values of $N$. Many classes of FFT algorithms are based on decomposing a DFT computation into smaller DFTs and a few simple operations to combine results. We develop one such algorithm here.

Starting with the definition of the DFT (2.107a), write

$$X_k = \sum_{n=0}^{N-1} x_n W_N^{kn}$$

(a) $\sum_{n=0}^{N-1} x_{2n} W_N^{k(2n)} + \sum_{n=0}^{N-1} x_{2n+1} W_N^{k(2n+1)}$

(b) $\sum_{n=0}^{N-1} x_{2n} W_{N/2}^{kn} + W_N^k \sum_{n=0}^{N-1} x_{2n+1} W_{N/2}^{kn}$

where (a) uses separates the summation over odd- and even-numbered terms; and (b) follows from $W_{2N}^k = W_N^k W_{N/2}^k$. Recognize the first sum as the length-$(N/2)$ DFT of the sequence $[x_0 \ x_2 \ \ldots \ x_{N-2}]^T$, and the second sum as the length-$(N/2)$ DFT of the sequence $[x_1 \ x_3 \ \ldots \ x_{N-1}]^T$. It is now apparent that the length-$N$ DFT computation can make use of $F_{N/2} D_2 x$ and $F_{N/2} C_2 x$, where $D_2$ is the downsampling-by-2 operator defined in (2.123b), and $C_2$ is a similar operator, except that it keeps the odd-indexed values. Since the length-$(N/2)$ DFT expression is $(N/2)$-periodic in $k$, the length-$(N/2)$ DFT is useful even for $k \in \{N/2, N/2 + 1, \ldots, N - 1\}$.

To help get a compact matrix representation, we define a diagonal matrix to encapsulate the $W_N^k$ factor:

$$A_{N/2} = \text{diag}([1 \ W_N \ W_N^2 \ \ldots \ W_N^{(N/2)-1}])$$

We can now express the length-$N$ DFT vector $X$ as

$$X = \begin{bmatrix} I_{N/2} & A_{N/2} \\ I_{N/2} & -A_{N/2} \end{bmatrix} \begin{bmatrix} F_{N/2} & 0 \\ 0 & F_{N/2} \end{bmatrix} \begin{bmatrix} D_2 \\ C_2 \end{bmatrix} x, \quad (2.165)$$

$^{27}$We are counting complex multiplications and complex additions. It is customary to not count multiplications by $(-1)$ and thus lump together additions and subtractions.
where the final twist is to realize that $W_N^k = -W_N^{k-N/2}$. If this turns out to be a useful factorization, then we can repeat it to represent $F_{N/2}$ using $F_{N/4}$, etc. So let us count computations in the factored form.

Let $\mu_N$ and $\nu_N$ denote the number of multiplications and additions in computing a length-$N$ DFT. The factorization (2.165) shows that a length-$N$ DFT can be accomplished with two length-$(N/2)$ DFTs, $N/2$ multiplications and $N$ additions, yielding the following recursions:

$$\mu_N = 2\mu_{N/2} + \frac{N}{2} = \frac{N}{2}\mu_2 + (\log N - 1)\frac{N}{2} = (\log N - 1)\frac{N}{2},$$

$$\nu_N = 2\nu_{N/2} + N = \frac{N}{2}\nu_2 + (\log N - 1)N = (\log N - 1)N,$$

solved with the initial conditions $\mu_2 = 0$ and $\nu_2 = 0$. Thus, recursive application of (2.165) has allowed us to reduce the number of multiplications from $N^2$ to $(\log N - 1)N/2$.

Other well-known FFT algorithms are split-radix which reduces the number of multiplications; Good-Thomas for $N = N_1N_2$ where $N_1$ and $N_2$ are coprime; Winograd, typically used for small factors; Rader’s for prime size $N$; and many others. Further Reading gives pointers to texts discussing this wealth of FFTs.

### 2.7.2 Convolution

(TBD)
Appendix

2.A Elements of Complex Analysis

A complex number \( z \in \mathbb{C} \), is a number of the form:

\[
z = a + jb, \quad a, b \in \mathbb{R},
\]

and \( j \) is by definition the positive root of \(-1\). (Mathematicians and physicists typically use \( i \) for \( \sqrt{-1} \), while \( j \) is more common in engineering.)

In (2.166), \( a \) is called the real part while \( b \) is called the imaginary part. The complex conjugate of \( z \) is denoted by \( z^* \) and is by definition:

\[
z^* = a - jb.
\]

Any complex number can be represented in its polar form as well:

\[
z = re^{j\theta},
\]

where \( r \) is called the modulus or magnitude and \( \theta \) is the argument. Using the Euler’s formula:

\[
e^{j\theta} = \cos \theta + j \sin \theta,
\]

we can express a complex number further as

\[
z = re^{j\theta} = r(\cos \theta + j \sin \theta).
\]

Euler’s formula is particularly useful, since then we can easily find a power of a complex number as

\[
(cos \theta + j \sin \theta)^n = (e^{j\theta})^n = e^{j n \theta} = \cos n \theta + j \sin n \theta.
\]

Two other useful relations using Euler’s formula are:

\[
\cos \theta = \frac{e^{j\theta} + e^{-j\theta}}{2}, \quad \sin \theta = \frac{e^{j\theta} - e^{-j\theta}}{2j}.
\]

We can easily go between standard and polar forms as follows:

\[
r = \sqrt{a^2 + b^2} \quad \theta = \arctan \frac{b}{a},
\]

\[
a = r \cos \theta \quad b = r \sin \theta.
\]

Just as a reminder, the basic operations on complex numbers are as follows:

\[
z_1 + z_2 = (a_1 + a_2) + j(b_1 + b_2),
\]

\[
z_1 - z_2 = (a_1 - a_2) + j(b_1 - b_2),
\]

\[
z_1 \cdot z_2 = (a_1 a_2 - b_1 b_2) + j(b_1 a_2 + a_1 b_2),
\]

\[
z_1 \overline{z_2} = \frac{a_1 a_2 + b_1 b_2}{a_2^2 + b_2^2} + j \frac{a_2 b_1 - a_1 b_2}{a_2^2 + b_2^2}.
\]

Complex numbers are typically shown in the complex plane (see Figure 2.26(a)).
2.A. Elements of Complex Analysis

(a) $z = r e^{i\theta}$

(b) $W_3^0$, $W_3^1$, $W_3^2$

Figure 2.26: (a) A pictorial representation of a complex number in a complex plane. (b) Third roots of unity. (TBD: Figure either to be redone or removed.)

Roots of Unity The same way $j$ was defined as the second root of unity, we can define the principal $N$th root of unity as

$$W_N = (1)^{\frac{1}{N}} = e^{-j2\pi/N}.$$  \hspace{1cm} (2.174)

You can easily check that $W_N^k$, for $k \in \{1, 2, \ldots, N - 1\}$, are also roots of unity. If we drew all $N$ roots of unity in the complex plane, we would see that they slice up the unit circle by equal angles. An example with $N = 3$ is given in Figure 2.26(b), while for $N = 4$ they are $1, -j, -1, j$. Here are some useful identities involving the roots of unity:

$$W_N^N = 1,$$

$$W_N^{kN+n} = W_N^n, \quad \text{with} \quad k, n \in \mathbb{Z},$$

$$\sum_{k=0}^{N-1} W_N^{nk} = \begin{cases} N, & n = \ell N, \quad \ell \in \mathbb{Z}; \\ 0, & \text{otherwise}. \end{cases}$$ \hspace{1cm} (2.175c)

The last relation is often referred to as orthogonality of the roots of unity. Its proof is given next.

Proof. We use the finite sum from (P1.20-6):

$$\sum_{k=0}^{N-1} (W_N^N)^k = \frac{1 - W_N^{nN}}{1 - W_N^n}.$$ \hspace{1cm} (2.176a)

If $n = \ell N + p$, this reduces to

$$\sum_{k=0}^{N-1} W_N^{kn} = 0.$$ \hspace{1cm} (2.176b)
For $n = \ell N$, substitute $n$ directly into (2.175c).

### 2.B Elements of Algebra: Discrete Polynomials

A *polynomial sequence* is a sequence whose $n$th element is a finite sum of the following form:

$$p_n = \sum_{k=0}^{N} a_k n^k.$$  \hspace{1cm} (2.177)

For example, a *constant* polynomial sequence is of the form $p_n = a$, a linear polynomial sequence is of the form $p_n = a_0 + a_1 n$, and a *quadratic* polynomial sequence is of the form $p_n = a_0 + a_1 n + a_2 n^2$. The $z$-transform of such a sequence is:

$$P(z) = \sum_{n \in \mathbb{Z}} p_n z^{-n} = \sum_{n \in \mathbb{Z}} \left( \sum_{k=0}^{N} a_k n^k \right) z^{-n}.$$  

When we study wavelets and filter banks, we will be concerned with the moment annihilating/preserving properties of such systems. In that, the following fact will be of use: Convolution of the polynomial sequence with a differencing filter $d_n = (\delta_n - \delta_{n-1})$, or, multiplication of $P(z)$ by $D(z) = (1 - z^{-1})$, reduces the degree of the polynomial by 1, as in

$$D(z)P(z) = (1 - z^{-1}) \sum_{n} p_n z^{-n} = (1 - z^{-1}) \sum_{n} \left( \sum_{k=0}^{N} a_k n^k \right) z^{-n}$$

$$= \sum_{n} \sum_{k=0}^{N} a_k n^k z^{-n} - \sum_{n} \sum_{k=0}^{N} a_k n^k z^{-(n+1)}$$

$$= \sum_{n} \sum_{k=0}^{N} a_k n^k z^{-n} - \sum_{n} \sum_{k=0}^{N} a_k (n-1)^k z^{-n}$$

$$= \sum_{n} \sum_{k=0}^{N} a_k (n^k - (n-1)^k) z^{-n}$$

$$= \sum_{n} \left( \sum_{k=0}^{N-1} b_k n^k \right) z^{-n} = \sum_{n} r_n z^{-n},$$

where $r_n$ is a polynomial of degree $(N - 1)$, and (a) follows from $(n^N - (n-1)^N)$ being a polynomial of degree $(N - 1)$. The above process can be seen as applying a differencing filter with a zero at $z = 1$. Extending the above argument, we see that by repeatedly applying the differencing filter, we will kill the constant term, then the linear, then the quadratic, and so on.
We now summarize the main concepts and results seen in this chapter in a tabular form.

### Properties of the Dirac Impulse Sequence

$$\sum_{n \in \mathbb{Z}} \delta_n = 1$$
$$\sum_{n \in \mathbb{Z}} x_{n_0 - n} \delta_n = \sum_{n \in \mathbb{Z}} \delta_{n_0 - n} x_n = x_{n_0}$$
$$x_n \star \delta_{n_0 - n} = x_{n+n_0}$$
$$x_n \delta_{n_0} = x_0 \delta_n$$
$$\delta_n \xrightarrow{\text{DTFT}} \frac{1}{2\pi} \delta(\omega)$$

### Concepts in Discrete-Time Processing

<table>
<thead>
<tr>
<th>Concept</th>
<th>Notation</th>
<th>Infinite Time</th>
<th>Finite Time</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>shift</td>
<td>$x_n$</td>
<td>$n \in \mathbb{Z}$</td>
<td>$n \in {0, 1, \ldots, N - 1}$</td>
<td>signal vector</td>
</tr>
<tr>
<td>sequence</td>
<td>$h_n$</td>
<td>$n \in \mathbb{Z}$</td>
<td>$n \in {0, 1, \ldots, N - 1}$</td>
<td>filter operator impulse response</td>
</tr>
<tr>
<td>LSI system</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>convolution</td>
<td>$(h \star x)$</td>
<td>$\sum_{k \in \mathbb{Z}} x_k h_{n-k}$</td>
<td>$\sum_{k=0}^{N-1} x_k h_{n-k}$</td>
<td></td>
</tr>
<tr>
<td>eigensequence</td>
<td>$v$</td>
<td>$e^{j\omega n}$</td>
<td>$e^{\frac{2\pi}{N} \alpha}$</td>
<td></td>
</tr>
<tr>
<td>satisfies</td>
<td>$(h \star v_\lambda) = \lambda v_\lambda$</td>
<td>$(h \star v_\omega) = H(e^{j\omega}) v_\omega$</td>
<td>$(h \star v_k) = H_k v_k$</td>
<td></td>
</tr>
<tr>
<td>eigenspace</td>
<td>$\mathbb{H}_\lambda$</td>
<td>$\mathbb{H}_\omega = {\alpha e^{j\omega n}}$</td>
<td>$\mathbb{H}_k = {\alpha e^{j\omega n}}$</td>
<td></td>
</tr>
<tr>
<td>frequency response</td>
<td>$\lambda$</td>
<td>$\lambda_\omega = H(e^{j\omega})$</td>
<td>$\lambda_k = H_k$</td>
<td>eigenvalue corresponding to $v$</td>
</tr>
<tr>
<td>Fourier transform</td>
<td>$\langle x, v \rangle$</td>
<td>DTFT $X(e^{j\omega}) = \sum_{n=-\infty}^{\infty} x_n e^{-j\omega n}$</td>
<td>DFT $X_k = \sum_{n=0}^{N-1} x_n e^{-j\frac{2\pi k}{N} n}$</td>
<td>spectrum</td>
</tr>
</tbody>
</table>
## Chapter 2. Discrete-Time Sequences and Systems

### Multirate, factor 2

<table>
<thead>
<tr>
<th>Expression</th>
<th>Reference</th>
</tr>
</thead>
<tbody>
<tr>
<td>$x_n, X(z), X(e^{j\omega})$</td>
<td>(2.121)</td>
</tr>
</tbody>
</table>

#### Downsampling by 2

<table>
<thead>
<tr>
<th>Expression</th>
<th>Reference</th>
</tr>
</thead>
<tbody>
<tr>
<td>$y_n = x_{2n}$</td>
<td></td>
</tr>
<tr>
<td>$Y(z) = \frac{1}{2} \left[ X(z^{1/2}) + X(-z^{1/2}) \right]$</td>
<td>(2.124)</td>
</tr>
<tr>
<td>$Y(e^{j\omega}) = \frac{1}{2} \left[ X(e^{j\omega/2}) + X(e^{j(-2\pi/2)}) \right]$</td>
<td>(2.125)</td>
</tr>
</tbody>
</table>

#### Upsampling by 2

<table>
<thead>
<tr>
<th>Expression</th>
<th>Reference</th>
</tr>
</thead>
</table>
| $y_n = \begin{cases} 
  x_{n/2}, & \text{n even;} \\
  0, & \text{otherwise.} 
\end{cases}$ | (2.122) |
| $Y(z) = X(z^2)$ | (2.131) |
| $Y(e^{j\omega}) = X(e^{j2\omega})$ | (2.132) |

#### Filtering by $h_n$

<table>
<thead>
<tr>
<th>Expression</th>
<th>Reference</th>
</tr>
</thead>
<tbody>
<tr>
<td>$Y(z) = \frac{1}{2} \left[ H(z^{1/2})X(z^{1/2}) + H(-z^{1/2})X(-z^{1/2}) \right]$</td>
<td>(2.140)</td>
</tr>
<tr>
<td>$Y(e^{j\omega}) = \frac{1}{2} \left[ H(e^{j\omega/2})X(e^{j\omega/2}) + H(e^{j(-2\pi/2)})X(e^{j(-2\pi/2)}) \right]$</td>
<td>(2.141)</td>
</tr>
</tbody>
</table>

#### Downsampling by $N$

<table>
<thead>
<tr>
<th>Expression</th>
<th>Reference</th>
</tr>
</thead>
<tbody>
<tr>
<td>$y_n = x_{nN}$</td>
<td>(2.126)</td>
</tr>
<tr>
<td>$Y(z) = \frac{1}{N} \sum_{k=0}^{N-1} X(\omega_k z^{1/N})$</td>
<td>(2.127)</td>
</tr>
<tr>
<td>$Y(e^{j\omega}) = \frac{1}{N} \sum_{k=0}^{N-1} X(e^{j\omega - 2\pi k/N})$</td>
<td>(2.128)</td>
</tr>
</tbody>
</table>

#### Upsampling by $N$

<table>
<thead>
<tr>
<th>Expression</th>
<th>Reference</th>
</tr>
</thead>
</table>
| $y_n = \begin{cases} 
  x_{n/N}, & \text{n = lN;} \\
  0, & \text{otherwise.} 
\end{cases}$ | (2.134) |
| $Y(z) = X(z^N)$ | (2.135) |
| $Y(e^{j\omega}) = X(e^{jN\omega})$ | (2.136) |
Cooley-Tukey FFT The impact signal processing has had in practical terms is perhaps due in large part to the advent of the fast Fourier transform algorithms, spurred by the paper of Cooley and Tukey in 1965 [43]. It breaks the computation of the DFT of size $N = N_1 N_2$ into a recursive computation of smaller DFTs of sizes $N_1$ and $N_2$, respectively (see Section 2.7.1). Unbeknownst to them, a similar algorithm was published by Gauss some 150 years earlier, in his attempt to track asteroid trajectories.

MP3, JPEG and MPEG These already household names, are all algorithms for coding and compressing various types of data (audio, images and video). The basic ideas in all three all stem from transform-based work in signal processing, called subband coding, closely-related to wavelets as we will see in later chapters.

Further Reading

Books and Textbooks A standard book on discrete-time processing is the one Oppenheim, Schafer and Buck [120]. For statistical signal processing, see the text by Porat [124]. An early account of multirate signal processing was given by Crochiere and Rabiner [44], a more recent one is by Vaidyanathan [154]. Dudgeon and Mersereau cover multidimensional signal processing in [63]. Blahut in [18] discusses fast algorithms for DSP, and in particular, various classes of FFTs.

Inverse $z$-transform Via Contour Integration The formal inversion process for the $z$-transform is given by contour integration using Cauchy’s integral formula when $X(z)$ is a rational function of $z$. When $X(z)$ is not rational, inversion can be quite difficult. A short account of inversion using contour integration is given in [99]; more details can be found in [120].

Filter Design Numerous filter design techniques exist. They all try to approximate the desired specifications of the system/filter by a realizable discrete-time system/filter. For IIR filters, one of the standard methods is to design the discrete-time filters from continuous-time ones using bilinear transformation. For FIR filters, windowing is often used to approximate the desired response by truncating it with a window, a topic we touched upon in Example 2.4. Then, linear phase is often incorporated as a design requirement. Kaiser window design method is a standard method for FIR filter design using windowing. Another commonly used design method is called Parks-McClellan. An excellent overview of filter design techniques is given in [120].

Algebraic Theory of Signal Processing Algebraic theory of signal processing is a recent development whose foundations can be found in [127]. It provides the framework for signal processing in algebraic terms, and allows for the development of other models but those based on time. For example, by introducing extensions different from the circular (periodic) one we discussed in Section 2.4.3, in particular the symmetric one, one can show that the appropriate transform is the well-known DCT. Moreover, it provides a recipe
for building a signal model bases on sequence and filter spaces, appropriate convolutions and appropriate Fourier transforms. As such, the existence and form of fast algorithms for computing such transforms (among which are the well-known trigonometric ones) are automatically given. Some of the observations in this chapter were inspired by the algebraic framework.

### Exercises with Solutions

#### 2.1. Filtering as a Projection

Given is a filter with impulse response $g_n$, $n \in \mathbb{Z}$. For the filters below, check whether they are orthogonal projections or not.

(Hint: Use the DTFT $G(e^{j\omega})$ of the filter’s impulse response.)

(i) $g_n = \delta_{n-k}$, $k \in \mathbb{Z}$;

(ii) $g_n = \frac{1}{2} \delta_{n+1} + \delta_n + \frac{1}{2} \delta_{n-1}$;

(iii) $g_n = \frac{1}{\sqrt{2}} \sin(n\pi/2)$.

Consider now the class of real-valued discrete-time filters that perform an orthogonal projection. Give a precise characterization of this class (be more specific than just repeating the conditions for an operator being an orthogonal projection.)

**Solution:**
We can represent our system as $y = Gx$, with the corresponding matrix notation as in (2.37). To check that a filter (operator) is an orthogonal projection, we must check that it is idempotent and self-adjoint as in Definitions 1.20 and 1.21.

Checking idempotency is easier in the Fourier domain since $G^2 = G$ has as its DTFT pair $G(e^{j\omega}) = G^2(e^{j\omega})$. Checking self-adjointness is equivalent to checking that the operator matrix $G$ is Hermitian.

(i) We have that $G(e^{j\omega}) = e^{-j\omega}$ and $G^2(e^{j\omega}) = e^{-j2\omega}$. Thus, $G^2(e^{j\omega}) \neq G(e^{j\omega})$, unless $k = 0$, and the filter operator is not a projection.

(ii) Similarly, we have

$$G(e^{j\omega}) = \frac{1}{2} e^{j\omega} + 1 + \frac{1}{2} e^{-j\omega},$$

and

$$G^2(e^{j\omega}) = \left( \frac{1}{2} e^{j\omega} + 1 + \frac{1}{2} e^{-j\omega} \right)^2 = \frac{1}{4} e^{2j\omega} + e^{j\omega} + \frac{3}{2} + e^{-j\omega} + \frac{1}{4} e^{-2j\omega}.$$  

Thus, $G^2(e^{j\omega}) \neq G(e^{j\omega})$, and this filter is not a projection operator either.

(iii) We can rewrite $g_n$ as

$$g_n = \frac{1}{\sqrt{2}} \sin(n\pi/2) = \frac{1}{2\pi} \int_{-\pi/2}^{\pi/2} \sqrt{2} e^{j\omega n} d\omega,$$

yielding $G(e^{j\omega})$ as

$$G(e^{j\omega}) = \begin{cases} \sqrt{2}, & \text{for } \omega \in [-\pi/2, \pi/2]; \\ 0, & \text{for } \omega \in [-\pi, -\pi/2] \cup [\pi/2, \pi], \end{cases}$$

as well as

$$G^2(e^{j\omega}) = \begin{cases} 2, & \text{for } \omega \in [-\pi/2, \pi/2]; \\ 0, & \text{for } \omega \in [-\pi, -\pi/2] \cup [\pi/2, \pi]. \end{cases}$$

Again, $G^2(e^{j\omega}) \neq G(e^{j\omega})$, and this filter is not a projection operator either. Note that for $g_n = \frac{\sin(n\pi/2)}{n\pi/2}$ we would have had a projection.
Exercises with Solutions

In the most general case, again trying to satisfy idempotency, we see that for $G(e^{j\omega}) = G(e^{j\omega})^2$, $G(e^{j\omega})$ can only be 1 or 0:

$$G(e^{j\omega}) = \begin{cases} 1, & \text{for } \omega \in R_1 \cup R_2 \cup \cdots \cup R_n; \\ 0, & \text{otherwise}, \end{cases}$$

where $R_1, R_2, \ldots, R_n$ are disjoint regions in $[-\pi, \pi]$.

Self-adjointness is satisfied if $g_n = g^*_n$. In Fourier domain:

$$G(e^{j\omega}) = \sum_n g_n e^{-j\omega n} = \sum_n g^*_n e^{-j\omega n} = \sum_n g^*_n (e^{-j\omega n})^* = G^*(e^{j\omega}).$$

Since, from the first condition, $G(e^{j\omega})$ is real, we conclude that self-adjointness is automatically satisfied.

2.2. Circulant Matrices

Given is an $N \times N$ circulant matrix $C$ as in (1.94).

(i) Give a simple test for the singularity of $C$.

(ii) Give a formula for $\det(C)$.

(iii) Prove that $C^{-1}$ is circulant.

(iv) Given two circulant matrices $C_1$ and $C_2$, show that they commute, $C_1 C_2 = C_2 C_1$ and that the result is circulant as well.

Solution:

Most of the developments in this problem follow from Proposition 2.5.

(i) Consider circularly convolving two length-$N$ sequences $c_n$ and $g_n$:

$$c \ast g \overset{(a)}{=} Cg \overset{(b)}{=} \frac{1}{N} F^*(CG) \overset{(c)}{=} \frac{1}{N} F^*(CFg) \overset{(d)}{=} \frac{1}{N} F^*(AFg) \overset{(e)}{=} \frac{1}{N} F^*AFg,$$

where $F$ is the DFT matrix given in (2.108), $C_1 = [C_0 \ C_1 \ \cdots \ C_{N-1}]^T$ is the DFT of $c$, and $A = \text{diag} C_1$. In the above, (a) is obtained by applying the circular convolution operator of $c$ to $g$; (b) follows from taking the IDFT of the DFT pair of $CG$ which is $CG$ (convolution in time corresponds to multiplication in frequency, also $F^{-1} = (1/N)F^*$); in (c) we write $G$ as the DFT of $g$; in (d) we write the pointwise multiplication in frequency as a diagonal matrix $A$ multiplying $FG$; and finally in (e) we recognize that $C = F^*AF$. In other words, we have rederived (2.112), stating that any circulant matrix $C$ can be diagonalized using the DFT.

This formulation implies that the columns of $F^*$ are the eigenvectors of $C$, with $C_k$ being the associated eigenvalues. So the matrix $C$ is nonsingular if and only if none of the DFT coefficients $C_k$ are zero.

(ii) From the diagonalization of $C$, and from (1.91):

$$\det(C) = \prod_{k=0}^{N-1} C_k.$$

(iii) Since

$$C^{-1} = \frac{1}{N} F^*(AF)^{-1} = \frac{1}{N} F^{-1}A^{-1}(F^*)^{-1} = \frac{1}{N} \left( \frac{1}{N} F^*A^* \right) A^{-1} F = \frac{1}{N} F^*A^{-1} F;$$

it has the same form as $C$ in (2.112), and is thus circulant as well.

(iv) Writing $C_1$ and $C_2$ as $C_1 = F^*A_1F$ and $C_2 = F^*A_2F$, we see that

$$C = C_1 C_2 = (F^*A_1F)(F^*A_2F) = F^*A_1A_2F = F^*AF,$$

where $A = \text{diag}\{C_0, C_2, C_4, \ldots, C_{N-2}, C_{N-1}\}$. So $C = C_1 C_2$ is a circulant matrix with eigenvalues $C_1 A_2, k \in \{0, 1, \ldots, N-1\}$.

Since $A_1 A_2 = A_2 A_1$, we have $C_1 C_2 = C_2 C_1$. 

2.3. Combinations of Upsampling and Downsampling

Using matrix notation, compare:

(i) $U_3D_2x$ to $D_2U_3x$;

(ii) $U_4D_2x$ to $D_2U_4x$.

Explain the outcome of these comparisons.

Solution:

We will solve the problem using matrix notation for $U_3$, $U_4$ and $D_2$:

\[
U_3 = \begin{bmatrix}
\ldots & \ldots & \ldots & \ldots \\
\ldots & 1 & 0 & 0 \\
\ldots & 0 & 0 & 0 \\
\ldots & 0 & 0 & 0 \\
\ldots & 0 & 1 & 0 \\
\ldots & 0 & 0 & 0 \\
\ldots & 0 & 0 & 1 \\
\ldots & \ldots & \ldots & \ldots
\end{bmatrix}
\]

\[
U_4 = \begin{bmatrix}
\ldots & \ldots & \ldots & \ldots \\
\ldots & 1 & 0 & 0 \\
\ldots & 0 & 0 & 0 \\
\ldots & 0 & 0 & 0 \\
\ldots & 0 & 1 & 0 \\
\ldots & 0 & 0 & 0 \\
\ldots & 0 & 0 & 1 \\
\ldots & \ldots & \ldots & \ldots
\end{bmatrix}
\]

\[
D_2 = \begin{bmatrix}
\ldots & \ldots & \ldots & \ldots \\
\ldots & 1 & 0 & 0 \\
\ldots & 0 & 0 & 0 \\
\ldots & 0 & 0 & 0 \\
\ldots & 1 & 0 & 0 \\
\ldots & 0 & 0 & 0 \\
\ldots & 0 & 0 & 1 \\
\ldots & \ldots & \ldots & \ldots
\end{bmatrix}
\]

(i) For the first comparison, compute

\[
U_3D_2 = \begin{bmatrix}
\ldots & \ldots & \ldots & \ldots \\
\ldots & 1 & 0 & 0 \\
\ldots & 0 & 0 & 0 \\
\ldots & 0 & 0 & 0 \\
\ldots & 0 & 1 & 0 \\
\ldots & 0 & 0 & 0 \\
\ldots & 0 & 0 & 1 \\
\ldots & \ldots & \ldots & \ldots
\end{bmatrix}
\]

\[
D_2U_3 = \begin{bmatrix}
\ldots & \ldots & \ldots & \ldots \\
\ldots & 1 & 0 & 0 \\
\ldots & 0 & 0 & 0 \\
\ldots & 0 & 0 & 0 \\
\ldots & 0 & 1 & 0 \\
\ldots & 0 & 0 & 0 \\
\ldots & 0 & 0 & 1 \\
\ldots & \ldots & \ldots & \ldots
\end{bmatrix}
\]

Thus $U_3D_2x$ and $D_2U_3x$ are identical since $U_3D_2 = D_2U_3$.

(ii) For the second comparison, compute

\[
U_4D_2 = \begin{bmatrix}
\ldots & \ldots & \ldots & \ldots \\
\ldots & 0 & 0 & 0 \\
\ldots & 0 & 0 & 0 \\
\ldots & 0 & 0 & 0 \\
\ldots & 0 & 0 & 0 \\
\ldots & 0 & 0 & 0 \\
\ldots & 0 & 0 & 0 \\
\ldots & \ldots & \ldots & \ldots
\end{bmatrix}
\]

\[
D_2U_4 = \begin{bmatrix}
\ldots & \ldots & \ldots & \ldots \\
\ldots & 0 & 1 & 0 \\
\ldots & 0 & 0 & 0 \\
\ldots & 0 & 0 & 0 \\
\ldots & 0 & 1 & 0 \\
\ldots & 0 & 0 & 0 \\
\ldots & 0 & 0 & 1 \\
\ldots & \ldots & \ldots & \ldots
\end{bmatrix}
\]

which are evidently not the same, and thus $U_4D_2x \neq D_2U_4x$.

The results of the two comparisons are different because in the first case, 3 and 2 are coprime, while in the second 4 and 2 are not (see Exercise 2.11).
2.4. Periodically Shift-Varying Systems

Show that a linear, periodically shift-varying system of period \( N \), can be implemented with a polyphase transform followed by upsampling by \( N \), \( N \) filter operations and a summation.

Solution:

A linear, shift-varying system is characterized by a two-variable impulse response \( h_{kn} \), the response of the system to the input \( x_n = \delta_{n-k} \). Since the input can be written as \( x_n = \sum_k x_k \delta_{n-k} \), the output is given by \( y_n = \sum_k x_k h_{kn} \).

When \( h_{kn} \) is periodic in \( k \) with period \( N \), we define the polyphase components \( x_k \) as in (2.161), to yield

\[
x_{kn} = x_{nN+k}, \quad x_k = \begin{bmatrix} \cdots & x_{-2N+k} & x_{-N+k} & x_N & x_{N+k} & x_{2N+k} & \cdots \end{bmatrix}^T,
\]

for \( k \in \{0, 1, \ldots, N-1\} \). Denote the upsampled version of \( x_{kn} \) by \( x_{kn}^{(N)} \), so

\[
x_{kn}^{(N)} = \begin{cases} x_{kn+N/N}, & n/N \in \mathbb{Z}; \\ 0, & \text{otherwise}; \end{cases} = \begin{cases} x_{n+k}, & n/N \in \mathbb{Z}; \\ 0, & \text{otherwise}. \end{cases}
\]

\[
x_k^{(N)} = [\cdots 0 \underbrace{x_k 0 0 \cdots 0}_{N-1} x_{N+k} 0 \cdots]^T.
\]

If we take the above upsampled components, delay each by \( k \), and sum them up, we get \( x \) back:

\[
x_n = \sum_{k=0}^{N-1} x_k^{(N)}.
\]

As this is a linear system, we can find the output as

\[
y_n = \sum_k h_{kn} \sum_{i=0}^{N-1} x_{i,n-i}^{(N)} = \sum_{i=0}^{N-1} \sum_k h_{kn} x_{i,n-i}^{(N)} = \sum_{i=0}^{N-1} \sum_{k} h_{kn} x_{i,n-i}^{(N)}
\]

\[
= \sum_{t=0}^{N-1} \sum_{k} h_{tn} x_{t,n-t}^{(N)} = \sum_{t=0}^{N-1} \sum_{k} h_{tn} x_{t,n-t}^{(N)}
\]

where (a) follows from the excluded terms being zero; (b) from the periodicity of \( h_{kn} \); and (c) because the added terms are zero. The final expression shows the output as a sum of \( N \) terms. The \( t \)th term is the \( t \)th polyphase component, upsampled and filtered by an \( i \)-sample delayed version of \( h_{kn} \).

Exercises

2.1. Properties of Periodic Sequences

Given are complex exponential sequences of the form

\[
x_n = e^{j\omega_0 n}.
\]

(i) Show that if \( \alpha = \omega_0/2\pi \) is a rational number, \( \alpha = p/q \), with \( p \) and \( q \) coprime integers, then \( x_n \) is periodic with period \( q \).

(ii) Show that if \( \alpha = \omega_0/2\pi \) is irrational, then \( x_n \) is not periodic

(iii) Show that if \( x \) and \( y \) are two periodic sequences with periods \( N \) and \( M \) respectively, then \( x+y \) is periodic with period \( NM \) (in general).

2.2. Linear and Shift-Invariant Difference Equations

Consider the difference equation (2.32).
Chapter 2. Discrete-Time Sequences and Systems

(i) Show, possibly using a simple example, that if the initial conditions are nonzero, then the system is not linear and not shift-invariant.

(ii) Show that if the initial conditions are zero, then (a) the homogeneous solution is zero, (b) the system is linear and (c) the system is shift invariant.

2.3. LSI System Acting on Signals in \( \ell^p(\mathbb{Z}) \)

Prove that if \( x \in \ell^p(\mathbb{Z}) \) and \( h \in \ell^1(\mathbb{Z}) \), the result of \( (h \ast x) \) is in \( \ell^p(\mathbb{Z}) \) as well.

2.4. Modulation Property of the DTFT

Given are two sequences \( x \) and \( h \), both in \( \ell^1(\mathbb{Z}) \). Verify that \( xh \text{DTFT} \leftrightarrow \frac{1}{2\pi} X^*H \).

2.5. Autocorrelation and Crosscorrelation

Consider autocorrelation and crosscorrelation sequences and their DTFTs as defined in (2.55) and (2.59). Show that:

(i) \( a_n = a_{-n} \) and \( |a_n| \leq a_0 \);

(ii) \( c_n \) is in general not symmetric and \( C(e^{j\omega}) = X^*(e^{j\omega})H(e^{j\omega}) \);

(iii) The generalized Parseval’s formula (2.61b) holds.

2.6. Circulant Matrices and the DFT

Given is the circulant matrix from Proposition 2.5:

\[
G = \begin{bmatrix}
g_0 & g_{N-1} & g_{N-2} & \cdots & g_1 \\
g_1 & g_0 & g_{N-1} & \cdots & g_2 \\
\vdots & \vdots & \vdots & \ddots & \vdots \\
g_{N-1} & g_{N-2} & g_{N-3} & \cdots & g_0 \\
\end{bmatrix}
\]

(i) Prove that the eigenvalues are given by (2.110) and that the right eigenvectors are the columns of \( F^* \).

(ii) Show that the left eigenvectors are the rows of \( F \).

2.7. DFT Properties

Verify the following DFT pairs from Table 2.5:

(i) Time reversal \( y_n = x_{-n} \);

(ii) Real symmetric \( x_n = x_{-n} \) and real antisymmetric \( x_n = -x_{-n} \) sequences;

(iii) Modulation property.

(Hint: Start with the convolution in frequency domain.)

2.8. Computing Linear Convolution with the DFT

Prove that the linear convolution of two sequences of length \( M \) and \( L \), can be computed using DFTs of size \( N \geq M + L - 1 \), and show how to do it.

2.9. ROC of z-Transform

Verify the properties of the ROCs listed in Table 2.3.

2.10. Interchange of Filtering and Sampling Rate Change

(i) Prove that downsampling by 2 followed by filtering with \( H(z) \) is equivalent to filtering with \( H(z^2) \) followed by downsampling by 2.

(ii) Prove that filtering with \( G(z) \) followed by upsampling by 2 is equivalent to upsampling by 2 followed by filtering with \( G(z^2) \).

2.11. Commutativity of Up- and Downsampling

Prove that downsampling by \( M \) and upsampling by \( N \) commute if and only if \( M \) and \( N \) are coprime.
Chapter 3

Continuous-Time Signals and Systems

Contents

3.1 Introduction ................................. 160
3.2 Continuous-Time Fourier Transform—CTFT . . . 160
3.3 Continuous-Time Fourier Series—CTFS . . . . . . 163
3.4 Shannon Sampling ............................ 165

Chapter at a Glance ............................ 170
Historical Remarks ............................ 170
Further Reading ............................... 170
Exercises with Solutions ...................... 170
Exercises ........................................ 170

This chapter reviews the Fourier transform and its variations when signals have particular properties (such as periodicity). The material in this section can be found in many sources, and we refer to [23, 65, 73, 122, 165] for details and proofs.

Various incarnations of the Fourier transform are named based on the domain they operate on (continuous or discrete) as well as the reconstruction formula (integral or series). For consistency with the long history of using Fourier transforms in engineering, we refer to the domain as “time” even though the signals of interest are not always functions of time. Thus a Fourier transform operating on the continuous-domain signal with an integral reconstruction formula will be called the continuous-time Fourier transform (CTFT) while the one where the reconstruction formula is given as a series (for periodic signals) is the continuous-time Fourier series (CTFS).
3.1 Introduction

3.2 Continuous-Time Fourier Transform—CTFT

Given an absolutely integrable function \( f(t) \), its continuous-time Fourier transform (CTFT)\(^{28}\) is defined by

\[
F(\omega) = \int_{-\infty}^{\infty} f(t) e^{-j\omega t} \, dt = \langle e^{j\omega t}, f(t) \rangle_t, \tag{3.1a}
\]
which is called the Fourier analysis formula. The inverse Fourier transform is given by the Fourier synthesis formula:

\[
f(t) = \frac{1}{2\pi} \int_{-\infty}^{\infty} F(\omega) e^{j\omega t} \, d\omega = \frac{1}{2\pi} \langle e^{-j\omega t}, F(\omega) \rangle_\omega. \tag{3.1b}
\]

Note that \( e^{j\omega t} \) is not in \( L^2(\mathbb{R}) \) and that the set \( \{e^{j\omega t}\} \) is not countable. The exact conditions under which (3.1b) is the inverse of (3.1a) depend on the behavior of \( f(t) \) and are discussed in standard texts on Fourier theory [34, 165]. For example, the inversion is exact if \( f(t) \) is continuous or if \( f(t) \) is defined as \( (f(t^+) + f(t^-))/2 \) at a point of discontinuity.\(^{29}\)

When \( f(t) \) is square-integrable, then the formulas above hold in the \( L^2 \) sense; that is, calling \( \hat{f}(t) \) the result of the analysis followed by the synthesis formula,

\[
\|f(t) - \hat{f}(t)\| = 0.
\]

Assuming that the Fourier transform and its inverse exist, we will denote by

\[
f(t) \leftrightarrow F(\omega)
\]

a Fourier transform pair. The Fourier transform satisfies a number of properties, some of which we briefly review below. For proofs, see [122].

3.2.1 Properties of the CTFT

Linearity

Since the Fourier transform is an inner product (see (3.1a)), it follows immediately from the linearity of the inner product that

\[
\alpha f(t) + \beta g(t) \leftrightarrow \alpha F(\omega) + \beta G(\omega).
\]

\(^{28}\)This is also known simply as the Fourier transform.

\(^{29}\)We assume that \( f(t) \) is of bounded variation. That is, for \( f(t) \) defined on a closed interval \([a, b]\), there exists a constant \( A \) such that \( \sum_{n=1}^{N} |f(t_n) - f(t_{n-1})| < A \) for any finite set \( \{t_i\} \) satisfying \( a \leq t_0 < t_1 < \ldots < t_N \leq b \). Roughly speaking, the graph of \( f(t) \) cannot oscillate over an infinite distance as \( t \) goes over a finite interval.
3.2. Continuous-Time Fourier Transform—CTFT

Symmetry

If $F(\omega)$ is the Fourier transform of $f(t)$, then

$$F(t) \longleftrightarrow 2\pi f(-\omega), \quad (3.2)$$

which indicates the essential symmetry of the Fourier analysis and synthesis formulas.

Shifting

A shift by $t_0$ in the time domain results in multiplication by a phase factor in the Fourier domain,

$$f(t - t_0) \longleftrightarrow e^{-j\omega t_0} F(\omega). \quad (3.3)$$

Conversely, a shift in frequency results in a phase factor, or modulation by a complex exponential, in the time domain,

$$e^{j\omega t} f(t) \longleftrightarrow F(\omega - \omega_0).$$

Scaling

Scaling in the time domain results in inverse scaling in frequency as given by the following transform pair ($a$ is a real constant):

$$f(at) \longleftrightarrow \frac{1}{|a|} F\left(\frac{\omega}{a}\right). \quad (3.4)$$

Differentiation/Integration

Derivatives in the time domain lead to multiplication by $(j\omega)$ in frequency,

$$\frac{\partial^n f(t)}{\partial t^n} \longleftrightarrow (j\omega)^n F(\omega), \quad (3.5)$$

if the transform actually exists. Conversely, if $F(0) = 0$, we have

$$\int_{-\infty}^{t} f(\tau) \, d\tau \longleftrightarrow \frac{F(\omega)}{j\omega}.$$

Differentiation in frequency leads to

$$(-jt)^n f(t) \longleftrightarrow \frac{\partial^n F(\omega)}{\partial \omega^n}.$$

Moments

Calling $m_n$ the $n$th moment of $f(t)$,

$$m_n = \int_{-\infty}^{\infty} t^n f(t) \, dt, \quad n = 0, 1, 2, \ldots, \quad (3.6)$$

the moment theorem of the Fourier transform states that

$$(-j)^n m_n = \frac{\partial^n F(\omega)}{\partial \omega^n} \big|_{\omega=0}, \quad n = 0, 1, 2, \ldots. \quad (3.7)$$
The convolution of two functions \( f(t) \) and \( g(t) \) is given by

\[
h(t) = \int_{-\infty}^{\infty} f(\tau) g(t - \tau) \, d\tau,
\]

(3.8)

and is denoted \( h(t) = f(t) * g(t) = g(t) * f(t) \) since (3.8) is symmetric in \( f(t) \) and \( g(t) \).

Denoting by \( F(\omega) \) and \( G(\omega) \) the Fourier transforms of \( f(t) \) and \( g(t) \), respectively, the convolution theorem states that

\[
f(t) * g(t) \longleftrightarrow F(\omega) \, G(\omega).
\]

This result is fundamental, and we will prove it for \( f(t) \) and \( g(t) \) being in \( L_1(\mathbb{R}) \). Taking the Fourier transform of \( f(t) * g(t) \),

\[
\int_{-\infty}^{\infty} \left[ \int_{-\infty}^{\infty} f(\tau) g(t - \tau) \, d\tau \right] e^{-j\omega t} \, dt,
\]

changing the order of integration (which is allowed when \( f(t) \) and \( g(t) \) are in \( L_1(\mathbb{R}) \); see Fubini’s theorem in [52, 130]) and using the shift property, we get

\[
\int_{-\infty}^{\infty} f(\tau) \left[ \int_{-\infty}^{\infty} g(t - \tau) e^{-j\omega t} \, dt \right] d\tau = \int_{-\infty}^{\infty} f(\tau) e^{-j\omega \tau} G(\omega) \, d\tau = F(\omega) \, G(\omega).
\]

The result holds as well when \( f(t) \) and \( g(t) \) are square-integrable, but requires a different proof [73].

An alternative view of the convolution theorem is to identify the complex exponentials \( e^{j\omega t} \) as the eigenfunctions of the convolution operator, since

\[
\int_{-\infty}^{\infty} e^{j\omega(t-\tau)} g(\tau) \, d\tau = e^{j\omega t} \int_{-\infty}^{\infty} e^{-j\omega \tau} g(\tau) \, d\tau = e^{j\omega t} G(\omega).
\]

The associated eigenvalue \( G(\omega) \) is simply the Fourier transform of the impulse response \( g(\tau) \) at frequency \( \omega \).

By symmetry, the product of discrete-time functions leads to the convolution of their Fourier transforms,

\[
f(t) \, g(t) \longleftrightarrow \frac{1}{2\pi} F(\omega) * G(\omega).
\]

(3.9)

This is known as the modulation theorem of the Fourier transform.

As an application of both the convolution theorem and the derivative property, consider taking the derivative of a convolution,

\[
h'(t) = \frac{\partial[f(t) * g(t)]}{\partial t}.
\]

The Fourier transform of \( h'(t) \), following (3.5), is equal to

\[
j\omega \, (F(\omega)G(\omega)) = (j\omega F(\omega)) \, G(\omega) = F(\omega) \, (j\omega G(\omega)) ,
\]
that is,
\[ h'(t) = f'(t) * g(t) = f(t) * g'(t). \]

This is useful when convolving a signal with a filter which is known to be the derivative of a given function such as a Gaussian, since one can think of the result as being the convolution of the derivative of the signal with a Gaussian.

**Parseval's Formula**

Because the Fourier transform is an orthogonal transform, it satisfies an energy conservation relation known as Parseval's formula. See also Chapter 1 where we proved Parseval's formula for orthonormal bases. Here, we need a different proof because the Fourier transform does not correspond to an orthonormal basis expansion (first, exponentials are not in \( L^2(\mathbb{R}) \) and also the complex exponentials are uncountable, whereas we considered countable orthonormal bases [74]). The general form of Parseval's formula for the Fourier transform is given by

\[
\int_{-\infty}^{\infty} f^*(t) g(t) \, dt = \frac{1}{2\pi} \int_{-\infty}^{\infty} F^*(\omega) G(\omega) \, d\omega, \tag{3.10a}
\]

which reduces, when \( g(t) = f(t) \), to

\[
\int_{-\infty}^{\infty} |f(t)|^2 \, dt = \frac{1}{2\pi} \int_{-\infty}^{\infty} |F(\omega)|^2 \, d\omega. \tag{3.10b}
\]

Note that the factor 1/2\(\pi\) comes from our definition of the Fourier transform (3.1a–3.1b). A symmetric definition, with a factor 1/\(\sqrt{2\pi}\) in both the analysis and synthesis formulas (see, for example, [52]), would remove the scale factor in (3.10b).

The proof of (3.10a) uses the fact that

\[ f^*(t) \longleftrightarrow F^*(-\omega) \]

and the frequency-domain convolution relation (3.9). That is, since \( f^*(t) \cdot g(t) \) has Fourier transform \((1/2\pi)(F^*(-\omega) * G(\omega))\), we have

\[
\int_{-\infty}^{\infty} f^*(t) g(t) e^{-j\omega t} \, dt = \frac{1}{2\pi} \int_{-\infty}^{\infty} F^*(-\Omega) G(\omega + \Omega) \, d\Omega, \]

where (3.10a) follows by setting \( \omega = 0. \)

### 3.3 Continuous-Time Fourier Series—CTFS

A periodic function \( f(t) \) with period \( T \),

\[ f(t + T) = f(t), \]

can be expressed as a linear combination of complex exponentials with frequencies \( n\omega_0 \) where \( \omega_0 = 2\pi/T \). In other words,

\[
f(t) = \sum_{k=-\infty}^{\infty} F_k e^{j2\pi ft}, \tag{3.11a}
\]
with
\[ F_k = \frac{1}{T} \int_{-T/2}^{T/2} f(t) e^{-jk\omega_0 t} \, dt. \] (3.11b)

If \( f(t) \) is continuous, then the series converges uniformly to \( f(t) \). If a period of \( f(t) \) is square-integrable but not necessarily continuous, then the series converges to \( f(t) \) in the \( L^2 \) sense; denoting by \( \hat{f}_N(t) \) the truncated series with \( k \) going from \(-N\) to \( N \), the error \( \|f(t) - \hat{f}_N(t)\| \) goes to zero as \( N \to \infty \). At points of discontinuity, the infinite sum (3.11a) equals the average \( (f(t^+) + f(t^-))/2 \). However, convergence is not uniform anymore but plagued by the Gibbs phenomenon. That is, \( \hat{f}_N(t) \) will overshoot or undershoot near the point of discontinuity. The amount of over/undershooting is independent of the number of terms \( N \) used in the approximation. Only the width diminishes as \( N \) is increased.\(^{30}\) For further discussions on the convergence of Fourier series, see [34, 165].

Of course, underlying the Fourier series construction is the fact that the set of functions used in the expansion (3.11a) is a complete orthonormal system for the interval \([-T/2, T/2]\) (up to a scale factor). That is, defining \( \varphi_k(t) = (1/\sqrt{T}) e^{jk\omega_0 t} \) for \( t \) in \([-T/2, T/2]\) and \( k \in \mathbb{Z} \), we can verify that
\[ \langle \varphi_k(t), \varphi_\ell(t) \rangle_{t \in [-T/2, T/2]} = \delta_{k-\ell}. \]

When \( k = \ell \), the inner product equals 1. If \( k \neq \ell \), we have
\[ \frac{1}{T} \int_{-T/2}^{T/2} e^{jk\omega_0(t-k)t} \, dt = \frac{1}{\pi(\ell - k)} \sin(\pi(\ell - k)) = 0. \]

That the set \( \{\varphi_k\} \) is complete is shown in [165] and means that there exists no periodic function \( f(t) \) with \( L^2 \) norm greater than zero that has all its Fourier series coefficients equal to zero. Actually, there is equivalence between norms, as shown below.

### 3.3.1 Properties of the CTFS

**Parseval's Relation**

With the Fourier series coefficients as defined in (3.11b), and the inner product of periodic functions taken over one period, we have
\[ \langle f(t), g(t) \rangle_{t \in [-T/2, T/2]} = T \langle F_k, G_k \rangle, \]

where the factor \( T \) is due to the normalization chosen in (3.11a–3.11b). In particular, for \( g(t) = f(t) \),
\[ \|f(t)\|_{[-T/2, T/2]}^2 = T\|F_k\|^2. \]

This is an example of Theorem 1.5, up to the scaling factor \( T \).

\(^{30}\)Again, we consider non-pathological functions (that is, of bounded variation).
3.4. Shannon Sampling

**Best Approximation Property**

While the following result is true in a more general setting (see Section 1.4.1), it is sufficiently important to be restated for Fourier series, namely

\[
\left\| f(t) - \sum_{k=-N}^{N} \langle \varphi_k, f \rangle \varphi_k(t) \right\| \leq \left\| f(t) - \sum_{k=-N}^{N} a_k \varphi_k(t) \right\|
\]

where \( \{a_k\} \) is an arbitrary set of coefficients. That is, the Fourier series coefficients are the best ones for an approximation in the span of \( \{\varphi_k(t)\}, k = -N, \ldots, N \). Moreover, if \( N \) is increased, new coefficients are added without affecting the previous ones.

Fourier series, beside their obvious use for characterizing periodic signals, are useful for problems of finite size through periodization. The immediate concern, however, is the introduction of a discontinuity at the boundary, since periodization of a continuous signal on an interval results, in general, in a discontinuous periodic signal.

Fourier series can be related to the Fourier transform seen earlier by using sequences of Dirac functions which are also used in sampling.

### 3.4 Shannon Sampling

The process of sampling is central to discrete-time signal processing, since it provides the link with the continuous time domain. Call \( f_T(t) \) the sampled version of \( f(t) \), obtained as

\[
f_T(t) = f(t) s_T(t) = \sum_{n=-\infty}^{\infty} f(nT) \delta(t - nT). \tag{3.12}
\]

Using the modulation theorem of the Fourier transform (3.9) and the transform of \( s_T(t) \), we get

\[
F_T(\omega) = F(\omega) \ast \frac{1}{T} \sum_{k=-\infty}^{\infty} \delta \left( \omega - k \frac{2\pi}{T} \right) = \frac{1}{T} \sum_{k=-\infty}^{\infty} F \left( \omega - k \frac{2\pi}{T} \right), \tag{3.13}
\]

where we used the sifting property of convolving with a Dirac delta function. Thus, \( F_T(\omega) \) is periodic with period \( 2\pi/T \), and is obtained by overlapping copies of \( F(\omega) \) at every multiple of \( 2\pi/T \). Another way to prove (3.13) is to use the Poisson formula. Taking the Fourier transform of (3.12) results in

\[
F_T(\omega) = \sum_{n=-\infty}^{\infty} f(nT) e^{-jnT\omega},
\]

since \( f_T(t) \) is a weighted sequence of Dirac functions with weights \( f(nT) \) and shifts of \( nT \). To use the Poisson formula, consider the function \( g_0(t) = f(t) e^{-j\Omega t} \), which has Fourier transform \( G_\Omega(\omega) = F(\omega + \Omega) \) according to the effect of modulation on
the CTFT. Now, applying the Poisson Sum Formula to $g_{\Omega}(t)$, we find

$$\sum_{n=-\infty}^{\infty} g_{\Omega}(nT) = \frac{1}{T} \sum_{k=-\infty}^{\infty} G_{\Omega}\left(\frac{2\pi k}{T}\right)$$

or changing $\Omega$ to $\omega$ and switching the sign of $k$,

$$\sum_{n=-\infty}^{\infty} f(nT) e^{-jnT\omega} = \frac{1}{T} \sum_{k=-\infty}^{\infty} F\left(\omega - \frac{2\pi k}{T}\right), \quad (3.14)$$

which is the desired result (3.13).

Equation (3.13) leads immediately to the famous sampling theorem of Whittaker, Kotelnikov and Shannon. If the sampling frequency $\omega_s = 2\pi/T_s$ is larger than $2\omega_m$ (where $F(\omega)$ is bandlimited\(^{31}\) to $\omega_m$), then we can extract one instance of the spectrum without overlap. If this were not true, then, for example for $k = 0$ and $k = 1$, $F(\omega)$ and $F(\omega - 2\pi/T)$ would overlap and reconstruction would not be possible.

### 3.4.1 Sampling Theorem

**Theorem 3.1 (Sampling Theorem).** If $f(t)$ is continuous and bandlimited to $\omega_m$, then $f(t)$ is uniquely defined by its samples taken at twice $\omega_m$ or $f(n\pi/\omega_m)$. The minimum sampling frequency is $\omega_s = 2\omega_m$ and $T = \pi/\omega_m$ is the maximum sampling period. Then $f(t)$ can be recovered by the interpolation formula

$$f(t) = \sum_{n=-\infty}^{\infty} f(nT) \text{sinc}_T(t - nT), \quad (3.15)$$

where

$$\text{sinc}_T(t) = \frac{\sin(\pi t/T)}{\pi t/T}.$$ 

Note that $\text{sinc}_T(nT) = \delta_n$, that is, it has the interpolation property since it is 1 at the origin but 0 at nonzero multiples of $T$. It follows immediately that (3.15) holds at the sampling instants $t = nT$.

**Proof.** The proof that (3.15) is valid for all $t$ goes as follows: Consider the sampled version of $f(t)$, $f_T(t)$, consisting of weighted Dirac functions (3.12). We showed that its Fourier transform is given by (3.13). The sampling frequency $\omega_s$ equals $2\omega_m$, where $\omega_m$ is the bandlimiting frequency of $F(\omega)$. Thus, $F(\omega - k\omega_s)$ and $F(\omega - l\omega_s)$ do not overlap for $k \neq l$. To recover $F(\omega)$, it suffices to keep the term with $k = 0$ in (3.13) and normalize it by $T$. This is accomplished with a function that has a Fourier transform which is equal to $T$ from $-\omega_m$ to $\omega_m$ and 0 elsewhere. This is

\(^{31}\)We will say that a function $f(t)$ is bandlimited to $\omega_m$ if its Fourier transform $F(\omega) = 0$ for $|\omega| \geq \omega_m$. 

3.4. Shannon Sampling

called an ideal lowpass filter. Its discrete-time impulse response, denoted $\text{sinc}_T(t)$ where $T = \pi/\omega_m$, is equal to (taking the inverse Fourier transform)

$$\text{sinc}_T(t) = \frac{1}{2\pi} \int_{-\omega_m}^{\omega_m} T \, e^{j\omega t} \, d\omega = \frac{T}{2\pi jt} \left[ e^{j\pi t/T} - e^{-j\pi t/T} \right] = \frac{\sin(\pi t/T)}{\pi t/T}. \quad (3.16)$$

Convolving $f_T(t)$ with $\text{sinc}_T(t)$ filters out the repeated spectra (terms with $k \neq 0$ in (3.13)) and recovers $f(t)$, as is clear in frequency domain. Because $f_T(t)$ is a sequence of Dirac functions of weights $f(nT)$, the convolution results in a weighted sum of shifted impulse responses,

$$\left[ \sum_{n=-\infty}^{\infty} f(nT) \delta(t - nT) \right] * \text{sinc}_T(t) = \sum_{n=-\infty}^{\infty} f(nT) \text{sinc}_T(t - nT),$$

proving (3.15) \qed

An alternative interpretation of the sampling theorem is as a series expansion on an orthonormal basis for bandlimited signals. Define

$$\varphi_{n,T}(t) = \frac{1}{\sqrt{T}} \text{sinc}_T(t - nT), \quad (3.17)$$

whose Fourier transform magnitude is $\sqrt{T}$ from $-\omega_m$ to $\omega_m$, and 0 otherwise. One can verify that $\varphi_{n,T}(t)$ form an orthonormal set using Parseval’s relation. The Fourier transform of (3.17) is (from (3.16) and the shift property (3.3))

$$\Phi_{n,T}(\omega) \longleftrightarrow \left\{ \begin{array}{ll}
\sqrt{\frac{\pi}{\omega_m}} e^{-j\omega n \pi/\omega_m}, & -\omega_m \leq \omega \leq \omega_m; \\
0, & \text{otherwise},
\end{array} \right.$$ 

where $T = \pi/\omega_m$. From (3.10a), we find

$$\langle \varphi_{n,T}, \varphi_{k,T} \rangle = \frac{1}{2\omega_m} \int_{-\omega_m}^{\omega_m} e^{j\omega(n-k)\pi/\omega_m} \, d\omega = \delta_{n-k}.$$ 

Now, assume a bandlimited signal $f(t)$ and consider the inner product $\langle \varphi_{n,T}, f \rangle$. Again using Parseval’s relation,

$$\langle \varphi_{n,T}, f \rangle = \frac{\sqrt{T}}{2\pi} \int_{-\omega_m}^{\omega_m} e^{j\omega nT} \, F(\omega) \, d\omega = \sqrt{T} f(nT),$$

because the integral is recognized as the inverse Fourier transform of $F(\omega)$ at $t = nT$ (the bounds $[-\omega_m, \omega_m]$ do not alter the computation of $F(\omega)$ because it is bandlimited to $\omega_m$). Therefore, another way to write the interpolation formula (3.15) is

$$f(t) = \sum_{n=-\infty}^{\infty} \langle \varphi_{n,T}, f \rangle \varphi_{n,T}(t) \quad (3.18)$$

(the only change is that we normalized the sinc basis functions to have unit norm).
What happens if \( f(t) \) is not bandlimited? Because \( \{\varphi_{n,T}\} \) is an orthogonal set, the interpolation formula (3.18) represents the orthogonal projection of the input signal onto the subspace of bandlimited signals. Another way to write the inner product in (3.18) is

\[
\langle \varphi_{n,T}, f \rangle = \int_{-\infty}^{\infty} \varphi_{0,T}(\tau - nT) f(\tau) \, d\tau = \varphi_{0,T}(-t) \ast f(t) \big|_{t=nT},
\]

which equals \( \varphi_{0,T}(t) \ast f(t) \) since \( \varphi_{0,T}(t) \) is real and symmetric in \( t \). That is, the inner products, or coefficients, in the interpolation formula are simply the outputs of an ideal lowpass filter with cutoff \( \pi/T \) sampled at multiples of \( T \). This is the usual view of the sampling theorem as a bandlimiting convolution followed by sampling and re-interpolation.

To conclude this section, we will demonstrate a fact that will be used in Chapter 8. It states that the following can be seen as a Fourier transform pair:

\[
\langle f(t), f(t+n) \rangle_t = \delta_n \longleftrightarrow \sum_{k \in \mathbb{Z}} |F(\omega + 2k\pi)|^2 = 1. \tag{3.19}
\]

The left side of the equation is simply the deterministic autocorrelation\(^{32}\) of \( f(t) \) evaluated at integers, that is, sampled autocorrelation. If we denote the autocorrelation of \( f(t) \) as \( p(\tau) = \langle f(t), f(t+\tau) \rangle_t \), then the left side of (3.19) is \( p_1(\tau) = p(\tau) s_1(\tau) \), where \( s_1(\tau) \) is a train of Diracs spaced by 1. The Fourier transform of \( p_1(\tau) \) is (apply (3.13))

\[
P_1(\omega) = \sum_{k \in \mathbb{Z}} P(\omega - 2k\pi).
\]

Since the Fourier transform of \( p(t) \) is \( P(\omega) = |F(\omega)|^2 \), we get that the Fourier transform of the right side of (3.19) is the left side of (3.19).

**Pictorial Summary**

**What Does the Sampling Theorem Teach Us?**

From Table 3.1, we can see that:

- \( \omega_m > \frac{\pi}{T} \): We have a problem—aliasing. The repeated spectra overlap, and there is no way to recover the lost information.

- \( \omega_m < \frac{\pi}{T} \): We are fine. The repeated spectra have “guard-bands” in between them and are completely recoverable.

- \( \omega_m = \frac{\pi}{T} \): This is called the critical or Nyquist case. The repeated spectra cover the domain of \( \omega \) tightly and the signal can be completely recovered.

**Example 3.1.** In Figure 3.1(a) and (b), we see the same two signals and the sequences of samples they produce if they are sampled correctly ((c) and (d)) versus incorrectly ((e) and (f)). What this means is that the signals in (a) and (b) can be reconstructed from their samples in (c) and (d), respectively, but not from those in (e) and (f). ■

\(^{32}\)The deterministic autocorrelation of a real function \( f(t) \) is \( f(t) \ast f(-t) = \int f(\tau) \, f(\tau + t) \, d\tau \).
3.4. Shannon Sampling

![Figure 3.1:](image)

(a) Continuous-time signal \( f_1(t) = \sin(8t) \). (b) Continuous-time signal \( f_2(t) = \sin(8t) + 0.1\sin(64t) \). (c) Discrete-time sequence obtained by sampling \( f_1(t) \) at multiples of \( \pi/8 \). This is sampling at Nyquist frequency and thus \( f_1(t) \) can be uniquely reconstructed from its samples. (d) Discrete-time sequence obtained by sampling \( f_2(t) \) at multiples of \( \pi/64 \). This is sampling at Nyquist frequency and thus \( f_2(t) \) can be uniquely reconstructed from its samples. (e) Discrete-time sequence obtained by sampling \( f_1(t) \) at multiples of \( \pi/4 \). This is sampling below Nyquist frequency and thus \( f_1(t) \) cannot be uniquely reconstructed from its samples. (f) Discrete-time sequence obtained by sampling \( f_2(t) \) at multiples of \( \pi/8 \). This is sampling below Nyquist frequency and thus \( f_2(t) \) cannot be uniquely reconstructed from its samples.

What Does This Mean in Practice?

In a real-world problem, you are typically given (a) a sequence of digitized samples and (b) the sampling frequency in Hz. So what does the sample \( n \) correspond to in time? This is an easy exercise, but since it is very important, we go through it step-by-step.

We are given the sampling frequency \( f_s \), which equals \( f_s = 1/T_s \), where \( T_s \) is the sampling period. That means that sample \( n \) corresponds to \( nT_s \) sec, or \( n/f_s \) sec. For example, if our sampling frequency is \( f_s = 1 \text{Hz} \), that means that \( T_s = 1 \) sec. This is the underlying assumption in discrete-time signal processing. However, sometimes it is necessary to know what the samples correspond to in time (for example, when dealing with a sequence of EEG samples, we must know what the actual time is). Then, sample \( n \) corresponds to \( n/f_s = n \) sec. If, on the other hand, \( f_s = 200 \text{ Hz} \), then sample \( n \) corresponds to \( n/f_s = 0.005n \) sec or \( n \cdot 5 \) msec. Thus, our sequence is in increments of 5 msec.

All this, of course, is valid if our signal was bandlimited to \( f_m = 2f_s \) before sampling, which is rarely the case in practice. What does this mean? It means that you must cut off (filter) everything after \( f_m \), otherwise you will get aliasing, as we explained earlier, and will not be able to reconstruct the signal properly.
Chapter at a Glance

Historical Remarks

A giant featured in the title of this chapter, Jean Baptiste Joseph Fourier (1768-1830) was a French mathematician, who proposed his famous Fourier series while working on the equations for heat flow. His interests were varied, his biography unusual. He followed Napoleon to Egypt and spent a few years in Cairo, even contributing a few papers to the Egyptian Institute Napoleon founded. He served as a permanent secretary of the French Academy of Science. He published “Théorie analytique de la chaleur”, in which he claimed that any function can be decomposed into sum of sines; while we know that is not true, it took mathematicians a long time to rephrase the result. Lagrange and Dirichlet both worked on it, with Dirichlet finally formulating conditions under which Fourier series existed.

Further Reading

Key references Koerner [97] Bracewell [23] Papoulis [122]

Exercises with Solutions

3.1. TBD.

Exercises

3.1. TBD.
### Table 3.1: Pictorial summary of sampling.

<table>
<thead>
<tr>
<th>$f(t)$</th>
<th>$s_T(t) = \sum_{n \in \mathbb{Z}} \delta(t - nT)$</th>
<th>$f_T(t) = f(t) s_T(t)$, where $f_T(t) = \sum_{n \in \mathbb{Z}} f(nT) \delta(t - nT)$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$F(\omega)$</td>
<td>$S_T(\omega) = \frac{1}{T} \sum_{k \in \mathbb{Z}} \delta(\omega - k \frac{2\pi}{T})$</td>
<td>$F_T(\omega) = F(\omega) * S_T(\omega)$, where $F_T(\omega) = \frac{1}{T} \sum_{k \in \mathbb{Z}} F(\omega - k \frac{2\pi}{T}) \delta(\omega - k \frac{2\pi}{T})$</td>
</tr>
</tbody>
</table>
Chapter 3. Continuous-Time Signals and Systems
Chapter 4
Sampling, Interpolation, and Approximation

Contents

4.1 Introduction ............................................................... 174
4.2 Functions on a Closed, Bounded Interval ...................... 175
4.3 Functions on the Real Line ........................................... 181
4.4 Functions on the Real Line Revisited: Splines ............... 181
4.5 Approximations with Finite Precision .......................... 181
4.6 Approximating Random Processes .............................. 181
4.7 Algorithms ............................................................... 181
Chapter at a Glance ........................................................... 181
Historical Remarks .......................................................... 181
Further Reading ............................................................. 181
Exercises with Solutions ................................................... 181
Exercises ..................................................................... 182

Chapters 1–3 concentrated, in turn, on abstract vectors, sequences, and functions of a real variable. While all three of these settings are important, this book (indeed, the field of signal processing) concentrates on sequences. On the one hand, even though most signal processing is done on finite vectors, the techniques are usually modified from those designed for infinite sequences since the lengths of the vectors are very large. On the other hand, computations with functions of real variables are extremely difficult to physically realize, so these functions are approximated with sequences.

The primary goal of this chapter is to show several ways that continuous-time signals (functions with \( \mathbb{R} \) or part of \( \mathbb{R} \) as the domain, and \( \mathbb{R} \) as the range) and discrete-time sequences (real-valued) can be related. Obtaining a sequence from a function is called sampling, while the converse is called interpolation. Sampling followed by interpolation results in an approximation of the original function; we study this and other forms of function approximation.

\[33\] These modifications are discussed in (TBD).
Digital computation requires not just the discretization of time but also “digi-
tization” of values (finite precision or quantization). A secondary goal of this chap-
ter is to discuss some elementary effects of quantization. On modern computing
platforms, discretization of values to enable computation is very fine. Quantization
is covered in more detail in the context of compression in Chapter 12, where it is
course and thus more significant.

4.1 Introduction

When one first encounters functions, it is not through a formal definition, but rather
through examples of elementary expressions: \( f(t) = 2t^3 + t - 1 \), \( f(t) = \log(t^2 + 1) \),
\( f(t) = 3\cos(e^t \tan(t - \pi/4)) \), etc. From such an expression, one can evaluate a
function but it may be very difficult to make desired computations. For example,
closed-form evaluation of \( \int_0^1 e^{-t^2} \, dt \) is impossible (without adding a new “named
function erf specifically for this purpose) even though the integrand is relatively
simple. One approach is to find an approximation to the integrand that is easy to
integrate. This approach raises several questions, e.g.: How should we measure the
approximation quality of the integrand so that we can derive an accuracy guarantee
on the value of the integral? How can we choose the approximation to get the
best accuracy guarantee? Integrating is of course just one possible computational
problem.

The abstract conception of a function as any (single-valued) mapping makes
the prospect of computation even more daunting: Merely specifying a function
requires evaluation at uncountably many points, so how could we possibly make
any computations? Fortunately, for engineering purposes we can restrict the class
of functions (e.g. to have finite energy and finitely many discontinuities) and it
suffices to have good approximations.

Chapter Outline

Most of this chapter is dedicated to the basics of representing or approximating a
function with a finite or countable set of parameters. Just as there are infinitely
many ways to combine elementary functions, there are infinitely many such approx-
imations, but most are esoteric. We concentrate on the most useful approximations,
starting in Section 4.2 with functions on a closed, bounded interval. Boundedness of
the interval allows us to use polynomials (which otherwise would blow up) or to pe-
riodize the function and use trigonometric polynomials. Section 4.3 presents global
techniques for functions on the entire real line, which includes classical sampling
and some of its generalizations. Section 4.4 covers spline approximations, which
also are for functions on the entire real line, but have locality properties detailed
later.

To use a computer we have to go beyond countable number of parameters to
also making those parameters each take on a finite number of values. This finite-
precision aspect to approximation is introduced in Section 4.5. Finally, Sections 4.6
and 4.7 cover stochastic and algorithmic aspects of approximation.
4.2 Functions on a Closed, Bounded Interval

We start by considering approximation of a function \( f \) defined on an arbitrary closed, bounded interval using polynomials or trigonometric polynomials. For convenience, the results and examples with polynomials are for the interval \([0, 1]\) and those for trigonometric polynomials are for \([0, 2\pi]\). All the results can be easily modified for an arbitrary closed, bounded interval \([a, b]\).

4.2.1 Approximation by Polynomials

Let \( f \) be defined on \([0, 1]\) and consider approximations of the form

\[
p_K(t) = a_K t^K + a_{K-1} t^{K-1} + \cdots + a_1 t + a_0.
\]

Assuming \( a_K \neq 0 \), \( p_K(t) \) is called a polynomial of degree \( K \). Since a degree \( K \) polynomial is defined by \( K + 1 \) parameters \( \{a_k\}_{k=0}^K \), an approximation can be determined by any \( K + 1 \) independent constraints.

**Lagrange Interpolation Formula**

One natural way to make \( p_K \) approximate \( f \) is to force \( p_K \) to match \( f \) at selected points in \([0, 1]\). Fix any \( K + 1 \) distinct points \( \{t_k\}_{k=0}^K \subset [0, 1] \), called nodes. Then exactly one degree-\( K \) polynomial matches \( f \) at the nodes:

\[
p_{\text{interp}}(t) = \sum_{k=0}^{K} f(t_k) \prod_{\substack{i = 0 \atop i \neq k}}^{K} \frac{t - t_i}{t_k - t_i}.
\]

(4.1)

This is called the interpolating polynomial for \( \{(t_k, f(t_k))\}_{k=0}^K \). It is easy to verify that \( p_{\text{interp}}(t_k) = f(t_k) \) for \( k = 0, 1, \ldots, K \), and the uniqueness of the interpolating polynomial is established in Exercise 4.2.

**Example 4.1 (Lagrange Interpolation).** Let us construct approximations using (4.1) for two functions on \([0, 1]\), one very smooth and the other not even continuous:

\[
f_1(t) = t \sin 5t \quad \text{and} \quad f_2(t) = \begin{cases} t, & t \in [0, 1/\sqrt{2}]; \\ t - 1, & t \in (1/\sqrt{2}, 1]. \end{cases}
\]

(4.2)

Take the interpolation points to be \( k/K \) for \( k = 0, 1, \ldots, K \) (though the Lagrange interpolation formula does not require evenly-spaced nodes). Figure 4.1 shows the functions (bold) and the interpolating polynomials for \( K = 1, 2, \ldots, 7 \). The continuous function \( f_1 \) is approximated much more closely than the discontinuous function \( f_2 \).

The example hints that the smoothness of a function affects the quality of approximation by interpolation. This is made precise after we introduce another elementary polynomial approximation.
Taylor Series

Another simple way to pick \( p_K \) is to match \( f \) and the first \( K \) derivatives of \( f \) at some point. Assuming \( f \) has \( K \) derivatives at \( t_0 \in [0, 1] \), let

\[
p_{\text{Taylor}}(t) = f(t_0) + f'(t_0)(t - t_0) + \frac{1}{2}f''(t_0)(t - t_0)^2 + \cdots + \frac{1}{K!}f^{(K)}(t - t_0)^K. \tag{4.3}
\]

This is called a Taylor series expansion about \( t_0 \) and has the desired property.

**Example 4.2 (Taylor series).** Consider Taylor series expansions about \( \frac{1}{2} \) of the functions \( f_1 \) and \( f_2 \) in (4.2). Both \( f_1 \) and \( f_2 \) are infinitely differentiable at \( \frac{1}{2} \), so the Taylor series are easy to compute. Figure 4.2 shows \( f_1 \) (bold) and its expansions of degree \( K = 1, 2, \ldots, 5 \). The Taylor series for \( f_2 \) is not interesting to plot because we have \( p_{\text{Taylor}}(t) = t \) for all degrees \( \geq 1 \). While this is exact for \( t \in [0, 1/\sqrt{2}] \), the error for \( t \in (1/\sqrt{2}, 1] \) cannot be reduced by increasing the degree.

Error Bounds for Lagrange Interpolation and Taylor Series

As Example 4.2 illustrated, a Taylor series approximation can differ from the original function by an arbitrary amount if the function is discontinuous. This is not surprising because a Taylor series gets all its information about \( f \) from an infinitesimal interval around \( t_0 \). Lagrange interpolation gets its information from samples distributed through the interval of interest, but as shown in Example 4.1, it also does not perform well for discontinuous functions.

For functions with sufficiently-many continuous derivatives, the two approximation methods we have seen have remarkably similar error bounds. We measure error here by the pointwise difference between \( f \) and \( p_K \). This is called uniform approximation or Chebyshev approximation and the maximum absolute value of the
4.2. Functions on a Closed, Bounded Interval

Figure 4.2: Taylor series expansions of \( f_1 \) in (4.2) for degrees \( K = 1, 2, \ldots, 5 \). Bold curve is the original function and light curves are labeled by the polynomial degree.

Pointwise difference is the \( \mathcal{L}^\infty \) norm \( \| f - p_K \|_\infty \). The first theorem is proven in most calculus texts. The second is proven, e.g., in [4].

**Theorem 4.1 (Taylor’s theorem).** Let \( f \) have \( K + 1 \) continuous derivatives on \([0, 1]\) for some \( K \geq 0 \). Then with \( p_{\text{Taylor}} \) defined in (4.3),

\[
f(t) - p_{\text{Taylor}}(t) = \frac{(t-t_0)^{K+1}}{(K+1)!} f^{(K+1)}(\xi)
\]

for some \( \xi \) between \( t \) and \( t_0 \). Thus furthermore

\[
|f(t) - p_{\text{Taylor}}(t)| \leq \frac{|t-t_0|^{K+1}}{(K+1)!} \max_{\xi \in [0,1]} |f^{(K+1)}(\xi)|
\]

for every \( t \in [0, 1] \).

**Theorem 4.2 (Error of Lagrange interpolation).** Let \( f \) have \( K + 1 \) continuous derivatives on \([0, 1]\) for some \( K \geq 0 \), and let \( \{t_k\}_{k=0}^K \subset [0,1] \) be distinct. Then with \( p_{\text{interp}} \) defined in (4.1),

\[
f(t) - p_{\text{interp}}(t) = \frac{(t-t_0)(t-t_1)\cdots(t-t_K)}{(K+1)!} f^{(K+1)}(\xi)
\]

for some \( \xi \) between the minimum and maximum of \( \{t, t_0, t_1, \ldots, t_K\} \). Thus furthermore

\[
|f(t) - p_{\text{interp}}(t)| \leq \frac{|t-t_0| \cdot |t-t_1| \cdots |t-t_K|}{(K+1)!} \max_{\xi \in [0,1]} |f^{(K+1)}(\xi)|
\]

for every \( t \in [0, 1] \).

The numerators in the bounds (4.5) and (4.7) suggest there will be greater error at the extremes of the interval of interest. This is shown in Figure 4.3 for \( f_1 \) in (4.2).
Chapter 4. Sampling, Interpolation, and Approximation

Figure 4.3: Errors of polynomial interpolation of $f_1(t) = t \sin 5t$ from Examples 4.1 and 4.2. Curves are labeled by the polynomial degree.

Minimax Polynomial Approximation

Our two constructions thus far are based on nothing but gathering enough data to uniquely determine a degree-$K$ polynomial—not on any goodness criterion for the polynomial. In fact, there are infinitely-differentiable functions for which these constructions do not give convergent approximations (see Exercise ??). This is a deficiency of the constructions rather than of polynomial approximation: the following theorem establishes that continuity of $f$ is enough for the existence of polynomial approximations that are arbitrarily close to $f$ over the entire interval $[0,1]$.

Theorem 4.3 (Weierstrass approximation theorem). Let $f$ be continuous on $[0,1]$ and let $\epsilon > 0$. Then there is a polynomial $p$ for which

$$|f(t) - p(t)| \leq \epsilon \quad \text{for every } t \in [0,1]. \quad (4.8)$$

Proof. See Exercise ??.

Theorem 4.3 asserts that the maximum error in approximating a continuous function $f$ can be made arbitrarily small. Conversely, the continuity of polynomials implies that a discontinuous $f$ certainly cannot be approximated arbitrarily well in the neighborhood of a discontinuity.

If we minimize the maximum error for a given degree $K$, we obtain a minimax approximation $p_K^*$. An algorithm for minimax approximation is presented in Section 4.7. To see a key property, we continue the previous examples.

Example 4.3 (Minimax approximation). The first four minimax polynomial ap-
4.2. Functions on a Closed, Bounded Interval

Figure 4.4: Errors of minimax polynomial approximations of \( f(t) = t \sin 5t \) for degrees \( K = 0, 1, 2, 3 \). Curves are labeled by the polynomial degree. Dashed lines mark the maxima and minima, highlighting that for each degree, the minimum is the negative of the maximum.

Approximations of \( f_1 = t \sin 5t \) are:

\[
\begin{align*}
p_0^*(t) &\approx -0.297492 \\
p_1^*(t) &\approx 0.402070 - 0.999803t \\
p_2^*(t) &\approx 0.0928916 + 1.40614t - 2.61739t^2 \\
p_3^*(t) &\approx -0.124060 + 3.22328t - 6.31406t^2 + 2.13186t^3
\end{align*}
\]

Their approximation errors are shown in Figure 4.4.

Comparing Figures 4.3 and 4.4 demonstrates that minimax errors can be substantially smaller than what we saw earlier with interpolation with evenly-spaced nodes and with Taylor series. Also, the error is evenly distributed over the interval. This is an essential feature of minimax approximation, as shown by the following theorem.

**Theorem 4.4 (Chebyshev equioscillation theorem [4]).** Let \( f \) be continuous on \([0, 1] \) and let \( K \geq 0 \). The minimax approximation \( p_K^* \) is unique. Denote its maximum approximation error by \( \rho_K(f) = \|f - p_K^*\|_{\infty} \). There are at least \( K + 2 \) points

\[
0 \leq t_0 < t_1 < \cdots < t_{K+1} \leq 1
\]

for which

\[
f(t_k) - p_K^*(t_k) = \sigma(-1)^k \rho_K(f), \quad k = 0, 1, \ldots, K + 1,
\]

where \( \sigma = \pm 1 \), independent of \( k \).

The dotted lines in Figure 4.4 highlight that the approximation error lies between “rails” at \( \pm \rho_K(f) \) and hits the rails at least \( K + 2 \) times. To not hit the rails would “waste” some of the margin for error. The proof of Theorem 4.4 is beyond the scope of the book.
Least-Squares Polynomial Approximation

If instead of looking at error pointwise, we measure error by $L^2$ norm, finding optimal approximations becomes much easier because we get to use the Hilbert space tools developed in Chapter 1. An optimal approximation in this context is called a least-squares approximation.

The set of polynomials of degree $\leq K$ is a subspace of $L^2([0,1])$. The desired least-squares approximation of $f$ is the orthogonal projection of $f$ to this subspace. It can be determined easily if we have an orthonormal basis $\{\varphi_0, \varphi_1, \ldots, \varphi_K\}$ for the subspace. In this case, the least-squares approximation is $\sum_{k=0}^K \langle f, \varphi_k \rangle \varphi_k$. The triviality of this result is because we developed the general framework in Chapter 1!

Orthonormal polynomials become uniquely determined (up to sign) if we require each $\varphi_k$ to have degree $k$. It is standard to consider the interval $[-1,1]$, and the resulting (unnormalized) polynomials are the Legendre polynomials

$$L_k(t) = \frac{(-1)^k}{2^k k!} \frac{d^k}{dt^k}((1-t^2)^k), \quad k \geq 0.$$ 

An orthonormal set can be obtained by dividing by the norms $\|L_k\| = \sqrt{2/(2k+1)}$. The first few Legendre polynomials are $1, t, \frac{1}{2}(3t^2 - 1), \frac{1}{2}(5t^3 - 3t), \frac{1}{8}(35t^4 - 30t^2 + 3)$.

Example 4.4 (Least-squares approximation). To continue the previous examples, we need orthogonal polynomials in $L^2([0,1])$. We can obtain these by shifting and scaling the Legendre polynomials. The first few, in normalized form, are $\varphi_0(t) = 1, \varphi_1(t) = \sqrt{3}(2t-1), \varphi_2(t) = \sqrt{5}(6t^2 - 6t + 1)$, and $\varphi_3(t) = \sqrt{7}(20t^3 - 30t^2 + 12t - 1)$. The best constant approximation is $q_0 = \langle f, \varphi_0 \rangle \varphi_0$, and higher-degree least-squares approximation can be found through $q_k = q_{k-1} + \langle f, \varphi_k \rangle \varphi_k$. The least-squares approximations of $f(t) = t \sin 5t$ up to degree 3 are

$$q_0(t) \approx -0.09508940809$$
$$q_1(t) \approx 0.4890890728 - 1.168356962t$$
$$q_2(t) \approx -0.1085812076 + 2.417664720t - 3.586021682t^2$$
$$q_3(t) \approx -0.2035086898 + 3.556794507t - 6.433846149t^2 + 1.898549645t^3$$

The resulting approximation errors are shown in Figure 4.5.

By inspection of Figures 4.4 and 4.5, we can verify that the $L^2$ and $L^\infty$ criteria give lead to different optimal approximations. As expected, the $L^\infty$ criterion minimizes the maximum pointwise error and the $L^2$ criterion minimizes the energy of the error.

---

34 Other orthogonal polynomials are obtained by changing the interval of interest and changing the inner product to include a nonnegative weight factor. See, e.g., ??.
4.3. Functions on the Real Line

Figure 4.5: Errors of least-squares polynomial approximations of $f(t) = t \sin 5t$ for degrees $K = 0, 1, 2, 3$. Curves are labeled by the polynomial degree.

4.2.2 Approximation by Trigonometric Polynomials

4.3 Functions on the Real Line

4.3.1 Basic Idea of Sampling

4.3.2 Best Approximation in Shift-Invariant Spaces

4.3.3 Interpolation

4.4 Functions on the Real Line Revisited: Splines

4.5 Approximations with Finite Precision

4.6 Approximating Random Processes

4.7 Algorithms

Chapter at a Glance

Historical Remarks

[4, 37, 55]

Further Reading

Exercises with Solutions

4.1. TBD.
Exercises

4.1. TBD.
Part II

Fourier and Wavelet Representations
Chapter 5

Time, Frequency, Scale and Resolution

Contents

5.1 Introduction ................................................. 186
5.2 Time and Frequency Localization ...................... 187
5.3 Heisenberg Boxes and the Uncertainly Principle .... 189
5.4 Scale and Scaling ........................................... 194
5.5 Resolution, Bandwidth and Degrees of Freedom ... 196
5.6 An Efficient Orthonormal Basis with Time-Frequency Structure ........................................... 197
5.7 Case Studies on Real Signals .............................. 201
Chapter at a Glance ................................................. 202
Historical Remarks ............................................. 203
Further Reading .................................................. 203
Exercises with Solutions ....................................... 203
Exercises ............................................................ 206

Over the next several chapters, we embark on the construction of various representations, both bases and frames, for signal analysis and synthesis. To create representations

\[ x = \sum_k \alpha_k \varphi_k, \]

we need \( \{ \varphi_k \} \) to “fill the space” (completeness), and sometimes for \( \{ \varphi_k \} \) to be an orthonormal set. However, these properties are not enough for a representation to be useful. For most applications, the utility of a representation is tied to time, frequency, scale, and resolution properties of the \( \varphi_k \)s—the topic of this chapter.

Time and frequency properties are perhaps the most intuitive. Think of a basis function as a note on a musical score. It has a given
frequency (for example, middle A has the frequency of 440Hz) and start time, and its type (♩, ♧, ♩) indicates its duration. We can think of the musical score as a time-frequency plane\(^{35}\) and notes as rectangles in that plane with horizontal extent determined by start and end times and vertical extent related in some way to frequency. Mapping a musical score to the time-frequency plane is far from the artistic view usually associated with music, but it helps visualize the notion of the time-frequency plane.

5.1 Introduction

Time and frequency views of a signal are intertwined in several ways. The duality of the Fourier transform gives a precise sense of interchangeability, since, if \( x(t) \) has the transform \( X(\omega) \), \( X(t) \) has the transform \( 2\pi x(-\omega) \). More relevant to this chapter, the uncertainty principle we introduce shortly, determines the trade-off between concentration in one domain and spread in the other; signals concentrated in time will be spread in frequency, and, by duality, signals concentrated in frequency will be spread in time. It also bounds the product of spreads in time and frequency, with the lower bound reached by Gaussian functions. This cornerstone result of Fourier theory is due to Heisenberg in physics and associated with Gabor in signal theory (see Historical Remarks).

Another natural notion for signals is scale. For example, given a portrait of a person, recognizing that person should not depend on whether they occupy one-tenth or one-half of the image.\(^{36}\) Thus, image recognition should be scale invariant. Signals that are scales of each other are often considered as equivalent. However, this scale invariance is a purely continuous-time property, since discrete-time sequences cannot be rescaled easily. In particular, while continuous-time rescaling can be undone easily, discrete-time rescaling, such as downsampling by a factor of 2, cannot be undone in general.

A final important notion we discuss is that of resolution. Intuitively, a blurred photograph does not have the resolution of a sharp one, even when the two prints are of the same physical size. Thus, resolution is related to bandwidth of a signal, or, more generally, to the number of degrees of freedom per unit time (or space). Clearly, scale and resolution interact; this will be most obvious in discrete time.

Chapter Outline

The purpose of the present chapter is to explore these four basic notions in detail, as well as their interactions. Section 5.2 discusses localization in time and frequency domains, while Section 5.3 is devoted to the uncertainty principle that links these two domains. Sections 5.4 and 5.5 talk about scale and resolution, as well as their interaction. In Section 5.6 we show how to build the first ONB with simple time and frequency localization requirements. Finally, Section 5.7 gives illustrative examples ranging from music to images and communications.

\(^{35}\)Note that the frequency axis is logarithmic rather than linear.

\(^{36}\)This is true within some bounds, linked to resolution; see below.
5.2. Time and Frequency Localization

Time Spread

Consider a function \( x(t) \) or a sequence \( x_n \), where \( t \) or \( n \) is a time index. We now discuss localization of the function or sequence in time. The easiest case is when the support of the signal is finite, that is, \( x(t) \) is nonzero only in \([T_1, T_2]\), or, \( x_n \) is nonzero only for \( \{N_1, N_1 + 1, \ldots, N_2\} \), often called compact support. If a function (sequence) is of compact support, then its Fourier transform cannot be of compact support (it can only have isolated zeros). That is, a function (sequence) cannot be perfectly localized in both time and frequency. This fundamental property of the Fourier transform is explored in Exercise 5.1 for the DFT.

If not of compact support, a signal might decay rapidly as \( t \) or \( n \) go to \( \pm\infty \). Such decay is necessary for working in \( L^2(\mathbb{R}) \) or \( \ell^2(\mathbb{Z}) \); for example, for finite energy (\( L^2(\mathbb{R}) \)), a function must decay faster than \( |t|^{-1/2} \) for large \( t \) (see Chapter 3).

A concise way to describe locality (or lack thereof), is to introduce a spreading measure akin to standard deviation, which requires normalization so that \( |x(t)|^2 \) can be interpreted as a probability density function.\(^{37}\) Its mean is then the time center of the function and its standard deviation is the time spread. Denote the energy of \( x(t) \) by

\[
E_x = \|x\|^2 = \int_{-\infty}^{\infty} |x(t)|^2 \, dt.
\]

Then define the time center \( \mu_t \) as

\[
\mu_t = \frac{1}{E_x} \int_{-\infty}^{\infty} t |x(t)|^2 \, dt,
\]

and the time spread \( \Delta^2_t \) as

\[
\Delta^2_t = \frac{1}{E_x} \int_{-\infty}^{\infty} (t - \mu_t)^2 |x(t)|^2 \, dt.
\]

**Example 5.1 (Time spreads).** Consider the following signals and their time spreads:

(i) The box function

\[
b(t) = \begin{cases} 
1, & -1/2 < t \leq 1/2; \\
0, & \text{otherwise};
\end{cases}
\]

has \( \mu_t = 0 \) and \( \Delta^2_t = 1/12 \).

(ii) For the sinc function from (1.72),

\[
x(t) = \frac{1}{\sqrt{\pi}}(t) \text{sinc}_\pi = \frac{1}{\sqrt{\pi}} \frac{\sin t}{t},
\]

\( |x(t)|^2 \) decays only as \( 1/|t|^2 \), so \( \Delta^2_t \) is infinite.

\(^{37}\)Note that this normalization is precisely the same as restricting attention to unit-energy functions.
(iii) The Gaussian function as in (1.76) with \( \gamma = (2\alpha/\pi)^{1/4} \) is of unit energy as we have seen in (1.78). Then, the version centered at \( t = 0 \) is of the form:

\[
g(t) = \left( \frac{2\alpha}{\pi} \right)^{1/4} e^{-\alpha t^2}, \tag{5.4}
\]

and has \( \mu_t = 0 \) and \( \Delta^2_t = 1 \).

From the above example, we see that the time spread can vary widely; the box function had the narrowest one, while the sinc function had an infinite spread. In particular, it can be unbounded, even for functions which are widely used, such as the sinc function. Exercise 5.2 explores functions based on the box function and convolutions thereof.

We restrict our definition of the time spread to continuous-time functions for now. While it can be extended to sequences, it will not be done here since its frequency-domain equivalent (where functions are periodic) is not easy to generalize. Solved Exercise 5.1 explores some of the properties of the time spread \( \Delta_t \): (a) it is invariant to time shifts \( x(t - t_0) \); (b) it is invariant to modulations by an exponential \( e^{j\omega_0 t} x(t) \); (c) energy-conserving scaling by \( s \), \( 1/\sqrt{s} x(t/s) \), increases \( \Delta_t \) by \( s \).

**Frequency Spread**

We now discuss the dual concept to time localization of \( x(t) \)—frequency localization of its DTFT pair \( X(\omega) \). Again, the simplest case is when \( X(\omega) \) is of compact support, or, \( |X(\omega)| = 0 \) for \( \omega \notin [-\Omega, \Omega] \). This is the notion of bandlimitidness used in the sampling theorem in Chapter 4. Since \( X(\omega) \) is of compact support, its time-domain counterpart will have infinite support.

Even if not compactly supported, the Fourier transform can have fast decay, and thus the notion of a frequency spread. Similarly to our discussion of the time spread, we normalize the frequency spread, so as to be able to interpret \( |X(\omega)|^2 \) as a probability density function. Denote the energy of \( X(\omega) \) by

\[
E_\omega = ||x||^2 = \int_{-\infty}^{\infty} |X(\omega)|^2 d\omega.
\]

The frequency center \( \mu_\omega \) is then

\[
\mu_\omega = \frac{1}{2\pi E_\omega} \int_{-\infty}^{\infty} \omega |X(\omega)|^2 d\omega, \tag{5.5}
\]

and the frequency spread \( \Delta_\omega \) is

\[
\Delta^2_\omega = \frac{1}{2\pi E_\omega} \int_{-\infty}^{\infty} (\omega - \mu_\omega)^2 |X(\omega)|^2 d\omega. \tag{5.6}
\]

Because of the 2\( \pi \)-periodicity of the Fourier transform \( X(e^{j\omega}) \), there is no natural definition of \( \Delta^2_\omega \) for sequences. Thus, we continue to focus on continuous-time functions for now.
5.3. Heisenberg Boxes and the Uncertainty Principle

**Figure 5.1:** The time-frequency plane and a Heisenberg box. (a) A time-domain function \( x(t) \). (b) Its Fourier transform magnitude \( |X(\omega)| \). (c) A Heisenberg box centered at \((\mu_t, \mu_\omega)\) and size \((\Delta_t, \Delta_\omega)\).

**Figure 5.2:** A time and/or frequency shift of a function, simply shifts the Heisenberg box (time-frequency tile) in time and/or frequency.

**Example 5.2.** Consider a bandlimited Fourier transform given by

\[
X(\omega) = \begin{cases} \sqrt{s}, & |\omega| < \frac{\pi}{s} \\ 0, & \text{otherwise} \end{cases}
\]

Then

\[
\Delta_\omega^2 = \frac{1}{2\pi} \int_{-\pi/s}^{\pi/s} s\omega^2 \, d\omega = \frac{\pi^2}{3s^2}.
\]

The corresponding time-domain function is a sinc, for which we have seen that \( \Delta_t^2 \) is unbounded.

Solved Exercise 5.1 explores some of the properties of the frequency spread \( \Delta_\omega \): (a) it is invariant to time shifts \( x(t - t_0) \); (b) it is invariant to modulations by an exponential \( e^{j\omega_0 t}x(t) \) (but not to modulations by a real function such as a cosine); (c) energy-conserving scaling by \( s \), \( 1/\sqrt{s}x(t/s) \), decreases \( \Delta_\omega \) by a factor \( s \). This is exactly the inverse of the effect on the time spread, as is to be expected from the scaling property of the Fourier transform.

## 5.3 Heisenberg Boxes and the Uncertainty Principle

Given a function \( x(t) \) and its Fourier transform \( X(\omega) \), we have the 4-tuple \((\mu_t, \Delta_t, \mu_\omega, \Delta_\omega)\) describing the function’s center in time and frequency \((\mu_t, \mu_\omega)\) and the function’s spread in time and frequency \((\Delta_t, \Delta_\omega)\). It is convenient to show this pictorially (see Figure 5.1), as it conveys the idea that there is a center of mass \((\mu_t, \mu_\omega)\) and a spread \((\Delta_t, \Delta_\omega)\) shown by a rectangular box with appropriate location and size. The plane on which this is drawn is called the time-frequency plane, and the box is usually called a Heisenberg box, or a time-frequency tile.

From our previous discussion on time shifting and complex modulation, we know that a function \( y(t) \) obtained by a shift and modulation of \( x(t) \),

\[
y(t) = e^{j\omega_0 t}x(t - t_0), \quad (5.7)
\]

**Figure 5.3:** The effect of scaling (5.8) on the associated Heisenberg box, for \( s = 2 \).
will simply have a Heisenberg box shifted by \( t_0 \) and \( \omega_0 \), as depicted in Figure 5.2.

If we rescale a function by a factor \( s \),

\[
y(t) = \frac{1}{\sqrt{s}} x\left(\frac{t}{s}\right).
\]

(5.8)

If \( x(t) \) has a Heisenberg box specified by \((\mu_t, \Delta_t, \mu_\omega, \Delta_\omega)\), then the box for \( y(t) \) is specified by \((s\mu_t, \mu_\omega/s, s\Delta_t, \Delta_\omega/s)\), as shown in Figure 5.3. The effects of shift, modulation and scaling on the Heisenberg boxes of the resulting functions are summarized in Table 5.1.

<table>
<thead>
<tr>
<th>Function</th>
<th>Time Center</th>
<th>Time Spread</th>
<th>Freq. Center</th>
<th>Freq. Spread</th>
</tr>
</thead>
<tbody>
<tr>
<td>( x(t) )</td>
<td>( \mu_t )</td>
<td>( \Delta_t )</td>
<td>( \mu_\omega )</td>
<td>( \Delta_\omega )</td>
</tr>
<tr>
<td>( x(t - t_0) )</td>
<td>( \mu_t + t_0 )</td>
<td>( \Delta_t )</td>
<td>( \mu_\omega )</td>
<td>( \Delta_\omega )</td>
</tr>
<tr>
<td>( e^{i\omega_0 t} x(t) )</td>
<td>( \mu_t )</td>
<td>( \Delta_t )</td>
<td>( \mu_\omega + \omega_0 )</td>
<td>( \Delta_\omega )</td>
</tr>
<tr>
<td>( \frac{1}{\sqrt{s}} x\left(\frac{t}{s}\right) )</td>
<td>( s \mu_t )</td>
<td>( s \Delta_t )</td>
<td>( \frac{1}{s} \mu_\omega )</td>
<td>( \frac{1}{s} \Delta_\omega )</td>
</tr>
</tbody>
</table>

Table 5.1: Effect of a shift, modulation and scaling on a Heisenberg box \((\mu_t, \Delta_t, \mu_\omega, \Delta_\omega)\).

So far, we have considered the effect of shifting in time and frequency as well as scaling on a Heisenberg box. What about the effect on the area of the Heisenberg box? The intuition, corroborated by what we saw with scaling, is that one can trade time spread for frequency spread. Moreover, from Examples 5.1 and 5.2, we know that a function that is narrow in one domain will be broad in the other. It is thus intuitive that the size of the Heisenberg box is lower bounded, so that no function can be arbitrarily narrow in both time and frequency. While the result is often called the Heisenberg uncertainty principle, it has been shown independently by a number of others, including Gabor (see Historical Remarks).

**Theorem 5.1 (Uncertainty Principle).** Given a function \( x \in L^2(\mathbb{R}) \), the product of its squared time and frequency spreads is lower bounded as

\[
\Delta_t^2 \Delta_\omega^2 \geq \frac{1}{4}.
\]

(5.9)

The lower bound is attained by Gaussian functions from (1.76):

\[
x(t) = \gamma e^{-\alpha t^2}, \quad \alpha > 0.
\]

(5.10)

**Proof.** We prove the theorem for real functions; see Exercise 5.3 for the complex case. Without loss of generality, assume that \( x(t) \) is centered at \( t = 0 \), and that \( x(t) \) has unit energy; otherwise we may shift and scale it appropriately. Since \( x(t) \) is real, it is also centered at \( \omega = 0 \), so \( \mu_t = \mu_\omega = 0 \).

Suppose \( x(t) \) has a bounded derivative \( x'(t) \); if not, \( \Delta_\omega^2 = \infty \) so the statement holds trivially. Consider the function \( t x(t) x'(t) \) and its integral. Using the Cauchy-
5.3. Heisenberg Boxes and the Uncertainty Principle

Schwarz inequality (1.20), we can write

\[
\left| \int_{-\infty}^{\infty} tx(t)x'(t) \, dt \right|^2 \leq \int_{-\infty}^{\infty} |tx(t)|^2 \, dt \int_{-\infty}^{\infty} |x'(t)|^2 \, dt
\]

\[
\Delta_t^2 \leq \frac{1}{2\pi} \int_{-\infty}^{\infty} |j\omega X(\omega)|^2 \, d\omega = \Delta_\omega^2 \Delta_t^2, \quad (5.11)
\]

where (a) follows from Parseval’s equality (1.45) and the fact that 
\( x'(t) \) has Fourier transform \( j\omega X(\omega) \). We now simplify the left side:

\[
\int_{-\infty}^{\infty} tx(t)x'(t) \, dt \overset{(a)}{=} \frac{1}{2} \int_{-\infty}^{\infty} t \frac{dx^2(\tau)}{d\tau} \, d\tau \overset{(b)}{=} \frac{1}{2} t x^2(t)|_{-\infty}^{\infty} - \frac{1}{2} \int_{-\infty}^{\infty} x^2(t) \, dt \overset{(c)}{=} -\frac{1}{2},
\]

where (a) follows from \( (x^2(t))' = 2x'(t)x(t) \); (b) is due to integration by parts; and (c) holds because \( x(t) \in L^2(\mathbb{R}) \) implies that it decays faster than \( 1/\sqrt{|t|} \) for \( t \to \pm\infty \), and thus \( \lim_{t \to \pm\infty} tx^2(t) = 0 \) (see Chapter 3). Substituting this into (5.11) yields (5.9).

To find functions that meet the bound with equality, recall that Cauchy-Schwarz inequality becomes an equality if and only if the two functions are collinear (scalar multiples of each other), or at least one of them is 0. In our case this means \( x'(t) = \beta t x(t) \). Functions satisfying this relation have the form \( x(t) = \gamma e^{\beta t^2/2} = \gamma e^{-\alpha t^2} \) —the Gaussian functions.

The uncertainty principle points to a fundamental limitation in time-frequency analysis. If we desire to analyze a signal with a “probing function” to extract information about the signal around a location \((\mu_t, \mu_\omega)\), the probing function will necessarily be imprecise. That is, if we desire very precise frequency information about the signal, its time location will be uncertain, and vice versa.

\textbf{Example 5.3 (Sinusoid plus Dirac impulse).} A generic example to illustrate the tension between time and frequency localization is the analysis of a signal containing a Dirac impulse in time and a Dirac impulse in frequency (complex sinusoidal/exponential):

\[
x(t) = \delta(t - \tau) + e^{j\Omega t},
\]

with the Fourier transform

\[
X(\omega) = e^{-j\omega \tau} + 2\pi \delta(\omega - \Omega).
\]

Clearly, to locate the Dirac impulse in time or in frequency, one needs to be as sharp as possible in that particular domain, thus compromising the sharpness in the other domain. We illustrate this with a numerical example which, while simple, conveys this basic, but fundamental, trade-off.

Consider a periodic sequence with period \( N = 256 \), containing a complex sinusoidal sequence and a Dirac impulse, as shown in Figure 5.4(a). In Figure 5.4(b),
Figure 5.4: Time-frequency resolution trade-off for a signal containing a Dirac impulse in time and a complex sinusoid (Dirac impulse in frequency). (a) Original signal of length 256. (b) Magnitude of size-256 DFT. (c) Same as (b), but with signal split into 16 pieces of length 16 each yielding a size-16 DFT. (d) Magnitude of samples.

we show the magnitude of the size-256 DFT of one period, which has 256 frequency bins, and perfectly identifies the exponential component, while missing the time-domain Dirac impulse completely. In order to increase the time resolution, we divide the period into 16 pieces of length 16, taking the size-16 DFT of each, as shown in Figure 5.4(c). Now we can identify approximately where the time-domain Dirac impulse occurs, but in turn, the frequency resolution has been reduced, since we now have only 16 frequency bins. Finally, Figure 5.4(d) shows the dual case to Figure 5.4(b), that is, we plot the magnitude of each sample over time. The Dirac impulse is now perfectly visible, while the sinusoid is just a background wave, with its frequency not easily identified.

Example 5.4 (Chirp signal). As another example of time-frequency analysis and the trade-off between time and frequency sharpness, we consider a signal consisting of a windowed chirp (complex exponential with a rising frequency). Instead of a fixed frequency $\omega_0$, the “local frequency” is linearly growing with time: $\omega_0 t$. The signal is

$$x(t) = w(t) e^{j\omega_0 t^2},$$

where $w(t)$ is an appropriate window function. Such a chirp signal is not as esoteric as it may look; bats use such signals to hunt for bugs. An example is given in Figure 5.5(a), and an idealized time-frequency analysis is sketched in Figure 5.5(b).

As analyzing functions, we choose windowed complex exponentials (but with a fixed frequency). The choice we have is the size of the window (assuming the analyzing function will cover all shifts and modulations of interest). A short window allows for sharp frequency analysis, however, no frequency is really present other than at one instant! A short window will do justice to the transient nature of “frequency” in the chirp, but will only give a very approximate frequency analysis due to the uncertainty principle. A compromise between time and frequency sharpness must be sought, and one such possible analysis is shown in Figure 5.6.

While the uncertainty principle uses a spreading measure akin to variance, other measures can be defined. Though they typically lack fundamental bounds of the kind given by the uncertainty principle (5.9), they can be quite useful as well as intuitive. One such measure, easily applicable to functions that are symmetric in both time and frequency, finds the centered intervals containing a given percentage $\alpha$ of the energy in time and frequency (where $\alpha$ is typically 0.90 or 0.95). For a function $x(t)$ of unit norm, $\|x\| = 1$, and symmetric around $\mu_t$, the time spread
5.3. Heisenberg Boxes and the Uncertainty Principle

Figure 5.5: Chirp signal and time-frequency analysis with a linearly increasing frequency. (a) An example of a windowed chirp with a linearly increasing frequency. Real and imaginary parts are shown as well as the raised cosine window. (b) Idealized time-frequency analysis.

Figure 5.6: Analysis of a chirp signal. (a) One of the analyzing functions, consisting of a (complex) modulated window. (b) Magnitude squared of the inner products between the chirp and various shifts and modulates of the analyzing functions, showing a blurred version of the chirp.

\[ \hat{\Delta}_t^{(\alpha)} \] is now defined such that
\[ \int_{\mu_t - \frac{1}{2} \hat{\Delta}_t^{(\alpha)}}^{\mu_t + \frac{1}{2} \hat{\Delta}_t^{(\alpha)}} |x(t)|^2 \, dt = \alpha. \] \hspace{1cm} (5.12)

Similarly, the frequency spread \( \hat{\Delta}_\omega^{(\alpha)} \) around the point of symmetry \( \mu_\omega \) is defined such that
\[ \frac{1}{2\pi} \int_{\mu_\omega - \frac{1}{2} \hat{\Delta}_\omega^{(\alpha)}}^{\mu_\omega + \frac{1}{2} \hat{\Delta}_\omega^{(\alpha)}} |X(\omega)|^2 \, d\omega = \alpha. \] \hspace{1cm} (5.13)

Solved Exercise 5.1(c) shows that \( \hat{\Delta}_t^{(\alpha)} \) and \( \hat{\Delta}_\omega^{(\alpha)} \) satisfy the same shift, modulation and scaling behavior as \( \Delta_t \) and \( \Delta_\omega \) in Table 5.1.

Uncertainty Principle for Discrete Time

Thus far, we have restricted our attention to the study of localization properties for (continuous-time) functions. Analogous results for (discrete-time) sequences are not as elegant, except in the case of strictly lowpass sequences.

For \( x \in \ell^2(\mathbb{Z}) \) with energy \( E_x = \sum_n |x_n|^2 \), define the time center \( \mu_n \) and time spread \( \Delta_n \) as
\[ \mu_n = \frac{1}{E_x} \sum_{n \in \mathbb{Z}} n |x_n|^2 \quad \text{and} \quad \Delta_n^2 = \frac{1}{E_x} \sum_{n \in \mathbb{Z}} (n - \mu_n)^2 |x_n|^2. \]

Define the frequency center \( \mu_\omega \) and frequency spread \( \Delta_\omega \) as
\[ \mu_\omega = \frac{1}{2\pi E_x} \int_{-\pi}^{\pi} \omega |X(e^{j\omega})|^2 \, d\omega \quad \text{and} \quad \Delta_\omega^2 = \frac{1}{2\pi E_x} \int_{-\pi}^{\pi} (\omega - \mu_\omega)^2 |X(e^{j\omega})|^2 \, d\omega. \]

Because of the interval of integration \([0, \pi]\), the value of \( \mu_\omega \) may not match the intuitive notion of frequency center of the sequence. For example, while a sequence supported in Fourier domain on \( \bigcup_{k \in \mathbb{Z}} [0.9\pi + k2\pi, 1.1\pi + k2\pi] \) seems to have its frequency center near \( \pi \), \( \mu_\omega \) may be far from \( \pi \).
With these definitions paralleling those for continuous-time functions, we can obtain a result very similar to Theorem 5.1. One could imagine that it follows from combining Theorem 5.1 with Nyquist-rate sampling of a bandlimited function. The proof is shown in Solved Exercise 5.2 and suggests the use of the Cauchy-Schwarz inequality similarly to our earlier proof.

**Theorem 5.2 (Discrete-time uncertainty principle).** Given a sequence \( x \in \ell^2(\mathbb{Z}) \) with \( X(e^{j\pi}) = 0 \), the product of its squared time and frequency spreads is lower bounded as

\[
\Delta_n^2 \Delta_\omega^2 > \frac{1}{4}.
\]

In addition to the above uncertainty principle for infinite sequences, there is a simple and powerful uncertainty principle for finite-dimensional sequences and their DFTs. This is explored in Exercise 5.6 and arises again in Chapter (TBD).

### 5.4 Scale and Scaling

In the previous section, we introduced the idea of signal analysis as computing an inner product with a “probing function.” Along with the Heisenberg box 4-tuple, another key property of a probing function is its scale. Scale is closely related to the time spread, but it is inherently a relative (rather than absolute) quantity. Before further describing scale, let us revisit scaling, especially to point out the fundamental difference between continuous- and discrete-time scaling operations.

An energy-conserving rescaling of \( x(t) \) by a factor \( s \in \mathbb{R}^+ \) yields

\[
y(t) = \sqrt{s} x(st) \quad \xrightarrow{\text{FT}} \quad Y(\omega) = \frac{1}{\sqrt{s}} X\left(\frac{\omega}{s}\right).
\]

Clearly, this is a reversible process, since rescaling \( y(t) \) by \((1/s)\) gives

\[
\frac{1}{\sqrt{s}} y\left(\frac{t}{s}\right) = x(t).
\]

The situation is more complicated in discrete time, where we need multirate signal processing tools introduced in Section 2.5. Given a sequence \( x_n \), a “stretching” by an integer factor \( N \) can be achieved by upsampling by \( N \) as in (2.134). This can be undone by downsampling by \( N \) as in (2.126) (see Figure 5.7).

If instead of stretching, we want to “contract” a sequence by an integer factor \( N \), we can do this by downsampling by \( N \). However, such an operation cannot be undone, as \((N-1)\) samples out of every \( N \) have been lost, and are replaced by zeros during upsampling (see an example for \( N = 2 \) in Figure 5.8).

Thus, scale changes are more complicated in discrete time. In particular, compressing the time axis cannot be undone since samples are lost in the process. Scale changes by rational factors are possible through combinations of integer upsampling and downsampling, but cannot be undone in general (see Exercise 5.4).

Let us go back to the notion of scale, and think of a familiar case where scale plays a key role—maps. The usual notion of scale in maps is the following:
5.4. Scale and Scaling

Figure 5.7: Stretching of a sequence by a factor 2, followed by contraction. This is achieved with upsampling by 2, followed by downsampling by 2, recovering the original. (a) Original sequence $x_n$. (b) Upsampled sequence $y_n$. (c) Downsampled sequence $z_n$.

Figure 5.8: Contraction of a sequence by a factor 2, followed by stretching, using downsampling and upsampling. The result equals the original only at even indices, $z_{2n} = x_{2n}$, and is zero elsewhere. (a) Original sequence $x_n$. (b) Downsampled sequence $y_n$. (c) Upsampled sequence $z_n$.

in a map at scale 1:100,000, an object of length 1 km is represented by a length of $(10^3 \text{m})/10^5 = 1\text{cm}$. That is, the scale factor $s = 10^5$ is used as a contraction factor, to map a reality $x(t)$ into a scaled version $y(t) = \sqrt{s}x(st)$ (with the energy normalization factor $\sqrt{s}$ of no realistic significance because reality and a map are not of the same dimension). However, reality does provide us with something important: a baseline scale against which to compare the map.

When we look at functions in $L^2(\mathbb{R})$, a baseline scale does not necessarily exist. When $y(t) = \sqrt{s}x(st)$, we say that $y$ is at a larger scale if $s > 1$, and at a smaller scale if $s \in (0,1)$. There is no absolute scale for $y$ unless we arbitrarily define a scale for $x$. We indeed do this sometimes, as we will see in Chapter (TBD).

Now consider the use of a real probing function $\varphi(t)$ to extract some information about $x(t)$. If we compute the inner product between the probing function and a scaled function, we get

$$\langle \sqrt{s}x(st), \varphi(t) \rangle = \sqrt{s} \int x(st)\varphi(t) \, dt = \frac{1}{\sqrt{s}} \int x(\tau)\varphi(\frac{\tau}{s}) \, d\tau$$

$$= \left\langle x(t), \frac{1}{\sqrt{s}} \varphi(t/s) \right\rangle. \quad (5.16)$$
Figure 5.9: Aerial photographs of the EPF Lausanne campus at various scales. (a) 5,000. (b) 10,000. (c) 20,000.

Figure 5.10: Signals with features at different scales require probing functions adapted to the scales of those features. (a) A wide-area feature requires a wide probing function. (b) A sharp feature requires a sharp probing function.

Probing a contracted function is equivalent to stretching the probe, thus emphasizing that scale is relative. If only stretched and contracted versions of a single probe are available, large-scale features in \( x(t) \) are seen using stretched probing functions, while small-scale features (fine details) in \( x(t) \) are seen using contracted probing functions.

In summary, large scales \( s \gg 1 \) correspond to contracted versions of reality, or to widely spread probing functions. This duality is inherent in the inner product (5.16). Figure 5.9 shows an aerial photograph with different scale factors as per our convention, while Figure 5.10 shows the interaction of signals with various size features and probing functions.

5.5 Resolution, Bandwidth and Degrees of Freedom

The notion of resolution is intuitive for images. If we compare two photographs of the same size depicting the same reality, one sharp and the other blurry, we say that the former has higher resolution than the latter. While this intuition is related to a notion of bandwidth, it is not the only interpretation.

A more universal notion is to define resolution as the number of degrees of freedom per unit time (or unit space for images) for a set of signals. Classical bandwidth is then proportional to resolution. Consider the set of functions \( x(t) \) with spectra \( X(\omega) \) supported on the interval \([-\Omega, \Omega]\). Then, the sampling theorem we have seen in Chapter 4 states that samples taken every \( T = \pi/\Omega \) sec, or \( x_n = x(nT) \), \( n \in \mathbb{Z} \), uniquely specify \( x(t) \). In other words, real functions of bandwidth \( 2\Omega \) have \( \Omega/\pi \) real degrees of freedom per unit time.

As an example of a set of functions that, while not bandlimited, do have a finite number of degrees of freedom per unit time, consider piecewise constant functions over unit intervals:

\[
x(t) = x_{[t]} = x_n, \quad n \leq t < n + 1, \quad n \in \mathbb{Z}.
\]  

(5.17)

Clearly, \( x(t) \) has 1 degree of freedom per unit time, but an unbounded spectrum since it is discontinuous at every integer. This function is part of a general class of functions belonging to the so-called shift-invariant subspaces we studied in Chapter 4 (see also Exercise 5.5).

Scaling affects resolution, and as in the previous section there is a difference between discrete and continuous time. For sequences, we start with a reference
5.6. An Efficient Orthonormal Basis with Time-Frequency Structure

Figure 5.11: Interplay of scale and resolution for continuous- and discrete-time signals. We assume the original signal is at scale \( s = 1 \) and of resolution \( \tau = 1 \), and indicate the resulting scale and resolution of the output by \( s' \) and \( \tau' \). (a) Filtering of signal. (b) Downsampling by 2. (c) Upsampling by 2. (d) Downsampling followed by upsampling. (e) Filtering and downsampling. (f) Upsampling and filtering.

sequence space \( S \) in which there are no fixed relationships between samples and the rate of the samples is taken to be 1 per unit time. In the reference space, the resolution is 1 per unit time because each sample is a degree of freedom. Downsampling a sequence from \( S \) by \( N \) leads to a sequence having a resolution of \( 1/N \) per unit time. Higher-resolution sequences can be obtained by combining sequences appropriately; see Example 5.5 below. For continuous-time functions, scaling is less disruptive to the time axis, so calculating the number of degrees of freedom per unit time is not difficult. If \( x(t) \) has resolution \( \tau \) (per unit time), then \( x(\alpha t) \) has resolution \( \alpha \tau \).

Filtering can affect resolution as well. If a function of bandwidth \([-\Omega, \Omega]\) is perfectly lowpass filtered to \([-\beta \Omega, \beta \Omega]\), \( 0 < \beta < 1 \), then its resolution changes from \( \Omega/\pi \) to \( \beta \Omega/\pi \). The same holds for sequences, where an ideal lowpass filter with support \([-\beta \pi, \beta \pi]\), \( 0 < \beta < 1 \), reduces the resolution to \( \beta \) samples per unit time.

Example 5.5 (Scale and resolution). First consider the continuous-time case. Let \( x(t) \) be a bandlimited function with frequency support \([-\Omega, \Omega]\). Then \( y(t) = \sqrt{2} x(2t) \) is at twice the scale as well as resolution. Inversely, \( y(t) = \frac{1}{\sqrt{2}} x(t/2) \) is at half the scale and of half the resolution. Finally, \( y(t) = (h * x)(t) \), where \( h(t) \) is an ideal lowpass filter with support \([-\Omega/2, \Omega/2]\), has unchanged scale but half the resolution.

Now let \( x_n \) be a discrete-time sequence. The downsampled sequence \( y_n \) as in (2.121) is at twice the scale and of half the resolution with respect to \( x_n \). The upsampled sequence \( y_n \) as in (2.122) is at half the scale and of same resolution. A sequence first downsampled by 2 and then upsamled by 2 keeps all the even samples and zeros out the odd ones, is at the same scale and of half the resolution. Finally, filtering with an ideal halfband lowpass filter with frequency support \([-\pi/2, \pi/2]\) leaves the scale unchanged, but halves the resolution. Some of the above relations are depicted in Figure 5.11.

5.6 An Efficient Orthonormal Basis with Time-Frequency Structure

Now that we have discussed what kind of structure we would like to impose upon the representations we build, how do we build, say, an ONB for \( \ell^2(\mathbb{Z}) \)?

We already have a standard basis consisting of shifted Dirac impulses given in (2.8). We can also look at the IDTFT given in (2.47b) as an appropriate representation. Table 5.2 summarizes these two expansions and shows, at a glance,
the duality of the two expansions: each has perfect localization properties in one domain and no localization in the other.

<table>
<thead>
<tr>
<th>Basis</th>
<th>Dirac Basis</th>
<th>DTFT Basis</th>
</tr>
</thead>
<tbody>
<tr>
<td>Expansion</td>
<td>$x_n = \sum_{k \in \mathbb{Z}} X_k \delta_{n-k}$</td>
<td>$x_n = \frac{1}{2\pi} \int_{-\pi}^{\pi} X(e^{j\omega}) e^{j\omega n} d\omega$</td>
</tr>
<tr>
<td>Expansion coef.</td>
<td>$X_K = x_k$</td>
<td>$X(e^{j\omega}) = \sum_{n \in \mathbb{Z}} x_n e^{-j\omega n}$</td>
</tr>
<tr>
<td>Basis functions in time</td>
<td>$\varphi_k = \delta_{n-k}$</td>
<td>$\varphi_\Omega = e^{j\Omega n}$, $</td>
</tr>
<tr>
<td>Basis functions in frequency</td>
<td>$\Phi_\omega(e^{j\omega}) = 1$</td>
<td>$\Phi_\Omega(e^{j\omega}) = \delta(\omega - \Omega)$</td>
</tr>
<tr>
<td>Tiling</td>
<td>(TBD) Fig of Dirac in time</td>
<td>(TBD) TF tiling of the Dirac</td>
</tr>
<tr>
<td>Tiling</td>
<td>(TBD) Fig of DTFT in time</td>
<td>(TBD) TF tiling of the DTFT</td>
</tr>
</tbody>
</table>

**Table 5.2:** The Dirac and DTFT bases: Two extremes of time-frequency localization properties. The Dirac basis offers perfect time localization and no frequency localization, while the DTFT basis does the opposite.

While we do draw time-frequency tilings for these two representations, in discrete time, these are only used as a conceptual tool, and do not have a rigorous interpretation as in continuous time. In particular, since the time line is discrete, we choose to associate to each point in time a tile (Heisenberg box) of width 1, arbitrarily set to last from $n$ to $n+1$.

So the question now is: Can we build an ONB for $\ell^2(\mathbb{Z})$, but with some localization in both domains? For example, starting from the Dirac basis, can we improve its localization in frequency slightly while not trading too much of its time locality? Figure 5.12 illustrates the desired tiling, where each tile has been divided in two in frequency, thereby improving frequency localization. Comparing this to the Dirac basis in Table 5.2, the price we pay is slightly worse time localization, where each tile has become twice as wide.

Given this tiling, can we now find functions to produce such a tiling? Let us concentrate first on the lower left-hand tile with the basis function $\varphi_0$. We will search for the simplest $\varphi_0$ which has roughly the time spread of 2 and frequency spread of $\pi/2$ (remember that we have not defined time and frequency spreads for discrete time, we use them for illustration only). Assume we ask for $\varphi_0$ to be exactly of length 2, that is,

$$\varphi_{0,n} = \cos \theta \delta_n + \sin \theta \delta_{n-1}, \quad (5.18)$$

where we have also imposed $\|\varphi_0\| = 1$, as we want it to be a part of an ONB. Such a $\varphi_0$ would indeed have the time spread of 2. How about its frequency behavior? We are looking for $\varphi_0$ to be roughly a halfband lowpass sequence; let us ask for it to block the highest frequency $\pi$:

$$\Phi_0(e^{j\omega})|_{\omega=\pi} = \cos \theta + \sin \theta e^{-j\omega}|_{\omega=\pi} = \cos \theta - \sin \theta = 0. \quad (5.19)$$
5.6. An Efficient Orthonormal Basis with Time-Frequency Structure

Solving the above equation yields \( \theta = k\pi + \pi/4 \) and

\[
\varphi_{0,n} = \frac{1}{\sqrt{2}}(\delta_n + \delta_{n-1}) \quad \varphi_0 = \left[ \begin{array}{cccc} \ldots & 0 & \frac{1}{\sqrt{2}} & 0 & 0 & \ldots \end{array} \right]^T.
\]

We now repeat the process and try to find a \( \varphi_1 \) of length 2 being a roughly halfband highpass sequence. We can use (5.18) as a general form of a sequence of length 2 and norm 1. What we look for now is a sequence which has to be orthogonal to the first candidate basis vector \( \varphi_0 \) (if they are to be a part of an ONB), that is

\[
\langle \varphi_0, \varphi_1 \rangle = \frac{1}{\sqrt{2}} \cos \theta + \frac{1}{\sqrt{2}} \sin \theta = 0,
\]

which yields \( \theta = k\pi/2 + \pi/4 \), and one possible form of \( \varphi_1 \):

\[
\varphi_{1,n} = \frac{1}{\sqrt{2}}(\delta_n - \delta_{n-1}) \quad \varphi_1 = \left[ \begin{array}{cccc} \ldots & 0 & \frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{2}} & 0 & 0 & \ldots \end{array} \right]^T.
\]

Note that while we did not specifically impose it, the resulting sequence is indeed highpass in nature, as \( \Phi_1(e^{j\omega})\big|_{\omega=0} = 0 \).

So far so good; we only have infinitely many more functions to find. To make this task less daunting, we search for an easier way, for example, by shifting \( \varphi_0 \) and \( \varphi_1 \) along the time axis by integer multiples of 2. Call \( \varphi_{2k,n} = \varphi_{0,n-2k} \) and \( \varphi_{2k+1,n} = \varphi_{1,n-2k} \). Then indeed, the set \( \Phi = \{ \varphi_k \}_{k \in \mathbb{Z}} \) is an orthonormal set, 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);
- is efficient (it is built from two template functions and their shifts).

As we well know from Chapter 1, orthonormality of a set is not enough for that set to form an ONB; it must be complete as well. We now show that this set is
The matrix is block diagonal, with blocks of size 2 × 2.

From (5.23), we can immediately see that the Haar ONB projects onto two subspaces: lowpass space $V$, spanned by the lowpass template $\varphi_0$ and its even shifts, and the highpass space $W$, spanned by the highpass template $\varphi_0$ and its even shifts:

$$V = \text{span}(\{\varphi_{0,n-2k}\}_{k \in \mathbb{Z}}), \quad W = \text{span}(\{\varphi_{1,n-2k}\}_{k \in \mathbb{Z}}).$$

(5.25)

From (5.23), the lowpass and highpass projections are:

$$x_V = \begin{bmatrix} \ldots & \frac{1}{2}(x_0 + x_1) & \frac{1}{2}(x_0 + x_1) & \frac{1}{2}(x_2 + x_3) & \frac{1}{2}(x_2 + x_3) & \ldots \end{bmatrix}^T$$

(5.26)

and

$$x_W = \begin{bmatrix} \ldots & \frac{1}{2}(x_0 - x_1) & \frac{1}{2}(x_0 - x_1) & -\frac{1}{2}(x_2 - x_3) & -\frac{1}{2}(x_2 - x_3) & \ldots \end{bmatrix}^T$$

(5.27)
5.7. Case Studies on Real Signals

Indeed, \( x_V \) is a “smoothed” version of \( x \) where every two samples have been replaced by their average, while \( x_W \) is the “detailed” version of \( x \) where every two samples have been replaced by the their difference (and its negative). As the sequences in \( V \) and \( W \) are orthogonal, and the expansion is a basis,

\[
\ell^2(\mathbb{Z}) = V \oplus W. \tag{5.28}
\]

In the next chapter, we will show efficient ways of implementing such ONBs, using filter banks.

5.7 Case Studies on Real Signals

So far, our discussion has been mostly conceptual, and the examples synthetic. We now look at case studies using real-world signals such as music, images and communication signals. Our discussion is meant to develop intuition rather than be rigorous. We want to excite you to continue studying the rich set of tools responsible for examples below.

5.7.1 Music and Time-Frequency Analysis
5.7.2 Images and Pyramids
5.7.3 Singularities, Denoising and Superresolution
5.7.4 Channel Equalization and OFDM
Chapter at a Glance

We now summarize the main concepts and results seen in this chapter in a tabular form.

<table>
<thead>
<tr>
<th>Uncertainty Principle</th>
</tr>
</thead>
<tbody>
<tr>
<td>For a function $x(t) \in \mathcal{L}^2(\mathbb{R})$ with Fourier transform $X(\omega)$</td>
</tr>
<tr>
<td>Energy</td>
</tr>
<tr>
<td>Time center</td>
</tr>
<tr>
<td>Time spread</td>
</tr>
<tr>
<td>Frequency center</td>
</tr>
<tr>
<td>Frequency spread</td>
</tr>
</tbody>
</table>

$\Rightarrow \Delta^2_t \Delta^2_\omega \geq 1/4$, with equality achieved by a Gaussian $x(t)$.

<table>
<thead>
<tr>
<th>Scale</th>
</tr>
</thead>
<tbody>
<tr>
<td>If $x(t)$ is defined to have scale 1, then $y(t) = \sqrt{s} x(st)$ has scale $s$.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Resolution</th>
</tr>
</thead>
<tbody>
<tr>
<td>The resolution of a set of functions is the number of degrees of freedom per unit time.</td>
</tr>
</tbody>
</table>
Historical Remarks

Uncertainty principles stemming from the Cauchy-Schwarz inequality have a long and rich history. The best known one is Heisenberg’s uncertainty principle in quantum physics, first developed in a 1927 essay [84]. Werner Karl Heisenberg (1901-1976) was a German physicist, credited as a founder of quantum mechanics, for which he was awarded the Nobel Prize in 1932. He had seven children, one of whom, Martin Heisenberg, was a celebrated geneticist. He collaborated with Bohr, Pauli and Dirac, among others. While he was initially attacked by the Nazi war machine for promoting Einstein’s views, he did head the Nazi nuclear project during the war. His role in the project has been a subject of controversy every since, with differing views on whether he was deliberately stalling Hitler’s efforts or not.

Kennard is credited with the first mathematically exact formulation of the uncertainty principle, and Robertson and Schrödinger provided generalizations. The uncertainty principle presented in Theorem 5.1 was proven by Weyl and Pauli and introduced to signal processing by Dennis Gabor (1900-1979) [71], a Hungarian physicist, and another winner of the Nobel Prize for physics (he is also known as inventor of holography). By finding a lower bound to $\Delta_t \Delta_\omega$, Gabor was intending to define an information measure or capacity for signals. Shannon’s communication theory [136] proved much more fruitful for this purpose, but Gabor’s proposal of signal analysis by shifted and modulated Gaussian functions has been a cornerstone of time-frequency analysis ever since. Slepian’s survey [138] is enlightening on these topics.

Further Reading

Many of the uncertainty principles for discrete-time signals are considerably more complicated than Theorem 5.2. We have given only a result that follows papers by Ishii and Furutaka [90] and Calvez and Vilbé [27].

Donoho and Stark [60] derived new uncertainty principles in various domains. Particularly influential was an uncertainty principle for finite-dimensional signals and a demonstration of its significance for signal recovery (see Exercises 5.6 and 5.7). More recently, Donoho and Huo [59] introduced performance guarantees for $l^2$ minimization-based signal recovery algorithms; this has sparked a large body of work.

Exercises with Solutions

5.1. Properties of Time and Frequency Spreads:
Consider the time and frequency spreads as defined in (5.2) and (5.6), respectively.
(i) Show that time shifts and complex modulations of \(x(t)\) as in (5.7) leave \(\Delta_t\) and \(\Delta_\omega\) unchanged.

(ii) Show that energy conserving scaling of \(x(t)\) as in (5.8) increases \(\Delta_t\) by \(s\), while decreasing \(\Delta_\omega\) by \(s\), thus leaving the time-frequency product unchanged.

(iii) Show (i)-(ii) for the time-frequency spreads \(\Delta_t^{(\alpha)}\) and \(\Delta_\omega^{(\alpha)}\) defined in (5.12) and (5.13).

Solution:

(i) Without loss of generality assume \(\|x\|^2 = 1\). Time shift \(y(t) = x(t - t_0)\), or, \(Y(\omega) = X(\omega)e^{j\omega t_0}\) in frequency domain, changes \(\mu_t\):

\[
\mu_{t,\text{new}} = \int_{-\infty}^{+\infty} t |y(t)|^2 dt = \int_{-\infty}^{+\infty} t |x(t-t_0)|^2 dt = \int_{-\infty}^{+\infty} (\tau + t_0) |x(\tau)|^2 d\tau
\]

\[
= \int_{-\infty}^{+\infty} \tau |x(\tau)|^2 d\tau + t_0 \int_{-\infty}^{+\infty} |x(\tau)|^2 d\tau = \mu_{t,\text{old}} + t_0.
\]

where we have introduced a substitution \(\tau = t - t_0\). The time spread \(\Delta_t^2\), however, remains unchanged:

\[
\Delta_{t,\text{new}}^2 = \int_{-\infty}^{+\infty} (t - \mu_{t,\text{new}})^2 |y(t)|^2 dt = \int_{-\infty}^{+\infty} (t - \mu_{t,\text{old}} - t_0)^2 |x(t-t_0)|^2 dt
\]

\[
= \int_{-\infty}^{+\infty} (\tau - \mu_{t,\text{old}})^2 |x(\tau)|^2 d\tau = \Delta_{t,\text{old}}^2.
\]

where we again have introduced a substitution \(\tau = t - t_0\). Since \(|Y(\omega)|^2 = |X(\omega)|^2\), the frequency spread is clearly not changed by time shift.

Because of duality of time and frequency, complex modulation (a shift in frequency domain) changes \(\mu_\omega\) and does not change the frequency spread. Let us write \(y(t) = x(t) e^{-j\omega_0 t}\), or, \(Y(\omega) = X(\omega - \omega_0)\) in frequency domain, yields:

\[
2\pi \mu_{\omega,\text{new}} = \int_{-\infty}^{+\infty} \omega |Y(\omega)|^2 d\omega = \int_{-\infty}^{+\infty} \omega |X(\omega - \omega_0)|^2 d\omega
\]

\[
= \int_{-\infty}^{+\infty} (\Omega + \omega_0) |X(\Omega)|^2 d\Omega
\]

\[
= \int_{-\infty}^{+\infty} \Omega |X(\Omega)|^2 d\Omega + \omega_0 \int_{-\infty}^{+\infty} |X(\Omega)|^2 d\Omega = 2\pi \cdot (\mu_{\omega,\text{old}} + \omega_0)
\]

where we have introduced a substitution \(\Omega = \omega - \omega_0\), while

\[
2\pi \Delta_{\omega,\text{new}}^2 = \int_{-\infty}^{+\infty} (\omega - \mu_{\omega,\text{new}})^2 |Y(\omega)|^2 d\omega
\]

\[
= \int_{-\infty}^{+\infty} (\omega - \mu_{\omega,\text{old}} - \omega_0)^2 |X(\omega - \omega_0)|^2 d\omega
\]

\[
= \int_{-\infty}^{+\infty} (\Omega - \mu_{\omega,\text{old}})^2 |X(\Omega)|^2 d\Omega = 2\pi \Delta_{\omega,\text{old}}^2.
\]

where we again have introduced a substitution \(\Omega = \omega - \omega_0\). Similarly to the time shift not changing the frequency spread, the frequency shift does not change the time spread.

(ii) Scaling by a factor of \(s\) increases \(\mu_t\) by the same factor:

\[
\mu_{t,\text{new}} = \int_{-\infty}^{+\infty} t |y(t)|^2 dt = \int_{-\infty}^{+\infty} t \left| \frac{1}{\sqrt{s}} x \left( \frac{\tau}{s} \right) \right|^2 dt = s \int_{-\infty}^{+\infty} \tau |x(\tau)|^2 d\tau = s \mu_{t,\text{old}}.
\]
5.2. Discrete-Time Uncertainty Principle:

Prove Theorem 5.2 for real sequences. Do not forget to provide an argument for the strictness of inequality (5.14).

(Hint: Use the Cauchy-Schwarz inequality to bound \( \left| \int_{-\pi}^{\pi} \omega X(e^{j\omega}) \left[ \frac{d}{d\omega} X(e^{j\omega}) \right] \, d\omega \right|^2 \).)

Solution:
Since \( x_n \) is real, \( |X(e^{j\omega})| \) is even, so \( \mu_\omega = 0 \). However, unlike in the continuous case, \( \mu_n \)
can not be removed by shifting, because it may be not integer. Consider:
\[
\alpha = \langle \omega X(e^{j\omega}), \frac{dX(e^{j\omega})}{d\omega} \rangle + j\mu_n X(e^{j\omega})
\]
\[
= \int_{-\pi}^{+\pi} \omega X^*(e^{j\omega}) \frac{dX(e^{j\omega})}{d\omega} d\omega + j\mu_n \int_{-\pi}^{+\pi} \omega |X(e^{j\omega})|^2 d\omega
\]
\[
= \int_{-\pi}^{+\pi} \omega X^*(e^{j\omega}) \frac{dX(e^{j\omega})}{d\omega} d\omega.
\]
The integral \[\int_{-\pi}^{+\pi} \omega |X(e^{j\omega})|^2 d\omega = 0\] because \[|X(e^{j\omega})|\] is even.
Summing \(\alpha\) and its complex conjugate, and then integrating by parts, we obtain:
\[
\alpha + \alpha^* = \int_{-\pi}^{+\pi} \omega \left[ X^*(e^{j\omega}) \frac{dX(e^{j\omega})}{d\omega} + X(e^{j\omega}) \frac{dX^*(e^{j\omega})}{d\omega} \right] d\omega = \int_{-\pi}^{+\pi} \omega \left| \frac{dX(e^{j\omega})}{d\omega} \right|^2 d\omega
\]
\[
= \omega |X(e^{j\omega})|^2 \bigg|_{-\pi}^{+\pi} - \int_{-\pi}^{+\pi} |X(e^{j\omega})|^2 d\omega = -2\pi.
\]
Taking into account that the absolute value of the real part of any complex number is always less or equal to the absolute value of that number, and then applying Cauchy-Schwarz inequality, we obtain:
\[
-2\pi^2 = |\alpha + \alpha^*|^2 \leq 4 \cdot |\alpha|^2 = 4 \cdot \left| \langle \omega X(e^{j\omega}), \frac{dX(e^{j\omega})}{d\omega} \rangle + j\mu_n X(e^{j\omega}) \right|^2
\]
\[
\leq 4 \cdot \int_{-\pi}^{+\pi} |\omega X(e^{j\omega})|^2 d\omega \cdot \int_{-\pi}^{+\pi} \left| \frac{dX(e^{j\omega})}{d\omega} \right|^2 d\omega.
\]
Since \(\mu_\omega = 0\), we have that
\[
2\pi \Delta_n^2 = \int_{-\pi}^{+\pi} |\omega X(e^{j\omega})|^2 d\omega.
\]
Since \(dX(e^{j\omega})/d\omega\) is the Fourier transform of the sequence \(-jn_n x_n\), then \(dX(e^{j\omega})/d\omega + j\mu_n X(e^{j\omega})\) is the Fourier transform of the sequence \(-jn_n x_n + j\mu_n x_n = -j(n - \mu_n)x_n\). Then the Parseval’s relation gives
\[
\int_{-\pi}^{+\pi} \left| \frac{dX(e^{j\omega})}{d\omega} + j\mu_n X(e^{j\omega}) \right|^2 d\omega = \sum_n |-j(n - \mu_n)x_n|^2 = \sum_n (n - \mu_n)^2 x_n^2 = 2\pi \cdot \Delta_n^2
\]
The above results show that
\[
1 \leq 4 \cdot \Delta_n^2 \Delta_n^2
\]
The equality could be achieved only when \(\beta \cdot \omega X(e^{j\omega}) = dX(e^{j\omega})/d\omega + j\mu_n X(e^{j\omega})\), which would give us the Gaussian function. But this Gaussian function can never achieve 0 value, which contradicts the theorem’s condition that \(X(e^{j\beta}) = 0\). Therefore, the equality can not be achieved.

**Exercises**

5.1. **Finite Sequences and Their DTFTs:**
Show that if a sequence has a finite number of terms, then its DTFT cannot be zero over an interval (that is, it can only have isolated zeros). Conversely, show that if a DTFT is zero over an interval, then the corresponding sequence has an infinite number of nonzero terms.

5.2. **Box Function, Its Convolution, and Limits:**
Given is the box function from (5.3).
Exercises

(i) What is the time spread $\Delta t^2$ of the triangle function $(b \ast b)$?
(ii) What is the time spread $\Delta t^2$ of the function $b$ convolved with itself $N$ times?

5.3. Uncertainty Principle for Complex Functions:
Prove Theorem 5.1 without assuming that $x(t)$ is a real function.
(Hint: The proof requires more than the Cauchy-Schwarz inequality and integration by parts. Use the product rule of differentiation, $\frac{d}{dt}|x(t)|^2 = x'(t)x^*(t) + x'(t)x(t)$. Also, use that for any $\alpha \in \mathbb{C}$, $|\alpha| \geq \frac{1}{2}|\alpha + \alpha^*|$.)

5.4. Rational Scale Changes on Sequences:
A scale change by a factor $M/N$ can be achieved by upsampling by $M$ followed by downsampling by $N$.
(i) Consider a scale change by $3/2$, and show that it can be implemented either by upsampling by $3$, followed by downsampling by $2$, or the converse.
(ii) Show that the scale change by $3/2$ cannot be undone, even though it is a stretching operation.
(iii) Using the fact that when $M$ and $N$ are coprime, upsampling by $M$ and downsampling by $N$ commute, show that a sampling rate change by $M/N$ cannot be undone unless $N = 1$.

5.5. Shift-Invariant Subspaces and Degrees of Freedom:
Define a shift-invariant subspace $S$ as

$$S = \text{span}\{\varphi(t-nT)\}_{n \in \mathbb{Z}}, \quad T \in \mathbb{R}^+.$$ 

(i) Show that the piecewise-constant function defined in (5.17) belongs to such a space when $\varphi(t)$ is the indicator function of the interval $[0, 1]$ and $T = 1$.
(ii) Show that a function in $S$ has exactly $1/T$ degrees of freedom per unit time.

5.6. Uncertainty Principle for the DFT:
Let $x$ and $X$ be a length-$N$ DFT pair, and let $N_t$ and $N_\omega$ denote the number of nonzero components of $x$ and $X$, respectively.
(i) Prove that $X$ cannot have $N_t$ consecutive zeros, where “consecutive” is interpreted mod $N$.
(Hint: For an arbitrary selection of $N_t$ consecutive components of $X$, form a linear system relating the nonzero components of $x$ to the selected components of $X$.)
(ii) Using the result of the first part, prove $N_tN_\omega \geq N$. This uncertainty principle is due to Donoho and Stark [60].

5.7. Signal Recovery Based on the Finite-Dimensional Uncertainty Principle:
Suppose the DFT of a length-$N$ signal $x$ is known to have only $N_\omega$ nonzero components. Using the result of Exercise 5.6, show that the limited DFT-domain support makes it possible to uniquely recover $x$ from any $M$ (time-domain) components as long as $2(N - M)N_\omega < N$.
(Hint: Show that nonunique recovery leads to a contradiction.)
Chapter 5. Time, Frequency, Scale and Resolution
Chapter 6

Filter Banks: Building Blocks of Time-Frequency Expansions

Contents

6.1 Introduction ........................................ 210
6.2 Theory of Orthogonal Two-Channel Filter Banks 215
6.3 Design of Orthogonal Two-Channel Filter Banks 225
6.4 Theory of Biorthogonal Two-Channel Filter Banks 231
6.5 Design of Biorthogonal Two-Channel Filter Banks 241
6.6 Theory of Stochastic Filter Banks ............... 245
6.7 Algorithms ........................................ 245
Chapter at a Glance .................................... 248
Historical Remarks ................................... 252
Further Reading ....................................... 252
Exercises with Solutions ............................. 253
Exercises ............................................. 255

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 7 and 8. 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 (a) the simplest; (b) reveal the essence of the $N$-channel case; and (c) 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. Pointers to generalizations, such as $N$-channel filter banks and multidimensional filter banks, are given in Further Reading.
6.1 Introduction

**Implementing a Haar ONB Expansion** At the end of the previous chapter, we constructed an ONB 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 (5.20) and (5.22) as:

$$g_n = \varphi_{0,n} = \frac{1}{\sqrt{2}} (\delta_n + \delta_{n-1}), \quad (6.1)$$

$$h_n = \varphi_{1,n} = \frac{1}{\sqrt{2}} (\delta_n - \delta_{n-1}). \quad (6.2)$$

This is done both for simplicity, as well as because it is the standard way these sequences are denoted. We start by looking at the reconstruction formula (5.23), and rewrite it as

$$x_n = \sum_{k \in \mathbb{Z}} \langle x, \varphi_{2k} \rangle \varphi_{2k,n} + \sum_{k \in \mathbb{Z}} \langle x, \varphi_{2k+1} \rangle \varphi_{2k+1,n}$$

$$= \sum_{k \in \mathbb{Z}} \alpha_k \varphi_{2k,n} + \sum_{k \in \mathbb{Z}} \beta_k \varphi_{2k+1,n}$$

$$= \sum_{k \in \mathbb{Z}} \alpha_k g_{n-2k} + \sum_{k \in \mathbb{Z}} \beta_k h_{n-2k}, \quad (6.3)$$

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

$$\langle x, \varphi_{2k} \rangle = \langle x, g_{n-2k} \rangle_n = \alpha_k, \quad (6.4a)$$

$$\langle x, \varphi_{2k+1} \rangle = \langle x, h_{n-2k} \rangle_n = \beta_k. \quad (6.4b)$$

Then, recognize each sum in (6.3) as (2.143) with the input sequences being $\alpha_k$ and $\beta_k$, respectively. Thus, each sum in (6.3) can be implemented as: the input sequence $\alpha$ (first sum) or $\beta$ (second sum) going through an upsampler by 2 followed by filtering by $g$ (first sum) and $h$ (second sum).

By the same token, we can identify the computation of the expansion coefficients in (6.4a)-(6.4b) as (2.138), that is, both $\alpha$ and $\beta$ sequences can be implemented 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 ONB expansion as in Figure 6.1(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, etc.
Figure 6.1: 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.

For example, in matrix notation, the analysis process (6.4a)-(6.4b) can be expressed as

\[
\begin{bmatrix}
\vdots \\
\alpha_0 \\
\beta_0 \\
\alpha_1 \\
\beta_1 \\
\alpha_2 \\
\beta_2 \\
\vdots
\end{bmatrix}
= \frac{1}{\sqrt{2}}
\begin{bmatrix}
\cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\
\cdot & 1 & 0 & 0 & 0 & 0 & \cdots \\
\cdot & 1 & -1 & 0 & 0 & 0 & \cdots \\
\cdot & 0 & 0 & 1 & 1 & 0 & \cdots \\
\cdot & 0 & 0 & 1 & -1 & 0 & \cdots \\
\cdot & 0 & 0 & 0 & 0 & 1 & 1 & \cdots \\
\cdot & 0 & 0 & 0 & 0 & 1 & -1 & \cdots \\
\vdots \\
\end{bmatrix}
\begin{bmatrix}
\vdots \\
x_0 \\
x_1 \\
x_2 \\
x_3 \\
x_4 \\
x_5 \\
\vdots
\end{bmatrix},
\tag{6.5}
\]

and the synthesis process (6.3) as

\[
\begin{bmatrix}
\vdots \\
x_0 \\
x_1 \\
x_2 \\
x_3 \\
x_4 \\
x_5 \\
\vdots
\end{bmatrix}
= \frac{1}{\sqrt{2}}
\begin{bmatrix}
\cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\
\cdot & 1 & 0 & 0 & 0 & 0 & \cdots \\
\cdot & 1 & -1 & 0 & 0 & 0 & \cdots \\
\cdot & 0 & 0 & 1 & 1 & 0 & \cdots \\
\cdot & 0 & 0 & 1 & -1 & 0 & \cdots \\
\cdot & 0 & 0 & 0 & 0 & 1 & 1 & \cdots \\
\cdot & 0 & 0 & 0 & 0 & 1 & -1 & \cdots \\
\vdots \\
\end{bmatrix}
\begin{bmatrix}
\vdots \\
\alpha_0 \\
\beta_0 \\
\alpha_1 \\
\beta_1 \\
\alpha_2 \\
\beta_2 \\
\vdots
\end{bmatrix},
\tag{6.6}
\]
Chapter 6. Filter Banks: Building Blocks of Time-Frequency Expansions

or

\[ x = \Phi \Phi^T x \implies \Phi \Phi^T = I. \quad (6.7) \]

Of course, the matrix \( \Phi \) is the same matrix we have seen in (5.24). Moreover, from (6.7), it is a unitary matrix, which we know from (1.58) implies that the Haar basis is an ONB (which we have already shown in Chapter 5).

**Implementing a General ONB Expansion**  What we have seen for the Haar ONB is true in general; we can construct an ONB for \( \ell^2(\mathbb{Z}) \) using two template basis sequences and their even shifts. As we have seen, such an ONB can be implemented using a two-channel filter bank, consisting of downsamplers, upsamplers and filters \( g \) and \( h \). Let \( g \) and \( h \) be two causal filters,\(^38\) where we implicitly assume that these filters have certain time and frequency localization properties, as discussed in Chapter 5 (\( g \) is lowpass and \( h \) is highpass). The synthesis (6.6) generalizes to

\[
\begin{bmatrix}
\vdots \\
x_0 \\
x_1 \\
x_2 \\
x_3 \\
x_4 \\
\vdots
\end{bmatrix}
= \begin{bmatrix}
\vdots \\
g_0 \\
g_1 \\
g_2 \\
g_3 \\
g_4 \\
\vdots
\end{bmatrix}
= \begin{bmatrix}
0 \\
h_0 \\
h_1 \\
h_2 \\
h_3 \\
h_4 \\
0 \\
\vdots
\end{bmatrix}
= \Phi X, \quad (6.8)
\]

with the basis matrix \( \Phi \) as before. To have an ONB, the basis sequences \( \{ \varphi_k \}_{k \in \mathbb{Z}} \)—even shifts of template sequences \( g \) and \( h \)—must form an orthonormal set as in (1.40). In matrix parlance, \( \Phi \) must be unitary, implying the columns are orthonormal:

\[
\langle g_n, g_{n-2k} \rangle_n = \delta_k, \quad \langle h_n, h_{n-2k} \rangle_n = \delta_k, \quad \langle g_n, h_{n-2k} \rangle_n = 0. \quad (6.9)
\]

We have seen in (2.155) that such filters are called orthogonal; how to design them is a central topic of this chapter.

As we are building an ONB, 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

\(^{38}\)While causality is not necessary to construct a filter bank, we impose it later and it improves readability here.
6.1. Introduction

to a multiplication by $\Phi^T$:

$$
\begin{bmatrix}
\alpha_0 \\
\beta_0 \\
\alpha_1 \\
\beta_1 \\
\alpha_2 \\
\vdots
\end{bmatrix}
\begin{bmatrix}
\langle x_n, g_n \rangle_n \\
\langle x_n, h_n \rangle_n \\
\langle x_n, g_{n-2} \rangle_n \\
\langle x_n, h_{n-2} \rangle_n \\
\langle x_n, g_{n-4} \rangle_n \\
\vdots
\end{bmatrix}
= \begin{bmatrix}
\ldots & \ldots & \ldots & \ldots & \ldots & \ldots \\
\ldots & g_0 & g_1 & g_2 & g_3 & \ldots \\
\ldots & h_0 & h_1 & h_2 & h_3 & h_4 & \ldots \\
\ldots & 0 & 0 & g_0 & g_1 & g_2 & \ldots \\
\ldots & 0 & 0 & h_0 & h_1 & h_2 & \ldots \\
\ldots & 0 & 0 & 0 & g_0 & \ldots \\
\ldots & \ldots & \ldots & \ldots & \ldots & \ldots & \ldots
\end{bmatrix}
\begin{bmatrix}
\ldots \\
2^0 \\
\vdots
\end{bmatrix}
$$

(6.10)

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 6.1(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 like in Chapter 5—an ONB 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 ONB projects; we implicitly assume that $V$ is the space of coarse
(lowpass) sequences and $W$ is the space of detail (highpass) sequences. Figure 6.3
illustrates that, where a synthetic signal 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 (5.25):

$$
V = \text{span}\{\varphi_{0,-2k} \mid k \in \mathbb{Z}\} = \text{span}\{g_{n-2k} \mid k \in \mathbb{Z}\},
$$

(6.11)

$$
W = \text{span}\{\varphi_{1,-2k} \mid k \in \mathbb{Z}\} = \text{span}\{h_{n-2k} \mid k \in \mathbb{Z}\},
$$

(6.12)

and produce the lowpass and highpass approximations, respectively:

$$
x_V = \sum_{k \in \mathbb{Z}} \alpha_k g_{n-2k},
$$

(6.13)

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

(6.14)

As the basis sequences spanning these spaces are orthogonal to each other and all
together form an ONB, the two projection subspaces together give back the original
space as in (5.28): $\ell^2(\mathbb{Z}) = V \oplus W$.

In this brief chapter preview, we introduced the two-channel filter bank as in
Figure 6.1(a). It uses orthogonal filters satisfying (6.9) and computes an expansion
with respect to the set of basis vectors $\{g_{n-2k}, h_{n-2k} \mid k \in \mathbb{Z}\}$, 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 FIR
filters with real coefficients, since they are dominant in practice.
Figure 6.2: 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 $\ell^2(\mathbb{Z})$ into two orthogonal complements $V$ (lowpass subspace) and $W$ (highpass subspace).

Figure 6.3: A signal and its two projections. (a) The signal $x$ with features at different scales (low-frequency sinusoid, high-frequency noise, piecewise polynomial and Dirac impulse). (b) The lowpass component $x_V$. (c) The highpass component $x_W$. 

6.2 Theory of Orthogonal Two-Channel Filter Banks

This section develops necessary conditions for the design of orthogonal two-channel filter banks implementing ONBs and the key properties of such filter banks. We assume that the system shown in Figure 6.1(a) implements an ONB 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.

6.2.1 A Single Channel and Its Properties

We now look at each channel of Figure 6.1 separately and determine their properties. As the lowpass and highpass channels are essentially symmetric, our approach is to establish (a) the properties inherent to each channel on its own; and (b) given one channel, establish the properties the other has to satisfy so as to build an ONB when combined. While we have seen most of the properties already, we repeat them here for completeness.

Consider the lower branch of Figure 6.1(a), projecting the input \( x \) onto its lowpass approximation \( x_V \), depicted separately in Figure 6.4. In (6.13), that lowpass approximation \( x_V \) was given as

\[
x_V = \sum_{k \in \mathbb{Z}} \alpha_k g_{n-2k}.
\]

Similarly, in (6.14), the highpass approximation \( x_W \) was given as

\[
x_W = \sum_{k \in \mathbb{Z}} \beta_k h_{n-2k}.
\]
Orthogonality of the Lowpass Filter

Since we started with an ONB, the set \( \{g_{n-2k}\}_{k \in \mathbb{Z}} \) is an orthonormal set. We have seen in Section 2.5.4 that such a filter is termed orthogonal and satisfies (2.155):

\[
\langle g_n, g_{n-2k} \rangle = \delta_k \quad \text{Matrix View} \quad D_2 G^T G U_2 = I
\]

\[
\begin{align*}
G(z)G(z^{-1}) + G(-z)G(-z^{-1}) & = 2 \\
|G(e^{j\omega})|^2 + |G(e^{j(\omega+\pi)})|^2 & = 2
\end{align*}
\] (6.16)

In the matrix view, we have used linear operators (infinite matrices) introduced in Section 2.5. These are: (a) downsampling by 2, \( D_2 \), from (2.123a); (b) upsampling by 2, \( U_2 \), from (2.129a); and (c) filtering by \( G \), from (2.37). The matrix view expresses the fact that the columns of \( GU_2 \) form an orthonormal set. The DTFT version is the quadrature mirror formula we have seen in (2.154).

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:

\[
\langle h_n, h_{n-2k} \rangle = \delta_k \quad \text{Matrix View} \quad D_2 H^T H U_2 = I
\]

\[
\begin{align*}
H(z)H(z^{-1}) + H(-z)H(-z^{-1}) & = 2 \\
|H(e^{j\omega})|^2 + |H(e^{j(\omega+\pi)})|^2 & = 2
\end{align*}
\] (6.17)

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 we have seen in (2.154).

Autocorrelation of the Lowpass Filter

As it is widely used in filter design, we rephrase (6.16) in terms of the autocorrelation of \( g \), given by (2.55):

\[
\langle g_n, g_{n-2k} \rangle = a_{2k} = \delta_k \quad \text{Matrix View} \quad D_2 A U_2 = I
\]

\[
\begin{align*}
A(z) + A(-z) & = 2 \\
A(e^{j\omega}) + A(e^{j(\omega+\pi)}) & = 2
\end{align*}
\] (6.18)

In the above, \( A = G^T G \) is the autocorrelation operator as in (2.58), 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

\[
A(z) = G(z)G(z^{-1}) = 1 + 2 \sum_{k=0}^{\infty} a_{2k+1} (z^{2k+1} + z^{-(2k+1)}).
\] (6.19)

Autocorrelation of the Highpass Filter

Similarly to the lowpass filter,

\[
\langle h_n, h_{n-2k} \rangle = a_{2k} = \delta_k \quad \text{Matrix View} \quad D_2 A U_2 = I
\]

\[
\begin{align*}
A(z) + A(-z) & = 2 \\
A(e^{j\omega}) + A(e^{j(\omega+\pi)}) & = 2
\end{align*}
\] (6.20)

Equation (6.19) holds for this autocorrelation as well.
6.2. Theory of Orthogonal Two-Channel Filter Banks

Orthogonal Projection Property of the Lowpass Channel

We now look at the lowpass channel as a composition of four linear operators we just saw:

\[ x_V = P_V x = G U_2 D_2 G^T x. \] (6.21)

The notation is evocative of projection onto \( V \), and we will now show that the lowpass channel accomplishes precisely this. Using (6.16), we check idempotency and self-adjointness of \( P \),

\[
P_V^2 = (G U_2 D_2 G^T) (G U_2 D_2 G^T) = G U_2 D_2 G^T = P_V,
\]

\[
P_V^T = (G U_2 D_2 G^T)^T = G (U_2 D_2)^T G^T = G U_2 D_2 G^T = P_V.
\]

Indeed, \( P_V \) is an orthogonal projection operator, with the range given in (6.11):

\[ V = \text{span}(\{g_{n-2k}\}_{k \in \mathbb{Z}}). \] (6.22)

The summary of properties of the lowpass channel is given in Chapter at a Glance.

Orthogonal Projection Property of the Highpass Channel

The highpass channel as a composition of four linear operators (infinite matrices) is:

\[ x_W = P_W x = H U_2 D_2 H^T x. \] (6.23)

It is no surprise that \( P_W \) is an orthogonal projection operator with the range given in (6.12):

\[ W = \text{span}(\{h_{n-2k}\}_{k \in \mathbb{Z}}). \] (6.24)

The summary of properties of the highpass channel is given in Chapter at a Glance (table for lowpass channel, just make appropriate substitutions).

6.2.2 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 ONB. 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 6.1(a) implements an ONB, 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 (6.9). We have already used the orthonormality of the set \( \{g_{n-2k}\}_{k \in \mathbb{Z}} \) in (6.16) as well as the orthonormality of the set \( \{h_{n-2k}\}_{k \in \mathbb{Z}} \) in (6.17). What is left is that these two sets are orthogonal to each other, the third equation in (6.9).
Orthogonality of the Lowpass and Highpass Filters

Using similar methods as before, we summarize the properties the lowpass and highpass sequences must satisfy as follows:

\[ \langle g_n, h_{n-2k} \rangle = 0 \]

Matrix View

\[ D_2 H^T G U_2 = 0 \]

\[ G(z)H(z^{-1}) + G(-z)H(-z^{-1}) = 0 \]

\[ G(e^{j\omega})H(e^{-j\omega}) + G(e^{j(\omega+\pi)})H(e^{-j(\omega+\pi)}) = 0 \]

(6.25)

Crosscorrelation of the Lowpass and Highpass Filters

Instead of autocorrelation properties of an orthogonal filter, we look at crosscorrelation properties of two filters orthogonal to each other. The crosscorrelation is given by (2.59):

\[ \langle g_n, h_{n-2k} \rangle = c_{2k} = 0 \]

Matrix View

\[ D_2 C U_2 = 0 \]

\[ ZT \rightarrow C(z) + C(-z) = 0 \]

\[ DTFT \rightarrow C(e^{j\omega}) + C(e^{j(\omega+\pi)}) = 0 \]

(6.26)

In the above, \( C = H^T G \) is the crosscorrelation operator. In particular, all the even terms of \( c \) are equal to zero.

### 6.2.3 Orthogonal Two-Channel Filter Bank

We are now ready to put together everything we have developed so far. What is left to show is completeness: any sequence from \( \ell^2(\mathbb{Z}) \) can be represented using the ONB 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, inspired by what we have seen in the Haar case. We can also help our intuition by considering a two-channel filter bank with ideal filters as in Figure 6.5. Say 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 ONB? 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 \( g \). While this will work as we see in the next theorem, other solutions are possible, and the most general ones are shown in Sections 6.2.4 and 6.3.3.

Based on the discussion above, we now show how, given an orthogonal filter \( g \), it completely specifies an orthogonal two-channel filter bank implementing an ONB for \( \ell^2(\mathbb{Z}) \):

**Theorem 6.1 (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:

\[ \langle g_n, g_{n-2k} \rangle = \delta_k \]

\[ ZT \rightarrow G(z)G(z^{-1}) + G(-z)G(-z^{-1}) = 2. \]

Choose

\[ h_n = (-1)^n g_{n+L-1} \]

\[ ZT \rightarrow H(z) = -z^{-L+1}G(-z^{-1}). \]
Then, \( \{g_{n-2k}, h_{n-2k}\}_{k \in \mathbb{Z}} \) is an ONB for \( \ell^2(\mathbb{Z}) \), and the expansion can be 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
\[
\ell^2(\mathbb{Z}) = V \oplus W, \quad \text{with} \quad V = \text{span}(\{g_{n-2k}\}_{k \in \mathbb{Z}}),\quad W = \text{span}(\{h_{n-2k}\}_{k \in \mathbb{Z}}).
\] (6.29)

**Proof.** To prove the theorem, we must prove that 1. \( \{g_{n-2k}, h_{n-2k}\}_{k \in \mathbb{Z}} \) is an orthonormal set; and 2. it is complete.

1. To prove that \( \{g_{n-2k}, h_{n-2k}\}_{k \in \mathbb{Z}} \) is an orthonormal set, we must prove (6.9). 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 (6.17). The definition of \( h \) in (6.28) implies
\[
H(z)H(z^{-1}) = G(-z)G(-z^{-1}).
\] (6.30)

Thus,
\[
H(z)H(z^{-1}) + H(-z^{-1})H(-z^{-1}) = G(-z)G(-z^{-1}) + G(z)G(z^{-1}) \overset{\text{(a)}}{=} 2,
\]
where (a) follows from (6.16).

To prove the third condition in (6.9), that is, \( h \) is orthogonal to \( g \) and all
its even shifts, we must prove one of the conditions in (6.25):

\[
G(z)H(z^{-1}) + G(-z)H(-z^{-1})
\]

\[
(a)\ G(z) \left[-z^{-L-1}G(-z)\right] + G(-z) \left[(1)^Lz^{-L-1}G(z)\right]
\]

\[
= [-1 + (1)^L] z^{-L-1}G(z)G(-z) \quad \text{if} \quad L = 2\ell \quad \text{is even.}
\]

2. We prove completeness by proving that perfect reconstruction holds for any \( x \in \ell^2(\mathbb{Z}). \) An alternative would be to prove Parseval\'s equality \( \|x\|^2 = \|x_V\|^2 + \|x_W\|^2. \)

We do this in ZT-domain expressions for \( X_V(z) \) and \( X_W(z) \) and prove they sum up to \( X(z). \) We start with the lowpass branch. Recall that downsampling by 2 followed by upsampling by 2 maps a signal with the \( z \)-transform \( X(z) \) to a signal with the \( z \)-transform \( \frac{1}{2} [X(z) + X(z^{-1})]. \) In the lowpass channel, the input \( X(z) \) is filtered by \( G(z^{-1}) \), and is then down- and upsampled, followed by filtering with \( G(z) \). Thus, the \( z \)-transforms of \( x_V \) (and similarly for \( x_W \)) are:

\[
X_V(z) = \frac{1}{2} G(z) \left[G(z^{-1})X(z) + G(-z^{-1})X(-z)\right], \quad \text{(6.31)}
\]

\[
X_W(z) = \frac{1}{2} H(z) \left[H(z^{-1})X(z) + H(-z^{-1})X(-z)\right]. \quad \text{(6.32)}
\]

The output of the filter bank is the sum of \( x_V \) and \( x_W \):

\[
X_V(z) + X_W(z) = \frac{1}{2} A(z) X(z) + \frac{1}{2} B(z) X(z), \quad \text{(6.33)}
\]

Substituting (6.28) into the above equation, we get:

\[
A(z) = G(z)G(z^{-1}) + H(z)H(z^{-1})
\]

\[
\equiv (a) G(z)G(z^{-1}) + \left[-z^{-L+1}G(z^{-1})\right]\left[-(-z^{-1})^{-L+1}G(z)\right]
\]

\[
= [1 + (1)^{L+1}] G(z)G(-z^{-1}) \quad \text{if} \quad L = 2\ell \quad \text{is even.}
\]

\[
B(z) = G(z)G(z^{-1}) + H(z)H(z^{-1})
\]

\[
\equiv (b) G(z)G(z^{-1}) + G(-z^{-1})G(-z) \quad \text{if} \quad L = 2\ell \quad \text{is even.}
\]

To show (6.29), we write (6.35) in the original domain as in (6.13)-(6.14):

\[
x_n = \sum_{k \in \mathbb{Z}} a_k g_{n-2k} + \sum_{k \in \mathbb{Z}} b_k h_{n-2k}, \quad \text{(6.36)}
\]
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 (6.25) proving (6.29).

This completes the proof.

A few comments are in order. In the theorem, \( L \) is an arbitrary even integer. However, \( L \) is evocative of length. In the most natural case, \( g \) is supported on \( \{0, 1, \ldots, L-1\} \), when, the time reversal and shift in (6.28) make \( h \) also supported on \( \{0, 1, \ldots, L-1\} \). Furthermore, for FIR filters with lengths greater than 1, the orthogonality condition (6.18) requires even length (see Exercise 6.1).

Along with the time reversal and shift, the other qualitative feature of (6.28) is modulation by \((-1)^n\) (mapping \( z \to -z \) in the ZT domain). As we said, this makes \( h \) a highpass filter when \( g \) is a lowpass filter. As an example, if we apply Theorem 6.1 to the Haar lowpass filter from (6.1), we obtain the Haar highpass filter from (6.2).

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 causal) by shifting them both by \((-L+1)\). Beware that such an implementation implies perfect reconstruction within a shift (delay), and the ONB 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:

\[
\hat{x}_n = x_{n-L+1}.
\]

The four filters and the \( z \)-transforms in an orthogonal, two-channel filter bank, as well as the summary of various properties discussed until now, are given in Chapter at a Glance.

### 6.2.4 Polyphase View of Orthogonal Filter Banks

As we saw in Section 2.5, downsampling introduces periodic shift variance into a system. To deal with this, multirate systems are often analyzed in polyphase domain, as discussed in Section 2.5.5. 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.

For two-channel filter banks, a polyphase decomposition is achieved by simply splitting both sequences and filters into their even- and odd-indexed subsequences. What follows now is a brief account of polyphase analysis of two-channel orthogonal filter banks.

Define the polyphase decomposition of the synthesis filters through\(^{39}\)

\[
G(z) = G_0(z^2) + z^{-1}G_1(z^2) = \sum_{n \in \mathbb{Z}} g_{2n}z^{-2n} + z^{-1}\sum_{n \in \mathbb{Z}} g_{2n+1}z^{-2n}, \quad (6.37a)
\]

\[
H(z) = H_0(z^2) + z^{-1}H_1(z^2) = \sum_{n \in \mathbb{Z}} h_{2n}z^{-2n} + z^{-1}\sum_{n \in \mathbb{Z}} h_{2n+1}z^{-2n}. \quad (6.37b)
\]

\(^{39}\)Remember that whether we use an advance \((z)\) or delay \((z^{-1})\) is purely a matter of convention.
We can now define a polyphase matrix $\Phi_p(z)$:

$$\Phi_p(z) = \begin{bmatrix} G_0(z) & H_0(z) \\ G_1(z) & H_1(z) \end{bmatrix}. \tag{6.38}$$

You can easily remember the ordering of the polyphase components in the matrix by remembering that we put basis functions as columns of the synthesis matrix. Here the basis functions are $g$, $h$ and all their even translates. A block diagram of the polyphase implementation of the system is given in Figure 6.6, where we have yet to justify the form of the polyphase analysis operator.

As we are dealing with a two-channel orthogonal filter bank where all the filters are obtained from one single filter $g$, we know from (6.28) that $H(z) = -z^{-L+1}G(-z^{-1})$. Based on this, the polyphase components of $H$ are given by

$$H_0(z) = z^{-L/2+1}G_1(z^{-1}), \tag{6.39a}$$

$$H_1(z) = -z^{-L/2+1}G_0(z^{-1}), \tag{6.39b}$$
6.2. Theory of Orthogonal Two-Channel Filter Banks

leading to the polyphase matrix

\[
\Phi_p(z) = \begin{bmatrix}
G_0(z) & z^{-L/2+1}G_1(z^{-1}) \\
G_1(z) & -z^{-L/2+1}G_0(z^{-1})
\end{bmatrix}.
\] (6.40)

Since \( g \) is orthogonal to its even translates, substitute (6.37a) into the ZT domain version of (6.16) to get the condition for orthogonality in polyphase form:

\[
G_0(z)G_0(z^{-1}) + G_1(z)G_1(z^{-1}) = 1.
\] (6.41)

Using this, the determinant of \( \Phi_p(z) \) becomes \( z^{-L/2+1} \) and \( \Phi_p^{-1}(z) \) becomes \( \Phi_p^T(z^{-1}) \).

In other words, the polyphase matrix \( \Phi_p(z) \) satisfies the following:

\[
\Phi_p(z)\Phi_p^T(z^{-1}) = I.
\] (6.42)

The above condition defines a paraunitary matrix. A paraunitary matrix is unitary on the unit circle, and if all matrix entries are stable—as in this case where all the filters are FIR—the matrix is called lossless. In fact, (6.40), together with (6.41), define the most general \( 2 \times 2 \), real-coefficient, causal FIR paraunitary matrix.

Since the inverse of the polyphase matrix \( \Phi_p(z) \) is \( \Phi_p^T(z^{-1}) \), the latter is the polyphase matrix on the analysis side, as shown in Figure 6.6. We can easily check that. Taking the columns of \( \Phi_p(z^{-1}) \) to be the polyphase components of analysis filters, and bearing in mind that the polyphase decomposition of a filter followed by a downsampler involves an advance \( z \) rather than the delay \( z^{-1} \), we get

\[
G_0(z^{-1}) + zG_1(z^{-1}) = G(z^{-1}) \quad \text{ZT} \quad g_{-n}
\]

\[
H_0(z^{-1}) + zH_1(z^{-1}) = H(z^{-1}) \quad \text{ZT} \quad h_{-n}
\]

which, of course, we already know to be true.

**Example 6.1 (Haar filter bank in polyphase form).** The Haar filters (6.1)-(6.2) 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

\[
\Phi_p(z) = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}.
\] (6.43)

The form of the polyphase matrix for the Haar ONB is exactly the same as the Haar ONB for \( \mathbb{R}^2 \) from (1.12), or one block of the Haar ONB infinite matrix \( \Phi \) from (5.24). 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 ONB, in polyphase terms.
Proposition 6.2 (Paraunitary Polyphase Matrix and ONB). A $2 \times 2$ polyphase matrix $\Phi_p(z)$ is paraunitary if and only if the associated two-channel filter bank implements an ONB for $\ell^2(\mathbb{Z})$.

Proof. If the polyphase matrix is paraunitary, then the expansion it implements is complete, due to (6.42). To prove that the expansion is an ONB, we must show that the basis functions form an orthonormal set. From (6.40) and (6.42), we get (6.41). Substituting this into the ZT domain version of (6.16), 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 (6.28), $h$ 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 ONB implemented by a two-channel filter bank. That means we have a template sequence $g$, a template sequence $h$ formed from $g$ through (6.28), and their even shifts, all together forming an ONB. We can now translate those conditions into ZT domain using (6.16) and derive the corresponding polyphase domain versions, such as the one in (6.41). These lead to the polyphase matrix being paraunitary. □

6.2.5 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 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 2.B: Convolution of a polynomial sequence $x$ with a differencing filter $h_n = (\delta_n - \delta_{n-1})$, or, multiplication of $X(z)$ by $H(z) = (1 - z^{-1})^N$, reduces the degree of the polynomial by 1. In general, to kill a polynomial of degree $(N - 1)$, $x_n = \sum_{k=0}^{N-1} a_k n^k$, we need a filter of the following form:

$$H(z) = (1 - z^{-1})^N R(z).$$

Let us now apply what we just learned to two-channel orthogonal filter banks with polynomial signals as inputs. We will construct the analysis filter in the highpass branch to have $N$ zeros at $z = 1$, thus killing 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; thus, the lowpass branch will reconstruct polynomials of degree $(N - 1)$. In other words, $x_V$ will be a polynomial approximation of a certain degree.

To construct such a filter bank, remember that our highpass filter is given by (6.28); thus, if it is of the form

$$H(z) = (1 - z^{-1})^N (-z)^{-L+1+N} R(-z^{-1}),$$

the corresponding lowpass is

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

(6.44)
6.3. Design of Orthogonal Two-Channel Filter Banks

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 \( z = -1 \) makes a lot of signal processing sense. After all, \( z = -1 \) corresponds to \( \omega = \pi \), or the maximum discrete frequency; it is thus natural for a lowpass filter to have a zero at \( z = -1 \). Putting more than one zero at \( z = -1 \) has further approximation advantages. This is made precise in the following proposition and also arises in wavelet constructions in later chapters.

**Proposition 6.3 (Polynomial reproduction).** Given is an orthogonal filter bank in which the synthesis lowpass filter \( G(z) \) has \( N \) zeros at \( z = -1 \). Then polynomial signals up to degree \( (N-1) \) 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 (6.44). From Chapter at a Glance, the analysis highpass filter is of the form \( -z^{L-1}G(-z) \), which means it has a factor \( (1 - z^{-1})^N \), that is, it has \( N \) zeros at \( z = 1 \). From our discussion, this factor thus annihilates a polynomial input with degree \( (N-1) \), resulting in \( \beta = 0 \) and \( x_W = 0 \). Because of the perfect reconstruction property, \( x = x_V \), showing that the polynomial signals are reproduced by a linear combination of \( \{g_{n-2k}\}_{k \in \mathbb{Z}} \), as in (6.13).

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:

\[
\sum_{k \in \mathbb{Z}} k^\ell h_{n-k} = 0, \quad \text{for } \ell < N \text{ and } n \in \mathbb{Z},
\]

that is, \( \ell \)th-order moments of \( h \) up to \( (N-1) \) are zero (see Exercise 6.2).

6.3 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 (6.16); we have seen how the other three filters follow (see Chapter at a Glance). The design is based on (a) finding an autocorrelation function satisfying (6.18) (it is symmetric, positive semi-definite and has a single nonzero even-indexed coefficient, see also (2.58)); and (b) factoring that autocorrelation \( A(z) = G(z)G(z^{-1}) \) into its spectral factors (many factorizations are possible, see Section 2.4.2).

We consider three different approaches to the design. The first is tries to approach an ideal halfband lowpass filter, the second aims at polynomial approximation, while the third uses lattice factorization in polyphase domain.
6.3.1 Lowpass Approximation Design

Assume we wish to get a $G(e^{j\omega})$ as close as possible to an ideal lowpass halfband filter:

$$
\begin{align*}
G(e^{j\omega}) &= 2, & |\omega| &< \pi/2; \\
&= 0, & \text{otherwise.}
\end{align*}
$$

(6.46)

The autocorrelation of such a filter is an ideal lowpass halfband function (6.46) as well with an impulse response

$$
f_n = \frac{1}{2\pi} \int_{-\pi/2}^{\pi/2} 2e^{j\omega} d\omega = \frac{\sin n\pi/2}{n\pi/2},
$$

(6.47)

a valid autocorrelation: It has a single nonzero even-indexed term ($f_0 = 1$) and is positive semi-definite. To get a realizable function, we apply a symmetric window function $w$ that decays to zero. The new autocorrelation $f'$ is the pointwise product

$$
f'_n = f_n \cdot w_n.
$$

(6.48)

Clearly $f'$ is symmetric and still has a single nonzero even term. In frequency domain, this leads to the convolution

$$
F'(e^{j\omega}) = \frac{1}{2\pi} (F(e^{j\omega}) * W(e^{j\omega})).
$$

(6.49)

In general, (6.49) is not nonnegative for all frequencies anymore, and thus not a valid autocorrelation. One easy way to enforce nonnegativity is to choose $W(e^{j\omega})$ itself positive, for example as the autocorrelation of another window $w'$, or

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

If $w'$ is of norm 1, then $w_0 = 1$, and from (6.48), $f'_0 = 1$ as well. Therefore, since $F(e^{j\omega})$ is real and positive, $F'(e^{j\omega})$ will be as well. The resulting sequence $f'$ and its $z$-transform $F'(z)$ can then be used in spectral factorization to obtain an orthogonal filter $g$.

**Example 6.2 (Window design of orthogonal filters).** We design a length-4 filter by the window method. Its autocorrelation is of length 7 with the target impulse response obtained by evaluating (6.47):

$$
f = \left[\ldots -\frac{3}{16\pi} 0 \frac{1}{4} \left[ \frac{1}{4} \frac{1}{2} 0 \frac{1}{4} \ldots \right] \right]^T.
$$

For the window, we take $w$ as the autocorrelation of the sequence $w'_n = \frac{1}{2}$ for $n = 0, \ldots, 3$, and $w'_0 = 0$ otherwise:

$$
w = \left[\ldots \frac{1}{4} \frac{1}{2} \frac{1}{4} \left[ \frac{1}{4} \frac{1}{2} \frac{1}{4} \frac{1}{2} \ldots \right] \right]^T.
$$

By pointwise multiplication, we obtain the autocorrelation of the lowpass filter as

$$
f' = \left[\ldots -\frac{1}{6\pi} 0 \frac{3}{2\pi} \left[ \frac{3}{2\pi} \frac{1}{2} 0 \frac{1}{6\pi} \ldots \right] \right]^T.
$$
Factoring this autocorrelation (which requires numerical polynomial root finding) gives

\[ g \approx [\ldots, 0.832, 0.549, 0.0421, -0.0637, \ldots]^T. \]

The impulse response and frequency response of \( g \) are shown in Figure 6.7.

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 (6.49) again, and define

\[ \min_{\omega \in [-\pi, \pi]} F'(e^{j\omega}) = \delta. \]

If \( \delta \geq 0 \), we are done, otherwise, we simply choose a new function

\[ F''(e^{j\omega}) = F'(e^{j\omega}) - \delta, \]

which now is nonnegative, and so we can take the spectral factor. Filters designed using this method are tabulated in [155].

### 6.3.2 Polynomial Approximation Design

Recall that a lowpass filter \( G(z) \) with \( N \) zeros at \( z = -1 \) (see (6.44)) reproduces polynomials up to degree \( (N - 1) \). Thus, the goal of this design procedure is to find an autocorrelation \( A(z) \) of the form

\[ A(z) = (1 + z^{-1})^N (1 + z)^N Q(z), \]

with \( Q(z) \) is chosen such that (6.18) is satisfied, that is,

\[ A(z) + A(-z) = 2, \quad (6.50) \]
$Q(z) = Q(z^{-1})$ (in time domain, $q_n$ is symmetric), 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. Let us work through an example.

**Example 6.3 (Polynomial approximation design of orthogonal filters).** Take $N = 2$:

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

Can we now find $Q(z)$ so as to satisfy (6.50), in particular, a minimum-degree solution? We try with $Q(z) = az + b + az^{-1}$ 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 (6.50), we need to solve the following pair of equations:

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

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

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

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 autocorrelation and can be written as $Q(z) = R(z)R(z^{-1})$. We can now extract its spectral factor $R(z)$ (take the causal part)

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

The causal orthogonal lowpass filter with 2 zeros at $z = -1$ is then

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

$$= \frac{1}{4\sqrt{2}} \left[ (1 + \sqrt{3}) + (3 + \sqrt{3})z^{-1} + (3 - \sqrt{3})z^{-2} + (1 - \sqrt{3})z^{-3} \right].$$

This filter is one of the filters from the Daubechies family of orthogonal filters. Its impulse and frequency responses are shown in Figure 6.8. The rest of the filters in the filter bank can be found in *Chapter at a Glance*. 

\[\square\]
6.3. Design of Orthogonal Two-Channel Filter Banks

From this example, we see that solving a linear system followed by spectral factorization are the key steps. In general, it can be shown that 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. Now, $Q(z) = R(z)R(z^{-1})$ is obtained from solving an $N \times N$ linear system (to satisfy $A(z) + A(z) = 2$), and spectral factorization produces the desired result. (It can be verified that $Q(e^{j\omega}) \geq 0$.) These steps are summarized in Table 6.1, while Table 6.2 gives examples of filters designed using this method.

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

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

where $Q(e^{j\omega})$ is real and positive. $A(e^{j\omega})$ and its $(2N - 1)$ derivatives are zero at $\omega = \pi$. From this it follows that the magnitude of the filter $|G(e^{j\omega})|$ and its $(N - 1)$ derivatives are zero at $\omega = \pi$. Finally, because of quadrature mirror formula, it also means that $|G(e^{j\omega})|$ has $(N - 1)$ zero derivatives at $\omega = 0$. These facts are verified in Exercise 6.3.

6.3.3 Lattice Factorization Design

In the polyphase view of filter banks (see Section 6.2.4), we saw that orthogonality of a two-channel filter bank is connected to its polyphase matrix being paraunitary. An elegant factorization result can be used in the design of that paraunitary matrix:
Chapter 6. Filter Banks: Building Blocks of Time-Frequency Expansions

<table>
<thead>
<tr>
<th>Step</th>
<th>Operation</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.</td>
<td>Choose (N), the number of zeros at (z = -1)</td>
</tr>
<tr>
<td>2.</td>
<td>(G(z) = (1 + z^{-1})^N R(z)), where (R(z)) is causal with powers ((0, -1, \ldots, -N + 1))</td>
</tr>
<tr>
<td>3.</td>
<td>(A(z) = (1 + z^{-1})^N (1 + z)^N Q(z)), where (Q(z)) is symmetric and has powers ((-N - 1), \ldots, 0, \ldots, (N + 1)))</td>
</tr>
<tr>
<td>4.</td>
<td>(A(z) + A(-z) = 2). This leads to (N) linear constraints on the coefficients of (Q(z))</td>
</tr>
<tr>
<td>5.</td>
<td>Solve the (N \times N) linear system for the coefficients of (Q(z))</td>
</tr>
<tr>
<td>6.</td>
<td>The minimum phase orthogonal filter is (G(z) = (1 + z^{-1})^N R(z))</td>
</tr>
</tbody>
</table>

Table 6.1: Design of orthogonal lowpass filters with maximum number of zeros at \(z = -1\).

<table>
<thead>
<tr>
<th>(L = 4)</th>
<th>(L = 6)</th>
<th>(L = 8)</th>
<th>(L = 10)</th>
<th>(L = 12)</th>
</tr>
</thead>
<tbody>
<tr>
<td>(g_0)</td>
<td>0.482962913</td>
<td>0.332670553</td>
<td>0.230377813309</td>
<td>0.160102398</td>
</tr>
<tr>
<td>(g_1)</td>
<td>0.836516304</td>
<td>0.806891509</td>
<td>0.714846570553</td>
<td>0.603829270</td>
</tr>
<tr>
<td>(g_2)</td>
<td>0.224143868</td>
<td>0.459877502</td>
<td>0.630880767930</td>
<td>0.724308528</td>
</tr>
<tr>
<td>(g_3)</td>
<td>-0.129409522</td>
<td>-0.135011020</td>
<td>-0.027983769417</td>
<td>0.138428146</td>
</tr>
<tr>
<td>(g_4)</td>
<td>-0.085441274</td>
<td>-0.187034811719</td>
<td>-0.242294887</td>
<td>-0.226264693965</td>
</tr>
<tr>
<td>(g_5)</td>
<td>0.035226292</td>
<td>0.030841381836</td>
<td>-0.032244870</td>
<td>-0.12976867567</td>
</tr>
<tr>
<td>(g_6)</td>
<td>0.032883011667</td>
<td>0.077571494</td>
<td>0.097501605587</td>
<td></td>
</tr>
<tr>
<td>(g_7)</td>
<td>-0.010597401785</td>
<td>-0.006241490</td>
<td>0.027522865530</td>
<td></td>
</tr>
<tr>
<td>(g_8)</td>
<td>-0.012580752</td>
<td>-0.031582039318</td>
<td></td>
<td></td>
</tr>
<tr>
<td>(g_9)</td>
<td>0.00335725</td>
<td>0.000553842201</td>
<td></td>
<td></td>
</tr>
<tr>
<td>(g_{10})</td>
<td>0.004777257511</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>(g_{11})</td>
<td>-0.001077301085</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table 6.2: Orthogonal filters with maximum number of zeros at \(z = -1\) (from [50]). For a lowpass filter of even length \(L = 2\ell\), there are \(L/2\) zeros at \(z = -1\).

**Theorem 6.4.** The polyphase matrix of any real-coefficient, causal FIR orthogonal two-channel filter bank can be written as

\[
\Phi_p(z) = R_0 \prod_{k=1}^{K-1} \begin{bmatrix} 1 & 0 \\ 0 & z^{-1} \end{bmatrix} R_k,
\]

(6.51)

where \(R_k, k = 0, 1, \ldots, K - 1\), are rotation matrices as in (1.13):

\[
R_k = \begin{bmatrix} \cos \theta_k & -\sin \theta_k \\ \sin \theta_k & \cos \theta_k \end{bmatrix}.
\]

(6.52)
6.4. Theory of Biorthogonal Two-Channel Filter Banks

Using the factored form, designing an orthogonal filter bank amounts to choosing a set of angles \((\theta_0, \theta_1, \ldots, \theta_{K-1})\). For example, the Haar filter bank in lattice form amounts to keeping only the constant-matrix term, \(R_0\) with \(\theta_0 = \pi/4\), as in (6.43). The factored form also suggests a structure, called a lattice, that is very convenient for hardware implementations (see Figure 6.9).

How do we impose particular properties, such as zeros at \(\omega = \pi\) for the lowpass filter \(G(z)\)? From (6.37a), we see that \(G(-1) = 0\) is equivalent to \(G_0(1) - G_1(1) = 0\). Similarly, we obtain \(\sqrt{2} = H(-1) = H_0(1) - H_1(1)\), so in terms of the polyphase matrix,

\[
\begin{bmatrix}
1 & -1 \\
0 & \sqrt{2}
\end{bmatrix}
\]

(6.53)

Now, using (6.51), \(\Phi_p(1)\) is simply the product of the rotations, and the requirement to satisfy (6.53) is

\[
\sum_{k=0}^{K-1} \theta_k = \frac{\pi}{4} + m2\pi,
\]

(6.54)

for some \(m \in \mathbb{Z}\). 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 6.3 (see Exercise 6.4). In general, design problems in lattice factored form are nonlinear and thus nontrivial.

6.4 Theory of Biorthogonal Two-Channel Filter Banks

While orthogonal filter banks have many nice features, there is one property that eludes them: When we restrict ourselves to FIR filters, none of the solutions have symmetric or antisymmetric impulse responses (except for the very short Haar filters), required for linear phase. This is one of the key motivations for looking beyond the orthogonal case, as well as for 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 1.22, we know that a system
Figure 6.10: A biorthogonal two-channel analysis/synthesis filter bank. The synthesis lowpass and highpass filters have z-transforms $G(z)$ and $H(z)$, respectively. The z-transforms of the analysis filters are $\tilde{G}(z)$ and $\tilde{H}(z)$. The output is the sum of the lowpass approximation $x_V$ and its highpass counterpart $x_W$.

$\{\varphi_k, \tilde{\varphi}_k\}$ constitutes a pair of biorthogonal bases of the Hilbert space $\ell^2(\mathbb{Z})$, if (a) they satisfy biorthogonality constraints (1.50a):

$$\langle \varphi_k, \tilde{\varphi}_i \rangle = \delta_{k-i} \iff \Phi\tilde{\Phi}^T = \tilde{\Phi}\Phi^T = I,$$

(6.55)

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 (b) it is complete:

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

(6.56)

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

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

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 (6.55) 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 6.10:

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

or

$$\begin{bmatrix} x_0 \\ x_1 \\ x_2 \\ x_3 \\ x_4 \\ \vdots \end{bmatrix} = \begin{bmatrix} \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ \cdots & g_0 & h_0 & 0 & 0 & 0 & 0 & \cdots \\ \cdots & g_1 & h_1 & 0 & 0 & 0 & 0 & \cdots \\ \cdots & g_2 & h_2 & g_0 & 0 & 0 & 0 & \cdots \\ \cdots & g_3 & h_3 & g_1 & h_0 & 0 & 0 & \cdots \\ \cdots & g_4 & h_4 & g_2 & h_1 & g_0 & h_0 & \cdots \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \end{bmatrix} \begin{bmatrix} \alpha_0 \\ \beta_0 \\ \alpha_1 \\ \beta_1 \\ \alpha_2 \\ \vdots \end{bmatrix} = \Phi X, \quad (6.57)$$
6.4. Theory of Biorthogonal Two-Channel Filter Banks

exactly the same as (6.8). As in (6.8), \( g_{n-2k} \) 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 the matrix \( \Phi \):

\[
\Phi = \{ \varphi_k \}_{k \in \mathbb{Z}} = \{ \varphi_{2k}, \varphi_{2k+1} \}_{k \in \mathbb{Z}} = \{ g_{n-2k}, h_{n-2k} \}_{k \in \mathbb{Z}},
\]

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 want this to implement a biorthogonal expansion, the transform coefficients \( \alpha_k \) and \( \beta_k \) are inner products between the dual basis sequences and the input signal: \( \alpha_k = \langle x, \tilde{\varphi}_{2k} \rangle, \beta_k = \langle x, \tilde{\varphi}_{2k+1} \rangle \). As we want these inner products to be implemented by filtering as in Figure 6.10, then

\[
\alpha_k = (\tilde{\varphi}_{2k} * x)_k = \langle x_n, \tilde{\varphi}_{2k-n} \rangle_n = \langle x, \tilde{\varphi}_{2k} \rangle \quad \leftrightarrow \quad \alpha = \tilde{\Phi}^T g x,
\]

\[
\beta_k = (\tilde{\varphi}_{2k+1} * x)_k = \langle x_n, \tilde{\varphi}_{2k+1-n} \rangle_n = \langle x, \tilde{\varphi}_{2k+1} \rangle \quad \leftrightarrow \quad \beta = \tilde{\Phi}^T h x,
\]

or, finally

\[
X = \tilde{\Phi}^T x.
\]

From above, we see that the dual basis sequences are

\[
\tilde{\Phi} = \{ \tilde{\varphi}_k \}_{k \in \mathbb{Z}} = \{ \tilde{\varphi}_{2k}, \tilde{\varphi}_{2k+1} \}_{k \in \mathbb{Z}} = \{ \tilde{g}_{2k-n}, \tilde{h}_{2k-n} \}_{k \in \mathbb{Z}},
\]

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 (6.9), here we have four biorthogonality relations:

\[
\langle g_n, \tilde{g}_{2k-n} \rangle_n = \delta_k, \quad (6.60a)
\]

\[
\langle h_n, \tilde{h}_{2k-n} \rangle_n = \delta_k, \quad (6.60b)
\]

\[
\langle h_n, \tilde{g}_{2k-n} \rangle_n = 0, \quad (6.60c)
\]

\[
\langle g_n, \tilde{h}_{2k-n} \rangle_n = 0. \quad (6.60d)
\]

A way to visualize these relations is given in Figure 6.11.

The purpose of this section is to explore families 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.
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.

### 6.4.1 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 6.12 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 (6.15a):

\[
x_V = \sum_{k \in \mathbb{Z}} \alpha_k g_{n-2k}.
\] (6.61a)

The highpass channel follows the lowpass exactly, substituting \( h \) for \( g \), \( \tilde{h} \) for \( \tilde{g} \), and \( x_W \) for \( x_V \) (see Figure 6.12). The highpass approximation \( x_W \) is

\[
x_W = \sum_{k \in \mathbb{Z}} \beta_k h_{n-2k}.
\] (6.61b)

**Biorthogonality of the Lowpass Filters** Since we started with a pair of biorthogonal bases, \( \{g_{n-2k}, \tilde{g}_{2k-n}\}_{k \in \mathbb{Z}} \) satisfies biorthogonality relations (6.60a). Similarly to the orthogonal case, these can be expressed in various domains as:

\[
\langle g_n, \tilde{g}_{2k-n} \rangle_n = \delta_k
\]

\[
\begin{align*}
D_2 \tilde{G}^T G U_2 &= I \\
G(z)\tilde{G}(z) + G(-z)\tilde{G}(-z) &= 2 \\
G(e^{j\omega})\tilde{G}(e^{j\omega}) + G(e^{j(\omega+\pi)})\tilde{G}(e^{j(\omega+\pi)}) &= 2
\end{align*}
\] (6.62)

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 columns of \( \tilde{G} U_2 \). 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.
6.4. Theory of Biorthogonal Two-Channel Filter Banks

Biorthogonality of the Highpass Filters

Matrix View
\[ \langle h_n, \tilde{h}_{2k-n} \rangle_n = \delta_k \]

\[ D_2 \bar{H}^T H U_2 = I \]
\[ H(z)\bar{H}(z) + H(-z)\bar{H}(-z) = 2 \]
\[ H(e^{j\omega})\bar{H}(e^{j\omega}) + H(e^{j(\omega+\pi)})\bar{H}(e^{j(\omega+\pi)}) = 2 \]

Crosscorrelation of the Lowpass Filters

In the orthogonal case, we rephrased relations as in (6.62) in terms of the autocorrelation of \( g \); here, as we have two sequences \( g \) and \( \tilde{g} \), we express it in terms of the crosscorrelation of \( g \) and \( \tilde{g} \), given by (2.59):

\[ \langle g_n, \tilde{g}_{2k-n} \rangle_n = c_{2k} = \delta_k \]

\[ D_2 C U_2 = I \]
\[ C(z) + C(-z) = 2 \]
\[ C(e^{j\omega}) + C(e^{j(\omega+\pi)}) = 2 \]

Crosscorrelation of the Highpass Filters

\[ \langle h_n, \tilde{h}_{2k-n} \rangle_n = c_{2k} = \delta_k \]

\[ D_2 C U_2 = I \]
\[ C(z) + C(-z) = 2 \]
\[ C(e^{j\omega}) + C(e^{j(\omega+\pi)}) = 2 \]

Projection Property of the Lowpass Channel

We now look at the lowpass channel as a composition of linear operators:

\[ x_V = P_V x = G U_2 D_2 \tilde{G}^T x. \]

While \( P_V \) is a projection, it is not an orthogonal projection:

\[ P_V^2 = (G U_2 D_2 \tilde{G}^T)(G U_2 D_2 \tilde{G}^T) = G U_2 D_2 G^T = P_V, \]
\[ P_V^T = (G U_2 D_2 \tilde{G})^T \neq G (U_2 D_2)^T G = \tilde{G} U_2 D_2 G^T. \]

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:

\[ V = \text{span}(\{g_{n-2k}\}_{k \in \mathbb{Z}}). \]

Note the interchangeable role 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 Chapter at a Glance.
Chapter 6. Filter Banks: Building Blocks of Time-Frequency Expansions

Projection Property of the Highpass Channel

The highpass projection operator $P_W$ is:

$$x_W = P_W x = HU_2D_2\tilde{H}^T x,$$  \hfill (6.68)

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

$$W = \text{span}\{h_{n-2k}\}_{k\in\mathbb{Z}}.$$  \hfill (6.69)

6.4.2 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 $\tilde{g}$ and $g$ satisfying (6.62), how can we choose $\tilde{h}$ and $h$ to complete the biorthogonal filter bank and thus implement a biorthogonal basis expansion? The sets of basis and dual basis sequences $\{g_{n-2k}, \tilde{h}_{n-2k}\}_{k\in\mathbb{Z}}$ and $\{\tilde{g}_{2k-n}, h_{2k-n}\}_{k\in\mathbb{Z}}$ must satisfy (6.60). We have already used (6.60a) in (6.62) and similarly for the highpass sequences in (6.63). What is left to use is that these lowpass and highpass sequences are orthogonal to each other as in (6.60c)-(6.60d).

Orthogonality of the Lowpass and Highpass Filters

We first summarize the properties the lowpass and highpass sequences must satisfy as follows:

$$\langle h_{n}, \tilde{g}_{2k-n} \rangle_n = 0 \quad \text{Matrix View}$$

$$D_2\tilde{G}^T HU_2 = 0$$

$$ZT \quad H(z)\tilde{G}(z) + H(-z)\tilde{G}(-z) = 0$$

$$\text{DTFT} \quad H(e^{j\omega})\tilde{G}(e^{j\omega}) + H(e^{j(\omega+\pi)})\tilde{G}(e^{j(\omega+\pi)}) = 0$$

$$= 0 \quad \text{(6.70)}$$

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

$$\langle g_{n}, \tilde{h}_{2k-n} \rangle_n = 0 \quad \text{Matrix View}$$

$$D_2\tilde{H}^T GU_2 = 0$$

$$ZT \quad G(z)\tilde{H}(z) + G(-z)\tilde{H}(-z) = 0$$

$$\text{DTFT} \quad G(e^{j\omega})\tilde{H}(e^{j\omega}) + G(e^{j(\omega+\pi)})\tilde{H}(e^{j(\omega+\pi)}) = 0$$

$$= 0 \quad \text{(6.71)}$$

6.4.3 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
6.4. Theory of Biorthogonal Two-Channel Filter Banks

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 6.5** (Biorthogonal two-channel filter bank). Given are two FIR filters \( g \) and \( \tilde{g} \), such that their impulse responses are orthogonal to each other and their even shifts as in (6.62), as well as \( L = 2\ell, \ell \in \mathbb{Z} \). Choose

\[
\begin{align*}
    h_n &= (-1)^n g_{n-2\ell+1} \quad \text{ZT,} \quad H(z) = -z^{-L+1}\tilde{G}(-z) \quad (6.72a) \\
    \tilde{h}_n &= (-1)^n g_{n+2\ell-1} \quad \text{ZT,} \quad \tilde{H}(z) = -z^{-L-1}G(-z) \quad (6.72b)
\end{align*}
\]

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 \( \{g, \tilde{g}\} \) and synthesis filters \( \{h, \tilde{h}\} \).

**Proof.** To prove the theorem, we must prove that 1. \( \{g_{n-2k}, h_{n-2k}\}_{k \in \mathbb{Z}} \) and \( \{\tilde{g}_{2k-n}, \tilde{h}_{2k-n}\}_{k \in \mathbb{Z}} \) satisfy biorthogonality relations (6.60); and 2. they are complete.

1. We first concentrate on proving (6.60). The first condition, (6.60a), is satisfied by assumption. To prove the second, (6.60b), we must prove one of the conditions in (6.63). The definitions of \( H \) and \( \tilde{H} \) in (6.72) imply

\[
H(z)\tilde{H}(z) = G(-z)\tilde{G}(-z).
\]

(6.73)

Thus,

\[
H(z)\tilde{H}(z) + H(-z)\tilde{H}(-z) = G(-z)\tilde{G}(-z) + G(z)\tilde{G}(z) \overset{(a)}{=} 2,
\]

where (a) follows from (6.62).

To prove (6.60c)–(6.60d), we must prove one of the conditions in (6.70)-(6.71), respectively. We prove (6.60c), (6.60d) follows similarly.

\[
H(z)\tilde{G}(z) + H(-z)\tilde{G}(-z) = -z^{-L-1}\tilde{G}(-z)G(z) - (-1)^{-L+1}z^{L-1}\tilde{G}(z)\tilde{G}(-z)
\]

\[
\overset{(a)}{=} -z^{-L-1}G(-z)\tilde{G}(z) + z^{L-1}\tilde{G}(z)\tilde{G}(-z) = 0,
\]

where (a) follows from the fact that \( L = 2\ell \) is even.

2. We prove completeness by proving that perfect reconstruction holds for any \( x \in \ell^2(\mathbb{Z}) \). What we do is find ZT-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.

\[
X_V(z) = \frac{1}{2}G(z) \left[ \tilde{G}(z)X(z) + \tilde{G}(-z)X(-z) \right], \quad (6.74)
\]

\[
X_W(z) = \frac{1}{2}H(z) \left[ \tilde{H}(z)X(z) + \tilde{H}(-z)X(-z) \right]. \quad (6.75)
\]
The output of the filter bank is the sum of $x_V$ and $x_W$:

$$X_V(z) + X_W(z) = \frac{1}{2} \left[ G(z)\tilde{G}(-z) + H(z)\tilde{H}(-z) \right] X(-z) \quad (6.76)$$

$$+ \frac{1}{2} \left[ G(z)\tilde{G}(z) + H(z)\tilde{H}(z) \right] X(z). \quad (6.77)$$

Substituting (6.72) into the above equation, we get:

$$A(z) = G(z)\tilde{G}(-z) + H(z)\tilde{H}(-z)$$

$$\equiv (a) G(z)\tilde{G}(z) + \left[ -z^{-L+1}\tilde{G}(-z) \right] \left[ -(z)^{-L-1}G(z) \right]$$

$$= [1 + (-1)^{L+1}] G(z)\tilde{G}(-z) \quad (b) 0,$$  

$$B(z) = G(z)\tilde{G}(z) + H(z)\tilde{H}(z)$$

$$\equiv (c) G(z)\tilde{G}(z) + \tilde{G}(-z)G(-z) \quad (d) 2,$$

where (a) follows from (6.72); (b) uses that $L = 2\ell$ is even; (c) follows from (6.72); and (d) follows from (6.62). Substituting this back into (6.76), we get

$$X_V(z) + X_W(z) = X(z), \quad (6.78)$$

completing the proof.

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 to each other 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. This was schematically shown in Figure 6.11.

### 6.4.4 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 (6.38) that the polyphase matrix of the synthesis bank is given by

$$\Phi_p(z) = \begin{bmatrix} G_0(z) & H_0(z) \\ G_1(z) & H_1(z) \end{bmatrix}, \quad G(z) = G_0(z) + z^{-1}G_1(z), \quad H(z) = H_0(z) + z^{-1}H_1(z). \quad (6.79)$$

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

$$\tilde{\Phi}_p(z) = \begin{bmatrix} \tilde{G}_0(z) & \tilde{H}_0(z) \\ \tilde{G}_1(z) & \tilde{H}_1(z) \end{bmatrix}, \quad \tilde{G}(z) = \tilde{G}_0(z) + z\tilde{G}_1(z), \quad \tilde{H}(z) = \tilde{H}_0(z) + z\tilde{H}_1(z). \quad (6.80)$$

---

40When 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”.
6.4. Theory of Biorthogonal Two-Channel Filter Banks

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:

\[ \Phi_p(z) \tilde{\Phi}_p^T(z) = I. \]  
(6.81)

From this

\[ \tilde{\Phi}_p(z) = (\Phi_p^T(z))^{-1} = \frac{1}{\det \Phi_p(z)} \begin{bmatrix} H_1(z) & -H_0(z) \\ -G_1(z) & G_0(z) \end{bmatrix}. \]  
(6.82)

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:

\[ \det \Phi_p(z) = G_0(z)H_1(z) - G_1(z)H_0(z) = z^{-k}. \]  
(6.83)

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**.

**Proposition 6.6** (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 is a common factor between \( G_0(z) \) and \( G_1(z) \), it will show up in the determinant. Sufficiency follows from the Euclidean algorithm or Bézout’s identity: given two coprime polynomials \( a(z) \) and \( b(z) \), the equation \( a(z)p(z) + b(z)q(z) = c(z) \) has a unique solution. Thus, choose \( c(z) = z^{-k} \) and then, the solution \( \{p(z), q(z)\} \) corresponds to the two polyphase components of \( H(z) \). \( \square \)

Note that the solution \( H(z) \) is not unique. Also, coprimeness of \( \{G_0(z), G_1(z)\} \) 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 6.5).

6.4.5 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.\(^{41}\) Thus, now we do just that: we build perfect reconstruction filter banks where all the filters involved are linear phase. Linear-phase filters were defined and discussed in (2.95).

\(^{41}\)If we allow filters to have complex-valued coefficients or if we lift the restriction of two channels, linear phase and orthogonality are both possible.
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 proposition, the proof of which is left as Exercise 6.6:

**Proposition 6.7.** In a two-channel, perfect reconstruction filter bank where all filters are linear phase, the synthesis filters have one of the following forms:

1. Both filters are symmetric and of odd lengths, differing by an odd multiple of 2.
2. One filter is symmetric and the other is antisymmetric; both lengths are even, and are equal or differ by an even multiple of 2.
3. 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.

We now 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:

**Proposition 6.8.** 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, (6.41)-(6.42) hold, and the filters are of even length. Therefore, following Proposition 6.7, one filter is symmetric and the other antisymmetric. Take the symmetric one, \( G(z) \) for example, and use (2.95)

\[
G(z) = G_0(z^2) + z^{-1}G_1(z^2) = z^{-L+1}G(z^{-1}) = z^{-L+1}(G_0(z^{-2}) + zG_1(z^{-2})) = z^{-L+2}G_1(z^{-2}) + z^{-1}(z^{-L+2}G_0(z^{-2})).
\]

This further means that for the polyphase components, the following hold:

\[
G_0(z) = z^{-L/2+1}G_1(z^{-1}), \quad G_1(z) = z^{-L/2+1}G_0(z^{-1}). \tag{6.84}
\]

Substituting the second equation from (6.84) into (6.41) we obtain

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

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 \( G_1(z) = \frac{1}{\sqrt{2}} z^{-k} \), and

\[
G(z) = \frac{1}{\sqrt{2}} (z^{-2l} + z^{-2k+1}), \quad H(z) = G(-z),
\]

yielding Haar filters \((m = k = 0)\) or trivial variations thereof.
6.5 Design of Biorthogonal Two-Channel Filter Banks

Figure 6.13: Various solutions to the perfect reconstruction filter bank problem. The Haar filters are the only solution that combines three desirable features, namely, orthogonality, finite impulse response, and symmetry or antisymmetry.

Figure 6.14: Design of filters based on factorization of $C(z)$.

The above result is highlighted in Figure 6.13, where various properties of real perfect reconstruction filter banks are given. While the outstanding features of the Haar filters make it a very special solution, Proposition 6.8 is a fundamentally negative result as the Haar filters have poor frequency localization and polynomial reproduction capabilities.

6.5 Design of Biorthogonal Two-Channel Filter Banks

Given that biorthogonal filters are less constrained than their orthogonal cousins, the design space is much more open. In both cases, one factors a (Laurent) polynomial $C(z)$ satisfying $C(z) + C(-z) = 2$ as in (6.64). In the orthogonal case, $C(z)$ was an autocorrelation, while in the biorthogonal case, it is a crosscorrelation and thus more general. In addition, the orthogonal case requires spectral factorization (or taking a “square root”), while in the biorthogonal case, any factorization will do. We show this in Figure 6.14.

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.

6.5.1 Factorization Design

From (6.62)-(6.64), we know that $C(z)$ 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 the ones used in orthogonal filter banks, in particular, the ones with a maximum number of zeros at $z = -1$. The factorization is done so that the resulting filters have linear phase.

Example 6.4 (Biorthogonal filter bank with linear-phase filters). We
reconsider Example 6.3, in particular \( C(z) \) given by

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

which satisfies \( C(z) + C(-z) = 2 \) by construction. This also means it satisfies (6.62) for any factorization of \( C(z) \) into \( G(z)G(z) \). 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. A 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 (6.72), 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 signals go through the lowpass channel only.

Another possible factorization of \( C(z) \) 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 translate, following (6.72b), into 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 6.5 (Design of 9/7 filter pair).** The next-higher order \( C(z) \) 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 6.3). The filters have odd length and even symmetry, and are part of the JPEG2000 image compression standard. 

6.5. Design of Biorthogonal Two-Channel Filter Banks

Table 6.3: Biorthogonal filters used in the still-image compression standard JPEG 2000. The lowpass filters are given; the highpass filters can be derived using (6.72a)–(6.72b). The first pair is from [3] and the second is from [108].

<table>
<thead>
<tr>
<th>n</th>
<th>( \tilde{g}_n )</th>
<th>( g_n )</th>
<th>( \tilde{g}_n )</th>
<th>( g_n )</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0.60294901823635790</td>
<td>1.11508705245699400</td>
<td>3/4</td>
<td>1</td>
</tr>
<tr>
<td>±1</td>
<td>0.26686411844287230</td>
<td>0.59127176311424700</td>
<td>1/4</td>
<td>1/2</td>
</tr>
<tr>
<td>±2</td>
<td>-0.07822326652898785</td>
<td>-0.05754352622849957</td>
<td>-1/8</td>
<td></td>
</tr>
<tr>
<td>±3</td>
<td>-0.01686411844287495</td>
<td>-0.09127176311424948</td>
<td>-1/4</td>
<td></td>
</tr>
<tr>
<td>±4</td>
<td>0.02674875741080976</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

6.5.2 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 (6.62) is satisfied, which can always be done if \( G(z) \) has coprime polyphase components (see Proposition 6.6). Then \( G(z) \) can be found by solving a linear system of equations.

**Example 6.6 (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.) Translating this polynomial constraint into a matrix equation gives the following system of equations for a length-5 symmetric filter \( c z^2 + b z + a + b z^{-1} + c z^{-2} \):

\[
\begin{bmatrix}
1 & 1/2 & 0 & 0 & 0 \\
0 & 1/2 & 1 & 1/2 & 0 \\
0 & 0 & 0 & 1/2 & 1
\end{bmatrix}
\begin{bmatrix}
c \\
b \\
a \\
b \\
c
\end{bmatrix}
= \begin{bmatrix}
0 \\
0 \\
1 \\
0 \\
0
\end{bmatrix}.
\]

The three equations give only two linearly independent constraints, for example:

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

If we impose that the filter has a zero at \( z = -1 \), we add the constraint

\[a - 2b + 2c = 0,
\]

The trivial solution \( \tilde{G}(z) = 1 \) is of no interest since it has no frequency selectivity.
leading to a unique solution

\[ \tilde{G}(z) = \frac{1}{8} \left(-z^2 + 2z + 6 + 2z^{-1} - z^{-2}\right). \]

All coefficients of the filter pair \((g, \tilde{g})\) are integer multiples of \(1/8\). This makes the analysis and synthesis exactly invertible even with finite-precision (binary) arithmetic. These filters are used in the JPEG2000 image compression standard; see Table 6.3.

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 length \(3 + 4m, m = 0, 1, 2, \ldots\) 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 \(G(z)\).

### 6.5.3 Lifting Design

While the original idea behind lifting was to build shift-varying perfect reconstruction filter banks, it has become more popular as it allows for building discrete-time bases with nonlinear operations. The trivial filter bank to start lifting is the polyphase transform which splits the signal into even- and odd-indexed components as in Figure 6.15. In the first lifting step, we use a prediction filter \(P\) to predict the odd samples from the even ones. The even samples remain unchanged, while the result of the predict filter applied to the even samples is subtracted from the odd samples yielding the highpass coefficients. In the second step, we use an update filter \(U\) to update the even samples based on the previously computed highpass coefficients. We start with a simple example.

**Example 6.7 (Haar filter bank obtained by lifting).** Call the even sequence \(x_0\) and the odd one \(x_1\). The purpose of the predict operator \(P\) is to predict odd samples based on the even ones. The simplest prediction is if we say that the even ones will predict that the odd ones are exactly the same, that is \(P(z) = 1\). The output of the highpass branch is thus \(X_1(z) - X_0(z)\) — a reasonable outcome. The purpose of the update operator \(U\) is to then update the even samples based on the
newly computed odd ones. As we are looking for a “lowpass”-like version in the other branch, the easiest it to subtract half of this difference from the even sequence, leading to $X_0(z) - 1/2(X_1(z) - X_0(z))$, that is, $1/2(X_0(z) + X_1(z))$, definitely a reasonable lowpass output. Within scaling, it is clear that the choice $P(z) = 1$, $U(z) = 1/2$ leads to the Haar filter bank.

More generally, let us now identify the polyphase matrix $\Phi_p(z)$:

$$
\Phi_g(z) = \alpha(z) - U(z)\beta(z),
\Phi_h(z) = \beta(z) + P(z)\Phi_g(z) = \beta(z) + P(z)(\alpha(z) - U(z)\beta(z))
= P(z)\alpha(z) + (1 - P(z)U(z))\beta(z),
$$

which we can write as

$$
\begin{bmatrix}
\Phi_g(z) \\
\Phi_h(z)
\end{bmatrix} =
\begin{bmatrix}
1 & -U(z) \\
P(z) & 1 - P(z)U(z)
\end{bmatrix}
\begin{bmatrix}
\alpha(z) \\
\beta(z)
\end{bmatrix} =
\Phi_p(z)
\begin{bmatrix}
\alpha(z) \\
\beta(z)
\end{bmatrix}.
\tag{6.85}
$$

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

$$
\tilde{\Phi}_p(z) = (\Phi_p^T(z))^{-1} =
\frac{1}{1 - P(z)U(z) + P(z)U(z)}
\begin{bmatrix}
1 - P(z)U(z) & -P(z) \\
P(z)U(z) & 1
\end{bmatrix},
\tag{6.86}
$$

and we immediately see that the $\text{det}(\Phi_p(z)) = 1$ which means that the inverse of $\Phi_p(z)$ does not involve actual inversion—one of the reasons why this technique is popular. Note, moreover, that we can write $\Phi_p$ as

$$
\Phi_p(z) =
\begin{bmatrix}
1 & -U(z) \\
P(z) & 1 - P(z)U(z)
\end{bmatrix}
= 
\begin{bmatrix}
1 & 0 \\
P(z) & 1
\end{bmatrix}
\begin{bmatrix}
1 & -U(z) \\
0 & 1
\end{bmatrix},
\tag{6.87}
$$

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:

$$
\begin{bmatrix}
1 & 0 \\
M_1 & 1
\end{bmatrix} \quad \text{or} \quad \begin{bmatrix}
1 & M_2 \\
0 & 1
\end{bmatrix},
$$

is

$$
\begin{bmatrix}
1 & 0 \\
-M_1 & 1
\end{bmatrix} \quad \text{or} \quad \begin{bmatrix}
1 & -M_2 \\
0 & 1
\end{bmatrix},
$$

respectively. In other words, to invert it, one needs only reverse the sequence of operations as shown in Figure 6.15. This is why this scheme allows for nonlinear operations; if $M_i$ is nonlinear, its inversion just amounts to reversing the sign in the matrix.

### 6.6 Theory of Stochastic Filter Banks

#### 6.7 Algorithms

Until now, we were only concerned with filtering an infinite sequence with a finite-length filter. But in signal processing as in life, most things are finite, that is, start
and end for some finite index. Thus, for this discussion, consider finite sequences of length $N$, or

$$x_n = \sum_{k=0}^{N-1} x_k \delta_{n-k},$$

as shown in Figure 6.16(a). To process this sequence with a filter or a filter bank, we need to make some assumptions about the signal outside the index range $\{0, 1, \ldots, N - 1\}$. We present several options and briefly discuss their pros and cons.
6.7. Algorithms

**The World Is Periodic**

From $x$, create a periodic signal $y$ so that

$$y_n = x_n \mod N.$$  

The attraction of this method is its simplicity, its direct relation with the discrete Fourier transform (which implicitly assumes a periodic-signal model) and the fact that it works for any signal length. The drawback is that the underlying signal is most likely not periodic, and thus, periodization creates artificial discontinuities at multiples of $N$; see Figure 6.16(b).

**The World Is Finite**

From the finite-length $x$, create $y$ as

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

Again, this is very simple and works for any signal length. But it too creates artificial discontinuities as in Figure 6.16(c). Also, by filtering, the signal is extended by the length of the filter (minus 1), which is often undesirable.

**The World Is Symmetric**

From $x$, create a version of double length,

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

which can then be periodized. As shown in Figure 6.16(d), this periodic signal of period $2N$ does not show the artificial discontinuities of the previous two cases. However, the signal 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 are variations on symmetric extension, depending on whether whole- or half-point symmetry is used. The formulation above is called *half-point symmetric* because there is (even) symmetry about the half-integer index value $N - \frac{1}{2}$. An alternative

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

gives a *whole-point symmetric* signal of length $2N - 1$ with even symmetry about the index value $N$.

---

43 Technically speaking, a discrete signal cannot be continuous or discontinuous. However, if the sequence is a densely sampled version of a smooth signal, periodization will destroy this smoothness.

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

Copyright 2008 M. Vetterli, J. Kovacevic, and V. K. Goyal
Chapter 6. Filter Banks: Building Blocks of Time-Frequency Expansions

The World Is Smooth

Another idea is to extend the signal by polynomial extrapolation. This is only lightly motivated at this point, but after we establish polynomial approximation properties of discrete wavelet transforms in Chapter 7, it will be clear that a signal extension by polynomial extrapolation will be a way to get zeros as detail coefficients. The order 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

We now summarize the main concepts and results seen in this chapter in a tabular form.

<table>
<thead>
<tr>
<th>Haar Two-Channel Filter Bank</th>
<th>Synthesis</th>
<th>Analysis</th>
</tr>
</thead>
<tbody>
<tr>
<td>Time domain</td>
<td>lowpass</td>
<td>highpass</td>
</tr>
<tr>
<td>$g_n$</td>
<td>$h_n$</td>
<td>$g_{-n}$</td>
</tr>
<tr>
<td>$\frac{1}{\sqrt{2}}(\delta_n + \delta_{n-1})$</td>
<td>$\frac{1}{\sqrt{2}}(\delta_n - \delta_{n-1})$</td>
<td>$\frac{1}{\sqrt{2}}(\delta_n + \delta_{n+1})$</td>
</tr>
<tr>
<td>ZT domain</td>
<td>lowpass</td>
<td>highpass</td>
</tr>
<tr>
<td>$G(z)$</td>
<td>$H(z)$</td>
<td>$G(z^{-1})$</td>
</tr>
<tr>
<td>$\frac{1}{\sqrt{2}}(1 + z^{-1})$</td>
<td>$\frac{1}{\sqrt{2}}(1 - z^{-1})$</td>
<td>$\frac{1}{\sqrt{2}}(1 + z)$</td>
</tr>
<tr>
<td>DTFT domain</td>
<td>lowpass</td>
<td>highpass</td>
</tr>
<tr>
<td>$G(e^{j\omega})$</td>
<td>$H(e^{j\omega})$</td>
<td>$G(e^{-j\omega})$</td>
</tr>
<tr>
<td>$\frac{1}{\sqrt{2}}(1 + e^{-j\omega})$</td>
<td>$\frac{1}{\sqrt{2}}(1 - e^{-j\omega})$</td>
<td>$\frac{1}{\sqrt{2}}(1 + e^{j\omega})$</td>
</tr>
</tbody>
</table>
### Lowpass Channel in a Two-Channel Orthogonal Filter Bank

<table>
<thead>
<tr>
<th>Domain</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Lowpass filter</strong></td>
<td>Impulse response orthogonal to its even translates</td>
</tr>
<tr>
<td>Original domain</td>
<td>$g_n$</td>
</tr>
<tr>
<td>Matrix domain</td>
<td>$G$</td>
</tr>
<tr>
<td>ZT domain</td>
<td>$G(z)$</td>
</tr>
<tr>
<td>DTFT domain</td>
<td>$G(e^{j\omega})$</td>
</tr>
<tr>
<td>Polyphase domain</td>
<td>$G(z) = G_0(z^2) + z^{-1}G_1(z^2)$</td>
</tr>
</tbody>
</table>

### Autocorrelation

<table>
<thead>
<tr>
<th>Domain</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Original domain</td>
<td>$a_n = \langle g_k, g_{k+n} \rangle_k$</td>
</tr>
<tr>
<td>Matrix domain</td>
<td>$A = G^T G$</td>
</tr>
<tr>
<td>ZT domain</td>
<td>$A(z) = G(z)G(z^{-1})$</td>
</tr>
<tr>
<td>DTFT domain</td>
<td>$A(e^{j\omega}) =</td>
</tr>
<tr>
<td></td>
<td>$A(e^{j\omega}) + A(e^{j(\omega+\pi)}) = 2$</td>
</tr>
</tbody>
</table>

### Projection onto smooth space $V = \text{span}\{g_{n-2k}\}_{k \in \mathbb{Z}}$

$x_V = P_V x$

$P_V = GU_2 D_2 G^T$

---

### Lowpass Channel in a Two-Channel Biorthogonal Filter Bank

<table>
<thead>
<tr>
<th>Domain</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Lowpass filters</strong></td>
<td>Impulse responses orthogonal to each other and their even translates</td>
</tr>
<tr>
<td>Original domain</td>
<td>$g_n, \tilde{g_n}$</td>
</tr>
<tr>
<td>Matrix domain</td>
<td>$G, \tilde{G}$</td>
</tr>
<tr>
<td>ZT domain</td>
<td>$G(z), \tilde{G}(z)$</td>
</tr>
<tr>
<td>DTFT domain</td>
<td>$G(e^{j\omega}), \tilde{G}(e^{j\omega})$</td>
</tr>
<tr>
<td>Polyphase domain</td>
<td>$G(z) = G_0(z^2) + z^{-1}G_1(z^2)$</td>
</tr>
</tbody>
</table>

### Crosscorrelation

<table>
<thead>
<tr>
<th>Domain</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Original domain</td>
<td>$c_n = \langle g_k, \tilde{g}_{k+n} \rangle_k$</td>
</tr>
<tr>
<td>Matrix domain</td>
<td>$C = G^T \tilde{G}$</td>
</tr>
<tr>
<td>ZT domain</td>
<td>$C(z) = G(z)\tilde{G}(z^{-1})$</td>
</tr>
<tr>
<td>DTFT domain</td>
<td>$C(e^{j\omega}) = G(e^{j\omega})\tilde{G}(e^{j\omega})$</td>
</tr>
</tbody>
</table>

### Projection onto smooth space $V = \text{span}\{g_{n-2k}\}_{k \in \mathbb{Z}}$

$x_V = P_V x$

$P_V = GU_2 D_2 \tilde{G}^T$
Chapter 6. Filter Banks: Building Blocks of Time-Frequency Expansions

Two-Channel Orthogonal Filter Bank

Relationship between lowpass and highpass filters

<table>
<thead>
<tr>
<th>Domain</th>
<th>Lowpass</th>
<th>Highpass</th>
</tr>
</thead>
<tbody>
<tr>
<td>Original</td>
<td>$\langle h_n, g_{n-2k} \rangle_{n=0} = 0$</td>
<td>$\langle h_n, g_{n+2k} \rangle_{n=0} = 0$</td>
</tr>
<tr>
<td>Matrix</td>
<td>$D_2H^T GU_2 = 0$</td>
<td>$D_2H^T GU_2 = 0$</td>
</tr>
<tr>
<td>ZT</td>
<td>$G(z)H(z^{-1}) + G(-z)H(-z^{-1}) = 0$</td>
<td>$G(z)H(z^{-1}) + G(-z)H(-z^{-1}) = 0$</td>
</tr>
<tr>
<td>DTFT</td>
<td>$G(e^{j\omega})H(e^{j\omega}) + G(e^{j(\omega+\pi)})H(e^{j(\omega+\pi)}) = 0$</td>
<td>$G(e^{j\omega})H(e^{j\omega}) + G(e^{j(\omega+\pi)})H(e^{j(\omega+\pi)}) = 0$</td>
</tr>
<tr>
<td>Polyphase</td>
<td>$G_0(z)G_1(z^{-1}) + H_0(z)H_1(z^{-1}) = 0$</td>
<td>$G_0(z)G_1(z^{-1}) + H_0(z)H_1(z^{-1}) = 0$</td>
</tr>
</tbody>
</table>

Sequences

<table>
<thead>
<tr>
<th>Domain</th>
<th>Basis</th>
</tr>
</thead>
<tbody>
<tr>
<td>Time</td>
<td>lowpass: ${g_{n-2k}}<em>{k \in \mathbb{Z}}$, highpass: ${h</em>{n-2k}}_{k \in \mathbb{Z}}$</td>
</tr>
<tr>
<td>ZT</td>
<td>$\Phi(z)$</td>
</tr>
<tr>
<td>DTFT</td>
<td>$\Phi(e^{j\omega})$</td>
</tr>
<tr>
<td>Polyphase</td>
<td>$\Phi_p(z)$</td>
</tr>
</tbody>
</table>

Filters

<table>
<thead>
<tr>
<th>Domain</th>
<th>Synthesis</th>
<th>Analysis</th>
</tr>
</thead>
<tbody>
<tr>
<td>Time</td>
<td>lowpass: $g_n = (-1)^n g_{n+2k-1}$, highpass: $G(z) = z^{-2k-1}G(-z^{-1})$</td>
<td>lowpass: $g_n = (-1)^n g_{n+2k-1}$, highpass: $G(z) = z^{-2k-1}G(-z^{-1})$</td>
</tr>
<tr>
<td>ZT</td>
<td>$\Phi(z)$</td>
<td></td>
</tr>
<tr>
<td>DTFT</td>
<td>$\Phi(e^{j\omega})$</td>
<td></td>
</tr>
<tr>
<td>Polyphase</td>
<td>$\Phi_p(z)$</td>
<td></td>
</tr>
</tbody>
</table>

Matrix representation

<table>
<thead>
<tr>
<th>Domain</th>
<th>Basis</th>
</tr>
</thead>
<tbody>
<tr>
<td>Matrix</td>
<td>$\Phi = \begin{bmatrix} \ldots &amp; g_{n-2k} &amp; h_{n-2k} &amp; \ldots \end{bmatrix}$</td>
</tr>
<tr>
<td>ZT</td>
<td>$\Phi(z)$</td>
</tr>
<tr>
<td>DTFT</td>
<td>$\Phi(e^{j\omega})$</td>
</tr>
<tr>
<td>Polyphase</td>
<td>$\Phi_p(z)$</td>
</tr>
</tbody>
</table>

Constraints

<table>
<thead>
<tr>
<th>Domain</th>
<th>Orthogonality relations</th>
<th>Perfect reconstruction</th>
</tr>
</thead>
<tbody>
<tr>
<td>Matrix</td>
<td>$\Phi^T \Phi = I$</td>
<td>$\Phi^T \Phi = I$</td>
</tr>
<tr>
<td>ZT</td>
<td>$\Phi(z)^T \Phi(z) = I$</td>
<td>$\Phi(z)^T \Phi(z) = I$</td>
</tr>
<tr>
<td>DTFT</td>
<td>$\Phi^T(e^{j\omega}) \Phi(e^{j\omega}) = I$</td>
<td>$\Phi(e^{j\omega})^T \Phi(e^{j\omega}) = I$</td>
</tr>
<tr>
<td>Polyphase</td>
<td>$\Phi_p(z)^T \Phi_p(z) = I$</td>
<td>$\Phi_p(z)^T \Phi_p(z) = I$</td>
</tr>
</tbody>
</table>
Chapter at a Glance

<table>
<thead>
<tr>
<th>Two-Channel Biorthogonal Filter Bank</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Relationship between lowpass and highpass filters</strong></td>
</tr>
<tr>
<td>Original domain</td>
</tr>
<tr>
<td>Matrix domain</td>
</tr>
<tr>
<td>ZT domain</td>
</tr>
<tr>
<td>DTFT domain</td>
</tr>
<tr>
<td>Polyphase domain</td>
</tr>
<tr>
<td><strong>Sequences</strong></td>
</tr>
<tr>
<td>Time domain</td>
</tr>
<tr>
<td>( g_n )</td>
</tr>
<tr>
<td>ZT domain</td>
</tr>
<tr>
<td>( G(z) )</td>
</tr>
<tr>
<td>DTFT domain</td>
</tr>
<tr>
<td>( G(e^{j\omega}) )</td>
</tr>
<tr>
<td><strong>Filters</strong></td>
</tr>
<tr>
<td>Matrix representation</td>
</tr>
<tr>
<td>Matrix domain</td>
</tr>
<tr>
<td>ZT domain</td>
</tr>
<tr>
<td>DTFT domain</td>
</tr>
<tr>
<td>Polyphase domain</td>
</tr>
<tr>
<td><strong>Constraints</strong></td>
</tr>
<tr>
<td>Matrix domain</td>
</tr>
<tr>
<td>ZT domain</td>
</tr>
<tr>
<td>DTFT domain</td>
</tr>
<tr>
<td>Polyphase domain</td>
</tr>
</tbody>
</table>
Historical Remarks

Filter banks have been popular in signal processing since the 1970s. At that time, the question of critically-sampled filter banks (where the number of channel samples per unit of time is conserved) came up in the context of subband coding of speech. In that method, a speech signal is divided up into downsampled frequency bands, which allow more powerful compression. However, downsampling can create aliasing, which is perceptually disturbing. In 1977, Esteban and Galand [68] came up with a simple and elegant solution to remove aliasing resulting in quadrature mirror filters (see Exercise 6.9 for a derivation of these filters and their properties). These filters do not allow perfect reconstruction, however, and a flurry of work followed to solve this problem. The orthogonal solution was found independently by Mintzer [117] and Smith and Barnwell [140] in the mid 1980’s, who presented design solutions as well. Vaidyanathan [151] made the connection to lossless systems, deriving the factorization of paraunitary matrices as in Section 6.3.3 and presented designs [155]. The design of filters with a maximum number of zeros at \( z = -1 \) was done by Daubechies [50] for wavelet purposes, and the solution goes back to Herrmann’s design of maximally flat FIR filters [86]. The equivalent IIR filter design problem leads to Butterworth filters, as derived by Herley and Vetterli [85].

The biorthogonal filter bank problem was solved by Vetterli [158, 159]. Solving for biorthogonal filter banks with maximum number of zeros at \( z = -1 \) was done by Cohen, Daubechies and Feauveau [38] and Vetterli and Herley [160]. The polyphase notation was used by many authors working on filter banks, but it really goes back to earlier work on transmultiplexers by Bellanger and Daguet [8]. The realization that perfect reconstruction subband coding can be used for perfect transmultiplexing appears in [159]. The idea of multichannel structures which can be inverted perfectly (including when quantization is used) goes back to ladder structures in filter design and implementation, in particular, the work of Bruckens and van den Enden, Marshall, Shah and Kalker [25, 115, 135]. The generalization of this idea under the name of lifting, was done by Sweldens [148] who derived a number of new schemes based on this concept (including nonlinear operators and nonuniform sampling).

Further Reading

Books and Textbooks  A few standard textbooks on filter banks exist written by Vaidyanathan [154], Vetterli and Kovačević [161], Strang and Nguyen [145], among others.

Theory and Design of \( N \)-Channel Filter Banks  Allowing the number of channels to be larger than two is yet another important generalization of two-channel filter banks. Numerous options are available, from directly designing \( N \)-channel filter banks, studied in detail by Vaidyanathan [152, 153], 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, with more freedom; for example, orthogonality and linear phase are much easier to achieve at the same time. The most obvious case is that of block transforms, those where the number of channels is equal
Exercises with Solutions

253
to the sampling factor \(N\) and equal to the filter length \(L = N\). These include well-known transforms such as the OFT, OFT. Those following closely are lapped orthogonal transforms (Blopts) proposed by Cassereau [33] and Malvar [113, 114], with the filter length \(L = 2N\); these were later extended to filter lengths \(L = kN\). Modulated filter banks mimicking the short-time Fourier transform as well as other types of modulations, also belong to this category, where all filters are obtained from a single prototype filter [125].

Theory and Design of Multidimensional Filter Banks

The first difference we encountered 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 [62], and have been used by Viscito and Allebach [162], Karlsson and Vetterli [93], Kovačević and Vetterli [102], Do and Vetterli [57], 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.

Theory and Design of IIR Filter Banks

While IIR filters are rightfully considered 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. In image processing they can still be used because of the finite length of the input signal, which 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 [128, 139, 85].

Theory and Design of Oversampled Filter Banks

While overcomplete filter banks implementing frame expansions exist, we have refrained from discussing them here as they will be the topic of Chapter 10.

Duality and Transmultiplexing

The scheme to a filter bank is known as a transmultiplexer, where one takes two signals and synthesizes a combined signal from which the two parts can be extracted perfectly. When the decomposition uses orthogonal filters, and there are typically many channels, then we have orthogonal frequency-division multiplexing (OFDM), which is the basis for many modulation schemes used in communications, such as 802.11. The analysis uses similar tools as for filter banks, see [159]. Solved Exercise 6.1 covers the analysis of the orthogonal case, while the biorthogonal case is left as Exercise 6.7. Exercise 6.8 considers frequency-division multiplexing with Haar filters.

Exercises with Solutions

6.1. Two-Channel Transmultiplexer

Given is a two-channel orthogonal filter bank. Reversing the order of the analysis and synthesis two-channel filter banks, we obtain a transmultiplexer. Call the inputs into the synthesis filter bank \(\alpha(z)\) and \(\beta(z)\) and call \(X(z)\) the output of the synthesis bank. Find the input-output relationship of the transmultiplexer and comment.

Solution:
A question we are trying to answer here is whether, given we know that a signal can be split into two bands and be perfectly reconstructed, we can do the converse, namely take two signals and synthesize a combined signal from which the two parts can be extracted perfectly. It is intuitive that such a scheme, shown in Figure E6.1-1, will work, and we derive this below. Beyond this algebraic result, the scheme is of great importance in practice, since it is the basis for frequency division multiplexing as will be shown as well.

When the decomposition uses orthogonal filters, and there are typically many channels, the above relations are satisfied if and only if $\alpha = \beta = 1$ and $\alpha = \beta = -1$. Of course, an easy way to state the result is to recall that matrices living in $\mathbb{R}^2$ have a left inverse if and only if they are square. Of course, in $\mathbb{R}^2$ the left inverse is also the right inverse.

Figure E6.1-1: Going from two-channel signals to a single upsampled signal, and back. This is the basis for frequency division multiplexing.

Figure E6.1-2: The synthesis filter bank combines two signals $x_1$ and $x_2$ living in orthogonal subspace $V$ and $W$. The analysis simply picks out these components.

\begin{align*}
\hat{\alpha}(z) & = \frac{1}{2} \begin{bmatrix} G(z^{-1/2}) & G(-z^{-1/2}) \\ H(z^{-1/2}) & H(-z^{-1/2}) \end{bmatrix} X(z^{1/2}) \\
\hat{\beta}(z) & = \frac{1}{2} \begin{bmatrix} G(z^{-1/2}) & G(-z^{-1/2}) \\ H(z^{-1/2}) & H(-z^{-1/2}) \end{bmatrix} X(-z^{1/2}) \quad \text{.} \tag{E6.1-2}
\end{align*}

Substituting (E6.1-1) into this, we see that the input-output relationship is given by the following matrix product (where we formally replaced $z^{1/2}$ by $z$):

\begin{align*}
\frac{1}{2} \begin{bmatrix} G(z^{-1}) & G(-z^{-1}) \\ H(z^{-1}) & H(-z^{-1}) \end{bmatrix} & = \begin{bmatrix} G(z) & H(z) \\ G(-z) & H(-z) \end{bmatrix} \quad \text{.} \tag{E6.1-3}
\end{align*}

For this to be identity, we require

\begin{align*}
G(z)G(z^{-1}) + G(-z)G(-z^{-1}) & = 2, \
H(z)H(z^{-1}) + H(-z)H(-z^{-1}) & = 2, \tag{E6.1-4a} \\
G(z)H(z^{-1}) + G(-z)H(-z^{-1}) & = 0, \tag{E6.1-4b}
\end{align*}

and a fourth relation which is simply the last one with $z$ replaced by $z^{-1}$. Of course, the above relations are satisfied if and only if $G(z)$ and $H(z)$ are orthogonal filters, since (E6.1-4a) and (E6.1-4b) are orthogonality with respect to even translates (see 6.18), and (E6.1-4c) is orthogonality of $\{g_n\}$ and $\{h_{n-2k}\}$ (see 6.25).

Thus, as to be expected, if we have a two-channel orthogonal filter bank, it does not matter if we cascade analysis followed by synthesis, or synthesis followed by analysis—both will lead to perfect reconstruction systems. The same applies to biorthogonal filter banks, which is left as Exercise 6.7. Of course, an easy way to state the result is to recall that when matrices are square, then the left inverse is also the right inverse.

Let us give some geometrical perspective. The output of the synthesis filter bank is the sum of two signals $x_1$ and $x_2$ living in $V = \text{span}(\{g_{n-2k}\}_{k \in \mathbb{Z}})$ and $W = \text{span}(\{h_{n-2k}\}_{k \in \mathbb{Z}})$, respectively. Because of the orthogonality of $g_n$ and $h_n$,

\begin{align*}
V \perp W. \quad \text{(E6.1-5)}
\end{align*}

The analysis filter bank on the right side of Figure E6.1-1 simply picks out the two components $x_1$ and $x_2$. This geometric view is shown in Figure E6.1-2.
Exercises

6.1. Even-Length Requirement for Orthogonality:
Given is an orthogonal FIR filter with impulse response \( g \) satisfying
\[
\langle g_n, g_{n-2k} \rangle = \delta_k.
\]
Prove that its length \( L \) is necessarily an even integer.

6.2. Zero-Moment Property of Highpass Filters:
Verify that a filter with the \( z \)-transform \( G(z) = (1 - z^{-1})^N R(z) \) has its first \((N - 1)\) moments zero, as in by (6.45).

6.3. Orthogonal Filters Are Maximally Flat:
Consider the design of an orthogonal filter with \( N \) zeros at \( z = -1 \). Its autocorrelation is of the form \( A(z) = (1 + z)^N (1 + z^{-1})^N Q(z) \) and satisfies \( A(z) + A(-z) = 2 \).
(i) Verify that \( A(e^{i\omega}) \) can be written as
\[
A(e^{i\omega}) = 2^N (1 + \cos \omega)^N Q(e^{i\omega}).
\]
(ii) Show that \( A(e^{i\omega}) \) and its \((2N - 1)\) derivatives are zero at \( \omega = \pi \), and show that \( A(e^{i\omega}) \) has \((2N - 1)\) zero derivatives at \( \omega = 0 \).
(iii) Show that the previous result leads to \( |G(e^{i\omega})| \) being maximally flat at \( \omega = 0 \) and \( \omega = \pi \), that is, having \((N - 1)\) zero derivatives.

6.4. Lattice Factorization Design:
Consider the factorization (6.51) for orthogonal filter banks.
(i) Prove that the sum of angles must satisfy (6.54) in order for the lowpass filter to have a zero at \( z = -1 \) or \( \omega = \pi \).
(ii) Consider the case \( K = 2 \), which leads to length 4 orthogonal filters. Show that imposing two zeros at \( z = -1 \) leads to \( \theta_0 = \pi/3 \) and \( \theta_1 = -\pi/12 \) as a possible solution and verify that this corresponds to Example 6.3.

6.5. Complementary Filters:
Using Proposition 6.6, prove that the filter \( H_0(z) = (1 + z^{-1})^N \) has always a complementary filter.

6.6. Structure of Linear-Phase Solutions:
Prove the three statements on the structure of linear phase solutions given in Proposition 6.7.

6.7. Biorthogonal Transmultiplexer:
Show that a perfect reconstruction analysis-synthesis system is also a perfect reconstruction synthesis-analysis system. Do this by mimicking the steps in Exercise 6.1, but generalizing to biorthogonal filter banks.

6.8. Frequency-Division Multiplexing with Haar Filters:
Consider the synthesis-analysis system as shown in Figure E6.1-2, and the Haar filters
\[
g_n = \frac{1}{\sqrt{2}} (\delta_n + \delta_{n-1}),
\]
\[
h_n = \frac{1}{\sqrt{2}} (\delta_n + \delta_{n-1}).
\]
(i) Characterize explicitly the spaces \( V \) and \( W \), and show that they are orthogonal.
(ii) Draw two example signals from \( V \) and \( W \), and their sum.
(iii) Verify explicitly the perfect reconstruction property, either by writing the \( z \)-transform relations or the matrix operators. The latter is more intuitive and thus to be preferred.
6.9. Quadrature Mirror Filters:

(i) Show that the following choice of filters:
   (i) Analysis: \( H(z), H(-z), \)
   (ii) Synthesis: \( H(z), -H(-z), \)
   where \( H(z) \) is a linear-phase FIR filter, automatically cancels aliasing in the output \( x_{\text{rec}}. \)

(iii) Give the input-output relationship.

(iv) Indicate why these cannot be perfect FIR reconstruction, and that FIR reconstruction would be unstable.
Chapter 7

Wavelet Series on Sequences

Contents

7.1 Introduction .................................................. 259
7.2 Iterated Filter Banks and Equivalent Filters .... 259
7.3 The DWT as an Orthonormal Expansion ......... 268
7.4 Properties of the DWT ................................. 269
7.5 Discrete Wavelets as a Basis for $\ell^2(\mathbb{Z})$ .... 275
7.6 Variations on the DWT ................................. 278
7.7 Processing Stochastic Signals ....................... 281
7.8 Algorithms .................................................. 281

Chapter at a Glance .......................................... 282

Historical Remarks .......................................... 282

Further Reading ............................................. 282

Exercises with Solutions .................................. 282

Exercises ......................................................... 282

If the split of a signal into two components is good, splitting into more components might be even better! This is obtained with multichannel filter banks, which come in various flavors. There are direct multichannel filter banks, with $N$ filters covering the entire spectrum, their outputs downsampled by $N$. There are also tree-based multichannel filter banks, where a two-channel filter bank is used as a building block for more complex structures. And among these tree structures, there is a particularly simple one that has some distinguishing features, from both mathematical and practical points of view. This elementary tree structure recursively splits the lowpass channel into highpass and lowpass components. In signal processing parlance, this is an octave-band filter bank. The input spectrum from 0 to $\pi$ is cut into a highpass part from $\pi/2$ to $\pi$, and the remainder is cut again into $\pi/4$ to $\pi/2$ and a new remainder, which at that point is a lowpass from 0 to $\pi/4$. As an example, iterating this three times leads to the following 4-channel split:

$$
\left[ \frac{0}{8}, \frac{\pi}{8} \right], \quad \left[ \frac{\pi}{8}, \frac{\pi}{4} \right], \quad \left[ \frac{\pi}{4}, \frac{\pi}{2} \right], \quad \left[ \frac{\pi}{2}, \pi \right].
$$
Figure 7.1: A two-channel analysis filter bank iterated three times to obtain one lowpass channel with support $[0, \pi/8]$, and three bandpass channels. (a) The filter bank. (b) The corresponding frequency division. [Note: In the second and third stages, the filters are mislabeled. $G(z)$ and $H(z)$ should be replaced with $G(z^{-1})$ and $H(z^{-1})$. Furthermore, for coherence with Fig. 6.1(a), the labels should be in time domain instead of $z$ domain.]

yielding a lowpass version and three bandpass versions, where each corresponds to an octave of the initial spectrum. This is shown in Figure 7.1.

Another way to state this 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 $\pi/2^i$, which means they have center frequency $3\pi/2^{i+2}$ and bandwidth $\pi/2^{i+1}$. The relative bandwidth $Q$ is then

$$Q = \frac{\omega_{\text{center}}}{\omega_{\text{bandwidth}}} = \frac{3\pi/2^{i+2}}{\pi/2^{i+1}} = \frac{3}{2}.$$ 

In classic circuit theory, the relative bandwidth is called the $Q$-factor, and the filter bank above has constant-$Q$ bandpass channels.

The unbalanced tree-structured filter bank shown (for three iterations) in Figure 7.1(a) is a central concept both in filter banks and in wavelets. This chapter is devoted to its study, its properties, and its geometrical interpretation. In wavelet parlance, when the lowpass filter is designed appropriately, the filter bank computes a discrete wavelet transform (DWT). Even more is true: the construction can be used to derive continuous-time wavelet bases, and the filter bank leads to an algorithm to compute wavelet series coefficients. The extension to continuous-time functions is the topic of Chapter 8.
7.1 Introduction

TBD: Haar

Chapter Outline

The structure of the present chapter is as follows. First, we build iterated filter banks from orthonormal two-channel filter banks and derive the equivalent filters in Section 7.2. Properties of the equivalent filters are studied, and the various subspaces spanned by the filters are derived. Section 7.3 establishes that the analysis filter bank implements an orthonormal expansion. Then, approximation and projection properties of orthonormal DWTs are derived in Section 7.4. Iterating the construction infinitely gives discrete wavelet bases for $\ell^2(\mathbb{Z})$, as discussed in Section 7.5. We prove a simple case of a discrete wavelet basis for $\ell^2(\mathbb{Z})$, namely the discrete Haar basis. Finally, Section 7.6 presents some additional topics including variations on orthonormal DWTs and complexity of DWT algorithms. A variation that is very important in practice, for example in image compression, is biorthogonal (non-orthonormal) DWTs. As in Chapter 6, we develop the orthonormal case thoroughly; extension to the biorthogonal case is then not difficult.

7.2 Iterated Filter Banks and Equivalent Filters

Start with a two-channel orthonormal filter bank with lowpass synthesis filter $g$. The filter’s $z$-transform $G(z)$ satisfies the usual orthonormality condition we saw in (6.16),

$$G(z)G(z^{-1}) + G(-z)G(-z^{-1}) = 2.$$  \hspace{1cm} (7.1)

The filter is FIR of length $L = 2l$, and the highpass is chosen as in (6.28):

$$H(z) = -z^{-L+1}G(-z^{-1}).$$  \hspace{1cm} (7.2)

The analysis filters are simply time-reversed versions of the synthesis filters. This is the building block from which we construct the DWT.

Definition 7.1 (Discrete wavelet transform). A $J$-stage (orthogonal) discrete wavelet transform (DWT) is a signal decomposition obtained by iterative use of $J$ two-channel (orthogonal) analysis filter banks, with all the iterations on the lowpass channel as shown for $J = 3$ in Figure 7.1(a). The corresponding inverse DWT is the transformation obtained by iterative use of the corresponding synthesis filter bank, as shown for $J = 3$ in Figure 7.2.

Consider the three-level decomposition in Figure 7.1. Just as in the basic two-channel case, the synthesis filters and their appropriate shifts are the basis vectors of the expansion. So, consider the synthesis corresponding to the three-level analysis, as shown in Figure 7.2. Some notation is needed at this point. The input and output are $x$ and $\hat{x}$, as usual. The lowpass channel is $X_g^{(3)}$, where the superscript $(3)$ indicates the number of iterations. Similarly, we have highpass channels $X_h^{(1)}$, $X_h^{(2)}$, and $X_h^{(3)}$. 

v3.0 [April 15, 2008] Do not distribute

Comments to book-errata@FourierAndWavelets.org
To start, let us verify that the overall system with analysis and synthesis as in Figures 7.1 and 7.2, is perfect reconstruction.

**Proposition 7.1 (Perfect Reconstruction of the DWT).** The analysis-synthesis DWT pair built from a two-channel orthonormal filter bank has the perfect reconstruction property.

**Proof.** We provide an informal proof based on the diagram in Figure 7.3. A two-channel filter bank is sketched as a block consisting of a fork and a merge. If it is perfect reconstruction (with zero delay\(^{45}\) and exact reconstruction), the elementary block can be replaced by a straight line. Repeating this procedure three times thus proves perfect reconstruction.

A key point is that the analysis filter bank is critically sampled, since each two-channel filter bank is. So, the system computes an expansion, and since each block itself is orthonormal, it is not hard to verify that the expansion is orthonormal. The question of interest is to find the basis vectors of this orthonormal expansion.

### 7.2.1 The Iterated Lowpass Filter

For this, it is necessary to consider equivalent filters corresponding to the synthesis tree. The idea is simple: if we have upsampling by 2 followed by filtering with \(G(z)\), the basis vectors are \(\{g_{n-2k}\}_{k \in \mathbb{Z}}\). What if we have this twice, as shown in Figure 7.4(a)? The trick is to use the identity in Figure 7.4(b) (we also saw that in Figure 2.22), which states that filtering with \(G(z)\) followed by upsampling by 2 is equivalent to upsampling followed by filtering with \(G(z^2)\). Applying this to the cascade in part (a) leads to the equivalent system in part (c) of Figure 7.4.

\(^{45}\) Note how it is critical for the elementary filter bank to have zero delay.
7.2. Iterated Filter Banks and Equivalent Filters

The basis vectors corresponding to this system are \( \{ g^{(2)}_{n-4k} \}_{k \in \mathbb{Z}} \), where \( g^{(2)} \) is the impulse response with \( z \)-transform \( G(z)G(z^2) \). More generally, a cascade of \( J \) times upsampling and filtering by \( G(z) \) leads to upsampling by \( 2^J \) followed by filtering with the equivalent filter

\[
G^{(J)}(z) = G(z)G(z^2)G(z^4) \cdots G(z^{2^{J-1}}) = \prod_{i=0}^{J-1} G(z^{2^i}), \tag{7.3}
\]

as shown in Figure 7.5. The output of this system is thus a linear combination of the vectors

\[
\left\{ g^{(J)}_{n-2^J k} \right\}_{k \in \mathbb{Z}}, \tag{7.4}
\]

where \( g^{(J)} \) is the impulse response of the equivalent filter with \( z \)-transform as given by (7.3). It is worthwhile to show a few properties of the iterated filter \( G^{(J)}(z) \).
Chapter 7. Wavelet Series on Sequences

Figure 7.5: Cascade of $J$ times upsampling and filtering. (a) Original system. (b) Equivalent system. (TBD: Upsampling by 2 should be upsampling by $2^J$.

Recursive Relation for Iterated Filters

Directly from its product form (7.3), we see that

$$G^{(J)}(z) = G(z) \cdot G^{(J-1)}(z^2) = G^{(J-1)}(z) \cdot G(z^{2^{J-1}}). \quad (7.5)$$

Orthogonality of Iterated Lowpass Filter

One 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 $2^J$. We now show this directly using the orthogonality of the basic lowpass filter $g$ to its even shifts.

**Proposition 7.2 (Orthogonality of Iterated Lowpass Filter).** The iterated lowpass filter

$$G^{(J)}(z) = \prod_{i=0}^{J-1} G(z^{2^i})$$

is orthonormal with respect to translates by $N = 2^J$.

**Proof.** We need to verify that

$$\langle g_{n-Nk}^{(J)}, g_n^{(J)} \rangle_n = \delta_k. \quad (7.6)$$

Similarly to what we did for $N = 2$ in (6.16), we see that the left-hand side of (7.6) is the autocorrelation of $g^{(J)}$ downsampled by $N$. Call this autocorrelation $p^{(J)}$. The right-hand side of (7.6) implies that all the coefficients of $p^{(J)}$ indexed by $k = 0 \mod N$ are zero, except at the origin. In other words, writing

$$P^{(J)}(z) = G^{(J)}(z)G^{(J)}(z^{-1}) \quad (7.7)$$

for the autocorrelation of the $J$th iterate, a direct translation of (7.6) to $z$-transform domain is

$$\frac{1}{N} \sum_{\ell=0}^{N-1} P^{(J)}(W_N^{\ell}z^{1/N}) = 1,$$
where \( W_N \) is the \( N \)th root of unity, \( W_N = e^{-j2\pi/N} \). Since this must hold for all \( z \), we can utilize it as

\[
\sum_{\ell=0}^{N-1} P^{(J)}(W_N^\ell z) = N. \tag{7.8}
\]

Note that from (7.5),

\[
P^{(J)}(z) = P(z)P^{(J-1)}(z^2).
\]

To prove (7.8), we will use induction. Assume the statement to be true for \( J-1 \), that is,

\[
\sum_{\ell=0}^{N/2-1} P^{(J-1)}(W_N^{\ell}z) = 2^{J-1} = \frac{N}{2}.
\]

Then

\[
\sum_{\ell=0}^{N-1} P^{(J)}(W_N^\ell z) = \sum_{\ell=0}^{N-1} P(W_N^\ell z)P^{(J-1)}(W_N^{2\ell}z^2)
\]

\[
= \sum_{\ell=0}^{N/2-1} \frac{(P(W_N^\ell z) + P(-W_N^\ell z)) P^{(J-1)}(W_N^{2\ell}z^2)}{2} \tag{a}
\]

\[
= 2 \sum_{\ell=0}^{N/2-1} P^{(J-1)}(W_N^{\ell}z^2) \tag{b}
\]

\[
= N, \tag{c}
\]

where (a) follows from splitting the sum into \( \ell = 0, \ldots, N/2-1 \) and \( \ell = N/2, \ldots, N-1 \) and using \( W_N^{N/2} = -1 \) and \( W_N^N = 1 \), (b) uses \( P(z) + P(-z) = 2 \) and \( W_N^{2\ell} = W_N^{\ell} \), and finally (c) uses orthogonality at level \( (J-1) \).

For the initialization of the induction, choose \( J = 2 \). Then \( P^{(J-1)}(z) = P(z) \), which satisfies \( P(z) + P(-z) = 2 \), and thus (7.8) is proved.

**Length of Iterated Filters**

It is also of interest to derive the length of the equivalent iterated filters. Recall that \( g \) is of length \( L \), that is, \( G(z) \) has powers of \( z^{-1} \) going from 0 to \( L-1 \). Then, from (7.3), \( G^{(J)}(z) \) has powers of \( z^{-1} \) going from 0 to \( (L-1)(1+2+\ldots+2^{J-1}) = (L-1)(2^J - 1) \). The length of \( g^{(J)} \) is therefore

\[
L^{(J)} = (L-1)(2^J - 1) + 1. \tag{7.9}
\]

Note that when \( L = 2 \) (the Haar case), this equals \( 2^J \). In general, \( L^{(J)} \) is upper bounded by

\[
L^{(J)} \leq (L-1)2^J. \tag{7.10}
\]
7.2.2 Bandpass Equivalent Filters

We are now ready to transform the synthesis filter bank in Figure 7.2 in a form that shows the basis vectors explicitly. Let us start from the top, where it is obvious that the vectors correspond to \( h \) and its even shifts, just as in the basic two-channel filter bank:

\[
\Phi^{(1)}_h = \{h_{n-2k}\}_{k \in \mathbb{Z}} = \{h_{n-2k}\}_{k \in \mathbb{Z}}.
\]

The second channel, with input \( X^{(2)}_h \), has an equivalent filter

\[
H^{(2)}(z) = H(z^2)G(z),
\]

with upsampling by 4 in front. This is obtained by shifting the upsampler to the left, across \( H(z) \). The basis vectors are

\[
\Phi^{(2)}_h = \{h_{n-4k}\}_{k \in \mathbb{Z}}.
\]

The equivalent filter for the channel with \( X^{(3)}_h \) as input is

\[
H^{(3)}(z) = G(z)G(z^2)H(z^4),
\]

with upsampling by 8, or basis vectors

\[
\Phi^{(3)}_h = \{h_{n-8k}\}_{k \in \mathbb{Z}},
\]

and finally, the lowpass branch with \( X^{(3)}_g \) as input has an equivalent filter

\[
G^{(3)}(z) = G(z)G(z^2)G(z^4),
\]

which follows directly from (7.3). The basis vectors are

\[
\Phi^{(3)}_g = \{g_{n-8k}\}_{k \in \mathbb{Z}}.
\]

The complete set of basis vectors is thus:

\[
\Phi = \{h^{(1)}_{n-2k}, h^{(2)}_{n-4k}, h^{(3)}_{n-8k}, g^{(3)}_{n-8k}\}_{k \in \mathbb{Z}}. \quad (7.11)
\]

We can now draw the filter bank shown in Figure 7.6, which is completely equivalent to the discrete wavelet synthesis we started with in Figure 7.2. The important point is that all upsampling is done right at the input, and the equivalent filters indicate the basis vectors explicitly. It is time to put the above transformations into practice by working through our favorite example—Haar.

Example 7.1 (The Haar DWT). Recall the Haar orthogonal filter bank in Chapter at a Glance of Chapter 6, with synthesis filters

\[
G(z) = \frac{1}{\sqrt{2}}(1 + z^{-1}), \quad H(z) = \frac{1}{\sqrt{2}}(1 - z^{-1}).
\]
Consider the three-level DWT shown in Figure 7.2 with the equivalent form depicted in Figure 7.6. The four equivalent filters are

\[
\begin{align*}
H^{(1)}(z) &= H(z) = \frac{1}{\sqrt{2}}(1 - z^{-1}), \\
H^{(2)}(z) &= G(z)H(z^2) = \frac{1}{2}(1 + z^{-1})(1 - z^{-2}) \\
 &= \frac{1}{2}(1 + z^{-1} - z^{-2} - z^{-3}), \\
H^{(3)}(z) &= G(z)G(z^2)H(z^4) = \frac{1}{2\sqrt{2}}(1 + z^{-1})(1 + z^{-2})(1 - z^{-4}) \\
 &= \frac{1}{2\sqrt{2}}(1 + z^{-1} + z^{-2} + z^{-3} - z^{-4} - z^{-5} - z^{-6} - z^{-7}), \\
G^{(3)}(z) &= G(z)G(z^2)G(z^4) = \frac{1}{2\sqrt{2}}(1 + z^{-1})(1 + z^{-2})(1 + z^{-4}) \\
 &= \frac{1}{2\sqrt{2}}(1 + z^{-1} + z^{-2} + z^{-3} + z^{-4} + z^{-5} + z^{-6} + z^{-7}).
\end{align*}
\]

Our matrix $\Phi$ now looks as follows:

\[
\Phi = \begin{bmatrix}
\vdots & \Phi_0 & \Phi_0 \\
\Phi_0 & \Phi_0 & \Phi_0 \\
\vdots & \vdots & \vdots
\end{bmatrix},
\]

that is, it is a block-diagonal matrix. This happens only when the length of the filters in the original filter banks is equal to the downsampling factor. The block in this case is of size $8 \times 8$, since the same structure repeats itself after every 8 samples. That is, $h^{(3)}$ and $g^{(3)}$ are repeated every 8 samples, $h^{(2)}$ repeats itself every 4 samples, while $h^{(1)}$ repeats itself every 2 samples. Then, there will be 2
Chapter 7. Wavelet Series on Sequences

Figure 7.7: Discrete-time Haar basis. Eight vectors forming $\Phi_0$ are given: (a) $h^{(1)}$ and its shifts by 2. (b) $h^{(2)}$ and its shift by 4. (c) $h^{(3)}$. (d) $g^{(3)}$.

instances of $h^{(2)}$ in block $\Phi_0$ and 4 instances of $h^{(1)}$. The matrix $\Phi_0$ is

$$
\Phi_0 = \begin{bmatrix}
1 & 0 & 0 & 0 & 1 & 0 & 1 & 1 \\
-1 & 0 & 0 & 0 & 1 & 0 & 1 & 1 \\
0 & 1 & 0 & 0 & -1 & 0 & 1 & 1 \\
0 & -1 & 0 & 0 & -1 & 0 & 1 & 1 \\
0 & 0 & 1 & 0 & 0 & 1 & -1 & 1 \\
0 & 0 & -1 & 0 & 0 & 1 & -1 & 1 \\
0 & 0 & 0 & -1 & 0 & -1 & -1 & 1 \\
0 & 0 & 0 & -1 & 0 & -1 & -1 & 1
\end{bmatrix}
\begin{bmatrix}
\frac{1}{\sqrt{2}} \\
\frac{1}{\sqrt{2}} \\
\frac{1}{\sqrt{2}} \\
\frac{1}{\sqrt{2}} \\
\frac{1}{\sqrt{2}} \\
\frac{1}{\sqrt{2}} \\
\frac{1}{2\sqrt{2}} \\
\frac{1}{2\sqrt{2}}
\end{bmatrix},
$$

where the right-side matrix contains the normalization. The basis vectors are the columns of the above matrix and all their shifts by 8 (which corresponds to other blocks $\Phi_0$ in $\Phi$). It is easy to verify that the above matrix is orthonormal, a property that holds in general as we shall see. These impulse responses are depicted in Figure 7.7.

A general expression for the lowpass equivalent filter $G^{(i)}(z)$ was derived in (7.3). Let us now write the counterpart for the bandpass equivalent filters. The filter $H^{(1)}(z)$ corresponds to a branch with a highpass filter followed by $(i - 1)$ lowpass filters (always with upsampling by 2 in between). The $(i - 1)$th lowpass filter branch has an equivalent filter $G^{(i - 1)}(z)$, preceded by upsampling by $2^{i - 1}$. Passing this upsampling across the initial highpass filter changes $H(z)$ into $H(z^{2^{i - 1}})$ and

$$
H^{(i)}(z) = H(z^{2^{i - 1}})G^{(i - 1)}(z), \quad i = 1, \ldots, J,
$$

follows. The basis vectors correspond to the impulse responses and the shifts given.
7.2. Iterated Filter Banks and Equivalent Filters

Figure 7.8: Impulse responses for a three-level DWT based on an orthonormal lowpass filter of length 4 with two zeros at $z = -1$. The basis vectors are the impulse responses $h^{(1)}$, $h^{(2)}$, $h^{(3)}$ and $g^{(3)}$, together with shifts by multiples of 2, 4, 8 and 8. [Note: These are discrete impulse responses. Connecting the dots may obscure things.]

by the upsampling. These upsampling factors are $2^j$ for the lowpass branch and $2^i$ for the bandpass branches.

**Example 7.2 (The Three-Level DWT with Daubechies Filters).** Let us look at Example 6.3, where a 4-tap orthogonal filter with two zeros at $z = -1$ was designed. The $z$-transform of this filter is

$$G(z) = \frac{1}{4\sqrt{2}} \left[ (1 + \sqrt{3}) + (3 + \sqrt{3})z^{-1} + (3 - \sqrt{3})z^{-2} + (1 - \sqrt{3})z^{-3} \right]. \quad (7.14)$$

A three-level DWT has the impulse responses of the following filters:

$$H^{(1)}(z) = H(z) = z^{-3}G(-z^{-1}),$$
$$H^{(2)}(z) = G(z)H(z^2),$$
$$H^{(3)}(z) = G(z)G(z^2)H(z^4),$$
$$G^{(3)}(z) = G(z)G(z^2)G(z^4),$$

together with shifts by multiples of 2, 4, 8 and 8, respectively. These impulse responses are shown in Figure 7.8.
7.3 The DWT as an Orthonormal Expansion

The iteration formulas derived above allow us to derive some useful properties of the DWT.

**Proposition 7.3 (The DWT).** Given is an orthogonal lowpass filter $G(z)$ satisfying the quadrature formula (7.1), and the highpass is chosen as in (7.2). Then, a $J$-level discrete wavelet transform computes an orthonormal expansion

$$x_n = \sum_{k \in \mathbb{Z}} X^{(J)}_g g_{n-2^Jk} + \sum_{i=1}^{J} \sum_{k \in \mathbb{Z}} X^{(i)}_h h^{(i)}_{n-2^Jk},$$

with

$$X^{(J)}_g = \langle g^{(J)}_{n-2^Jk}, x_n \rangle_n, \quad X^{(i)}_h = \langle h^{(i)}_{n-2^Jk}, x_n \rangle_n,$$

where $X^{(i)}_h$ are called the wavelet coefficients and $X^{(J)}_g$ are the scaling coefficients.

**Proof.** In Proposition 7.1 we saw the perfect reconstruction property. To prove that the set of filters and their translates forms an orthonormal basis, we first prove that they form an orthonormal set; that is, for all $k, \ell \in \mathbb{Z}$:

$$\langle g^{(J)}_{n-2^Jk}, g^{(J)}_{n-2^J\ell} \rangle_n = \delta_{k-\ell},$$

$$\langle g^{(J)}_{n-2^Jk}, h^{(i)}_{n-2^J\ell} \rangle_n = 0, \quad i = 1, 2, \ldots, J,$$

$$\langle h^{(i)}_{n-2^Jk}, h^{(j)}_{n-2^J\ell} \rangle_n = \delta_{i-j}\delta_{k-\ell}, \quad i = 1, 2, \ldots, J, \quad j = 1, 2, \ldots, J.$$

The first assertion was proved in Proposition 7.2. To gain some intuition, consider (7.18) for $J = i = 3$. We thus consider the two lowest branches of the equivalent filter bank in Figure 7.4, with the two filters:

$$H^{(3)}(z) = G(z)G(z^2)H(z^4), \quad G^{(3)}(z) = G(z)G(z^2)G(z^4),$$

and we need to show that their impulse responses are orthogonal with respect to shifts by 8. Note that the common part $G^{(2)}(z) = G(z)G(z^2)$ is orthonormal with respect to shifts by 4. The equivalent filters have impulse responses

$$h^{(3)}_n = \sum_{i=0}^{L-1} h gn_{n-4i}, \quad g^{(3)}_n = \sum_{i=0}^{L-1} g gn_{n-4i}.$$
7.4. Properties of the DWT

since there is a term $H(z^4)$ and $G(z^4)$ in the equivalent filters. We now verify orthogonality,

$$\langle h_{n-8\ell}, g_n \rangle_n = \left( \sum_{i=0}^{L-1} h_{i} g_{n-2\ell+4i} \right) \sum_{j=0}^{L-1} g_{n-4j}$$

$$= \sum_{i=0}^{L-1} \sum_{j=0}^{L-1} h_{i} g_{j} \langle g_{n-2\ell+4i}, g_{n-4j} \rangle_n$$

$$= (a) \sum_{i=0}^{L-1} h_{i} g_{2\ell+i} (b) = 0,$$

where in (a) we used the fact that the inner product is zero, unless $8\ell + 4i = 4j$ or $j = 2\ell + i$; and in (b) we used the fact that $h$ and $g$ are orthogonal with respect to even shifts.

Figure 7.9 gives a graphical representation. The upsamplers have been shifted to highlight the common part $G^{(2)}(z)$. Consider an input $X^{(3)} g_n = \delta_n$. The corresponding output is the impulse response of the filter obtained by convolving $g$ upsamplled by 4 and $g^{(2)}$ as follows:

$$\begin{bmatrix} \cdots & 0 & g_0 & 0 & 0 & 0 & g_1 & 0 & 0 & 0 & g_2 & 0 & \cdots \end{bmatrix} \ast g^{(2)}.$$  

Similarly, the input $X^{(3)} h_n = \delta_n$ leads to the following output:

$$\begin{bmatrix} \cdots & 0 & h_0 & 0 & 0 & 0 & h_1 & 0 & 0 & 0 & h_2 & 0 & \cdots \end{bmatrix} \ast g^{(2)}.$$  

Since $g^{(2)}$ is orthogonal to shifts by multiples of 4, only the orthogonality of the sequences corresponding to $H(z^4)$ and $G(z^4)$ matters, and this with respect to shifts by multiples of 8. The latter follows since $H(z)$ and $G(z)$ are orthogonal for even shifts. It is easy to see how the argument extends to any $J = i$ in (7.18). Finally, if $i < J$, the orthogonality at stage $i$ between highpass and lowpass branch is sufficient. This is because previous lowpass filters (for $J$ up to $i+1$) simply create linear combinations of $g^{(i)}$, which is orthogonal to $h^{(i)}$. Finally, (7.19) can be shown using similar arguments, and we leave this as an exercise.

To complete the orthonormal basis proof, we need to show completeness. This can be done by verifying Parseval’s formula. Here, we follow an argument similar to the one used to prove Proposition 7.1. Namely, Parseval is verified for any elementary two-channel block, so recursively, it is verified between the input and the channels of the DWT, completing the proof that we have a discrete-time orthonormal expansion.

---

7.4 Properties of the DWT

So far, we have verified perfect reconstruction and orthonormality. We now consider projection and approximation properties of the DWT. The former follows directly
from the orthonormal expansion performed by the transform, while the latter comes from the polynomial approximation property of two-channel orthogonal filter banks seen in Section 6.2.5.

As an example, consider our usual three-level decomposition and reconstruction, with the equivalent filters $h^{(1)}, h^{(2)}, h^{(3)}$ and $g^{(3)}$. Define the following spaces:

$$W_1 = \text{span} \left\{ h^{(1)}_{n-2\ell} \right\}_{\ell \in \mathbb{Z}}, \quad W_2 = \text{span} \left\{ h^{(2)}_{n-4\ell} \right\}_{\ell \in \mathbb{Z}},$$

$$W_3 = \text{span} \left\{ h^{(3)}_{n-8\ell} \right\}_{\ell \in \mathbb{Z}}, \quad V_3 = \text{span} \left\{ g^{(3)}_{n-8\ell} \right\}_{\ell \in \mathbb{Z}}.$$

Clearly, following (7.18)–(7.19), these spaces are orthogonal to each other. And, because we have perfect reconstruction, they sum up to $\ell^2(\mathbb{Z})$, or

$$\ell^2(\mathbb{Z}) = V_3 \oplus W_3 \oplus W_2 \oplus W_1.$$

This is shown pictorially in Figure 7.10, where $x^{(3)}_g$ denotes the projection onto $V_3$, and $x^{(i)}_h$ denotes the projection onto $W_i, i = 1, 2, 3$. Figure 7.11 shows an example input signal and the resulting channel signals. 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.

**Projection Property**

In general, we have the following proposition for which we skip the proof as it follows from the previous results, especially the orthonormal basis property proved in Proposition 7.3.

**Proposition 7.4 (Projection Property of the DWT).** A $J$-level DWT constructed with an orthonormal filter bank projects the input onto one lowpass space

$$V_J = \text{span} \left\{ g^{(J)}_{n-2\ell} \right\}_{\ell \in \mathbb{Z}},$$

and $J$ bandpass spaces

$$W_i = \text{span} \left\{ h^{(i)}_{n-2\ell} \right\}_{\ell \in \mathbb{Z}}, \quad i = 1, \ldots, J,$$
7.4. Properties of the DWT

Figure 7.10: Projection of the input signal onto $V_3, W_3, W_2$ and $W_1$, respectively, and perfect reconstruction as the sum of the projections.

Figure 7.11: Approximation properties of the discrete wavelet transform. (a) Original signal $x$ with various components. (b) The highpass channel after the first iteration $X_h^{(1)}$. (c) The highpass channel after the second iteration $X_h^{(2)}$. (d) The highpass channel after the third iteration $X_h^{(3)}$. (e) The lowpass channel $X_g^{(3)}$. [Note: The labels on the bottom four subplots should be $x_h^{(1)}, x_h^{(2)}, x_h^{(3)}$, and $x_g^{(3)}$; note the lower case for being back to the original sampling rate.]
where \( g^{(J)} \) and \( h^{(i)} \) are the equivalent filters given in (7.3) and (7.13), respectively. The input space \( \ell^2(\mathbb{Z}) \) is split into the following \((J + 1)\) spaces:

\[
\ell^2(\mathbb{Z}) = V_J \oplus W_J \oplus W_{J-1} \oplus \ldots \oplus W_1.
\]

**Approximation of Polynomials**

Next, we add a constraint on the lowpass filter \( g \) in the orthogonal filter bank, namely it must have \( m \geq 1 \) zeros at \( z = -1 \), or

\[
G(z) = (1 + z^{-1})^m R(z),
\]

as we did in Section 6.2.5 and in particular in Proposition 6.3. Note that \( R(z)|_{z=1} \) cannot be zero because of the orthogonality constraint (7.1). Remember that the highpass filter, being a modulated version of the lowpass as in (7.3), has \( m \) zeros at \( z = 1 \). In other words, it annihilates polynomials up to degree \((m - 1)\) since it takes an \( m \)th order difference of the sequence.

In the DWT, each bandpass channel annihilates polynomials, which are therefore carried by the lowpass branch. Or, if \( x \) is a polynomial signal of degree smaller than \( m \), the channel signals \( X^{(i)}_h \) are all zero, and the polynomial sequence is projected onto \( V_J \), the lowpass approximation space. This means that

\[
n^k = \sum_{i \in \mathbb{Z}} X^{(J)}_{g_i} g^{(J)}_{n-2^J i} \quad 0 < k < m,
\]

that is, the equivalent lowpass filter reproduces polynomials up to degree \((m - 1)\). In the orthogonal case, the coefficients follow from the orthogonal projection formula,

\[
X^{(J)}_{g_i} = \sum_{n \in \mathbb{Z}} g^{(J)}_{n-2^J i} n^k.
\]

**Example 7.3 (Reproduction of Discrete Polynomials).** Consider our usual 4-tap orthogonal filter as given in (7.14). Then, consider its sixth iteration, or

\[
G^{(6)}(z) = G(z)G(z^2)G(z^4)G(z^8)G(z^{16})G(z^{32}).
\]

According to (7.9), the equivalent filter is of length

\[
L^{(6)} = (L - 1)(2^J - 1) + 1 = 3 \cdot 63 + 1 = 190
\]

The impulse response is shown in Figure 7.12(a) while the reproduction of the linear polynomial is depicted in part (b) of the same figure.

**Effect of Singularities**

We saw that polynomials are represented by the lowpass filter. What about discontinuities? For simplicity, we consider a single nonzero sample in the input, at
7.4. Properties of the DWT

![Figure 7.12: Polynomial reproduction. (a) Equivalent filter impulse response after 6 iterations. (b) Reproduction of the linear function. [Note that the reproduction in (b) is exact over a certain range because a finite number of terms was used.]](image)

some \( k \), or \( x_n = \delta_{n-k} \). This discrete-time Dirac now excites each equivalent filter impulse response. In the analysis, we have equivalent filters at level \( i \) of length

\[
(L - 1)(2^i - 1) + 1,
\]

which are then downsampled by \( 2^i \) (see Figure 7.10 for an illustration with \( J = 3 \)). This indicates that a single nonzero input creates at most \( L - 1 \) nonzero coefficients in each channel.

Let us go back to our favorite example:

**Example 7.4 (The Haar DWT (cont’d)).** Consider the Haar case as in Example 7.1, but now up to iteration \( i \). The lowpass equivalent filter has the \( z \)-transform

\[
G^{(i)}(z) = \prod_{k=0}^{i-1} G(z^{2^k}) = \frac{1}{2^{i/2}} \sum_{\ell=0}^{2^i-1} z^{-\ell},
\]

that is, it is a length-\( 2^i \) averaging filter

\[
g_n^{(i)} = \frac{1}{2^{i/2}} \sum_{k=0}^{2^i-1} \delta_{n-k}.
\]

To verify this formula, use the recursion

\[
G^{(i)}(z) = G(z^{2^{i-1}})G^{(i-1)}(z),
\]

where

\[
G(z^{2^{i-1}}) = \frac{1}{\sqrt{2}}(1 + z^{-2^{i-1}}),
\]
and $G^{(i-1)}(z)$ is a length-$2^{i-1}$ averaging filter. The bandpass equivalent filter follows from

$$H^{(i)}(z) = H(z^{2^{i-1}})G^{(i-1)}(z) = \frac{1}{\sqrt{2}} \left( G^{(i-1)}(z) - z^{-2^{i-1}} G^{(i-1)}(z) \right), \quad (7.21)$$

with an impulse response

$$h^{(i)}_n = \frac{1}{2^{i/2}} \left( \sum_{k=0}^{2^{i-1}} \delta_{n-k} - \sum_{k=2^{i-1}}^{2^{i-1}} \delta_{n-k} \right).$$

Let us start by considering an impulse at some instant $k$. Convolution with the analysis filter $H^{(i)}(z^{-1})$ generates $2^i$ coefficients, and downsampling by $2^i$ leaves a single coefficient of size $2^{-i/2}$, as indicated earlier. Next, look at a step function delayed by $k$. First, a single wavelet coefficient will be different from zero at each scale, namely the one corresponding to the wavelet that straddles the discontinuity, see Figure 7.13. At scale $2^i$, this corresponds to the wavelet with support from $2^i \lfloor k/2^i \rfloor$ to $2^i \lfloor (k/2^i) + 1 \rfloor$. All other wavelet coefficients are zero; on the left of the discontinuity because the function is zero, 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 $2^{i-1}$. Namely, when $k$ is multiple of $2^i$, this magnitude is zero, and when $k$ is equal to $\ell 2^i + 2^{i/2}$, it is maximum. This latter case appears when the discontinuity is aligned with the discontinuity of the wavelet itself; then, the inner product is $2^{i-1} 2^{-i/2} = 2^{(i/2)-1}$. 

In the example above, we obtained precisely $X^{(i)}_b = 2^{-i/2}$ for the one nonzero wavelet coefficient in scale $i$. Figure 7.14 gives another example, with a signal with more modes and a DWT with longer filters. Notice that there is again $\sim 2^{-i/2}$
7.5 Discrete Wavelets as a Basis for $\ell^2(\mathbb{Z})$

Consider a real sequence $\{x_n\}_{n \in \mathbb{Z}}$ which has a finite $\ell^2$ norm, or

$$\sum_{n \in \mathbb{Z}} |x_n|^2 < \infty,$$

in other words, it belongs to $\ell^2(\mathbb{Z})$, the Hilbert space of finite-energy sequences. Can we represent such a sequence in terms of wavelets alone? We know we can do it if we use a finite number of wavelet decomposition steps and keep the scaling coefficients as well. In that case, we have an orthonormal basis composed of wavelets at various scales and the scaling coefficients at the coarsest scale. To do it with wavelets alone we need to iterate the decomposition indefinitely and show that the scaling coefficients, as we iterate farther and farther, vanish in $\ell^2$ norm. That is, the energy of the original sequence is entirely captured by the wavelet coefficients, thus proving Parseval’s formula for the wavelet basis.

Figure 7.14: A piecewise constant function plus an impulse, and its DWT. (a) The original signal. (b)–(e) The wavelet coefficients at scales $2^i$ for $i = 1, 2, 3, 4$. (f) The scaling coefficients. Note: in order to compare the different channels, all the coefficients have been upsampled by a factor $2^i$ ($i = 1, 2, 3, 4$).

scaling of wavelet coefficient magnitudes and a roughly constant number of nonzero wavelet coefficients per scale. We will study the variation of wavelet coefficient magnitudes across scale in more detail in Chapter 12. There, bounds on the coefficient magnitudes will play a large role in quantifying approximation performance.
To keep matters simple, we present a proof for the Haar case. The result holds more generally, but needs some additional technical conditions.

**Theorem 7.5** (Discrete Haar Wavelets as a Basis for $\ell^2(\mathbb{Z})$). The discrete-time filters $h^{(i)}$ with impulse responses

$$h^{(i)}_n = \frac{1}{2^{i/2}} \left( \sum_{k=0}^{2^i - 1} \delta_{n-k} - \sum_{k=2^i-1}^{2^{i+1} - 1} \delta_{n-k} \right), \quad i = 1, 2, 3, \ldots$$

and their shifts by $2^i$, as derived in Example 7.4, constitute an orthonormal basis for the space of finite-energy sequences, $\ell^2(\mathbb{Z})$.

**Proof.** The fact that it is an orthonormal set is shown in two steps. First, at a given scale $i$, we have

$$\left\langle h^{(i)}_{n-2^i \ell}, h^{(i)}_{n-2^i k} \right\rangle_n = \delta_{k-\ell},$$

since the filters do not overlap for $\ell \neq k$, and the filter is of unit norm. Then, considering two different scales, the impulse responses either do not overlap, or if they do, then the smaller-scale impulse response has a change of sign over an interval where the large-scale one is constant, making the inner product zero (see Figure 7.15).

To prove completeness, we will show that Parseval’s identity is verified, that is, for an arbitrary input $x \in \ell^2(\mathbb{Z})$, we have

$$\|x\|^2 = \sum_{i=1}^{\infty} \|X^{(i)}_h\|^2,$$

(7.22)

where $X^{(i)}_h$ is the sequence of wavelet coefficients at scale $i$, or

$$X^{(i)}_{h_k} = \left\langle h^{(i)}_{n-k2^i}, x_n \right\rangle_n.$$
For the sake of simplicity, consider a norm-1 sequence $x$ that is supported on \{0, 1, \ldots, 2^j - 1\}. Now consider the $j$-level DWT of $x$. Because of the length of $x$, the sequence of scaling coefficients $X_g^{(j)}$ has a single nonzero coefficient, namely
\[
X_g^{(j)} = \left\langle g^{(j)}, x \right\rangle = \frac{1}{2^{j/2}} \sum_{n=0}^{2^j-1} x_n,
\]
(7.23)
since all other scaling functions \{g^{(j)}_{n-2^j k}\}_{k \neq 0} do not overlap with the support of the sequence $x$. Also, that nonzero coefficient satisfies
\[
\left| X_g^{(j)} \right| \leq 1
\]
because $\|X_g^{(j)}\|^2 \leq \|x\|^2 = 1$ by Parseval’s relation.

We will now consider further levels of decomposition beyond $j$. After one more stage, the lowpass output $X_g^{(j+1)}$ again has a single nonzero coefficient, namely
\[
X_g^{(j+1)} = \frac{1}{\sqrt{2}} \left( X_g^{(j)} + X_h^{(j)} \right) = \frac{1}{\sqrt{2}} X_g^{(j)}.\]
(7.24)
Similarly, $|X_h^{(j+1)}|$ also equals $\frac{1}{\sqrt{2}} |X_g^{(j)}|$; that is, the energy has been split evenly between the highpass and lowpass channels. Iterating this lowpass decomposition $J$ times yields
\[
X_g^{(j+J)} = \frac{1}{2^{j/2}} X_g^{(j)},
\]
and thus
\[
\lim_{k \to \infty} |X_g^{(j+J)}| = 0.
\]
At the same time, the energy has been captured by the successive outputs $X_h^{(i)}$, $i > j$, since each step is energy conserving. More formally,
\[
|X_g^{(j)}|^2 = |X_g^{(j+J)}|^2 + \sum_{k=1}^{J} |X_h^{(j+k)}|^2,
\]
and as $J \to \infty$,
\[
|X_g^{(j)}|^2 = \sum_{k=1}^{\infty} |X_h^{(j+k)}|^2,
\]
verifying (7.22).

Albeit a bit technical, the above proof contains an ingredient which will be at the center of Chapter 8; namely, what happens to a DWT as the number of levels in the decomposition tends to infinity.
7.6 Variations on the DWT

So far, all elementary two-channel filter banks were orthogonal, and the iteration always decomposed only the lowpass branch further. Neither of these choices are necessary, so we explore the alternatives below. In addition, the computational complexity of the DWT is considered, and we show an elementary but astonishing result, namely that the complexity is linear in the input size. Finally, practical matters related to boundary effects, size and initialization of the discrete wavelet transform are discussed.

7.6.1 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 (6.60a)–(6.60d), we obtain perfect reconstruction by the same argument as in the orthogonal case (see Figure 7.3).

What is more interesting is that by using either \( g \) or \( \tilde{g} \) in the synthesis tree, we get iterated lowpass filters (see (7.3))

\[
G^{(J)}(z) = \prod_{i=0}^{J-1} G(z^{2^i})
\]

or

\[
\tilde{G}^{(J)}(z) = \prod_{i=0}^{J-1} \tilde{G}(z^{2^i}),
\]

respectively. As we will see, this iterated product plays a crucial role in the construction of wavelets (see the next chapter), and the two iterated filters above can exhibit quite different behaviors.

**Example 7.5 (Iterated Biorthogonal Filters).** In Example 6.4, we derived a biorthogonal pair with lowpass filters

\[
G(z) = 1 + 3z^{-1} + 3z^{-2} + z^{-3}, \\
\tilde{G}(z) = \frac{1}{4} (-1 + 3z + 3z^2 - z^3).
\]

Figure 7.16 shows the first few iterated \( G^{(J)}(z) \) and \( \tilde{G}^{(J)}(z) \), indicating a very different behavior. Recall that both filters are lowpass filters as are their iterated versions. However, the iteration of \( \tilde{G}(z) \) does not look “smooth,” indicating possible problems as we iterate further. This will be considered further in the Chapter 8.

7.6.2 Discrete Wavelet Packets

So far, the iterated decomposition was always applied to the lowpass filter, and often, there are good reasons to do so. But sometimes, it is of interest to decompose the highpass channel as well. More generally, we can consider an arbitrary tree decomposition. In other words, start with a signal \( x \) and decompose it into a
7.6. Variations on the DWT

Figure 7.16: Iteration of a biorthogonal pair of lowpass filters. On the left, the iteration of [1 3 3 1] leads to a “smooth” sequence, while on the right, the iteration of (1/4)[-1 3 3 -1] does not look “smooth.”

lowpass and a highpass version. Then, decide if the lowpass, the highpass, or both, are decomposed further, and keep going until a given depth $J$. The discrete wavelet transform is thus the particular case when only the lowpass version is repeatedly decomposed. Various possibilities are depicted in Figure 7.17.

How many such trees are there? Calling $N_J$ the number of trees of depth $J$, we have the recursion

$$N_{J+1} = N_J^2 + 1,$$  \hspace{1cm} (7.25)

since each branch of the initial two-channel filter bank can have $N_J$ 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

$$N_J \sim 2^{2^J}$$  \hspace{1cm} (7.26)

possible trees. Of course, many of these trees will be poor matches to real-life signals, but an efficient search algorithm allowing to find the best match between a given signal and a tree-structured expansion is possible.
Figure 7.17: Two-channel filter banks and variations together with the corresponding time-frequency tilings. (a) Two-channel filter bank. (b) N-channel filter bank. (c) The discrete wavelet transform tree. (d) The wavelet packet tree. (e) The short-time Fourier transform tree. (f) The time-varying tree.
7.7 Processing Stochastic Signals

7.8 Algorithms

Computing a DWT amounts to computing a set of convolutions but with a twist; as the decomposition progresses down the tree (see Figure 7.1), the sampling rate decreases. This twist is the key to the computational efficiency of the transform. Consider a discrete-time convolution with a causal filter of finite length $L$, or

$$y_n = \sum_{\ell=0}^{L-1} h_\ell x_{n-\ell}.$$  

This amounts to $L$ multiplications and $(L - 1)$ additions per input or output sample.\(^{46}\) Downsampling by $K$ clearly divides the complexity by $K$. A two channel filter bank downsampled by 2 has therefore a complexity of order $L$ both in terms of multiplications and additions, always per input sample (or 2 output samples).

Considering the iteration in the DWT tree, the second stage has similar complexity, but at half the sampling rate. Continuing this argument, we see that the overall transform has complexity for $J$ stages given by

$$L + L/2 + L/4 + \cdots + L/2^{J-1} < 2L$$

in both multiplications and additions. Given that the synthesis is simply the transpose (or reverse) of the analysis, we conclude that a complete analysis/synthesis DWT of a signal of length $N$ has a complexity of the order of at most

$$C_{\text{DWT}} \sim 4NL,$$  \hspace{1cm} (7.27)

that is, it is linear in the input size with a constant depending on the filter length.

What happens for more general trees? Clearly, the worst case is for a full tree (see Figure 7.17(e)). As the sampling rate goes down, the number of channels goes up, and the two effects cancel each other. Therefore, for $J$ levels of analysis and synthesis, the complexity amounts to

$$C_{\text{full}} \sim 2NLJ$$  \hspace{1cm} (7.28)

multiplications or additions, again for a signal of length $N$ and a filter of length $L$. To stress the efficiency of this tree-structured computation, let us consider a naive implementation of a $2^J$-channel filter bank, downsampled by $2^J$. Recall that, according to (7.9), the equivalent filters (in the discrete wavelet transform or the full tree) are of length

$$L^{(J)} = (L - 1)(2^J - 1) + 1 \sim L2^J.$$  

Computing each filter and downsampling by $2^J$ leads to $L$ operations per channel, or for $2^J$ channels including analysis and synthesis, we obtain

$$C_{\text{direct}} \sim 2NL2^J$$

which becomes exponentially worse for large $J$. Therefore, it pays off to study the computational structure, taking sampling rate changes carefully into account.

\(^{46}\)For long filters, improvements can be achieved by using fast Fourier transform techniques.
Chapter at a Glance
TBD

Historical Remarks
TBD

Further Reading
TBD

Exercises with Solutions
7.1. TBD

Exercises
7.1. TBD
Chapter 8

Wavelet Series on Functions

Contents

8.1 Introduction ........................................ 283
8.2 Wavelet Bases from Filter Banks .............. 286
8.3 The Wavelet Series and its Properties ........ 306
8.4 Multiresolution Analysis ......................... 310
8.5 Algorithms ....................................... 325
8.6 Biorthogonal Wavelets from Filter Banks .... 332

Chapter at a Glance ................................. 338
Historical Remarks ................................. 338
Further Reading .................................. 338
Exercises with Solutions ......................... 338
Exercises ............................................ 339

TBD: Opening text.

8.1 Introduction

Iterated filter banks, as seen in Chapter 7, pose interesting theoretical and practical questions. The key question is quite simple: what happens if we iterate a discrete wavelet transform to infinity? Of course, the question needs to be made more precise by indicating how this iteration process to infinity actually takes place. When done properly, and under certain conditions on the filters used in the filter bank, the limit leads to an orthonormal wavelet basis for the space of square integrable functions, $L^2(\mathbb{R})$. The key is that we take a basis for $\ell^2(\mathbb{Z})$, or a discrete-time basis, and derive from it a continuous-time basis. This connection between discrete and continuous time is reminiscent of the sampling theorem. The iteration process itself is fascinating, but the resulting basis is even more so: it has a scale invariance
property so that all basis vectors are obtained from a single function $\psi(t)$ through shift and scales,

$$
\psi_{m,n}(t) = 2^{-m/2} \psi(2^{-m} t - n) = \frac{1}{2^{m/2}} \psi\left(\frac{t - n2^m}{2^m}\right), \quad m, n \in \mathbb{Z}.
$$

(8.1)

The scale parameter $m$ shows that the basis functions are scaled by powers of 2, or stretched for $m > 0$ and compressed for $m < 0$.\(^\text{47}\) The shift parameter $n$ is itself multiplied by the scale $2^m$. This is quite natural, since long basis functions are shifted by large steps, while the short ones are shifted in small steps. This is shown in Figure 8.1. Part (a) depicts a wavelet $\psi(t)$, while part (b) shows a few of the basis functions $\psi_{m,n}(t)$.

The scaling property of the wavelets is not only very elegant, but also useful. In particular, properties of functions related to their scales will appear naturally in their wavelet expansion, just as properties related to modulation appear naturally in the Fourier transform. Another elegant property of wavelets, directly related to their construction through an iterative process, is the two-scale equation property. Specifically, a wavelet is obtained through an intermediate function, called a scaling function $\varphi(t)$, in such a way that

$$
\psi(t) = \sqrt{2} \sum_n h_n \varphi(2t - n).
$$

(8.2)

The “helper” function $\varphi(t)$ itself satisfies

$$
\varphi(t) = \sqrt{2} \sum_n g_n \varphi(2t - n),
$$

(8.3)

or in words, $\varphi(t)$ can be written as a linear combination of itself scaled by a factor of 2, properly shifted and weighted.

The magic is that the coefficients $g_n$ and $h_n$ (in (8.2)–(8.3)) are the taps of the discrete-time filter bank used to construct the continuous-time basis through iteration. (So it is no coincidence that we used the symbols $g_n$ and $h_n$ as in Chapters 6 and 7!) But even stronger, every wavelet construction leads to a two-scale equation and an associated filter bank. A graphical representation of the two-scale equation property is given in Figure 8.2, where part (a) depicts a scaling function $\varphi(t)$ and part (b) its representation through a two-scale equation.

But most fundamentally, 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. It is the continuous-time equivalent of what was seen in Section 7.4, where $\ell^2(\mathbb{Z})$ was split into spaces $W_j$. Very similarly, we will show that

$$
L^2(\mathbb{R}) = \bigoplus_{m \in \mathbb{Z}} W_m.
$$

(8.4)

\(^{47}\)We follow the convention of [52] that positive $m$ corresponds to large scales, or long basis functions, while negative $m$ corresponds to small scales, or short basis functions.
8.1. Introduction

Figure 8.1: Scale and shift relations in a wavelet basis. (a) Prototype wavelet $\psi(t)$. (b) A few of the basis functions, at various scales and shifts. The scale parameter $m$ corresponds to scaling by $2^m$, and the shift parameter $n$ corresponds to a shift by $n2^m$.

Figure 8.2: Example of a function satisfying a two-scale equation. (a) The scaling function $\phi(t)$. (b) The same function written as the sum of $\frac{1}{2}\phi(2t-1)$, $\phi(2t)$, and $\frac{1}{2}\phi(2t+1)$.

where the space $W_m$ is spanned by the wavelets at scale $m$, or

$$W_m = \text{span} \{ \psi_{m,n}(t) \}_{n \in \mathbb{Z}}.$$ (8.5)

And just like wavelets were built from scaling functions, there are scaling spaces, spanned by scaling functions at various scales or resolutions. These spaces, denoted $V_m$, are embedded into each other like Russian dolls,

$$\cdots \subset V_1 \subset V_0 \subset V_{-1} \subset \cdots$$

and the wavelet spaces $W_i$ are simply differences between two subsequent “dolls,”
This intuitive, geometrical view will be made formal in the body of this chapter.

Chapter Outline

The chapter starts with wavelets from filter banks, that is, by considering the infinite iteration of a discrete wavelet transform, properly defined and constrained so as to find an interesting limit. The prototypical example is the Haar filter, which leads to the Haar wavelet. In Section 8.2.2, we derive conditions on the filters to find interesting limits for the scaling function \( \varphi(t) \) and see how the conditions influence properties of the resulting continuous-time functions. In particular, smoothness of the limit function (for example, is the function continuous, or even differentiable) is studied. In Section 8.2.3, we construct wavelets \( \psi(t) \) from the scaling function \( \varphi(t) \). A key property of the wavelet, namely whether it has a certain number of zero moments, is also considered, showing how it is inherited from the discrete-time zero moment property seen in Chapter 7. Next, Section 8.2.4 shows that the iterated construction leads to orthonormal bases for \( \mathcal{L}^2(\mathbb{R}) \), when an orthonormal filter bank is used and the filters have the right properties. The Haar case is studied in detail, both because of its simplicity and its historical importance. The section concludes with a definition of the scaling spaces \( V_m \) and the wavelet spaces \( W_m \).

In Section 8.3, we consider the properties of wavelet series (WS). First, localization of wavelet series are given. Then, zero moment properties and decay of wavelet coefficients are explored, before considering the characterization of singularities by the decay of the associated wavelet series coefficients.

Section 8.4 is a bit formal, since it revisits the wavelet construction from an axiomatic point of view. This is done by first examining in detail the now-familiar Haar case and then pointing out a fundamental structure that leads to multiresolution analysis. This is then used to construct the famous Meyer wavelets.

Section 8.5 is devoted to computational issues, in particular to derive Mallat’s algorithm, which allows to compute wavelet coefficients with an initial continuous-time projection followed by a discrete-time, filter bank algorithm.

The body of the chapter develops the orthogonal case—orthogonal bases for \( \mathcal{L}^2(\mathbb{R}) \) derived from orthogonal bases from \( \ell^2(\mathbb{Z}) \), in turn generated from orthogonal two-channel filter banks, and orthogonal multiresolution analysis. The biorthogonal case is outlined in Appendix 8.B.

8.2 Wavelet Bases from Filter Banks

In the opening we saw the basic structure of wavelets in (8.1) and some of the properties of wavelets. It is time to construct actual wavelet functions. To achieve this, we will essentially study “infinitely” iterated discrete wavelet transforms. This leads to scaling and wavelet functions, and their associated properties.
8.2. Wavelet Bases from Filter Banks

8.2.1 Limits of Iterated Filters

We come back to filters and their iterations, but with a new angle, since we associate a continuous-time function with the discrete-time filter. Consider a length \( L \) orthonormal lowpass filter \( g_n, \; n = 0, 1, \ldots, L - 1 \), with \( z \)-transform \( G(z) \). Because of orthonormality,

\[
\langle g_n, g_{n-2k} \rangle_n = \delta_k \quad \text{for} \; k \in \mathbb{Z}
\]

and in particular (taking \( k = 0 \)), \( \|g\| = 1 \).

The iteration of upsampling by 2, followed by convolution with \( G(z) \), and this \( J \) times, was shown in Figure 7.5, where part (a) depicts the original system, and part (b) the equivalent system. This latter system consists of an upsampling by \( 2^J \) followed by an equivalent filter \( G^{(J)}(z) \) given by

\[
G^{(J)}(z) = G(z) G(z^2) G(z^4) \cdots G(z^{2^{J-1}}) = \prod_{i=0}^{J-1} G(z^{2^i}). \tag{8.6}
\]

By construction, this filter satisfies orthonormality with respect to shifts by \( 2^J \), or (see Proposition 7.2)

\[
\langle g^{(J)}_n, g^{(J)}_{n-2^Jk} \rangle_n = \delta_k \quad \text{for} \; k \in \mathbb{Z}. \tag{8.7}
\]

The equivalent filter \( g^{(J)} \) has norm 1 and length \( L^{(J)} \), which can be upper bounded by (see (7.10))

\[
L^{(J)} \leq (L - 1)2^J. \tag{8.8}
\]

Now, we will associate a piecewise constant function \( \varphi^{(J)}(t) \) with \( g^{(J)}_n \) in such a way that \( \varphi^{(J)}(t) \) is of finite length and of norm 1. The piecewise constant function is a continuous-time staircase function, so the question reduces to determining the width and height of the stairs. Since the number of steps (or nonzero coefficients of \( g^{(J)}_n \)) grows exponentially in \( J \) (see (8.8)), we pick the stair width as \( 1/2^J \). Then, the length of \( \varphi^{(J)}(t) \) will be upper bounded by \( (L - 1) \), or

\[
\text{support} \left( \varphi^{(J)}(t) \right) \subset [0, L - 1] \quad \text{for} \; J = 1, 2, \ldots, \tag{8.9}
\]

where 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 \( g^{(J)}_n \), we pick the height of the stairs as \( 2^{J/2} g^{(J)}_n \). Then, each piece of the staircase function contributes

\[
\int_{n/2^J}^{(n+1)/2^J} |\varphi^{(J)}(t)|^2 \, dt = \int_{n/2^J}^{(n+1)/2^J} 2^J (g^{(J)}_n)^2 \, dt = (g^{(J)}_n)^2.
\]

Summing up all contributions, we indeed get

\[
|\varphi^{(J)}(t)|^2 = \sum_{n=0}^{L^{(J)}-1} \int_{n/2^J}^{(n+1)/2^J} |\varphi^{(J)}(t)|^2 \, dt = \sum_{n=0}^{L^{(J)}-1} (g^{(J)}_n)^2 = 1.
\]
Figure 8.3: Iterated filters and associated staircase functions based on a 4-tap lowpass filter. (a) From top to bottom: The initial filter $g_n$ and the iterates $g_n^{(2)}$, $g_n^{(3)}$, and $g_n^{(4)}$. (b) The associated staircase functions $\varphi^{(1)}(t)$, $\varphi^{(2)}(t)$, $\varphi^{(3)}(t)$, and $\varphi^{(4)}(t)$. [Note: This figure is qualitative. It will be replaced with an exact figure. In particular, the sketched $\varphi^{(1)}(t)$ is too long.]

We have thus defined the staircase function as

$$\varphi^{(J)}(t) = 2^{J/2} \cdot g_n^{(J)} \quad \text{for} \quad \frac{n}{2^J} \leq t < \frac{n + 1}{2^J} \quad (8.10)$$

and verified along the way that it is supported on a finite interval and has unit norm.\footnote{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.}

In Figure 8.3, we show a few iterations of a 4-tap filter in part (a) and its associated staircase function in part (b). The filter is as in Example 7.2, equation (7.14). Note how the staircase function $\varphi^{(J)}$ has geometrically decreasing steps and a support contained in the 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 stairs become finer and finer. On the contrary, if $g_n^{(J)}$ has too much variation as $J \to \infty$, the sequence
of functions \( \varphi^{(J)}(t) \) might not have a limit as \( J \to \infty \). This leads to the following necessary condition for the filter \( g_n \).

**Proposition 8.1 (Necessity of a Zero at \( \pi \)).** For \( \lim_{J \to \infty} \varphi^{(J)}(t) \) to exist, it is necessary for the filter \( g_n \) to have a zero at \( z = -1 \) or \( \omega = \pi \).

**Proof.** The overall strategy of the proof is to show that convergence implies that the sum of the even polyphase component of \( g, \sum_n g_{2n} \), equals the sum of the odd polyphase component, \( \sum_n g_{2n+1} \). This implies a zero at \( z = -1 \).

To obtain a useful time-domain expression from the \( z \)-domain expression describing the iteration process, \( G^{(J)}(z) = G(z)G^{(J-1)}(z^2) \), let \( f \) be the sequence with \( z \) transform \( G^{(J-1)}(z^2) \). (\( f \) is \( g^{(J-1)} \) upsampled by 2.) Then

\[
\varphi_{g}^{(J)}(n) = \sum_{k \in \mathbb{Z}} g_k f_{n-k} = \sum_{\ell \in \mathbb{Z}} g_{2\ell} f_{n-2\ell} + \sum_{\ell \in \mathbb{Z}} g_{2\ell+1} f_{n-(2\ell+1)}, \tag{8.11}
\]

by breaking the convolution sum into \( k = 2\ell \) and \( k = 2\ell + 1 \) terms. Now since \( f_n = 0 \) for all odd values of \( n \), in the right side of (8.11), the first sum is zero for odd \( n \) and the second sum is zero for even \( n \). Thus

\[
\begin{align*}
g_{2n}^{(J)} &= \sum_{\ell \in \mathbb{Z}} g_{2\ell} f_{2n-2\ell} = \sum_{\ell=0}^{(L/2)-1} g_{2\ell} g_{n-\ell}^{(J-1)}, \tag{8.12a} \\
g_{2n+1}^{(J)} &= \sum_{\ell \in \mathbb{Z}} g_{2\ell+1} f_{2n-2\ell} = \sum_{\ell=0}^{(L/2)-1} g_{2\ell+1} g_{n-\ell}^{(J-1)}, \tag{8.12b}
\end{align*}
\]

where we have also used the length of \( g \) to write finite sums.

Now suppose that \( \lim_{J \to \infty} \varphi^{(J)}(t) \) exists for all \( t \), and denote the limit by \( \varphi(t) \). It must also be true that \( \varphi(\tau) \neq 0 \) for some \( \tau \); otherwise we contradict that \( \|\varphi^{(J)}\| = 1 \) for every \( J \). With \( n_J \) chosen as any function of \( J \) such that \( \lim_{J \to \infty} 2n_J/2^J = \tau \), we must have

\[
\lim_{J \to \infty} 2^{J/2} g_{2n_J}^{(J)} = \lim_{J \to \infty} 2^{J/2} g_{2n_{J+1}}^{(J)}
\]

because both sides equal \( \varphi(\tau) \). Multiplying by \( 2^{J/2} \) and taking limits in (8.12) and then subtracting, we find

\[
\sum_{\ell=0}^{(L/2)-1} (g_{2\ell} - g_{2\ell+1}) \left( \lim_{J \to \infty} 2^{J/2} g_{n_J-\ell}^{(J-1)} \right) = 0.
\]

Since the limit above exists equals \( \varphi(\tau) \neq 0 \) for every \( \ell \in \{0, 1, (L/2) - 1\} \), we conclude 0 = \( \sum_{\ell \in \mathbb{Z}} (g_{2\ell} - g_{2\ell+1}) = G(-1) \).

**Definition 8.1 (Scaling function).** If the limit exists, define

\[
\varphi(t) = \lim_{J \to \infty} \varphi^{(J)}(t). \tag{8.13}
\]
Let us calculate the Fourier transform of \( \varphi^J(t) \), denoted by \( \Phi^J(\omega) \). Remember that it is a linear combination of box functions, each of width \( 1/2^J \) and height \( 2^{J/2} \). The unit box function (equal to 1 between 0 and 1) has Fourier transform

\[
B(\omega) = e^{-j\omega/2} \frac{\sin(\omega/2)}{\omega/2},
\]

and using the scaling property of the Fourier transform, the transform of a box of height \( 2^{J/2} \) between \( 0 \) and \( 1/2^J \) is

\[
B^J(\omega) = 2^{-J/2} e^{-j\omega/2^{J+1}} \frac{\sin(\omega/2^{J+1})}{\omega/2^{J+1}}. \tag{8.14}
\]

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

\[
\Phi^J(\omega) = \sum_{n=0}^{L(J)-1} e^{-j\omega n/2^J} g_n^J \cdot B^J(\omega). \tag{8.15}
\]

Now, notice that

\[
\sum_{n=0}^{L(J)-1} e^{-j\omega n/2^J} g_n^J = G^J(e^{j\omega/2^J})
\]

since the summation is a discrete-time Fourier transform of the sequence \( g_n^J \) evaluated at \( \omega/2^J \). Using (8.6),

\[
G^J(e^{j\omega/2^J}) = G^J(z)|_{z=e^{j\omega/2^J}} = \prod_{i=0}^{J-1} G(e^{j\omega/2^{i+1}}) = \prod_{i=1}^{J} G(e^{j\omega/2^{i}}).
\]

It follows from (8.15) that

\[
\Phi^J(\omega) = G^J(e^{j\omega/2^J}) \cdot B^J(\omega) = \prod_{i=1}^{J} G(e^{j\omega/2^{i}}) \cdot B^J(\omega). \tag{8.16}
\]

In the sequel, we are interested in what happens in the limit, when \( J \to \infty \). First, notice that for any finite \( \omega \), the effect of the interpolation function \( B^J(\omega) \) becomes negligible as \( J \to \infty \). Indeed in (8.14), both terms dependent on \( \omega \) tend to 1 as \( J \) grows unbounded and only the factor \( 2^{-J/2} \) remains. So, in (8.16), the key term is the product, which becomes an infinite product. Let us define this limit specifically.

**Definition 8.2 (Infinite Product).** We call \( \Phi(\omega) \) the limit, if it exists, of the infinite product, or

\[
\Phi(\omega) = \lim_{J \to \infty} \Phi^J(\omega) = \lim_{J \to \infty} 2^{-J/2} G^J(e^{j\omega/2^J}) \prod_{i=1}^{J} G(e^{j\omega/2^{i}}). \tag{8.17}
\]
It follows immediately that a necessary condition for the product to be well behaved is that

\[ G(1) = \sqrt{2}, \]

leading to \( \Phi(0) = 1 \). If \( G(1) \) were larger, \( \Phi(0) \) would blow up; if \( G(1) \) were smaller, \( \Phi(0) \) would be zero, contradicting the fact that \( \Phi(\omega) \) is the limit of lowpass filters and hence a lowpass function.\(^{49}\)

A more difficult question is to understand when the limits of the time domain iteration \( \varphi^{(J)}(t) \) (in (8.10)) and of the Fourier domain product \( \Phi^{(J)}(\omega) \) (in (8.16)) are themselves related by the Fourier transform. That this is a non-trivial question will be shown using a counterexample below (see Example 8.4). The exact conditions are technical and beyond the scope of our treatment, so we will concentrate on “well behaved” cases. These cases are when all limits are well defined (see Definitions 8.1 and 8.2) and related by Fourier transform relationships, or

\[ \varphi^{(J)}(t) \xrightarrow{CTFT} \Phi^{(J)}(\omega), \]

which we verified, and

\[ \varphi(t) \xrightarrow{CTFT} \Phi(\omega), \]

which we will assume.

A more interesting question is to understand the behavior of the infinite product. If \( \Phi(\omega) \) decays sufficiently fast with \( \omega \), the scaling function \( \varphi(t) \) will be smooth. How this can be done while maintaining other desirable properties (like compact support and orthogonality) is the key result for designing wavelet bases from iterated filter banks. In order to gain some intuition, we consider some examples.

**Example 8.1 (Iteration of the Haar lowpass filter).** Going back to Example 7.4, we recall that the equivalent lowpass filter, after \( J \) iterations, is a length \( 2^J \) averaging filter

\[ g^{(J)} = \frac{1}{2^{J/2}} [1, 1, 1, \ldots, 1]. \]

The staircase function \( \varphi^{(J)}(t) \) in (8.10) leads simply to the indicator function of the unit interval \([0, 1]\), independently of \( J \), and the limit \( \varphi(t) \) is

\[ \varphi(t) = \begin{cases} 1, & 0 \leq t < 1; \\ 0, & \text{otherwise}. \end{cases} \]

Convergence is achieved without any problem, actually in one step! That the frequency domain product converges to the sinc function is left as an exercise (see Exercise 8.1). □

\(^{49}\)Notice that \( G(1) = \sqrt{2} \) is automatically satisfied for orthogonal filters with at least 1 zero at \( \omega = \pi \), since the quadrature formula in (6.16) states \( G^2(1) + G^2(-1) = 2 \) or \( G(1) = \sqrt{2} \) when \( G(-1) = 0 \).
Chapter 8. Wavelet Series on Functions

Figure 8.4: Schematic depiction of $|G(e^{i\omega/2})|$, $|G(e^{i\omega/4})|$, and $|G(e^{i\omega/8})|$, which appear in the Fourier domain product $\Phi^{(J)}(\omega)$.

Example 8.2 (Orthonormal 4-tap lowpass with 2 zeros at $\omega = \pi$). Consider the filter designed in Example 7.4, or

$$G(z) = \frac{1}{4\sqrt{2}} (1 + \sqrt{3} + (3 + \sqrt{3})z^{-1} + (3 - \sqrt{3})z^{-2} + (1 - \sqrt{3})z^{-3}), \quad (8.18)$$

for which the staircase function was shown in Figure 8.3. We consider the Fourier domain product

$$\Phi^{(J)}(\omega) = \prod_{i=1}^{J} G(e^{i\omega/2^i})B^{(J)}(\omega)$$

and notice that the terms in the product are periodic with periods $4\pi, 8\pi, \ldots, 2^J \cdot 2\pi$, since $G(e^{i\omega})$ is $2\pi$-periodic. This is shown schematically in Figure 8.4. Similarly, we show the product for the actual filter given in (8.18) in Figure 8.5. As can be seen, the terms are oscillating depending on their periodicity, but the product decays rather nicely. This decay will be studied in detail below.

Example 8.3 (Necessity of a zero at $\omega = \pi$). Consider the orthogonal filter designed using the window method in Example 6.2. This filter does not have a zero at $\omega = \pi$, since

$$G(-1) \approx 0.389.$$  

Its iteration is shown in Figure 8.6, where it can be noticed that high frequency oscillations are present. This will not go away, in essence prohibiting convergence of the iterated function $\varphi^{(J)}(t)$. □
8.2. Wavelet Bases from Filter Banks

Figure 8.5: Fourier product with $J$ terms, $G(e^{j\omega/2}) \cdot G(e^{j\omega/4}) \cdots G(e^{j\omega/2^J})$. (a) The individual terms. (b) The product.

Example 8.4 (Stretched Haar filter). Instead of the regular Haar filter, consider a filter with $z$-transform

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

and impulse response

$$g = \left[ \frac{1}{\sqrt{2}}, 0, 0, \frac{1}{\sqrt{2}} \right].$$

Clearly, it is an orthogonal lowpass filter, and it has one zero at $\omega = \pi$ or $z = -1$. However, unlike the Haar filter, its iteration is highly unsmooth. Consider the three-stage filter bank equivalent lowpass filter

$$G^{(3)}(z) = \frac{1}{2^{3/2}}(1 + z^{-3})(1 + z^{-6})(1 + z^{-12})$$

with impulse response

$$g^{(3)} = \frac{1}{2^{3/2}} \left[ 1, 0, 0, 1, 0, 0, 1, \ldots, 1, 0, 0, 1 \right].$$

Similarly, all iterations are sequences of the form

$$[\ldots, 0, 1, 0, 0, 1, 0, 0, 1, 0, \ldots].$$
The staircase function \( \phi^{(J)}(t) \) inherits this lack of smoothness, and does not converge pointwise to a proper limit, as shown graphically in Figure 8.7. Considering the frequency domain and the infinite product, it turns out that \( L^2 \) convergence fails also (see Exercise 8.1).

The examples above indicated that iterated filters and their associated graphical functions have interesting behaviors. The Haar case was trivial, the 4-tap filters showed a smooth behavior, and the stretched Haar filter pointed out potential convergence problems.

### 8.2.2 The Scaling Function and its Properties

In the sequel, we concentrate on orthonormal filters with \( N \geq 1 \) zeros at \( \omega = \pi \), or

\[
G(z) = \left( \frac{1 + z^{-1}}{2} \right)^N R(z)
\]

(8.19)

where \( R(1) = \sqrt{2} \). We assume pointwise convergence of the iterated graphical function \( \phi^{(J)}(t) \) to \( \phi(t) \), as well as of \( \Phi^{(J)}(\omega) \) to \( \Phi(\omega) \), and finally that \( \phi(t) \) and \( \Phi(t) \) are a Fourier transform pair. In one word, we avoid all tricky issues of convergence and concentrate on the nicely behaved cases, which are the ones of most interest for constructing “nice” bases.
8.2. Wavelet Bases from Filter Banks

Figure 8.7: Counterexample to convergence using the stretched Haar filter. The discrete-time filter has impulse response $g = \left[ \frac{1}{\sqrt{2}}, 0, 0, \frac{1}{\sqrt{2}} \right]$. (a) $\varphi^{(1)}(t)$. (b) $\varphi^{(2)}(t)$. (c) $\varphi^{(i)}(t)$.

Smoothness of the Scaling Function

As seen earlier, the key is to understand the infinite product (8.17) which becomes, using (8.19),

$$\Phi(\omega) = \prod_{i=1}^{\infty} 2^{-1/2} \left( \frac{1 + e^{-j\omega/2^i}}{2} \right)^N R(e^{j\omega/2^i})$$

$$= \left( \prod_{i=1}^{\infty} \left( \frac{1 + e^{-j\omega/2^i}}{2} \right) \right)^N \prod_{i=1}^{\infty} 2^{-1/2} R(e^{j\omega/2^i}). \tag{8.20}$$

The goal is to see if $\Phi(\omega)$ has a sufficiently fast decay for large $\omega$. Specifically, if $|\Phi(\omega)|$ decays faster than $1/|\omega|$, or

$$|\Phi(\omega)| < \frac{C}{(1 + |\omega|)^{(1+\epsilon)}} \quad \text{for all } \omega \tag{8.21}$$

for some $C < \infty$ and $\epsilon > 0$, then $\varphi(t)$ is guaranteed to be continuous (see Chapter 3). Consider first the product

$$\prod_{i=1}^{\infty} \left( \frac{1 + e^{-j\omega/2^i}}{2} \right).$$
This corresponds to a Haar filter \( G(z) = \frac{1}{\sqrt{2}}(1 + z^{-1}) \) iterated according to (8.17) and converges to the box function of the unit interval. Its Fourier transform is the sinc function, thus

\[
\prod_{i=1}^{\infty} \left( \frac{1 + e^{-j\omega/2^i}}{2} \right) = e^{-j\omega/2} \frac{\sin(\omega/2)}{\omega/2}.
\]

The decay of this Fourier transform is of order \( 1/|\omega| \). Thus, the first half of (8.20) decays as \( 1/|\omega|^N \). In time domain, it is the convolution of \( N \) box splines, or a \( B \) spline of order \( N - 1 \). Denote the second half of (8.20) by \( \gamma(\omega) \):

\[
\gamma(\omega) = \prod_{i=1}^{\infty} 2^{-1/2} R(e^{j\omega/2^i}).
\]

So, as long as \( |\gamma(\omega)| \) does not grow faster than \( |\omega|^{N-1-\epsilon} \), \( \epsilon > 0 \), the product will indeed decay fast enough to satisfy (8.21), leading to a continuous scaling function \( \varphi(t) \). The above discussion is made precise in the following proposition.

**Proposition 8.2 (Smoothness of scaling function).** Consider the factorization of the lowpass filter \( G(z) \) in (8.19). Let \( B = \sup_{\omega \in [0, 2\pi]} |R(e^{j\omega})| \). If

\[
B < 2^{N-1/2}
\]

then the iterated function \( \varphi^{(i)}(t) \) converges pointwise as \( i \to \infty \), to a continuous function \( \varphi(t) \) with Fourier transform

\[
\Phi(\omega) = \prod_{i=1}^{\infty} 2^{-1/2} G(e^{j\omega/2^i}).
\]

For a proof of this result, see Appendix 8.A. Note that the condition (8.24) is sufficient, but not necessary: many filters that fail the test still lead to continuous limits (and more sophisticated tests can be used).

If we strengthen the bound to

\[
B < 2^{N-k-1/2} \quad \text{for } k \in \mathbb{N}
\]

then \( \varphi(t) \) will be continuous and \( k \)-times differentiable (see Exercise 8.3).

**Example 8.5 (Smoothness of the limit function).** Let us test the continuity condition given in Proposition 8.1 on filters we have encountered so far. First, the Haar filter

\[
G(z) = \frac{1}{\sqrt{2}}(1 + z^{-1}) = \left( \frac{1 + z^{-1}}{2} \right) \cdot \sqrt{2}
\]

has one zero at \( \omega = \pi \) and \( R(z) = \sqrt{2} \). So \( B = \sqrt{2} \), which does not meet the inequality in (8.24). So according to Proposition 8.2, \( \varphi(t) \) may or may not be continuous (and we know it is not!).
Second, consider the 4-tap filter with two zeros at $\omega = \pi$ first derived in Example 6.4 and seen again in Example 8.2. It can be written as
\[
G(z) = \left(\frac{1 + z^{-1}}{2}\right)^2 \cdot \frac{1}{\sqrt{2}} (1 + \sqrt{3} + (1 - \sqrt{3}) z^{-1}).
\]
The supremum of the second factor is attained at $\omega = \pi$ or $z = -1$, where it equals
\[
B = \sqrt{6},
\]
which is smaller than $2^{3/2} = \sqrt{8}$. Therefore, the limit function must be continuous.

**Two-Scale Equation Property**

By construction, the scaling function $\varphi(t)$ has a distinguishing feature in that it satisfies a two-scale equation. This can be shown in the Fourier domain, starting with the infinite product (8.17):
\[
\Phi(\omega) = \prod_{i=1}^{\infty} 2^{-1/2} G(e^{j\omega/2^i}) = 2^{-1/2} G(e^{j\omega/2}) \cdot \prod_{i=2}^{\infty} 2^{-1/2} G(e^{j\omega/2^i}) = 2^{-1/2} G(e^{j\omega/2}) \Phi(\omega/2). \tag{8.25}
\]

To take the inverse Fourier transform, note that
\[
\Phi(\omega) = 2^{-1/2} \sum_{n=0}^{L-1} g_n e^{-j\omega n/2} \Phi(\omega/2).
\]

Then, use the scaling property, $\Phi(\omega/2) \leftrightarrow 2\varphi(2t)$, and the shift property, $e^{-j\omega n/2} F(\omega) \leftrightarrow f(t - n/2)$, of the Fourier transform. Together, these lead to
\[
\varphi(t) = \sqrt{2} \sum_{n=0}^{L-1} g_n \varphi(2t - n). \tag{8.26}
\]

This two-scale relation is shown in Figure 8.8 for the usual 4-tap filter seen in Examples 8.2 and 8.5.

**Reproduction of Polynomials**

It is well known that splines of order $N$ and their shifts can reproduce polynomials of degree up to $N$. For example, the linear spline or hat function, defined as
\[
\beta^{(1)}(t) = \begin{cases} 
1 - |t|, & |t| < 1; \\
0, & \text{otherwise}
\end{cases}
\]
reproduces first degree polynomials, like
\[
t = \sum_{n=-\infty}^{\infty} n\beta^{(1)}(t - n).
Figure 8.8: Two-scale equation. The scaling function obtained from the iteration of the 4-tap filter with two zeros at $\omega = \pi$. It can be written as a linear combination of itself, scaled by a factor of 2, and appropriately shifted and weighted. [This figure is accurate, but it is at some arbitrary scale. It will be redone at scale 0.]

We have seen that the scaling function based on a filter having $N$ zeros at $\omega = \pi$ contains a “spline part” of order $N - 1$. This leads to the property that linear combinations of $\{\varphi(t - n)\}_{n \in \mathbb{Z}}$ can reproduce polynomials of degree $N - 1$. Pictorially, this is shown in Figure 8.9, where the usual filter with two zeros at $\pi$ reproduces the linear function $f(t) = t$. (The proof of this property will be shown later.)

Orthogonality with Respect to Integer Shifts

Another property inherited from the underlying filter is orthogonality with respect to integer shifts:

$$\langle \varphi(t), \varphi(t - n) \rangle_t = \delta_n. \quad (8.27)$$

Since $\varphi(t)$ is defined through a limit and the inner product is continuous in both arguments, the orthogonality (8.27) follows from the orthogonality of $\varphi^{(J)}(t)$ and its integer translates for any iterate $J$:

$$\left\langle \varphi^{(J)}(t), \varphi^{(J)}(t - n) \right\rangle_t = \delta_n, \quad \text{for } J = 1, 2, \ldots. \quad (8.28)$$
8.2. Wavelet Bases from Filter Banks

Figure 8.9: Reproduction of polynomials by the scaling function and its shifts. Here, \( \varphi(t) \) is based on a 4-tap filter with two zeros at \( \omega = \pi \), and the linear function is reproduced. [Note: Something must be said about the finiteness of a sum leading to polynomial reproduction only on an interval.]

This follows from the equivalent property of the iterated filter \( g_{n}^{(J)} \) in Proposition 7.2. Specifically,

\[
\left\langle \varphi^{(J)}(t), \varphi^{(J)}(t - k) \right\rangle_t = a \sum_{n=0}^{L(J)-1} \int_{n/2^J}^{(n+1)/2^J} 2^{J/2} g_n^{(J)} \cdot 2^{J/2} g_{n-k}^{(J)} \, dt \\
= b \sum_{n=0}^{L(J)-1} g_n^{(J)} g_{n-k}^{(J)} \\
= c \left\langle g_n^{(J)}, g_{n-k}^{(J)} \right\rangle_n = \delta_k
\]

where we used (a) the fact that a shift by \( k \) on \( \varphi^{(J)}(t) \) corresponds to a shift by \( k2^J \) on the filter \( g_n^{(J)} \); (b) that the functions are constant over intervals of size \( 1/2^J \); and (c) that the filters \( g_n^{(J)} \) are orthogonal with respect to shifts by multiples of \( 2^J \) (see (7.6)).

Note that the orthogonality (8.27) at scale 0 has counterparts at other scales. It is easy verify by changing the variable of integration that

\[
\left\langle \varphi(2^J t), \varphi(2^J t - n) \right\rangle_t = 2^{-i} \delta_n.
\]
8.2.3 The Wavelet and its Properties

We have seen the construction and properties of the scaling function \( \phi(t) \), but our aim is to construct the wavelet \( \psi(t) \). Just as in Chapter 7 for the discrete wavelet transform, the scaling function is a lowpass filter while the wavelet is a bandpass filter. We recall that the iterated bandpass filter in the discrete transform is given by (see (7.13))

\[
H(J)(z) = H(z^{2^{J-1}})G(J-1)(z) = H(z^{2^{J-1}})G(z)G(z^2)\cdots G(z^{2^{J-2}}).
\]  
(8.30)

Comparing this to \( G(J)(z) \) given in (8.6), we see that only one filter is changed in the cascade, namely \( G(z^{2^{J-1}}) \) by \( H(z^{2^{J-1}}) \). We introduce an iterated function \( \psi(J)(t) \) associated with \( H(J)(z) \), similarly to \( \phi(J)(t) \) associated with \( G(J)(z) \) (see (8.10)),

\[
\psi(J)(t) = 2^{J/2}h_n, \quad \text{for} \quad \frac{n}{2^J} \leq t < \frac{n + 1}{2^J},
\]  
(8.31)

where \( h_n \) is the impulse response of \( H(J)(z) \).

Just like \( \phi(J)(t) \) and using the same arguments, this function is piecewise constant (with pieces of size \( 1/2^J \)), has unit norm, and is supported on a finite interval. Unlike \( \phi(J)(t) \), our new object of interest \( \psi(J)(t) \) is a bandpass function. In particular, its Fourier transform \( \Psi(\omega) \) satisfies

\[
\Psi(0) = 0,
\]
which follows from the fact that \( H(1) = 0 \).

Again, we are interested in the case of \( J \to \infty \). Clearly, this involves an infinite product, but it is the same infinite product we studied for the convergence of \( \phi^{(I)}(t) \) towards \( \phi(t) \). 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 \( G(z^{2^{J-1}}) \) by the highpass \( H(z^{2^{J-1}}) \). We do not repeat the details, but rather indicate the main points. So (8.16) becomes

\[
\Psi(J)(\omega) = H(e^{j\omega/2})\prod_{i=2}^{J} G(e^{j\omega/2^i})B^{(J)}(\omega).
\]  
(8.32)

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 8.3 (Wavelet).** Assuming the limit to exist, we define the wavelet in time and frequency domains to be

\[
\psi(t) = \lim_{J \to \infty} \psi(J)(t), \quad (8.33)
\]

\[
\Psi(\omega) = \lim_{J \to \infty} \Psi(J)(\omega). \quad (8.34)
\]

From (8.32) and using the steps leading to (8.17), we can write

\[
\Psi(\omega) = 2^{-1/2}H(e^{j\omega/2})\prod_{i=2}^{\infty} 2^{-1/2}G(e^{j\omega/2^i}).
\]  
(8.35)
8.2. Wavelet Bases from Filter Banks

Two-Scale Relation

Similarly to (8.25), we can rewrite (8.35) as

\[ \Psi(\omega) = 2^{-1/2} H(e^{j\omega/2}) \Phi(\omega/2). \]  

(8.36)

Taking the inverse Fourier transform, we get a relation similar to (8.26), namely

\[ \psi(t) = \sqrt{2} \sum_{n=0}^{L-1} h_n \varphi(2t - n), \]  

(8.37)

which is the two-scale equation for the wavelet, with the scaling function involved. From the support of \( \varphi(t) \) in (8.9), it also follows that \( \psi(t) \) has the same support on \([0, L-1]\). To illustrate the two-scale relation and also show (finally!) a wavelet, consider the following example.

**Example 8.6 (Wavelet and Two-scale Equation).** Take the highpass filter associated with the usual 4-tap orthonormal lowpass. It has a double zero at \( \omega = 0 \), and its impulse response is (from time reversal and modulation by \((-1)^n\) of \(g_n\) from Example 8.2,

\[ H(z) = \frac{1}{4\sqrt{2}}(\sqrt{3} - 1 + (3 - \sqrt{3})z^{-1} - (3 + \sqrt{3})z^{-2} + (1 + \sqrt{3})z^{-3}). \]  

(8.38)

In Figure 8.10, the wavelet \( \psi(t) \) is shown in part (a), while part (b) depicts the two-scale relationship.

Smoothness of the Wavelet

Since the wavelet is a finite linear combination of scaling functions and their shifts (see (8.37)), the smoothness is inherited from the scaling function. This can be seen in Figure 8.10(a).

Zero Moment Property of the Wavelet

Recall that the lowpass filter \( G(z) \) had \( N \) zeros \((N \geq 1)\) at \( \omega = \pi \) or \( z = -1 \). Because of time reversal and modulation (see (6.28)) to obtain the highpass, and given the form of \( G(z) \) in (8.19), \( H(z) \) can be written as

\[ H(z) = z^{-L+1} \left( \frac{1 - z^{-1}}{2} \right)^N R(-z^{-1}). \]  

(8.39)

It has therefore \( N \) zeros at \( \omega = 0 \) or \( z = 1 \). These \( N \) zeros carry over directly to \( \Psi(\omega) \), since (see (8.35))

\[ \Psi(\omega) = 2^{-1/2} H(e^{j\omega/2}) \Phi(\omega/2) \]

and \( \Phi(0) = 1 \). Recall the moment theorem of the Fourier transform, which relates the \( n \)th moment of a function to the \( n \)th derivative of its Fourier transform at
the origin (see Chapter 3). Because \( \Psi(\omega) \) has \( N \) zeros at the origin, \( \Psi(0) \) and its derivatives of order 1 to \( N - 1 \) are zero. This implies that

\[
\int_{-\infty}^{\infty} t^n \psi(t) \, dt = 0 \quad \text{for} \quad n = 0, 1, \ldots, N - 1 \tag{8.40}
\]

or equivalently, that any polynomial \( p(t) \) of the degree \( N - 1 \) will have an inner product with a wavelet (of any shift or scale) equal to zero, or

\[
\langle p(t), \psi(at - b) \rangle_t = 0 \quad \text{for any} \quad a, b \in \mathbb{R}. \tag{8.41}
\]

Remembering that \( \varphi(t) \) is able to reproduce polynomials up to degree \( N - 1 \), we note a nice role split between wavelets (that annihilate polynomials) and the scaling function (that catches them).

**Orthogonality of the wavelet**

On our quest towards orthonormal bases of wavelets, we show orthogonality between a wavelet and its integer shifts. The development uses the orthogonality of the
scaling function and its integer shifts (8.27), as well as the two-scale equation (8.37).

\[
\langle \psi(t), \psi(t-n) \rangle_t \overset{(a)}{=} 2 \left\langle \sum_k h_k \varphi(2t-k), \sum_m h_m \varphi(2t-m-2n) \right\rangle_t \\
= 2 \sum_k \sum_m h_k h_m \langle \varphi(2t-k), \varphi(2t-m-2n) \rangle_t \\
\overset{(b)}{=} \sum_m h_{m+2n} \overset{(c)}{=} \delta_n,
\]

(8.42)

where (a) uses the two-scale equation (8.37); (b) uses the orthogonality (8.29) at scale \(-1\); and (c) follows from \(h\) being orthonormal to its even shifts.

A very similar development leads to

\[
\langle \varphi(t), \psi(t-n) \rangle_t = 0.
\]

(8.43)

### 8.2.4 Orthonormal Bases of Wavelets

So far, we have considered only a single scale, with the two functions \(\varphi(t)\) and \(\psi(t)\). Yet, multiple scales where already lurking in the background through the two-scale equations. And just like in the discrete wavelet transform in Chapter 7, the real action appears when all scales are considered. Thus, let us recall the following family of functions (see (8.1)):

\[
\psi_{m,n}(t) = 2^{-m/2} \psi(2^{-m} t - n) = \frac{1}{2^{m/2}} \psi \left( \frac{t - n2^m}{2^m} \right) \quad \text{for } m, n \in \mathbb{Z}.
\]

(8.44)

For \(m = 0\), we have the usual wavelet and its integer shifts; for \(m > 0\), the wavelet is stretched by a power of 2, and the shifts are proportionally increased; and for \(m < 0\), the wavelet is compressed by a power of 2, with appropriately reduced shifts.

This was shown pictorially in Figure 8.1 at the start of this chapter. Of course, underlying the wavelet is a scaling function, as indicated by the two-scale equation (8.37), so we define also

\[
\varphi_{m,n}(t) = 2^{-m/2} \varphi(2^{-m} t - n) = \frac{1}{2^{m/2}} \varphi \left( \frac{t - n2^m}{2^m} \right) \quad \text{for } m, n \in \mathbb{Z},
\]

(8.45)

which has the same scale and shift relations as discussed for the wavelet above. Note that the factor \(2^{-m/2}\) ensures

\[
\| \psi_{m,n} \| = \| \varphi_{m,n} \| = 1 \quad \text{for } m, n \in \mathbb{Z}.
\]

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 which are powers of 2. The idea is simple: take the original two-scale equation for \(\Phi(\omega)\) in (8.25), and write it at \(\omega/2\):

\[
\Phi(\omega/2) = 2^{-1/2} G(e^{j\omega/4}) \Phi(\omega/4).
\]
Replace this into (8.25) to get
\[ \Phi(\omega) = 2^{-1} G(e^{i\omega/2}) G(e^{i\omega/4}) \Phi(\omega/4) \]
\[ = 2^{-1} G(2)(e^{i\omega/4}) \Phi(\omega/4), \]
where we replaced the product of two filters by the equivalent filter evaluated at \( z = e^{i\omega/4} \); see (8.6). It is not hard to verify (see Exercise 8.2) that iterating the above leads to
\[ \Phi(\omega) = 2^{-k/2} G^{(k)}(e^{i\omega/2^k}) \Phi(\omega/2^k) \quad \text{for} \quad k = 1, 2, \ldots, \quad (8.46) \]
which by inverse Fourier transform leads to
\[ \varphi(t) = 2^{k/2} \sum_n g_n^{(k)} \varphi(2^k t - n) \quad \text{for} \quad k = 1, 2, \ldots. \quad (8.47) \]

Similarly, but starting with the two-scale equation for the wavelet (8.37), we obtain
\[ \Psi(\omega) = 2^{-k/2} H^{(k)}(e^{i\omega/2^k}) G^{(k-1)}(e^{i\omega/2^k}) \Phi(\omega/2^k) \]
\[ = 2^{-k/2} H^{(k)}(e^{i\omega/2^k}) \Phi(\omega/2^k), \quad \text{for} \quad k = 2, 3, \ldots, \quad (8.48) \]
where we used the expression (7.13) for the equivalent filter \( H^{(k)}(z) \), evaluated at \( z = e^{i\omega/2^k} \). Taking an inverse Fourier transform gives
\[ \psi(t) = 2^{k/2} \sum_n h_n^{(k)} \varphi(2^k t - n). \quad (8.49) \]

The attractiveness of the above formulae stems from the fact that any \( \varphi_{m,n}(t) \) or \( \psi_{m,n}(t) \) can be expressed 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 \( g^{(k)} \) or \( h^{(k)} \). We are therefore ready for the main result of this chapter.

**Theorem 8.3 (Orthonormal basis for \( L^2(\mathbb{R}) \)).** Consider a length-\( L \) orthonormal lowpass filter with coefficients \( g \) and a sufficient number of zeros at \( \omega = \pi \), so that the scaling function
\[ \Phi(\omega) = \prod_{i=1}^{\infty} 2^{-1/2} G(e^{i\omega/2^i}) \]
is well defined, and its inverse Fourier transform \( \varphi(t) \) is smooth. Using the coefficients of the orthonormal highpass filter \( h \), the wavelet is given by
\[ \psi(t) = \sqrt{2} \sum_{n=0}^{L-1} h_n \varphi(2t - n). \]

Then, the wavelet family given by
\[ \psi_{m,n}(t) = 2^{-m/2} \psi(2^{-m} t - n), \quad m, n \in \mathbb{Z} \]
is an orthonormal basis of \( L^2(\mathbb{R}) \).
8.2. Wavelet Bases from Filter Banks

Proof. The good news is that most of the hard work has already been done while studying the discrete wavelet transform in Chapter 7.

Consider orthonormality first. The orthogonality of wavelets at the same scale was already shown in (8.42), which means that
\[
\langle \psi_{m,n}, \psi_{m,\ell} \rangle = \delta_{n-\ell}. \tag{8.50}
\]
So we are interested in showing that
\[
\langle \psi_{m,n}(t), \psi_{k,\ell}(t) \rangle = 0, \quad m \neq k. \tag{8.51}
\]
Assume without loss of generality \(m > k\), or \(m = k + i\) with \(i > 0\). Make the change of variable \(t = 2^m \tau\), so (8.51) becomes
\[
\langle \psi_{0,n}(\tau), \psi_{-i,\ell}(\tau) \rangle = 0. \tag{8.52}
\]
Using (8.49), write
\[
\psi_{0,n}(\tau) = 2^{-i/2} \sum_k h_k^{(i)} \phi(2^i(\tau - n) - k) = 2^{-i/2} \sum_k h_k^{(i)} \phi_{-i,2^i(n+k)}(\tau).
\]
Substitute this into (8.52) and use linearity of the inner product to get
\[
2^{-i/2} \sum_k h_k^{(i)} \langle \phi_{-i,2^i(n+k)}(\tau), \psi_{-i,\ell}(\tau) \rangle = 0
\]
as what we must show for all \(i > 0\) and all \(n \) and \(\ell\). In fact, (8.43) implies that every term in the sum is zero, so orthonormality is verified.

We restrict our proof of completeness to the Haar case for simplicity. Consider a causal signal \(x(t)\) of norm 1, with finite length at most \(2^J\) for some \(J \in \mathbb{Z}\). Both of these restriction are inconsequential; the former because a general signal can be decomposed into a causal and an anticausal part, the latter because the fraction of the energy of \(x(t)\) outside of the interval under consideration can be made arbitrarily small by making \(J\) large.

We now approximate \(x(t)\) by a piecewise constant approximation at scale \(i\), (where \(i \ll J\)) or
\[
\hat{x}^{(i)}(t) = 2^{-i} \int_{2^i n}^{2^i(n+1)} x(\tau) d\tau, \quad \text{for} \quad 2^i n \leq t < 2^i(n + 1). \tag{8.53}
\]
Using the Haar scaling function and its shifts and scales as in (8.45), or
\[
\varphi_{m,n}(t) = 2^{-m/2}, \quad \text{for} \quad 2^m n \leq t < 2^m(n + 1),
\]
we can rewrite (8.53) as
\[
\hat{x}^{(i)}(t) = \sum_{n \in \mathbb{Z}} a_n^{(i)} \varphi_{i,n}(t), \tag{8.54}
\]
\[50\]The general case is more involved, but follows a similar intuition. We refer the interested reader to [50, 52], for example.
where
\[ \alpha_n^{(i)} = \langle \varphi_{i,n}, x \rangle. \] (8.55)

Because of the finite length assumption of \( x(t) \), the sequence \( \alpha_n^{(i)} \) is also of finite length (of order \( 2^{J-i} \)). And since \( x(t) \) is of norm 1 and the approximation in (8.54) is a projection, the sequence \( \alpha_n^{(i)} \) has norm smaller or equal to 1. Thus, we can apply Theorem 7.5 and represent the sequence \( \alpha_n^{(i)} \) by discrete Haar wavelets only. The expression (8.54) of \( \hat{x}^{(i)}(t) \) is the piecewise constant interpolation of the sequence \( \alpha_n^{(i)} \), together with proper scaling and normalization. By linearity, we can apply this interpolation to the discrete Haar wavelets used to represent \( \alpha_n^{(i)} \), which leads to a continuous-time Haar wavelet representation of \( \hat{x}^{(i)}(t) \). Again by Theorem 7.5, this representation is exact.

It remains to be shown that \( \hat{x}^{(i)}(t) \) can be arbitrarily close, in \( L^2 \) norm, to \( x(t) \). This is achieved by letting \( i \to -\infty \) and using the density of such piecewise constant functions in \( L^2(\mathbb{R}) \); we get
\[ \lim_{i \to -\infty} \| x(t) - \hat{x}^{(i)}(t) \| = 0. \]

\vspace{1em}

\section*{8.3 The Wavelet Series and its Properties}

An orthonormal basis of wavelets leads to a wavelet series representation of continuous-time functions,
\[ x(t) = \sum_{m \in \mathbb{Z}} \sum_{n \in \mathbb{Z}} \beta^{(m)}_{n} \psi_{m,n}(t) \] (8.56)
where
\[ \beta^{(m)}_{n} = \langle \psi_{m,n}, x \rangle, \quad m, n \in \mathbb{Z}, \] (8.57)
are the wavelet series coefficients. We derived such bases in the previous section and we will see other constructions in Section 8.4.

Let us consider some of the properties of the wavelet series. Many follow from the properties of the wavelet (see Section 8.2.3) or of the discrete wavelet transform (see Chapter 7), and thus our treatment will be brief.

\subsection*{8.3.1 Time-frequency Localization}

Assume that the wavelet \( \psi(t) \) is centered around the origin in time and around \( 3\pi/4 \) in frequency (that is, it is a bandpass filter covering approximately \([\pi/2, \pi]\)). Then, from (8.44), \( \psi_{m,n}(t) \) lives around \( 2^{-m}3\pi/4 \) in frequency. This is shown schematically in Figure 8.11.

With our assumption of \( g \) being a causal FIR filter of length \( L \), the support in time of the wavelets is easy to characterize. The wavelet \( \psi(t) \) lives on \([0, L-1]\), thus
\[ \text{support}(\psi_{m,n}(t)) \subseteq [2^m n, 2^m (n + L - 1)]. \] (8.58)
8.3. The Wavelet Series and its Properties

Figure 8.11: Time-frequency localization of wavelet basis functions. Starting from \( \psi(t) \), we show a higher frequency wavelet \( \psi_{-1,7}(t) \) and a lower frequency wavelet \( \psi_{2,1}(t) \). Note also the dyadic sampling grid given by the points \( (2^m n, 2^{-m} 3\pi/4), m, n \in \mathbb{Z} \). [There are two flaws here: (1) The “centering” of the wavelets is incorrect; e.g., \( \Psi(t) \) should start at \( t = 0 \) not be centered at \( t = 0 \). (2) The length of a wavelet and sampling density should not be confused; the length at scale 0 is \( L - 1 \) while the sampling density is 1.]

By the FIR assumption, the frequency localization is less precise (no compact support in frequency), but the center frequency is around \( 2^{-m} \cdot 3\pi/4 \) and the passband is mostly in an octave band,

\[
\text{support}(\Psi_{m,n}(\omega)) \sim [2^{-m} \frac{\pi}{2}, 2^{-m} \pi].
\] (8.59)

8.3.2 Zero Moments and Decay of Wavelet Coefficients

When the lowpass filter \( g(p) \) has \( N \) zeros at \( \omega = \pi \), we verified that \( \psi(t) \) has \( N \) zero moments (8.40). This property carries over to all scaled versions, and thus we have that for any polynomial signal \( p(t) \) of degree smaller than \( N \),

\[
\beta_{n}^{(m)} = \langle \psi_{m,n}, p \rangle = 0.
\]

This allows us to prove the following result.

**Proposition 8.4 (Decay of WS Coefficients for Functions in \( C^N \)).** For a signal \( x(t) \) with \( N \) continuous and bounded derivations, that is, \( x \in C^N \), the wavelet series coefficients decay as

\[
|\langle \psi_{m,n}, x \rangle| \leq c 2^{mN}
\]

for some constant \( c > 0 \) and \( m \to -\infty \).
Proof. Consider the Taylor series expansion of \( x(t) \) around some point \( t_0 \). Since \( x(t) \) has \( N \) continuous derivatives,

\[
x(t_0 + \epsilon) = x(t_0) + \frac{x'(t_0)}{1!} \epsilon + \frac{x''(t_0)}{2!} \epsilon^2 + \cdots + \frac{x^{(N-1)}(t_0)}{(N-1)!} \epsilon^{N-1} + R_N(\epsilon),
\]

where

\[
|R_N(\epsilon)| \leq \epsilon^N \sup_{t_0 \leq t \leq t_0 + \epsilon} |x^{(N)}(t)|.
\]

View this as a polynomial of degree \( N - 1 \) and a remainder \( R_N(\epsilon) \). Because of the zero moment property of the wavelet, the inner product with the polynomial term is zero, and only the remainder matters. To minimize an upper bound on \( |\langle \psi_{m,n}, R_N(\epsilon) \rangle| \), we want \( t_0 \) close to the center of the wavelet. Since the spacing of the sampling grid at scale \( m \) is \( 2^m \), we see that \( \epsilon \) is at most \( 2^m \) and thus \( |\langle \psi_{m,n}, R_N(\epsilon) \rangle| \) has an upper bound of order \( 2^{mN} \).

A stronger result, in which \( N \) is replaced by \( N + \frac{1}{2} \), follows from Proposition 11.4, which is shown in the context of the continuous wavelet transform.

8.3.3 Characterization of Singularities

Consider the simplest singularity, namely a Dirac at a location \( t_0 \) or \( x(t) = \delta(t - t_0) \). At scale \( m \), only wavelets having their support (given in (8.58)) straddling \( t_0 \) will produce nonzero coefficients. Thus

\[
\beta_n^{(m)} \neq 0 \quad \text{for} \quad \lfloor t_0/2^m \rfloor - L < n \leq \lfloor t_0/2^m \rfloor,
\]

so there are \( L \) nonzero coefficients at each scale. These coefficients correspond to a region of size \( 2^m(L - 1) \) around \( t_0 \), or, as \( m \to -\infty \), they focus arbitrarily closely on the singularity. What about the size of the coefficients at scale \( m \)? The inner product of the wavelet with a Dirac simply picks out a value of the wavelet. Because of the scaling factor \( 2^{-m/2} \) in (8.44), the nonzero coefficients will be of order

\[
|\beta_n^{(m)}| \sim 2^{-m/2}
\]

for the range of \( n \) indicated in (8.60). That is, as \( m \to -\infty \), the nonzero wavelet series coefficients zoom in to the discontinuity, and they grow at a specific rate given by (8.61). This is shown schematically in Figure 8.12.

Generalizing the simple Dirac singularity, a signal is said to have a \( k \)th-order singularity at a point \( t_0 \) when its \( k \)th-order derivative has a Dirac component at \( t_0 \). The scaling (8.61) for a zeroth-order singularity is an example of the following result.

**Proposition 8.5 (Scaling Behavior Around Singularities).** Consider analysis with a wavelet \( \psi \) with \( N \) zero moments. Around a singularity of order \( k \), \( 0 \leq k \leq N \), the wavelet coefficients behave, as \( m \to -\infty \), like

\[
|\beta_n^{(m)}| \sim 2^{m(k-1/2)}.
\]
8.3. The Wavelet Series and its Properties

\[ x(t) \]

\[ t_0 \]

\[ n \]

\[ \beta^{-3} \]

\[ \beta^{-2} \]

\[ \beta^{-1} \]

\[ \beta^{0} \]

\textbf{Figure 8.12:} Singularity at \( t_0 \) and wavelet coefficients across a few scales. Note the focusing on the point \( t_0 \) and the growth of the coefficients as \( m \) grows negative.

\textit{Proof.} The case \( k = 0 \) was shown above. Here we give a proof of the \( k = 1 \) case; generalizing to larger \( k \)s is left as an exercise (see Exercise 8.6).

Assume the wavelet has at least one zero moment (\( N \geq 1 \)) so that there is something to prove for \( k = 1 \). A function with a first-order singularity at \( t_0 \) looks locally at \( t_0 \) like a Heaviside or step function:

\[ x(t) = \begin{cases} 0, & t < t_0; \\ 1, & t \geq t_0. \end{cases} \]

We can bring this back to the previous case by considering the derivative \( x'(t) \), which is a Dirac at \( t_0 \). We use the fact that \( \psi \) has (at least) one zero moment and is of finite support. Then, using integration by parts, one can show that

\[ \langle \psi(t), x(t) \rangle_t = -\langle \psi^{(1)}(t), x'(t) \rangle_t \]

where \( \psi^{(1)}(t) \) is the primitive of \( \psi(t) \) and \( x'(t) \) is the derivative of \( x(t) \). Because \( \psi(t) \) has one (or more) zeros at \( \omega = 0 \) and is of finite support, its primitive is well defined and also has finite support. The key is now the scaling behavior of the primitive of \( \psi_{m,n}(t) \) with respect to the primitive of \( \psi(t) \). Evaluating

\[ \psi^{(1)}_{m,0}(\tau) = \int_0^\tau 2^{-m/2} \psi(2^{-m} t) dt = 2^{m/2} \int_0^{2^{-m}\tau} \psi(t') dt' = 2^{m/2} \psi^{(1)}(2^{-m}\tau), \]

The World of Fourier and Wavelets
Copyright 2008 M. Vetterli, J. Kovačević, and V. K. Goyal

v3.0 [April 15, 2008] Do not distribute
Comments to book-errata@FourierAndWavelets.org
we see that the primitive scales with a factor \(2^{m/2}\). Therefore, the inner product scales as
\[
\left\langle \psi_{m,n}^{(1)}(t), \delta(t-t_0) \right\rangle_t \sim 2^{m/2}
\] (8.63)
at fine scales and close to \(t_0\).

8.4 Multiresolution Analysis

8.4.1 Introduction

How to take a discrete filter and construct a continuous-time basis was the theme of Section 8.2. The converse is one of the themes in the current section! More concretely, we saw that the continuous-time wavelet basis generated a partition of \(L^2(\mathbb{R})\) into a sequence of nested spaces
\[
\cdots \subset V_2 \subset V_1 \subset V_0 \subset V_{-1} \subset V_{-2} \subset \cdots
\] (8.64)
and that these spaces where all scaled copies of each other:
\[
V_i \sim \text{scale } V_0 \text{ by } 2^i.
\] (8.65)

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 filter bank? The answer, as you have guessed, is yes; the framework is multiresolution theory. Rather than give an abstract, axiomatic formulation, we will construct multiresolution theory concretely through the Haar example. This case is prototypical, and leads then easily to the general formulation. This allows to revisit some of the classic wavelet constructions, like the one of Meyer.

The embedded spaces in (8.64) are very natural for piecewise polynomial signals over uniform intervals of lengths \(2^i\). For example, the Haar case leads to piecewise constant signals. The next higher order is for piecewise linear signals, and so on. The natural bases for such spaces are splines, actually Bezier or B-splines. Such bases are not orthogonal, and this leads to orthogonalization methods.

8.4.2 The Haar Case Revisited

Let us start with a simple but enlightening case. Consider the space \(V_0\) of piecewise constant signals over unit intervals, or
\[
f(t) \in V_0
\]
if and only if \(f(t)\) is constant for \(t \in [n, n+1[.\) In addition, \(f(t)\) has to be of finite \(L^2\) norm. Another way to phrase the above is to note that
\[
V_0 = \text{span} \left\{ \{ \varphi(t-n) \}_{n \in \mathbb{Z}} \right\}
\] (8.66)
where \(\varphi(t)\) is our usual Haar scaling function, or
\[
\varphi(t) \begin{cases} 1, & 0 \leq t < 1; \\ 0, & \text{otherwise}, \end{cases}
\]
8.4. Multiresolution Analysis

and since \( \langle \varphi(t-n), \varphi(t-m) \rangle = \delta_{n-m} \), \( \varphi(t) \) and its integer translates form an orthonormal basis for \( V_0 \). So \( f(t) \) can be written as a linear combination

\[
f(t) = \sum_{n \in \mathbb{Z}} \alpha_n^{(0)} \varphi(t-n).
\]

and \( \alpha_n^{(0)} \) is simply the value of \( f(t) \) in the interval \([n, n+1)\). Since \( \|\varphi(t)\|_2 = 1 \), we also have

\[
\|f(t)\|_2 = \|\alpha_n^{(0)}\|_2.
\]

Let us introduce scaled versions of \( V_0 \), namely \( V_m \), which is the space of piecewise constant signals over intervals of size \( 2^m \), that is \([2^mn, 2^m(n+1))\). An orthonormal basis for \( V_m \) is given by

\[
\{ \varphi_{m,n} \}_{n,m \in \mathbb{Z}},
\]

where

\[
\varphi_{m,n}(t) = 2^{-m/2} \varphi\left(\frac{t-n2^m}{2^m}\right) = \begin{cases} 2^{-m/2}, & n2^m \leq t < (n+1)2^m; \\ 0, & \text{otherwise} \end{cases} \tag{8.67}
\]

is the indicator function of the interval \([2^mn, 2^m(n+1))\), weighted by \( 2^{-m/2} \) to be of unit norm. So \( V_m \) is a stretched version of \( V_0 \) (by \( 2^m \), for \( m > 0 \)) or a compressed version of \( V_0 \) (for \( m < 0 \)).

In addition, there is an inclusion relation, since functions which are constant over \([2^mn, 2^m(n+1))\) are also constant over \([2^{\ell}n, 2^{\ell}(n+1))\), \( \ell < m \). Phrased in terms of spaces,

\[
V_m \subset V_{\ell} \quad \text{for} \quad \ell < m. \tag{8.68}
\]

In particular, we have

\[
V_0 \subset V_{-1}.
\]

An orthonormal basis for \( V_{-1} \) is given by (following (8.67))

\[
\left\{ \sqrt{2}\varphi(2t-n) \right\}_{n \in \mathbb{Z}}.
\]

Since \( \varphi(t) \in V_0 \) and \( V_0 \subset V_{-1} \), we can express \( \varphi(t) \) in the basis for \( V_{-1} \) or

\[
\varphi(t) = \sqrt{2} \sum_{n \in \mathbb{Z}} g_n \varphi(2t-n), \tag{8.69}
\]

where

\[
g_n = \langle \varphi(2t-n), \varphi(t) \rangle_t = \begin{cases} 1/\sqrt{2}, & n = 0, 1; \\ 0, & \text{otherwise}. \end{cases} \tag{8.70}
\]

So we have obtained naturally a two-scale equation (8.69), and the coefficients \( g_n \) in (8.70) correspond to the discrete-time Haar filters (see Chapter at a Glance of Chapter 6). Graphically, we show the spaces \( V_0, V_{-1} \), and their basis functions in Figure 8.13, and the two-scale equation in Figure 8.14.

What about the difference spaces? Consider the space of functions which are constant over half integer intervals (or in \( V_{-1} \)) but are not so over integer intervals.
Figure 8.13: The spaces and basis functions for the Haar case. (a) A function $f_0(t)$ belonging to $V_0$. (b) The basis functions for $V_0$. (c) A function $f_{-1}(t)$ in $V_{-1}$. (d) The basis functions for $V_{-1}$.

Figure 8.14: A graphical representation of the two-scale equation in the Haar case. (a) The scaling function $\varphi(t)$. (b) $\varphi(t)$ expressed as a linear combination of $\{\varphi(2t - n)\}$.

Take such a function $f_{-1}(t)$ and decompose it as the sum of its projection onto $V_0$ and onto the orthogonal complement to $V_0$ in $V_{-1}$, which we call $W_0$, or

$$f_{-1}(t) = P_{V_0}(f_{-1}) + P_{W_0}(f_{-1}), \quad (8.71)$$

which is shown graphically in Figure 8.15. Let us calculate the projection of $f_{-1}(t)$ onto $V_0$, calling it $f_0(t)$. This is

$$f_0(t) = \sum_{n \in \mathbb{Z}} \alpha_n^{(0)} \varphi(t - n)$$

with

$$\alpha_n^{(0)} = \langle \varphi(t - n), f_{-1}(t) \rangle_t.$$
Now, express both \( \varphi(t - n) \) and \( f_{-1}(t) \) in terms of the orthonormal basis for \( V_{-1} \),

\[
\varphi(t - n) = \varphi(2t - 2n) + \varphi(2t - 2n - 1),
\]

\[
f_{-1}(t) = \sqrt{2} \sum_{\ell \in \mathbb{Z}} \alpha_{\ell}^{(-1)} \varphi(2t - \ell),
\]

to get

\[
\alpha_{n}^{(0)} = \sqrt{2} \sum_{\ell \in \mathbb{Z}} \alpha_{n}^{(-1)} \langle \varphi(2t - 2n) + \varphi(2t - 2n - 1), \varphi(2t - \ell) \rangle_t
\]

\[
= \frac{1}{\sqrt{2}} \left( \alpha_{2n}^{(-1)} + \alpha_{2n+1}^{(-1)} \right),
\]

(8.72)

where we used the fact that the inner product is zero unless \( \ell = 2n \) or \( 2n + 1 \), in which case it is 1/2. In other words, \( f_0(t) \) is simply the average of \( f_{-1}(t) \) over the successive intervals. Indeed, if \( f_{-1}(t) \) has value \( f_{-1}(n) \) and \( f_{-1}(n + 1/2) \) over two successive intervals, \( [n, n + 1/2) \) and \( [n + 1/2, n + 1) \), then

\[
f_0(t) = \frac{f_{-1}(n) + f_{-1}(n + 1/2)}{2}, \quad n \leq t < n + 1
\]

(8.73)

as can be checked (beware of the normalization!). This is the best least squares approximation of \( f_{-1}(t) \) by a function in \( V_0 \) (Exercise 8.4).

To see the structure of \( W_0 \), subtract the projection \( f_0(t) \) from \( f_{-1}(t) \), and call this difference \( d_0(t) \). Since \( f_0(t) \) is an orthogonal projection,

\[
d_0(t) = f_{-1}(t) - f_0(t)
\]

(8.74)

is orthogonal to \( V_0 \); see also Figure 8.15. Using (8.73) and (8.74) leads to

\[
d_0(t) = \begin{cases} \frac{1}{2}(f_{-1}(n) - f_{-1}(n + 1/2)), & n \leq t < n + 1/2; \\ -\frac{1}{2}(f_{-1}(n) - f_{1}(n + 1/2)), & n + 1/2 \leq t < n + 1, \end{cases}
\]

(8.75)
and clearly $f_{-1}(t) = f_0(t) + d_0(t)$. Noting that $d_0(t)$ has alternating signs and using the Haar wavelet

$$\psi(t) = \begin{cases} 
1, & 0 \leq t < 1/2; \\
-1, & 1/2 \leq t < 1; \\
0, & \text{otherwise}, 
\end{cases}$$

we can rewrite $d_0(t)$ as

$$d_0(t) = \sum_n \beta_n^{(0)} \psi(t - n) \quad (8.76)$$

with

$$\beta_n^{(0)} = \frac{1}{\sqrt{2}} \left( \alpha_{2n}^{(-1)} - \alpha_{2n+1}^{(-1)} \right). \quad (8.77)$$

This last expression is verified similarly to (8.72).

Figure 8.16 shows schematically what we have derived so far. Informally, we have shown that

$$V_{-1} = V_0 \oplus W_0 \quad (8.78)$$
8.4. Multiresolution Analysis

and gave bases for all of these spaces, namely scaling functions for $V_{-1}$, $V_{0}$, and wavelets for $W_{0}$. There is one more property that follows immediately from the inclusion $W_{0} \subset V_{-1}$, namely that $\psi(t)$ can be written in terms of a linear combination of $\{\varphi(2t - n)\}_{n \in \mathbb{Z}}$, or

$$\psi(t) = \sqrt{2} \sum h_n \varphi(2t - n) \quad (8.79)$$

with

$$h_n = \begin{cases} 1/\sqrt{2}, & n = 0 \\ -1/\sqrt{2}, & n = 1 \\ 0, & \text{otherwise} \end{cases} \quad (8.80)$$

similarly to (8.69)–(8.70).

8.4.3 Axioms of Multiresolution Analysis

Let us summarize the fundamental characteristics of the spaces and basis functions seen in the Haar case. First, there is a sequence of embedded spaces

$$\cdots \subset V_{2} \subset V_{1} \subset V_{0} \subset V_{-1} \subset V_{-2} \subset \cdots, \quad (8.81)$$

where $V_{m}$ is the space of function with finite $L^2$ norm that are piecewise constant over $[2^m n, 2^m (n+1)]_{n \in \mathbb{Z}}$. Such spaces are just “scaled” copies of each other, and the embedding is obvious. We can call the $V_{m}$s successive approximation spaces, since as $m \to -\infty$, we get finer and finer approximations. Since piecewise constant functions of arbitrarily-fine pieces are dense in $L^2$, \footnote{That is, any $L^2$ function can be approximated arbitrarily closely by a piecewise constant function over intervals that tend to 0, as they do in $V_{m}$ when $m \to -\infty$.} we can say that

$$\lim_{m \to -\infty} V_{m} = \bigcup_{m \in \mathbb{Z}} V_{m} = L^2(\mathbb{R}). \quad (8.82)$$

As $m \to +\infty$, we get coarser and coarser approximations. Given a function $f$ in $L^2(\mathbb{R})$, its projection onto $V_{m}$ tends to zero as $m \to \infty$, since we lose all the details. More formally, we can say

$$\bigcap_{m \in \mathbb{Z}} V_{m} = \{0\}. \quad (8.83)$$

The fact that the spaces $V_{m}$ are just scales of each other means that

$$f(t) \in V_{m} \iff f(2t) \in V_{m-1}, \quad (8.84)$$

and similarly for any scaling by a power of 2. Because $f$ is piecewise constant over intervals $[2^m n, 2^m (n+1)]$, it is invariant to shifts by multiples of $2^m$, or

$$f(t) \in V_{m} \iff f(t - 2^m n) \in V_{m}. \quad (8.85)$$

Finally, with the indicator function of the unit interval, $\varphi(t)$, we have that

$$\{\varphi(t - n)\}_{n \in \mathbb{Z}} \text{ is a basis for } V_{0}. \quad (8.86)$$

The above six characteristics, which follow naturally from the Haar case, are defining characteristics of a broad class of wavelet systems.
Definition 8.4 (Multiresolution analysis). A sequence \( \{V_m\}_{m \in \mathbb{Z}} \) of subspaces of \( L^2(\mathbb{R}) \) satisfying (8.81)–(8.86) is called a multiresolution analysis. The spaces \( V_m \) are called the successive approximation spaces. The spaces \( W_m \), defined as the orthogonal complements of \( V_m \) in \( V_{m-1} \), are called the successive detail spaces, where

\[
V_{m-1} = V_m \oplus W_m. \tag{8.87}
\]

Note that in (8.86), we had an orthonormal basis. This is not necessary; a general (Riesz) basis is sufficient, and we denote it by \( \{\theta(t-n)\}_{n \in \mathbb{Z}} \). Given the above, the following holds.

Proposition 8.6. Consider a multiresolution analysis with a Riesz basis \( \{\theta(t-n)\}_{n \in \mathbb{Z}} \) for \( V_0 \). Then there is a scaling function \( \phi(t) \) such that \( \{\phi(t-n)\}_{n \in \mathbb{Z}} \) is an orthonormal basis for \( V_0 \). Furthermore, this scaling function satisfies a two-scale equation

\[
\phi(t) = \sqrt{2} \sum_{n \in \mathbb{Z}} g_n \phi(2t-n). \tag{8.88}
\]

Proof. A function \( f(t) \) is orthonormal to its integer translates,

\[
\langle f(t), f(t-n) \rangle_t = \delta_n,
\]

if and only if its Fourier transform satisfies

\[
\sum_{k \in \mathbb{Z}} |F(\omega + 2\pi k)|^2 = 1. \tag{8.89}
\]

Since \( \theta(t) \) and its integer shifts is a Riesz basis, it can be shown that its Fourier transform \( \Theta(\omega) \) satisfies

\[
0 < \sum_{k \in \mathbb{Z}} |\Theta(\omega + 2\pi k)|^2 < \infty.
\]

So, as a good guess for an orthonormal scaling function, pick its Fourier transform as

\[
\Phi(\omega) = \frac{\Theta(\omega)}{\left(\sum_{k \in \mathbb{Z}} |\Theta(\omega + 2\pi k)|^2\right)^{1/2}}, \tag{8.90}
\]

which satisfies

\[
\sum_{\ell \in \mathbb{Z}} |\Phi(\omega + 2\pi \ell)|^2 = \frac{1}{\sum_{k \in \mathbb{Z}} |\Theta(\omega + 2\pi k)|^2} \sum_{\ell \in \mathbb{Z}} |\Theta(\omega + 2\pi \ell)|^2 = 1,
\]

where the denominator was taken outside the sum since it is \( 2\pi \) periodic. So \( \varphi(t) \) is orthogonal to its integer translates. The fact that it satisfies a two-scale equation was shown in (8.69) for the Haar case, but the argument based on inclusion of \( V_0 \) in \( V_{-1} \) is general.

Thus, though we did not require it as part of the definition of a multiresolution analysis, assume an orthogonal basis \( \{\varphi(t-n)\}_{n \in \mathbb{Z}} \) for \( V_0 \). What can we say about
the coefficients \( g_n \) of the two-scale equation in (8.88)? Evaluating the inner product \( \langle \varphi(t), \varphi(t - \ell) \rangle_t \) and replacing both operands by their equivalent two-scale equation leads to

\[
\langle \varphi(t), \varphi(t - \ell) \rangle_t = 2 \sum_{n} \sum_{m} g_n g_m \langle \varphi(2t - n), \varphi(2t - 2\ell - m) \rangle_t
\]

\[
\overset{(a)}{=} \sum_{n} g_n g_{n - 2\ell} \overset{(b)}{=} \delta_{\ell},
\]

where (a) follows since \( \langle \varphi(2t - n), \varphi(2t - 2\ell - m) \rangle_t \) is zero except for \( n = 2\ell + m \) when it is \( 1/2 \); and (b) comes from orthogonality of \( \varphi(t) \) and its integer translates. So the sequence \( \{g_n\} \) corresponds to an orthogonal filter in a filter bank. Assuming that the Fourier transform \( \Phi(\omega) \) of \( \varphi(t) \) is continuous and satisfies

\[
|\Phi(0)| = 1,
\]

it follows from the two-scale equation in Fourier domain that

\[
|G(1)| = \sqrt{2},
\]

and so \( \{g_n\} \) is a lowpass sequence. Consider it to be of finite length \( L \) and derive the equivalent highpass filter

\[
h_n = (-1)^n g_{L - n - 1}, \quad n = 0, 1, \ldots, L - 1.
\]

Defining a wavelet as

\[
\psi(t) = \sqrt{2} \sum_{n=0}^{L-1} h_n \varphi(2t - n), \quad (8.91)
\]

we then have:

**Proposition 8.7.** The wavelet given by (8.91) satisfies

\[
\langle \psi(t), \psi(t - n) \rangle_t = \delta_n
\]

\[
\langle \psi(t), \varphi(t - n) \rangle_t = 0,
\]

and \( W_0 = \text{span} \{\psi(t - n)\}_{n \in \mathbb{Z}} \) is the orthogonal complement of \( V_0 \) in \( V_{-1} \),

\[
V_{-1} = V_0 \oplus W_0. \quad (8.92)
\]

We do not formally prove this here. The orthogonality relations follow easily from the orthogonality of the sequences \( g_n \) and \( h_n \) by using the two-scale equation. That \( \{\psi(t - n)\}_{n \in \mathbb{Z}} \) is an orthonormal basis for \( W_0 \) requires checking completeness and is more technical. By construction, the \( W_m \) spaces are also simply scales of each other, or

\[
f(t) \in W_m \iff f(2t) \in W_{m - 1}. \quad (8.93)
\]

Putting all the pieces above together, we have:

This can be shown to follow from completeness (8.82), for example, if \( \varphi(t) \) is integrable.
Theorem 8.8 (Wavelet bases for \( L^2(\mathbb{R}) \)). Consider a multiresolution analysis of \( L^2(\mathbb{R}) \). Then the family
\[
\psi_{m,n}(t) = \frac{1}{2^{m/2}} \psi \left( \frac{t - n2^m}{2^m} \right), \quad m, n \in \mathbb{Z},
\]
with \( \psi(t) \) as defined in (8.91), is an orthonormal basis of \( L^2(\mathbb{R}) \).

Proof. By scaling of (8.92),
\[
V_m = V_{m+1} \oplus W_{m+1}.
\]
Iterating this \( k \) times leads to
\[
V_m = W_{m+1} \oplus W_{m+2} \oplus \ldots W_{m+k} \oplus V_{m+k}.
\]
As \( k \to \infty \) and because of (8.83), we get
\[
V_m = \bigoplus_{i=m+1}^\infty W_i,
\]
and finally, letting \( m \to -\infty \) and because of (8.82), we obtain
\[
L^2(\mathbb{R}) = \bigoplus_{i=-\infty}^\infty W_i.
\]
Since \( \{\psi(t - n)\}_{n \in \mathbb{Z}} \) is an orthonormal basis for \( W_0 \), by scaling, \( \{\psi_{m,n}(t)\} \) is an orthonormal basis for \( W_m \). Then, following (8.94), the family \( \{\psi_{m,n}(t)\}_{m,n \in \mathbb{Z}} \) is an orthonormal basis for \( L^2(\mathbb{R}) \).

In sum, and in complementary fashion to Section 8.2, we obtain a split of \( L^2(\mathbb{R}) \) into a collection \( \{W_m\}_{m \in \mathbb{Z}} \) as a consequence of the axioms of multiresolution analysis given by (8.81)–(8.86). A graphical representation of the spaces \( \{V_m\} \) and \( \{W_m\} \) is given in Figure 8.17. Let us see the principles at work through an example.

Example 8.7 (Piecewise-linear Embedded Spaces). Consider the space \( V_0 \) of \( L^2 \) functions which are continuous and piecewise linear over intervals \([n, n+1)\), or \( f(t) \in V_0 \) if \( \|f\|_2 < \infty \) and \( f'(t) \) is piecewise constant over intervals \([n, n+1)\). For simplicity, consider causal functions, or \( f(t) = 0, t < 0 \). Then \( f'(t) \) is specified by a sequence \( \{a_n\}, n = 0, 1, \ldots \) which indicates the slopes of \( f(t) \). The nodes of \( f(t) \), that is, the values at the integers, are given by
\[
f(n) = \begin{cases} 
0, & n \leq 0; \\
\sum_{i=0}^{n-1} a_i, & n > 0,
\end{cases}
\]
and the piecewise linear function is
\[
f(t) = (f(n+1) - f(n))(t - n) + f(n) = a_n(t - n) + \sum_{i=0}^{n-1} a_i \tag{8.95}
\]
for \( t \in [n, n+1) \). See Figure 8.18 for a graphical representation of such a function.

\footnote{In the infinite sum, we imply closure, or inclusion of all limit points.}
8.4. Multiresolution Analysis

The spaces $V_m$ are simply scaled versions of $V_0$; they contain functions which are continuous and piecewise linear over intervals $[2^m n, 2^m (n + 1))$. Let us verify the axioms of multiresolution. Embedding as in (8.81) is clear, and similarly to the piecewise constant case, piecewise linear functions are dense in $L^2(\mathbb{R})$. Thus

$$\lim_{m \to -\infty} V_m = L^2(\mathbb{R}).$$

Conversely, as $m \to \infty$, the approximation gets coarser and coarser, ultimately verifying (8.83). Scaling (8.84) and shift-invariance (8.85) are clear from the definition of the piecewise linear functions over intervals scaled by powers of 2. It remains to find a basis for $V_0$. Take as a good guess the hat function, defined (in its causal

Figure 8.17: A graphical representation of the embedding of the $V_m$ spaces, and the successive detail spaces $W_m$.

Figure 8.18: A continuous and piecewise linear function and its derivative. (a) The function $f(t)$. (b) The derivative $f'(t)$.
version) as
\[
\theta(t) = \begin{cases} 
  t, & 0 \leq t < 1; \\
  1 - t, & 1 \leq t < 2; \\
  0, & \text{otherwise.}
\end{cases}
\]

Then \(f(t)\) in (8.95) can be written as
\[
f(t) = \sum_{n=0}^{\infty} b_n \theta(t - n),
\]
with \(b_0 = a_0\) and
\[
b_n = a_n + b_{n-1}.
\]

This is verified by taking derivatives. First,
\[
\theta'(t) = I(t) - I(t - 1),
\]
where \(I(t)\) is the indicator function of the unit interval. Thus, \(f'(t)\) is piecewise constant. Then, the value of the constant between \(n\) and \(n+1\) is \(b_n - b_{n-1}\) and thus equals \(a_n\) as desired. The only detail is that \(\theta(t)\) is not orthogonal to its integer translates, since
\[
\langle \theta(t), \theta(t-n) \rangle_t = \begin{cases} 
  1/6, & n = -1, 1; \\
  2/3, & n = 0; \\
  0, & \text{otherwise.}
\end{cases}
\]

We can apply the orthogonalization procedure in (8.90). The discrete-time Fourier transform of the sequence \([1/6, 2/3, 1/6]\) is
\[
\frac{2}{3} + \frac{1}{6}e^{j\omega} + \frac{1}{6}e^{-j\omega} = \frac{1}{6} (4 + z + z^{-1})|_{z = e^{j\omega}}.
\]

Since it is an autocorrelation, it is positive on the unit circle and can thus be factored into its spectral roots. So, write
\[
4 + z^{-1} + z = (2 + \sqrt{3})(1 + (2 - \sqrt{3})z)(1 + (2 - \sqrt{3})z^{-1}).
\]

Take as spectral root the causal part,
\[
\left(\frac{2 + \sqrt{3}}{6}\right)^{1/2} (1 + (2 - \sqrt{3})z^{-1}).
\]

Its inverse has impulse response
\[
\alpha_n = \left(\frac{2 + \sqrt{3}}{6}\right)^{1/2} \cdot (-1)^n (2 - \sqrt{3})^n.
\]

Finally, \(\varphi_c(t)\) is given by
\[
\varphi_c(t) = \sum_{n=0}^{\infty} \alpha_n \theta(t - n),
\]
8.4. Multiresolution Analysis

![Graphs illustrating basis functions for piecewise linear spaces.](image)

**Figure 8.19:** Basis function for piecewise linear spaces. (a) The non-orthogonal basis function \( \theta(t) \). (b) An orthogonalized causal basis function \( \varphi_c(t) \). (c) An orthogonalized symmetric basis function \( \varphi_s(t) \).

which is a causal function orthonormal to its integer translates. It is piecewise linear over integer pieces, but of infinite extent (see Figure 8.19).

Instead of the spectral factorization, we can just take the square root as in (8.90). 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 8.19).

Because of the embedding of the spaces \( V_m \), the scaling functions all satisfy two-scale equations (see Exercise 8.5). 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 (see Exercise 8.6).
8.4.4 The Sinc and Meyer Wavelets

The principles of multiresolution analysis come nicely into play when considering one of the earliest wavelet constructions, due to Meyer. The idea is to improve the sinc approximation, which we briefly review. Consider as
\[ \mathcal{V}_0 \]
the space of \( L^2 \) functions bandlimited to \([-\pi, \pi]\), or \( BL[-\pi, \pi] \), for which
\[
\varphi(t) = \frac{\sin(\pi t)}{\pi t} \quad (8.96)
\]
and its integer shifts form an orthonormal basis. Define
\[
\mathcal{V}_m = BL[-2^{-m}\pi, 2^{-m}\pi] \quad (8.97)
\]
as the nested spaces of bandlimited functions, which obviously satisfy
\[
\cdots \subset \mathcal{V}_2 \subset \mathcal{V}_1 \subset \mathcal{V}_0 \subset \mathcal{V}_{-1} \subset \mathcal{V}_{-2} \subset \cdots
\]
The axioms of multiresolution analysis in Definition 8.4 are clearly satisfied, i.e. the union of the \( \mathcal{V}_m \)'s is \( L^2(\mathbb{R}) \), their intersection is empty, the spaces are scales of each other (by powers of 2) and they are shift invariant with respect to shifts by integer multiples of \( 2^m \). Finally, \( \{\varphi(t-n)\}_{n \in \mathbb{Z}} \) is an orthonormal basis of \( \mathcal{V}_0 \).

The details are left as Exercise 8.7, including the derivation of the wavelet and the detail spaces \( \mathcal{W}_m \), which are given by the space of \( L^2 \) bandpass functions,
\[
\mathcal{W}_m = [-2^{-m+1}\pi, -2^{-m}\pi] \cup [2^{-m}\pi, 2^{-m+1}\pi], \quad (8.98)
\]
We give an illustration in Figure 8.20.

While the perfect bandpass spaces lead to a bona fide multiresolution analysis of \( L^2(\mathbb{R}) \), 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 (8.96)). The idea behind Meyer’s wavelet construction is to “smooth” the sinc solution, so as to obtain faster decay of the time domain basis functions. We will examine this through a specific example.

Example 8.8 (Meyer Wavelet). The idea is to construct a scaling function \( \varphi(t) \), orthogonal to its integer translates, and with a faster decay than the sinc function. In addition, this function is an orthonormal basis for a space \( \mathcal{V}_0 = \text{span}(\{\varphi(t-n)\}_{n \in \mathbb{Z}}) \), and this space and its scaled versions \( \mathcal{V}_m \) need to satisfy the multiresolution constraints.

The simplest case appears when the Fourier transform magnitude of the scaling function, \( |\Phi(\omega)|^2 \), is piecewise linear, and given by
\[
|\Phi(\omega)|^2 = \begin{cases} 
1, & |\omega| < \frac{2\pi}{3}; \\
2 - \frac{3|\omega|}{2\pi}, & \frac{2\pi}{3} < |\omega| < \frac{4\pi}{3}.
\end{cases} \quad (8.99)
\]

This function is shown in Figure 8.21, where we also show graphically that
\[
\sum_{k \in \mathbb{Z}} |\Phi(\omega + 2\pi k)|^2 = 1,
\]
8.4. Multiresolution Analysis

Figure 8.20: Scaling and wavelet in the sinc case. (a) Scaling function $\varphi(t)$. (b) Fourier transform magnitude $|\Phi(\omega)|$. (c) Wavelet $\psi(t)$. (d) Fourier transform magnitude $|\Psi(\omega)|$.

which proves that $\{\varphi(t-n)\}_{n \in \mathbb{Z}}$ is an orthonormal set (see (8.89)).

Define

$$V_0 = \text{span} \{\{\varphi(t-n)\}_{n \in \mathbb{Z}}\}$$

as well as all scaled versions $V_m$. The conditions (8.82) and (8.83) hold, similarly to the sinc case. By construction, scaling (8.84) and shift invariance (8.85) hold as well. We now verify inclusion (8.81), or

$$V_0 \subset V_{-1}.$$ Again, a graphical explanation is easiest, since we see in Figure 8.22 that $V_0$ is perfectly represented in $V_{-1}$. This means we can find a $2\pi$-periodic function $G(e^{j\omega})$ so as to satisfy the two-scale equation (8.88), which becomes in Fourier domain (see (8.25))

$$\Phi(\omega) = \frac{1}{\sqrt{2}} G(e^{j\omega/2})\Phi(\omega/2).$$

See Figure 8.23 for a pictorial illustration.

From (8.99) and the figure, the discrete-time filter $g$ has a Fourier transform given by

$$|G(e^{j\omega})| = \begin{cases} \sqrt{2}, & |\omega| \leq \pi/3; \\ \sqrt{4 - \frac{6|\omega|}{\pi}}, & \pi/3 \leq |\omega| < 2\pi/3; \\ 0, & 2\pi/3 < |\omega| \leq \pi. \end{cases}$$ (8.100)
Figure 8.21: Meyer scaling function, with a piecewise linear squared Fourier transform magnitude. (a) The function $|\Phi(\omega)|^2$. (b) Proof of orthogonality by verifying (8.89).

Figure 8.22: Inclusion relation $V_0 \subset V_{-1}$ for the Meyer wavelet. We simply show the Fourier transform magnitude $|\Phi(\omega)|^2$ and $|\Phi(\omega/2)|^2$ as representations of $V_0$ and $V_{-1}$, respectively.

While the phase is not specified, it can be chosen as zero, or $G(e^{j\omega})$ real and symmetric. Such a filter has an infinite impulse response, and its $z$-transform is non-rational (since it is exactly zero over an interval of nonzero measure). It satisfies the quadrature formula for an orthogonal lowpass filter from (6.16),

$$|G(e^{j\omega})|^2 + |G(e^{j(\omega+\pi)})|^2 = 2.$$  

Choosing the highpass filter in the standard way

$$H(e^{j\omega}) = e^{-j\omega} G(e^{j(\omega+\pi)}),$$

where we used the fact that $G(e^{j\omega})$ is real, leads to the wavelet

$$\Psi(\omega) = \frac{1}{\sqrt{2}} H(e^{j\omega/2}) \Phi(e^{j\omega/2}).$$

Since $H(e^{j\omega/2})$ is (up to a phase factor) a stretched and shifted version of $G(e^{j\omega})$,  

8.5. Algorithms

The two-scale equation for the Meyer wavelet in frequency domain. Note how the $4\pi$-periodic function $G(e^{j\omega/2})$ “carves out” $\Phi(\omega)$ from $\Psi(\omega/2)$.

The construction and resulting wavelet are shown in Figure 8.24. Note the bandpass characteristic of the wavelet. Finally, the scaling function $\phi(t)$ and wavelet $\psi(t)$ are shown, together with their Fourier transforms, in Figure 8.25.

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 $V_m$ are embedded. Since $\Phi(\omega)$ is continuous, $\psi(t)$ has $O(1/t^2)$ decay. Smoother $\Phi(\omega)$’s can be constructed, so as to have faster decay of $\phi(t)$ (see Exercise 8.8).

8.5 Algorithms

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.

8.5.1 Mallat’s Algorithm

Given a wavelet basis $\{\psi_{m,n}(t)\}_{m,n \in \mathbb{Z}}$, any function $f(t)$ can be written as

$$ f(t) = \sum_{m \in \mathbb{Z}} \sum_{n \in \mathbb{Z}} \beta_n^{(m)} \psi_{m,n}(t). $$

where

$$ \beta_n^{(m)} = \langle \psi_{m,n}, f \rangle. $$
Assume that only a finite-resolution version of \( f(t) \) can be acquired, in particular the projection of \( f(t) \) onto \( V_0 \), denoted \( f^{(0)}(t) \). Because

\[
V_0 = \bigoplus_{m=1}^{\infty} W_m,
\]

we can write

\[
f^{(0)}(t) = \sum_{m=1}^{\infty} \sum_{n \in \mathbb{Z}} \beta^{(m)}_n \psi_{m,n}(t). \tag{8.103}
\]

Since \( f^{(0)}(t) \in V_0 \), we can also write

\[
f^{(0)}(t) = \sum_{n \in \mathbb{Z}} \alpha^{(0)}_n \varphi(t - n), \tag{8.104}
\]

where

\[
\alpha^{(0)}_n = \langle \varphi(t - n), f(t) \rangle_t = \langle \varphi_{0,n}, f \rangle.
\]

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

\[
\alpha^{(1)}_n = \left\langle \frac{1}{\sqrt{2}} \varphi(t/2 - n), f^{(0)}(t) \right\rangle_t, \quad n \in \mathbb{Z}. \tag{8.105}
\]
From (8.88), we can write
\[
\frac{1}{\sqrt{2}} \varphi(t/2 - n) = \sum_{k \in \mathbb{Z}} g_k \varphi(t - 2n - k). \tag{8.106}
\]

Replacing this and (8.104) into (8.105) leads to
\[
\alpha_n^{(1)} = \sum_{k \in \mathbb{Z}} \sum_{\ell \in \mathbb{Z}} g_k \alpha_\ell^{(0)} \langle \varphi(t - 2n - k), \varphi(t - \ell) \rangle_t
\]
\[
\overset{(a)}{=} \sum_{\ell \in \mathbb{Z}} g_{\ell - 2n} \alpha_\ell^{(0)} \overset{(b)}{=} (\tilde{g} * \alpha^{(0)})_{2n}, \tag{8.107}
\]

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 subsampling by 2. The same development for the wavelet series coefficients
\[
\beta_n^{(1)} = \left\langle \frac{1}{\sqrt{2}} \psi(t/2 - n), f^{(0)}(t) \right\rangle_t
\]
yields

\[ \tilde{g}^{(1)}_n = (\tilde{h} * \alpha^{(0)})_{2n}, \]

(8.108)

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 signal \( f^{(m)} \) in \( V_m \) 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 \( \alpha^{(m)} \) with \( \tilde{g} \) and \( \tilde{h} \), respectively. Likewise, this process can be iterated \( k \) times, to go from \( V_m \) to \( V_{m+k} \), while splitting off \( W_{m+1}, W_{m+2}, \ldots, W_{m+k} \), or

\[ V_m = W_{m+1} \oplus W_{m+2} \oplus \ldots \oplus W_{m+k} \oplus V_{m+k}. \]

The key insight is of course that once we have an initial projection, e.g. \( f^{(0)}(t) \) with expansion coefficients \( \alpha^{(0)}_n \), then all the other expansion coefficients can be computed using discrete-time filtering. This is shown in Figure 8.27, where the sequence \( \alpha^{(0)}_n \), corresponding to an initial projection of \( f(t) \) 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.

### 8.5.2 Initialization

How do we initialize Mallat’s algorithm, that is, compute the initial sequence \( \alpha^{(0)}_n \)? There is no escape from computing the inner products

\[ \alpha^{(0)}_n = \langle \varphi(t-n), f(t) \rangle_t = \langle \tilde{\varphi} * f \rangle_{t=n}, \]

where \( \tilde{\varphi}(t) = \varphi(-t) \). This is shown in Figure 8.28.

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 (8.102), only a single scale has to be computed, namely the one leading to \( \alpha^{(0)}_n \). All the subsequent inner products are obtained from that sequence, using only discrete-time processing.
Figure 8.27: Mallat’s algorithm. From the initial sequence $\alpha_n^{(0)}$, all of the wavelet series coefficients are computed through a discrete filter bank algorithm.

$$\alpha_n^{(0)} = \langle \phi_{1,n} \rangle$$

$$\beta_n^{(1)} = \langle \psi_{1,n} \rangle$$

Figure 8.28: Initialization of Mallat’s algorithm. The function $f(t)$ is convolved with $\tilde{\varphi}(t) = \varphi(-t)$ and sampled at $t = n$.

The question is: How well does $f^{(0)}(t)$ approximate the function $f(t)$? The key is that if the error $\|f^{(0)} - f\|$ is too large, we can go to finer resolutions $f^{(m)}$, $m < 0$, until $\|f^{(m)} - f\|$ 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 8.29, we show two different initial approximations and the resulting errors,

$$e^{(m)}(t) = f(t) - f^{(m)}(t).$$

Clearly, the smoother the function, the faster the decay of $\|e^{(m)}\|$ as $m \to -\infty$. Exercise 8.9 explores this further.

8.5.3 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 (8.103), how to synthesize $f^{(0)}(t)$. One way is to effectively add wavelets at different scales and shifts, with the appropriate weights (8.102).

The other way is to synthesize $f^{(0)}(t)$ as in (8.104), 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 $d^{(1)}(t) \in W_1$. 

$$\sum_{n \in \mathbb{Z}} \delta(t-n)$$
Figure 8.29: Initial approximation in Mallat’s algorithm. (a) Signal $f(t)$. (b) Approximation $f^{(0)}(t)$ with Haar scaling function in $V_0$ and error $e^{(0)}(t) = f(t) - f^{(0)}(t)$. (c) Same but in $V_{-3}$, or $f^{(-3)}(t)$ and $e^{(-3)}(t) = f(t) - f^{(-3)}(t)$.

There are two ways to write $f^{(0)}(t)$, namely

$$f^{(0)}(t) = \sum_{n \in \mathbb{Z}} \alpha_n^{(0)} \varphi(t - n)$$

$$= \frac{1}{\sqrt{2}} \sum_{n \in \mathbb{Z}} \alpha_n^{(1)} \varphi(t/2 - n) + \frac{1}{\sqrt{2}} \sum_{n \in \mathbb{Z}} \beta_n^{(1)} \psi(t/2 - n),$$

where the latter is the sum of $f^{(1)}(t)$ and $d^{(1)}(t)$. Now,

$$\alpha_n^{(0)} = \left< \varphi(t - \ell), f^{(0)}(t) \right>_{\ell}.$$

Using the two-scale equation (8.106) 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

$$\alpha_0(t) = \langle \varphi(t-\ell), f^{(1)}(t) \rangle_t + \langle \varphi(t-\ell), d^{(1)}(t) \rangle_t$$

$$(a) \sum_{n \in \mathbb{Z}} \sum_{k \in \mathbb{Z}} a_n^{(1)} g_k \langle \varphi(t-\ell), \varphi(t-2n-k) \rangle_t$$

$$+ \sum_{n \in \mathbb{Z}} \sum_{k \in \mathbb{Z}} b_n^{(1)} h_k \langle \varphi(t-\ell), \varphi(t-2n-k) \rangle_t$$

$$(b) \sum_{n \in \mathbb{Z}} a_n(1) g_{\ell-2n} + \sum_{n \in \mathbb{Z}} b_n(1) h_{\ell-2n}$$

(8.111)

where (a) follows from (8.110) using the two-scale equation; and (b) is obtained from the orthogonality of the $\varphi$s, unless $k = \ell - 2n$. The obtained expression for $\alpha_0(t)$ indicates that the two sequences $a_n^{(1)}$ and $b_n^{(1)}$ 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 (8.109). The argument above can be extended to any number of scales and leads to the synthesis version of Mallat’s algorithm, shown in Figure 8.30.

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.
Appendices

8.A Proof of Proposition 8.2

We need to prove that

$$|\Phi(\omega)| < \frac{c}{1 + |\omega|^{1+\epsilon}}, \quad \epsilon > 0,$$

which amounts to prove that (see (8.23))

$$|\gamma(\omega)| = \prod_{i=1}^{\infty} 2^{-1/2} R(e^{j\omega/2^i}) < c'(1 + |\omega|)^{(N-1-\epsilon)}$$

(8.112)

since we have shown that there is a “smoothing term” of order $1/|\omega|^N$ for large $\omega$ (see (8.22)). Recall that $R(e^{j\omega})$ is $2\pi$-periodic and that $R(1) = \sqrt{2}$. Because $|R(e^{j\omega})| < 2^{N-1/2}$ by assumption, we can find a constant $\alpha$ such that

$$|R(e^{j\omega})| \leq \sqrt{2}(1 + \alpha |\omega|).$$

Using a Taylor series for the exponential,

$$|R(e^{j\omega})| \leq \sqrt{2} e^{\alpha |\omega|}.$$  (8.113)

Consider now $\gamma(\omega)$ for $|\omega| \leq 1$, and let us find an upper bound based on the bound on $|R(e^{j\omega})|$:

$$\sup_{|\omega| \leq 1} |\gamma(\omega)| = \sup_{|\omega| \leq 1} \prod_{k=1}^{\infty} 2^{-1/2} R(e^{j\omega/2^k})$$

(a) \leq \prod_{k=1}^{\infty} e^{\alpha |\omega|/2^k} = e^{\alpha |\omega|(1/2+1/4+\cdots)}

(b) \leq e^{\alpha},$$

(8.114)

where (a) follows from (8.113); and (b) comes from $|\omega| \leq 1$.

For $|\omega| \geq 1$, there exists a $J \geq 1$ such that

$$2^{J-1} \leq |\omega| < 2^J.$$

We can then split the infinite product into two parts, namely

$$\prod_{k=1}^{\infty} 2^{-1/2} |R(e^{j\omega/2^k})| = \prod_{k=1}^{J} 2^{-1/2} |R(e^{j\omega/2^k})| \cdot \prod_{k=1}^{\infty} 2^{-1/2} |R(e^{j\omega/2^{k+J}})|.$$

Because $|\omega| < 2^J$, we can bound the second product by $e^\alpha$ according to (8.114). The first product has $J$ terms and can be bounded by $2^{-J/2} \cdot B^J$. Using $B < 2^{N-1/2}$, we can upper bound the first product by $(2^{N-1-\epsilon})^J$. This leads to

$$\sup_{2^{J-1} \leq |\omega| < 2^J} |\gamma(\omega)| \leq c^* 2^{J(N-1-\epsilon)} \leq c''(1 + |\omega|)^{N-1-\epsilon},$$

where we used the fact that $\omega$ is between $2^{J-1}$ and $2^J$. Thus, the growth of $|\gamma(\omega)|$ is sufficiently slow and therefore $\Phi(\omega)$ decays faster than $1/|\omega|$, proving continuity of $\varphi(t)$. 
8.B  Biorthogonal Wavelets from Filter Banks

As we saw already with filter banks, not all cases of interest are necessarily orthogonal. In Chapter 6, we designed biorthogonal filter banks to obtain symmetric/antisymmetric finite impulse response (FIR) filters. As can be guessed, a similar situation exists for wavelets. Except for the Haar case, there are no orthonormal and compactly-supported wavelets that are symmetric or antisymmetric. Since symmetry is often a desirable feature, we need to relax orthonormality. Thus, we seek two functions, a wavelet \( \psi(t) \) and its dual \( \tilde{\psi}(t) \), such that the families

\[
\psi_{m,n}(t) = \frac{1}{2^{m/2}} \psi\left( \frac{t-2^m n}{2^m} \right)
\]

(8.115)

\[
\tilde{\psi}_{m,n}(t) = \frac{1}{2^{m/2}} \tilde{\psi}\left( \frac{t-2^m n}{2^m} \right)
\]

(8.116)

form a biorthogonal set

\[
\langle \tilde{\psi}_{m,n}, \psi_{k,\ell} \rangle = \delta_{m-k} \delta_{n-\ell}
\]

and are complete in \( L^2(\mathbb{R}) \). That is, any \( f \in L^2(\mathbb{R}) \) can be written as

\[
f = \sum_{m \in \mathbb{Z}} \sum_{n \in \mathbb{Z}} \beta^{(m)}_n \psi_{m,n} \quad \text{with} \quad \beta^{(m)}_n = \langle \tilde{\psi}_{m,n}, f \rangle
\]

or

\[
f = \sum_{m \in \mathbb{Z}} \sum_{n \in \mathbb{Z}} \tilde{\beta}^{(m)}_n \tilde{\psi}_{m,n} \quad \text{with} \quad \tilde{\beta}^{(m)}_n = \langle \psi_{m,n}, f \rangle.
\]

Among the various approaches to design such biorthogonal bases, we choose the one based on iterated biorthogonal filter banks. To start, we assume a quadruple \((h, g, \tilde{h}, \tilde{g})\) of biorthogonal sequences satisfying the four biorthogonality relations (6.60a)–(6.60d). We further require that both lowpass filters have at least one zero at \( \omega = \pi \), or

\[
G(z) = \left( \frac{1+z^{-1}}{2} \right)^N R(z)
\]

(8.117)

\[
\tilde{G}(z) = \left( \frac{1+z^{-1}}{2} \right)^{\tilde{N}} \tilde{R}(z).
\]

(8.118)

By construction, the highpass filters \( H(z) \) and \( \tilde{H}(z) \) will have \( \tilde{N} \) and \( N \) zeros at \( \omega = 0 \), respectively. In the biorthogonal case, unlike the orthonormal case, there is no implicit normalization, so we will assume that

\[
G(1) = \tilde{G}(1) = \sqrt{2},
\]

which can be enforced by normalizing \( H(z) \) and \( \tilde{H}(z) \) accordingly.
The iterated filter $G^{(J)}(z)$ is given by

$$G^{(J)}(z) = \prod_{i=0}^{J-1} G(z^{2^i}),$$

and the analogous definition holds for $\tilde{G}^{(J)}(z)$. So, we can define scaling functions in the Fourier domain, as in the orthogonal case, as

$$\Phi(\omega) = \prod_{i=0}^{\infty} 2^{-1/2} G(e^{j\omega/2^i}),$$

(8.119)

$$\tilde{\Phi}(\omega) = \prod_{i=0}^{\infty} 2^{-1/2} \tilde{G}(e^{j\omega/2^i}).$$

(8.120)

In the sequel, we will concentrate on well-behaved cases, 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 $\tilde{\varphi}(t)$, which are related to (8.119)–(8.120) by Fourier transform.

The two-scale relation follows directly from the infinite product (8.119):

$$\Phi(\omega) = \sqrt{\frac{1}{2}} G(e^{j\omega/2}) \Phi(\omega/2),$$

(8.121)

or in time domain

$$\varphi(t) = \sqrt{2} \sum_{n \in \mathbb{Z}} g_n \varphi(2t - n).$$

(8.122)

In terms of spaces, let

$$V_0 = \text{span} \{ \{\varphi(t-n)\}_{n \in \mathbb{Z}} \}.$$

We have also the usual scaled spaces

$$V_m = \text{span} \left( \{2^{-m/2} \varphi(2^{-m}t - n)\}_{n \in \mathbb{Z}} \right), \quad m \in \mathbb{Z}.$$ 

(8.123)

For a given $\varphi(t)$—e.g., the hat function—one can verify that the axioms of multiresolution analysis are verified (Exercise 8.10). Similarly, we define $\tilde{\varphi}(t)$ as in (8.121)–(8.123) but using $\tilde{g}$ instead of $g$,

$$\tilde{\varphi}(t) = \sqrt{2} \sum_{n \in \mathbb{Z}} \tilde{g}_n \tilde{\varphi}(2t - n),$$

(8.124)

as well as $\tilde{V}_0$ and its scaled versions $\tilde{V}_m$. The basic biorthogonality relation between the scaling function $\varphi(t)$ and its dual,

$$\langle \tilde{\varphi}(t-n), \varphi(t) \rangle_t = \delta_n,$$

(8.125)

can be verified using the two-scale equation of the iterated graphical functions $\varphi^{(i)}(t)$ and $\tilde{\varphi}^{(i)}(t)$ (analogous to (8.26)) and using induction. Define the wavelet $\psi(t)$ and
8.B. Biorthogonal Wavelets from Filter Banks

its dual \( \tilde{\psi}(t) \) as

\[
\psi(t) = \sqrt{2} \sum_{n \in \mathbb{Z}} h_n \varphi(2t - n), \\
\tilde{\psi}(t) = \sqrt{2} \sum_{n \in \mathbb{Z}} \tilde{h}_n \tilde{\varphi}(2t - n).
\] (8.126) (8.127)

From there, define the wavelet families given in (8.115)–(8.116), which then lead to the wavelet spaces \( W_m \) and \( \tilde{W}_m \). All this seems very natural, but the geometry is more complicated than in the orthogonal case. On the one hand, we have the decompositions

\[
V_m = V_{m+1} \oplus W_{m+1},  \quad (8.128) \\
\tilde{V}_m = \tilde{V}_{m+1} \oplus \tilde{W}_{m+1},  \quad (8.129)
\]

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, \( V_m \) is not orthogonal to \( W_m \). Instead,

\[ \tilde{W}_m \perp V_m \quad \text{and} \quad W_m \perp \tilde{V}_m, \]

similarly to a biorthogonal basis (see Figure 6.11). These relationship are explored further in Exercise 8.11, where it is shown that

\[ \tilde{W}_m \perp W_k, \quad m \neq k. \]

In sum, we have two multiresolution analyses based on \( \varphi(t) \) and \( \tilde{\varphi}(t) \), respectively:

\[ \cdots \subset V_2 \subset V_1 \subset V_0 \subset V_{-1} \subset V_{-2} \subset \cdots \quad \text{with detail spaces} \quad \{W_m\}_{m \in \mathbb{Z}} \]

and

\[ \cdots \subset \tilde{V}_2 \subset \tilde{V}_1 \subset \tilde{V}_0 \subset \tilde{V}_{-1} \subset \tilde{V}_{-2} \subset \cdots \quad \text{with detail spaces} \quad \{\tilde{W}_m\}_{m \in \mathbb{Z}}. \]

The detail spaces allow us to write

\[ L^2(\mathbb{R}) = \bigoplus_{m \in \mathbb{Z}} W_m = \bigoplus_{m \in \mathbb{Z}} \tilde{W}_m. \]

The diagram in Figure 8.31 indicates these two splits, and the biorthogonality between them.

**Example 8.9 (Biorthogonal Multiresolution Analysis Based on Linear B-spline).**

Consider the lowpass filter

\[ G(z) = \frac{1}{2\sqrt{2}}(z + 2 + z^{-1}), \]

which has a double zero at \( z = -1 \) and satisfies the normalization \( G(1) = \sqrt{2} \). Then let

\[ \Phi(\omega) = \prod_{i=0}^{\infty} 2^{-i/2} G(e^{j\omega/2^i}), \]
Figure 8.31: The space $L^2(\mathbb{R})$ is split according to two different ladder of spaces. (a) Ladder of spaces $V_m$ based on the scaling function $\varphi(t)$. (b) Ladder of spaces $\tilde{V}_m$ based on the dual scaling function $\tilde{\varphi}(t)$. Note that orthogonality is “across” the spaces and their duals, e.g. $W_m \perp V_m$.

which is the Fourier transform of

$$\varphi(t) = \begin{cases} 1 - |t|, & |t| \leq 1; \\ 0, & \text{otherwise}, \end{cases}$$

that is, the hat function or linear B-spline. To verify this, notice that $G(z)$ is (up to a normalization and shift) the convolution of the Haar filter with 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. Thus, we have

$$V_0 = \text{span} \{ \varphi(t-n) \}_{n \in \mathbb{Z}}$$

as the space of $L^2$ functions which are continuous and piecewise linear over integer intervals. Instead of orthogonalizing the basis given by $\varphi(t)$ and its shifts (which was done in Example 8.7), we search for a biorthogonal scaling function $\tilde{\varphi}(t)$. We can do this by finding a biorthogonal lowpass filter $\tilde{G}(z)$. Note that this filter is not unique, as we see below. First, pick the smallest length biorthogonal lowpass satisfying the quadrature formula

$$\tilde{G}(z^{-1})G(z) + \tilde{G}(-z^{-1})G(-z) = 2.$$ (8.130)

It is easy to verify that, besides the trivial solution $\tilde{G}(z) = 1$, the following filter is a solution:

$$\tilde{G}(z) = \frac{1}{4\sqrt{2}}(1+z)(1+z^{-1})(-z + 4 - z^{-1})$$

$$= \frac{1}{4\sqrt{2}}(-z^2 + 2z + 6 + 2z^{-1} - z^{-2}).$$

This solution can be obtained from a different factorization of $A(z)$ in Example 6.4. The resulting dual scaling function

$$\tilde{\Phi}(\omega) = \prod_{i=1}^{\infty} 2^{-1/2} \tilde{G}(e^{i\omega/2^i})$$
Figure 8.32: The hat 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 $\frac{1}{4\sqrt{2}}[-1, 2, 6, 2, -1]$.

This turns out to be quite “irregular” in time (see Figure 8.32). If instead we look for a “better” $\tilde{G}(z)$, that is, a biorthogonal lowpass with respect to $G(z)$, but with more zeros at $\omega = \pi$, we can obtain a smoother dual scaling function. As an example, choose

$$\tilde{G}(z) = \frac{1}{64\sqrt{2}}(1 + z)^2(1 + z^{-1})^2(3z^2 - 18z + 38 - 18z^{-1} + 3z^{-2}).$$

Again, it is easy to check that it satisfies the quadrature formula (8.130). This filter can be obtained from a factorization of an autocorrelation with 6 zeros at $\omega = \pi$.

The other filters follow from the usual relationships between biorthogonal filters:

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

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

where we use only a minimal shift, since the lowpass filters are centered around the origin and symmetric. The wavelet and its dual follow from (8.126)–(8.127), and the quadruple $(\varphi(t), \psi(t), \tilde{\varphi}(t), \tilde{\psi}(t))$ are shown in Figure 8.33.

As can be seen, the scaling function $\varphi(t)$ and wavelet $\psi(t)$ are piecewise linear. In particular, the wavelet is piecewise linear over half-integer intervals, since it provides the details to go from $V_0$ to $V_{-1}$ (see (8.128)), that is, from integer intervals to half-integer intervals.
Figure 8.33: Biorthogonal linear spline basis. (a) The linear $B$-spline or hat function $\varphi(t)$. (b) The linear $B$-spline wavelet $\psi(t)$. (c) The dual scaling function $\tilde{\varphi}(t)$. (d) The dual wavelet $\tilde{\psi}(t)$.

Chapter at a Glance

Historical Remarks

TBD

Further Reading

TBD

Exercises with Solutions

8.1. TBD.


<table>
<thead>
<tr>
<th></th>
<th>Lowpass &amp; scaling function</th>
<th>Highpass &amp; wavelet</th>
</tr>
</thead>
<tbody>
<tr>
<td>Filter</td>
<td>$G(z) = \left(\frac{1+z^{-1}}{2}\right)^N \cdot R(z)$</td>
<td>$H(z) = z^{-L+1} \left(\frac{1+z^{-1}}{2}\right)^N \cdot R(-z^{-1})$</td>
</tr>
<tr>
<td>Function</td>
<td>$\Phi(\omega) = \prod_{i=1}^{\infty} \frac{1}{\sqrt{2}} G(e^{j\omega/2^i})$</td>
<td>$\Psi(\omega) = \frac{1}{\sqrt{2}} H(e^{j\omega/2^i}) \prod_{i=1}^{\infty} \frac{1}{\sqrt{2}} G(e^{j\omega/2^i})$</td>
</tr>
<tr>
<td>Two-scale equation</td>
<td>$\Phi(\omega) = 2^{-1/2} G(e^{j\omega/2}) \Phi(\omega/2)$</td>
<td>$\Psi(\omega) = 2^{-1/2} H(e^{j\omega/2}) \Phi(\omega/2)$</td>
</tr>
<tr>
<td>Orthogonality</td>
<td>$\langle \varphi(t), \varphi(t-n) \rangle_t = \delta_n$</td>
<td>$\langle \psi(t), \psi(t-n) \rangle_t = \delta_n$</td>
</tr>
<tr>
<td>Smoothness</td>
<td>Can be tested, increases with $N$</td>
<td>Same as for $\varphi(t)$</td>
</tr>
<tr>
<td>Moments</td>
<td>Polynomials of degree $N-1$ are in span${\varphi(t-n)}_{n \in \mathbb{Z}}$</td>
<td>Wavelets has $N$ zero moments</td>
</tr>
<tr>
<td>Size and support</td>
<td>support$(g) = {0, \ldots, L-1}$ support$(\varphi) = [0, L-1]$</td>
<td>support$(h) = {0, \ldots, L-1}$ support$(\psi) = [0, L-1]$</td>
</tr>
</tbody>
</table>

Table 8.1: 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 $\omega = \pi$.

### Exercises

8.1. **Fourier Domain Iteration of Haar Filter:**
Consider the Fourier domain iteration of the stretched Haar filter

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

(i) Verify that

$$\Phi(\omega) = e^{-j3\omega/2} \sin(3\omega/2) \frac{3\omega/2}{3\omega/2}$$

(ii) Verify that each finite iteration is of norm 1, while the limit is not, showing failure of $L^2$ convergence.

8.2. **Multiscale Equation:**
Based on the two scale equation (8.25), verify the expressions (8.46) and (8.48)

$$\Phi(\omega) = 2^{-k/2} G^{(k)}(e^{j\omega/2^k}) \Phi(\omega/2^k),$$

$$\Psi(\omega) = 2^{-k/2} H^{(k)}(e^{j\omega/2^k}) \Phi(\omega/2^k),$$

as well as their time domain equivalents given in (8.47) and (8.49).

8.3. **Scaling Behavior of Wavelet Coefficients around Singularities:**
In Proposition 8.5, it is stated that wavelet coefficients close to singularities of order $k$
behave as
\[ \beta_n^{(m)} \sim 2^m(k-1/2) \]
for \( m \to -\infty \). This was shown for \( k = 0 \) and 1 in the text. Extend the method used to prove the case \( k = 1 \) to include larger \( k \)'s and thus prove (8.62) in general.

8.4. Best Least Squares Approximation in the Haar Case:
Consider a function \( f_{-1}(t) \) in \( V_{-1} \), the space of functions constant over half integer intervals. Show that the best least squares approximation \( f_0(t) \) in \( V_0 \), the space of functions constant over integer intervals, is given by the average over two successive intervals, see also (8.72).

8.5. Two-Scale Equation for Piecewise Linear Spaces:
In Example 8.7, we saw various bases for piecewise linear spaces. From the fact that \( \varphi(t) \) satisfies a two scale equation, derive the two scale equation for the orthonormal scaling function \( \varphi(t) \).

(i) Give the two scale equation for \( \theta(t) \).
(ii) From the expression for \( \Phi(\omega) \) in (8.90) and the two scale equation for \( \Theta(\omega) \), derive the two scale equation for \( \Phi(\omega) \).
(iii) Derive the expression for \( G(e^{j\omega}) \), the Fourier transform of the sequence of coefficients of the two scale equation.

Note: This can be done for either the causal case, \( \varphi_c(t) \), or the symmetric case, \( \varphi_s(t) \).

8.6. Wavelets for Piecewise Linear Spaces: Given the coefficients of the two scale equation for the orthonormal scaling function \( \varphi(t) \) (see Exercise 8.5), derive an expression for the wavelet, based on the Fourier expression
\[ \Psi(\omega) = -\frac{1}{\sqrt{2}} e^{-j\omega/2} G^* \left( e^{j(\omega/2+\pi)} \right) \cdot \Phi(\omega/2) \]
where \( G(e^{j\omega}) \) is the discrete Fourier transform of the coefficients of the two scale equation for \( \varphi(t) \).

8.7. Sinc Multiresolution Analysis: Consider the sequence of spaces of bandlimited functions
\[ V_m = BL \left( [-2^{-m+1}\pi, 2^{-m}\pi] \right) \cup [-2^{-m}\pi, 2^{-m+1}\pi] \]
with an orthonormal basis for \( V_0 \) given by
\[ \varphi(t) = \frac{\sin(\pi t)}{\pi t} \]
and its integer shifts.

(i) Verify that the axioms of multiresolution analysis in Definition 8.4 are satisfied.
(ii) Given the embedding \( V_0 \subset V_{-1} \), derive the two-scale equations coefficients \( g_n \) in
\[ \varphi(t) = \sqrt{2} \sum_{n \in \mathbb{Z}} g_n \varphi(2t - n). \]
(iii) Derive the wavelet based on the highpass filter\(^{54}\) and give its expressions in both time and frequency domains.
(iv) Verify that the wavelet spaces are
\[ W_m = BL \left( [-2^{-m+1}\pi, -2^{-m}\pi] \cup [2^{-m}\pi, 2^{-m+1}\pi] \right). \]

\(^{54}\)This differs from (8.91) in that we skip the shift by \( L \), since here we have a two-sided infinite filter impulse response.
8.8. Meyer Scaling Function and Wavelet: In Example 8.8, we derived one of the simplest Meyer wavelets, based on a continuous \( \Phi(\omega) \). We generalize this to smoother \( \Phi(\omega) \)'s. For this purpose, introduce a helper function \( a(x) \) that is 0 for \( x \leq 0 \) and 1 for \( x \geq 1 \), and satisfies

\[
0 \leq x \leq 1 \quad \Rightarrow \quad a(x) + a(1-x) = 1
\]

An example of such a function is

\[
a(x) = \begin{cases} 
0 & x \leq 0 \\
3x^2 - 2x^3 & 0 \leq x \leq 1 \\
1 & x \geq 1 
\end{cases}
\]

(P8.8-2)

Construct the scaling function \( \Phi(\omega) \) as

\[
\Phi(\omega) = \sqrt{\alpha(2 - 3|\omega|/2\pi)}
\]

(P8.8-3)

(i) Verify that \( a(x) \) in (P8.8-2) satisfies (P8.8-1) and that it has a continuous first derivative.

(ii) Verify that \( \Phi(\omega) \) given by (P8.8-3) satisfies

\[
\sum_{k \in \mathbb{Z}} |\Phi(\omega + 2\pi k)|^2 = 1
\]

and thus, that \( \{\phi(t-n)\}_{n \in \mathbb{Z}} \) is an orthonormal set. Hint: start by using \( a(t) \) given in (P8.8-2), and then a general \( a(t) \) as in (P8.8-1).

(iii) With \( V_0 = \text{span}(\{\phi(t-n)\}_{n \in \mathbb{Z}}) \) and \( V_m \) defined the usual way, prove that

\[ V_0 \subset V_{-1} \]

(iv) Show that there exists a 2\( \pi \)-periodic function \( G(\text{e}^{j\omega}) \) such that

\[
\Phi(\omega) = \frac{1}{\sqrt{2}} G(\text{e}^{j\omega/2}) \Phi(\omega/2)
\]

and that

\[
G(\text{e}^{j\omega}) = \sqrt{2} \sum_{k \in \mathbb{Z}} \Phi(2\omega + 4\pi k)
\]

(P8.8-5)

(v) Verify (8.82) by showing that

\[
\langle f, \phi_{m,n} \rangle = 0, \quad m, n \in \mathbb{Z}
\]

implies necessarily that \( f = 0 \).

(vi) Verify (8.83) by showing that if

\[
f \in \bigcap_{m \in \mathbb{Z}} V_m
\]

then necessarily \( f = 0 \).

(vii) From (P8.8-5) and the usual construction of the wavelet, give an expression for \( \Psi(\omega) \) in terms of \( \Phi(\omega) \).

(viii) For \( a(x) \) given in (P8.8-2), what decay is expected for \( \phi(t) \) and \( \psi(t) \)?

8.9. Initialization of Mallat’s Algorithm:

Create an approximation problem for a smooth function (for example, bounded with bounded derivative) and compare rate of decay for Haar and piecewise linear approximation. Details later.

8.10. Biorthogonal Multiresolution Analysis:

Consider the hat function

\[
\phi(t) = \begin{cases} 
1 - |t| & |t| \leq 1 \\
0 & \text{else}
\end{cases}
\]

and the family \( \{\phi(t-n)\}_{n \in \mathbb{Z}} \).
(i) Characterize \( V_0 = \text{span}(\{\varphi(t-n)\}_{n \in \mathbb{Z}}) \)

(ii) Evaluate the autocorrelation sequence

\[ a_n = \langle \varphi(t), \varphi(t-n) \rangle \]

verifying that \( \varphi(t) \) is not orthogonal to its integer translates.

(iii) Define the usual scaled versions of \( V_0, V_m \). Verify that the axioms of multiresolution analysis, or (8.66 - 8.70), are verified.

8.11. Geometry of Biorthogonal Multiresolution Analysis:
Consider the biorthogonal family \( \{\varphi(t), \tilde{\varphi}(t), \psi(t), \tilde{\psi}(t)\} \) as defined in (8.122), (8.124), (8.126) and (8.127), as well as the associated multiresolution spaces \( \{V_m, \tilde{V}_m, W_m, \tilde{W}_m\} \).

(i) Verify that

\[ V_m = V_{m+1} \oplus W_{m+1} \]

and similarly for \( \tilde{V}_m \).

(ii) Show that \( V_m \) and \( W_m \) are not orthogonal to each other.

(iii) Verify the orthogonality relations

\[ \tilde{W}_m \perp V_m \]

and

\[ W_m \perp \tilde{V}_m \]

(iv) Show further that

\[ \tilde{W}_m \perp W_{m+k} \quad k \neq 0 \]

Hint: Show this first for \( k = 1, 2, \ldots \) using part 1.
Chapter 9

Localized Fourier Series on Sequences and Functions

Contents

<table>
<thead>
<tr>
<th>Section</th>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>9.1</td>
<td>Introduction</td>
<td>343</td>
</tr>
<tr>
<td>9.2</td>
<td>N-channel filter banks</td>
<td>344</td>
</tr>
<tr>
<td>9.3</td>
<td>Fourier-based N-channel filter banks</td>
<td>350</td>
</tr>
<tr>
<td>9.4</td>
<td>Cosine Modulated N-channel Filter Banks</td>
<td>356</td>
</tr>
<tr>
<td>9.5</td>
<td>Examples</td>
<td>363</td>
</tr>
<tr>
<td>9.6</td>
<td>What have we accomplished in this chapter?</td>
<td>364</td>
</tr>
<tr>
<td></td>
<td>Chapter at a Glance</td>
<td>364</td>
</tr>
<tr>
<td></td>
<td>Historical Remarks</td>
<td>364</td>
</tr>
<tr>
<td></td>
<td>Further Reading</td>
<td>364</td>
</tr>
<tr>
<td></td>
<td>Exercises with Solutions</td>
<td>364</td>
</tr>
<tr>
<td></td>
<td>Exercises</td>
<td>365</td>
</tr>
</tbody>
</table>

TBD: Opening text.

9.1 Introduction

Think of a piece of music. Notes appear at different instants of time, and then fade away. These are short-time frequency events, that the human ear picks out easily but are a challenge for a computer to understand.

The intuition is clear: the notes are well identified frequencies, but they are short lived. Thus, we would like to have a local Fourier transform, that is, a time-frequency analysis tool that understands the spectrum locally in time. Such a transform has many names, like short-time Fourier transform (STFT) or Gabor transform.\(^{55}\) The local energy distribution over frequency, which can be obtained by squaring the magnitude of the STFT coefficients, is called the spectrogram. It is widely used in speech processing and time series analysis.

\(^{55}\)The name honors Dennis Gabor, who studied a localized version of the Fourier transform in the 1940's.
The purpose of this chapter is to understand what is possible in terms of obtaining a localized version of the Fourier transform of a sequence. There is good and bad news on this topic. Let us give away an unfortunate fact right at the start: There is no good local Fourier basis! Now, on to the good news. If we relax the basis constraint and allow for a redundant system or frame, then there exist good local Fourier transforms. If we replace the complex exponential modulation of the Fourier transform by a cosine modulation instead, then we can obtain good orthonormal bases, called local cosine transform. Thus, as can be seen from this short overview, the situation for local frequency analysis is subtle: some desirable features are impossible, but relaxing some of the constraints allows very interesting constructions.

Chapter Outline

This leads naturally to the following structure of the chapter. In Section 9.2, we first review the two obvious constructions for \( N \)-channel filter banks, namely ideal filters on the one hand, and block transforms on the other hand. We then briefly overview general results on \( N \)-channel filter banks, based on the polyphase representation.

In Section 9.3, we consider the local Fourier transform in more detail. Block transform and sliding window cases are analyzed. For both, limitations are pointed out. We then provide the analysis of such schemes using the polyphase representation. This permits us to easily prove a discrete-time version of the Balian-Low theorem, which states the impossibility of good local orthonormal Fourier bases. As alternatives, redundant local Fourier systems are described, including the spectrogram.

Section 9.4 considers what happens if we use cosine modulation to obtain a local frequency analysis. In the block transform case, the discrete cosine transform (DCT) plays an eminent role. 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. Variations on this construction are also discussed.

Finally, Section 9.5 discusses a number of examples of local Fourier transforms and cosine modulated filter banks.

9.2 \( N \)-channel filter banks

A filter bank is a set of filters where each “picks out” some spectral region. A uniform division of the spectrum is desirable, so divide the (periodic) spectrum from 0 to \( 2\pi \) into \( N \) pieces of width \( 2\pi/N \). This can be achieved with ideal filters

\[
H_k(e^{j\omega}) = \begin{cases} \sqrt{N}, & \frac{2\pi k}{N} \leq \omega < \frac{2\pi (k+1)}{N} \\ 0, & \text{otherwise} \end{cases},
\]

(9.1)
9.2. N-channel filter banks

where the factor $\sqrt{N}$ is a normalization such that the filter has unit $\ell^2$ norm. By inverting the DTFT, their impulse responses are given by

\[
    h_{k,n} = \frac{1}{2\pi} \int_{-\pi}^{\pi} \frac{\sqrt{Ne^{j\omega n}}}{\sqrt{N}} e^{j2\pi(k+1/2)n/N} \sin(\pi n/N) \pi n.
\]

These are modulated sinc filters, with infinite impulse responses having slow, order $1/n$, decay. It can be verified that such filters and their shifts by $N$ form an orthonormal basis for $\ell^2(\mathbb{Z})$ (see Exercise 9.1). However, the slow decay of the impulse responses makes such filters impractical.

Alternatively, we can build the following orthonormal basis. Split the sequence into blocks of length $N$, and take a discrete Fourier transform of each block. Since the transform is orthonormal on each block and the blocks cover the entire sequence, we have an orthonormal basis for $\ell^2(\mathbb{Z})$. In terms of a matrix view, such a transform corresponds to a block diagonal matrix

\[
    \begin{bmatrix}
        \ddots & T \\
        T & \ddots & T \\
        & \ddots & \ddots & \ddots
    \end{bmatrix}
\]

where $T$ is an elementary, $N$ by $N$ block. In terms of basis vectors, we can rephrase this construction in the following way. Consider $N$ filters $g_k$, $k = 0, 1, \ldots, N-1$, with impulse responses

\[
    g_{k,n} = \begin{cases} 
        \frac{1}{\sqrt{N}} e^{j\pi nk/N}, & n = 0, 1, \ldots, N-1 \\
        0, & \text{otherwise.}
    \end{cases}
\]

Then, the set of impulse responses and their shifts by multiples of $N$,

\[
    \{g_{k,n-mN}\}_{k=0,\ldots,N-1, m\in\mathbb{Z}}
\]

form an orthonormal basis. Unfortunately, such basis functions, which are in some sense the $N$-channel extension of the two-channel Haar filters, have a poor frequency resolution. To see this, consider $g_0$; the other $g_k$s are modulated versions and therefore have the same frequency resolution.

\[
    G_0(e^{j\omega}) = \frac{1}{\sqrt{N}} \sum_{n=0}^{N-1} e^{-j\omega n} = \frac{1}{\sqrt{N}} \frac{1-e^{-j\omega N}}{1-e^{-j\omega}}
    = \frac{e^{-j\omega(N-1)/2}}{\sqrt{N}} \frac{\sin(\omega N/2)}{\sin(\omega/2)}.
\]
Chapter 9. Localized Fourier Series on Sequences and Functions

Figure 9.1: Time and frequency domain behavior of two orthonormal bases with $N = 8$ channels. (a) Ideal bandpass filters, with a sinc impulse response in time. (b) Ideal frequency response of (a). (c) Block transform, or rectangle window. (d) Frequency response of (c), with periodized sinc behavior.

This Fourier transform has zeros at $\omega = 2\pi k/N$, $k = 1, 2, \ldots, N - 1$, but decays slowly in between. Figure 9.1 shows time and frequency behaviors of the two constructions just shown.

Now, the question is: Are there any constructions “in between” these two extreme cases? Specifically, are there filters 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 9.2, where the input is analyzed by $N$ filters $h_k, k = 0, 1, \ldots, N - 1$, the output being subsampled by $N$. The synthesis is done by upsampling by $N$, followed by interpolation with $g_k, k = 0, 1, \ldots, N - 1$.

The analysis of $N$-channel filter banks can be done in complete analogy to the two-channel case, simply using the relevant equations for sampling rate changes by $N$. Rather than redoing the entire analysis, which is not particularly difficult but somewhat tedious, we concentrate on the polyphase representation. The polyphase transform of order $N$ splits signals and filter impulse responses into subsequences according to the index taken modulo $N$. For the sake of simplicity, we use $N = 3$; the general case following easily.

Consider a synthesis filter with impulse response $g_{k,n}$ and $z$-transform $G_k(z)$. This can be written as

$$G_k(z) = G_k^{(0)}(z^3) + z^{-1}G_k^{(1)}(z^3) + z^{-2}G_k^{(2)}(z^3),$$

where $G_k^{(i)}(z)$ is the $i$th polyphase component of the $(k$th) filter, or the $z$-transform of

$$g_k^{(i)} = g_{k,3n+i}$$

(taken as a sequence over $n$). Given some input $Y_k(z)$ to an upsampler by 3 followed
by interpolation by $G_k(z)$, we can write the output as

$$\hat{X}(z) = \begin{pmatrix} 1 & z^{-1} & z^{-2} \end{pmatrix} \begin{bmatrix} G_k^{(0)}(z^3) \\ G_k^{(1)}(z^3) \\ G_k^{(2)}(z^3) \end{bmatrix} \cdot Y_k(z^3),$$

and similarly, in a 3-channel synthesis bank, with 3 inputs driving 3 filters, the output is given by

$$\hat{X}(z) = \begin{pmatrix} 1 & z^{-1} & z^{-2} \end{pmatrix} \begin{bmatrix} G_0^{(0)}(z^3) & G_1^{(0)}(z^3) & G_2^{(0)}(z^3) \\ G_0^{(1)}(z^3) & G_1^{(1)}(z^3) & G_2^{(1)}(z^3) \\ G_0^{(2)}(z^3) & G_1^{(2)}(z^3) & G_2^{(2)}(z^3) \end{bmatrix} \begin{bmatrix} Y_0(z^3) \\ Y_1(z^3) \\ Y_2(z^3) \end{bmatrix}.$$

The matrix above is the polyphase matrix $G_p(z)$ of the filter bank, with the $(k, \ell)$ entry given by the $k$th polyphase component of the $\ell$th filter. Figure 9.3 shows the above equation in a flow diagram form, together with the related analysis bank.

Let us now write the polyphase decomposition of the analysis filter bank. While very similar to what we just did, there is a twist: the polyphase components of the analysis filter bank are defined in reverse order.\footnote{Note that using different polyphase decompositions in the analysis and synthesis is merely a convention, albeit a very convenient one that leads to the most compact formulation. We did the same thing in earlier chapters.}
Consider the input $x_n$ with $z$-transform $X(z)$, decomposed into polyphase components:

$$X(z) = X^{(0)}(z^3) + z^{-1}X^{(1)}(z^3) + z^{-2}X^{(2)}(z^3),$$

or

$$x^{(k)}_n = x_{3n+k}.$$

When filtering $X(z)$ with $H_k(z)$, followed by subsampling by 3, only the components with zero phase are kept. This is easiest to see if $H_k(z)$ is decomposed with the polyphase components defined in reverse order,

$$H_k(z) = H^{(0)}_k(z^3) + zH^{(1)}_k(z^3) + z^2H^{(2)}_k(z^3),$$

or

$$h^{(i)}_{k,n} = h_{k,3n-i}.$$

Then

$$(X(z) \cdot H_k(z)) \downarrow 3 = X^{(0)}(z)H^{(0)}_k(z) + X^{(1)}(z)H^{(1)}_k(z) + X^{(2)}(z)H^{(2)}_k(z),$$

where $\downarrow 3$ denotes subsampling by 3. Writing this for each filter $H_k(z), k = 0, 1, 2,$ we can express the channel signals as

$$\begin{bmatrix} Y_0(z) \\ Y_1(z) \\ Y_2(z) \end{bmatrix} = \begin{bmatrix} H^{(0)}_0(z) & H^{(1)}_0(z) & H^{(2)}_0(z) \\ H^{(0)}_1(z) & H^{(1)}_1(z) & H^{(2)}_1(z) \\ H^{(0)}_2(z) & H^{(1)}_2(z) & H^{(2)}_2(z) \end{bmatrix} \begin{bmatrix} X^{(0)}(z) \\ X^{(1)}(z) \\ X^{(2)}(z) \end{bmatrix},$$
Figure 9.4: Analysis-synthesis filter bank in polyphase domain. The product of the polyphase matrices $G_p(z)H_p(z)$ is between a forward and an inverse polyphase transform.

where the matrix above is the polyphase matrix $H_p(z)$ of the analysis bank. The $(k, l)$ entry is $l$th polyphase component of the $k$th filter (note the implicit transpose with respect to the synthesis polyphase matrix).

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:

$$
\hat{X}(z) = \begin{pmatrix}
1 & z^{-1} & z^{-2}
\end{pmatrix} G_p(z^3) H_p(z^3) \begin{bmatrix} X^{(0)}(z^3) \\
X^{(1)}(z^3) \\
X^{(2)}(z^3)
\end{bmatrix}.
$$

This formulation allows us to characterize classes of solutions. We state these results without detailed proofs since they follow easily from the equivalent two-channel filter bank results. Figure 9.4 sets the stage, since it depicts the cascade of an analysis and a synthesis filter bank. First, if

$$G_p(z)H_p(z) = I,$$

then the system reduces to a polyphase transform followed by its inverse, and we clearly have perfect reconstruction. Now consider special cases of polyphase matrices. As usual, we concentrate on the synthesis matrix $G_p(z)$, since it defines the basis of vectors of the expansion.

If the set of filters forms an orthonormal basis, then it can be verified (similarly to the two-channel case) that $G_p(z)$ is a paraunitary matrix,

$$G_p(z)G_p^T(z^{-1}) = G_p^T(z^{-1})G_p(z) = I,$$

or it is unitary on the unit circle,

$$G_p(e^{j\omega})G_p^*(e^{j\omega}) = G_p^*(e^{j\omega})G_p(e^{j\omega}) = I.$$

Perfect reconstruction is obvious with

$$H_p(z) = G_p^T(z^{-1}).$$
Because of the definition of $H_p(z)$ (with reversed order of polyphase components and a transposition), it can be checked that the analysis and synthesis filters are related by time reversal

$$H_i(z) = G_i(z^{-1}).$$

Now, consider an FIR synthesis filter bank, or $G_p(z)$ is a matrix of polynomials in $z^{-1}$. If

$$\det[G_p(z)] = \alpha z^{-k},$$

that is, $G_p(z)$ is unimodular,\(^{57}\) then there is an FIR analysis filter bank defined by

$$H_p(z) = \frac{z^k}{\alpha} \cdot \text{adjugate } [G_p(z)].$$

The result is clear, since $H_p(z)$ is the inverse of $G_p(z)$, and up to a factor $z^k$, is a polynomial matrix as well. It can be verified that the unimodularity condition is also necessary. We summarize the above results in the following theorem.

**Theorem 9.1.** Given is an analysis-synthesis filter bank with $N$ channels subsampled by $N$, and the polyphase matrices $H_p(z)$ and $G_p(z)$. Then

(i) The system is perfect reconstruction if and only if

$$G_p(z) \cdot H_p(z) = z^{-k} \cdot I.$$

(ii) The system performs an orthonormal expansion if and only if $G_p(z)$ is paraunitary, i.e.,

$$G_p(z)G_p^T(z^{-1}) = I.$$

(iii) The system is an FIR perfect reconstruction system if and only if $G_p(z)$ is unimodular, i.e.,

$$\det[G_p(z)] = \alpha z^{-k}.$$

As noted previously, the proofs follow from the two-channel case and can be found in the literature on filter banks. This concludes our general overview of $N$-channel filter banks.

### 9.3 Fourier-based $N$-channel filter banks

At the start of the previous section, we considered two extreme cases of $N$-channel filter banks, namely one based on ideal bandpass filters and the other based on block transforms. In addition, both performed a uniform split of the frequency range using $N$ filters obtained from a single prototype through modulation. This is obvious for $g_k$ in (9.3), where the prototype is $g_{0,n} = [1, 1, \ldots, 1]$ and

$$g_{k,n} = g_{0,n} \cdot e^{j2\pi nk/N} = W_N^{-nk}, \quad n = 0, 1, \ldots, N - 1, \quad (9.6)$$

---

\(^{57}\)The usual definition of unimodular requires $k = 0$, or the determinant is a constant. We relax this definition, since it allows a delay between input and output of the analysis-synthesis system.
9.3. Fourier-based \(N\)-channel filter banks

with \(W_N = e^{-j2\pi/N}\) as usual (and the \(N\) is dropped when it is clear from context), or

\[
G_k(z) = G_0(W_N^k z).
\] (9.7)

This can be verified for the ideal filter in (9.2) as well (see Exercise 9.1).

In general, consider a real prototype filter \(g_0\) with Fourier transform \(G_0(e^{j\omega})\) centered around the origin. Furthermore, assume bandwidth \(2\pi/N\), that is, the main lobe of \(|G_0(e^{j\omega})|\) is in the interval \(-\pi/N\) to \(\pi/N\). Since \(g_0\) is real, \(G_0(e^{j\omega})\) is Hermitian symmetric, \(G_0(e^{-j\omega}) = (G_0(e^{j\omega}))^*\). Now, if the \(k\)th filter is chosen according to (9.7), we obtain on the unit circle

\[
G_0(W^k z)|_{z = e^{j\omega}} = G_0(e^{-j2\pi k/N} e^{j\omega}) = G_0(e^{j(\omega-2\pi k/N)}),
\]

or a filter centered around \(\omega = 2\pi k/N\). Therefore, we define the Fourier-modulated filter bank as an \(N\)-channel filter bank with filters

\[
G_0(z), \ G_0(W z), \ G_0(W^2 z), \ldots, \ G_0(W^{N-1} z).
\]

Such a filter bank is desirable since it performs a “local” Fourier analysis with a window given by the prototype filter \(G_0(z)\). It would be nice to construct “good” orthonormal bases from such modulated filter banks. Unfortunately, this cannot be done, due to a negative result from harmonic analysis known as the Balian-Low theorem. Here, we will prove a weaker version of this result, which still excludes any good Fourier-modulated filter banks with critical sampling and finite impulse response filters.

**Theorem 9.2 (Balian-Low theorem for FIR filter banks).** There does not exist an \(N\)-channel Fourier-modulated basis for \(\ell^2(\mathbb{Z})\) having finite-length vectors, except for the trivial case of a prototype filter having \(N\) nonzero taps.

**Proof.** The way to prove this is to show the inexistence of an FIR filter bank solution, since there is a one-to-one relationship between perfect reconstruction filter banks and bases. To do this, we analyze the structure of the polyphase matrix of a Fourier-modulated filter bank. Given the polyphase representation of the prototype filter,

\[
G_0(z) = G_0^{(0)}(z^N) + z^{-1}G_0^{(1)}(z^N) + z^{-2}G_0^{(2)}(z^N) + \cdots + z^{-N+1}G_0^{(N-1)}(z^N),
\]

the modulated version becomes

\[
G_k(z) = G_0(W^k z)
= G_0^{(0)}(z^N) + W^{-k}z^{-1}G_0^{(1)}(z^N) + \cdots + W^{-(N-1)k}z^{-N+1}G_0^{(N-1)}(z^N).
\]

To make matters specific, we again pick \(N = 3\) and write the corresponding polyphase matrix,

\[
G_p(z) = \begin{pmatrix}
G_0^{(0)}(z) & G_0^{(0)}(z) & G_0^{(0)}(z) \\
G_0^{(1)}(z) & W^{-1}G_0^{(1)}(z) & W^{-2}G_0^{(1)}(z) \\
G_0^{(2)}(z) & W^{-2}G_0^{(2)}(z) & W^{-4}G_0^{(2)}(z)
\end{pmatrix}.
\]
which can be written in factorized form as

\[
G_p(z) = \begin{bmatrix}
G_0^{(0)}(z) \\
G_0^{(1)}(z) \\
G_0^{(2)}(z)
\end{bmatrix}
\begin{bmatrix}
1 & 1 & 1 \\
1 & W^{-1} & W^{-2} \\
1 & W^{-2} & W^{-4}
\end{bmatrix}.
\tag{9.8}
\]

The matrix on the right is recognized as a Fourier matrix conjugated, or \(F^*\), and the one on the left is diagonal with the polyphase components of the prototype filter as entries. According to Theorem 9.1, the requirement for perfect FIR reconstruction is that the determinant of \(G_p(z)\) is a monomial. Here,

\[
\det[G_p(z)] = \prod_{l=0}^{N-1} G_0^{(l)}(z) \cdot \det[F^*],
\]

which is a monomial if and only if each polyphase component is; in other words, each polyphase component of \(G_0(z)\) has exactly one nonzero term, or \(G_0(z)\) has \(N\) nonzero taps (one from each polyphase component). \(\Box\)

Unfortunately, this means there are no good FIR solutions for Fourier-modulated filter banks (\(N\)-tap filters, like the one in the block transform case in (9.5), have poor frequency resolution). On a more positive note, the factorization in (9.8), rewritten here in general as,

\[
G_p(z) = \text{diag}[G_0^{(0)}(z), G_0^{(1)}(z), \ldots, G_0^{(N-1)}(z)] \cdot F^*,
\]

leads straightforwardly to a fast algorithm for the implementation of modulated filter banks. The above formula was derived for the synthesis filter bank. A similar derivation gives the following factorization for modulated analysis filter banks, again for \(N = 3\) for clarity:

\[
H_p(z) = \begin{bmatrix}
1 & 1 & 1 \\
1 & W & W^2 \\
1 & W^2 & W^4
\end{bmatrix}
\begin{bmatrix}
H_0^{(0)}(z) \\
H_0^{(1)}(z) \\
H_0^{(2)}(z)
\end{bmatrix},
\]

where \(H_k(z) = H_0(W^k z)\). In general,

\[
H_p(z) = F \cdot \text{diag}[H_0^{(0)}(z), H_0^{(1)}(z), \ldots, H_0^{(N-1)}(z)],
\]

that is, the product of a Fourier matrix and a diagonal matrix of polyphase components. This leads to the following fast algorithm for modulated filter banks.

**Initialization**

Decompose the filter into its \(N\) polyphase components \(g_{0,n}^{(k)} = g_{0,nN+k}\).
9.3. Fourier-based $N$-channel filter banks

Given:

- a prototype filter $G_0(z)$ from which an $N$-channel modulated filter bank with upsampling by $N$ is constructed.
- $N$ channel signals $Y_0(z), Y_1(z), \ldots, Y_{N-1}(z)$

Given:

- A prototype filter $H_0(z)$ from which an $N$-channel modulated analysis bank with subsampling by $N$ is constructed.
- An input signal $X(z)$ to be analyzed.

Computation

1. **Fourier transform**: Transform the channel signals with an inverse Fourier transform:

\[
\begin{bmatrix}
y'_0,n \\
y'_1,n \\
\vdots \\
y'_{N-1,n}
\end{bmatrix} =
\begin{bmatrix}
1 & 1 & \cdots & 1 \\
1 & W^{-1} & \cdots & W^{-(N+1)} \\
1 & W^{-2} & \cdots & \\
1 & W^{-N+1} & \cdots & W^{-(N-1)^2}
\end{bmatrix} \cdot
\begin{bmatrix}
y_0,n \\
y_1,n \\
\vdots \\
y_{N-1,n}
\end{bmatrix}.
\]

2. **Convolution**: Convolve each transformed signal $y'_k,n$ with the $k$th polyphase component of $g_0$, denoted $g^{(k)}_0,n$:

\[
\begin{bmatrix}
y''_0,n \\
y''_1,n \\
\vdots \\
y''_{N-1,n}
\end{bmatrix} =
\begin{bmatrix}
g^{(0)}_0,n \\
g^{(1)}_0,n \\
\vdots \\
g^{(N-1)}_0,n
\end{bmatrix} \ast
\begin{bmatrix}
y'_0,n \\
y'_1,n \\
\vdots \\
y'_{N-1,n}
\end{bmatrix}.
\]

3. **Inverse polyphase transform**: Upsample the channel signals and interleave them to obtain the output

\[
x_{nN+k} = y''_{k,n}.
\]

The algorithm is schematically shown in Figure 9.5, together with the equivalent analysis filter bank, described next. This algorithm is the dual of the previous one.

Initialization

Decompose the analysis filter into its polyphase components

\[
h^{(k)}_{0,n} = h_{0,nN-k}.
\]
Figure 9.5: Fourier modulated filter bank with 4 channels. (a) Analysis with $H_i(z) = H_0(W_kz)$. (b) Synthesis with $G_i(z) = G_0(W_kz)$.

Computation

1. **Polyphase transform**: from the input sequence derive the $N$ polyphase components

   $$x^{(k)}_n = x_{nN+k}.$$

2. **Convolution**: convolve the polyphase components

   $$\begin{bmatrix}
   y_{0,n}^{(0)} \\
   y_{1,n}^{(1)} \\
   \vdots \\
   y_{N-1,n}^{(N-1)}
   \end{bmatrix} = 
   \begin{bmatrix}
   h_{0,n}^{(0)} \\
   h_{0,n}^{(1)} \\
   \vdots \\
   h_{0,n}^{(N-1)}
   \end{bmatrix} \ast 
   \begin{bmatrix}
   x_{0,n}^{(0)} \\
   x_{1,n}^{(1)} \\
   \vdots \\
   x_{N-1,n}^{(N-1)}
   \end{bmatrix}.$$  

3. **Fourier transform**: derive the channel signals by taking a forward Fourier transform

   $$\begin{bmatrix}
   y_{0,n} \\
   y_{1,n} \\
   \vdots \\
   y_{N-1,n}
   \end{bmatrix} = 
   \begin{bmatrix}
   1 & 1 & \cdots & 1 \\
   1 & W & \cdots & W^{N-1} \\
   1 & W^2 & \cdots & W^{2(N-1)} \\
   1 & W^{N-1} & \cdots & W^{(N-1)^2}
   \end{bmatrix} \cdot 
   \begin{bmatrix}
   y_{0,n}^{(0)} \\
   y_{1,n}^{(1)} \\
   \vdots \\
   y_{N-1,n}^{(N-1)}
   \end{bmatrix}.$$  

For a schematic illustration, see Figure 9.5.
9.3. Fourier-based $N$-channel filter banks

**Complexity**

We consider the complexity of the analysis. The synthesis, being dual, has the same complexity. Consider a prototype filter of length $L = KN$, such that each polyphase component is of length $K$.

First, we need to compute $N$ convolutions, but on polyphase components of the input signal, that is, at $N$ times slower sampling rate. Therefore, this is equivalent to a single convolution at full sampling rate, or of order $K$ operations per input sample.

Then, a fast Fourier transform (FFT) is computed, again at the subsampled rate. Assuming an order $N \log N$ computational load for the FFT, this requires $\log N$ operations per input sample. In total, we have

$$C_{MFB} \sim K + \log N \text{ operations per sample.} \quad (9.9)$$

This is very efficient, since simply taking a size-$N$ FFT for each consecutive block of $N$ samples would already require $\log N$ operations per input sample. Thus, the price for the “window” given by the prototype filter is $K$ operations per input sample, or the length of the window divided by $N$. So what are typical values for $K$? This depends on the frequency selectivity that is desired, but a typical value can be of order $\log N$.

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 where it can be unitary (like in the block transform case where it is the identity), it is highly dependent on the window. See Exercise 9.2 for an exploration of this issue.

**Non-critically subsampled Fourier modulated filter banks**

Because of the difficulty of finding good modulated bases as exemplified in Theorem 9.2, one often relaxes the requirement of critical sampling in cases where redundancy can be tolerated. Such is the case in signal analysis, for example in speech processing, where the spectrogram is a widely used tool. Because of redundancy, we are not talking about bases anymore, but rather about frames. These are studied in more detail in Chapter 10, so for now, we just take a glimpse at such redundant representations, mostly because they are the prevailing utilization of Fourier modulated filter banks.

For simplicity, we will assume a modulated filter bank with $2N$ channels. Instead of critical sampling, we subsample only by $N$, or a redundancy factor of 2. The easiest way to think about such a filter bank is to consider two versions of the input, namely $x_n$ and $x_{n-N}$, each feeding into a $2N$-channel modulated filter bank subsampled by $2N$.

Denoting the desired outputs of the redundant filter bank by $y_{i,n}$ (where $i$ is the channel number and $n$ the time index), one can verify that $y_{i,2n}$ comes from the filter bank with $x_n$ as input, while $y_{i,2n+1}$ comes from the one with $x_{n+N}$ as input (see Exercise 9.3). It then follows also that the computational complexity is doubled per input sample.
In general, for a modulated filter bank with $MN$ channels and subsampling by $N$ (or a redundancy of $M$), we have the following computational complexity (assuming a prototype filter of length $L = K \cdot MN$):

$$C_{\text{MFB},M} \sim M(K + \log MN) \text{ operations per sample.} \quad (9.10)$$

In the above formula, we recognize the contribution of the size-$MN$ FFT algorithm, the length of the polyphase filter $K$, and the redundancy $M$.

It should be noted that oversampling helps conditioning. For example, the Balian-Low theorem holds at critical sampling only. For oversampled systems, FIR solutions that lead to tight frames do exist (see Exercise 9.4).

### 9.4 Cosine Modulated $N$-channel Filter Banks

#### 9.4.1 Uniform modulation

One possible escape from the Balian-Low theorem is to replace complex modulation (multiplication by $e^{j2\pi k/N}$) with an appropriate cosine modulation. This has the added advantage that all filters are now real if the prototype window is real. To see what is the appropriate cosine modulation, we compare in Figure 9.6 a complex modulation with a cosine modulation for $N = 6$. In general, we have the following bandwidth and modulation for the $k$th filter in the Fourier case:

- Bandwidth: $\frac{2\pi}{N}$
- Modulation: $e^{j2\pi nk/N}$; $k = 0, 1, \ldots, N - 1$; $n \in \mathbb{Z}$,

while for the cosine case, we have

- Bandwidth: $\frac{\pi}{N}$
- Modulation: $\cos \left( \frac{n(2k+1)\pi}{2N} \right)$; $k = 0, 1, \ldots, N - 1$; $n \in \mathbb{Z}$.

The difference for the cosine case appears because the modulated filters 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 $\pi/N$.

With this, we indicated necessary conditions for a uniformly modulated cosine filter bank. But can we obtain an orthonormal basis? One solution is as usual an ideal lowpass prototype, with support $[-\pi/2N, \pi/2N]$. But as we know, this leads to a sinc-like basis with infinite and slowly-decaying impulse responses. Another solution is a block transform, with filters of length $N$, that is, too short for an interesting analysis.

#### 9.4.2 Modulated lapped orthogonal transforms

We develop a solution with filters of length $2N$, or just a “nearest-neighboring block” overlap; see Figure 9.7(a). Because of this overlapping structure, they are called lapped orthogonal transforms (LOT). Consider the $N$ filters of length $2N$ given by

$$g_{k,n} = \frac{1}{\sqrt{N}} \cos \left( \frac{n(2k+1)\pi}{2N} + \Theta_k \right), \quad k = 0, 1, \ldots, N - 1, \quad n = 0, 1, \ldots, 2N - 1. \quad (9.11)$$
9.4. Cosine Modulated \( N \)-channel Filter Banks

Figure 9.6: Complex versus cosine modulated filter bank for 6 channels. (a) In the Fourier case, the bandwidth is \( 2\pi/6 \), and the center frequencies \( 2\pi k/6, k = 0, \ldots, 5 \). Note that the prototype filter is identical with the first filter in the bank. (b) In the cosine case, the bandwidth of the prototype is \( 2\pi/12 \), and the center frequencies for the cosine modulation are \( \pm (2k+1)\pi/12 \) \( k = 0, \ldots, 5 \). The prototype is centered at the origin, while all filters are modulated.

where \( \Theta_k \) is a phase factor to be specified, and the factor \( 1/\sqrt{N} \) ensures \( \|g_k\|_2 = 1 \).

We first verify that \( g_{k,n} \) and \( g_{\ell,n} \) are orthogonal for \( k \neq \ell \). Using Euler’s formula, we rewrite \( g_{k,n} \) as

\[
g_{k,n} = \frac{1}{2\sqrt{N}} \left( e^{j\frac{2\pi n(2k+1)}{4N}} \cdot e^{j\Theta_k} + e^{-j\frac{2\pi n(2k+1)}{4N}} \cdot e^{-j\Theta_k} \right)
\]

\[
= \frac{1}{2\sqrt{N}} \left( C_k W^{-n(2k+1)}_N + C^*_k W^n(2k+1)_N \right),
\]

where \( C_k = e^{j\Theta_k} \) and \( W_{4N} = e^{-j2\pi/4N} \). In proving orthogonality, we will make repeated use of the partial sum of geometric series:

\[
\sum_{n=0}^{N-1} \alpha^n = \frac{1 - \alpha^N}{1 - \alpha}.
\]

We the expansion above, we can express the inner product as

\[
\langle g_{k,n}, g_{\ell,n} \rangle = \frac{1}{4N} \sum_{n=0}^{2N-1} \left( C_k W^{-n(2k+1)}_N + C^*_k W^n(2k+1)_N \right) \left( C_\ell W^{-n(2\ell+1)}_N + C^*_\ell W^n(2\ell+1)_N \right).
\]

Of the four terms resulting from the product inside the sum, consider the first:

\[
\sum_{n=0}^{2N-1} C_k C_\ell W^{-n(2k+2\ell+2)}_N = C_k C_\ell \frac{1 - W^{-4N(k+\ell+1)}_N}{1 - W^{-4N(2k+2\ell+2)}_N} = 0,
\]
Figure 9.7: Lapped orthogonal transforms of size $N$. (a) The filters are of length $2N$, and thus, they overlap with their nearest neighbors. (b) Orthogonality of the tails. The left half is symmetric, the right half anti-symmetric. The product of the overlapping tails is antisymmetric, leading to the orthogonality of the tails.

since $W_{4N}^{1i} = 1$ for any integer $i$. The other three terms similarly are zero.

So the $N$ impulse responses are orthogonal to each other, but what about their shifts by $N$? Fortunately, because there is only nearest-neighbor overlap, it suffices to check

$$\langle g_{k,n}, g_{\ell,n-N} \rangle = 0, \quad k, \ell \in \{0, 1, \ldots, N-1\};$$

see Figure 9.7(a). Now comes the magic trick. Assume that all left tails (from $n = 0$ to $N - 1$) are symmetric around the midpoint $(N - 1)/2$, while all right tails (from $n = N$ to $2N - 1$) are antisymmetric around the midpoint $(3N - 1)/2$. Then, the product of the tails is antisymmetric, leading automatically to a zero inner product. This geometric intuition is shown in Figure 9.7(b). To obtain such symmetric/antisymmetric tails, one needs to choose the right phase factor $\Theta_k$. This turns out to be given by $\Theta_k = (2k + 1)(1 - N)\pi/4N$, or

$$g_{k,n} = \frac{1}{2\sqrt{N}} \cos \left( \frac{(2k + 1)(2n - N + 1)\pi}{4N} \right), \quad k = 0, 1, \ldots, N-1, \quad n = 0, 1, \ldots, 2N-1. \quad (9.12)$$

Let us verify symmetry of the left tail:

$$g_{k,N-n-1} = \frac{1}{2\sqrt{N}} \cos \left( \frac{(2k + 1)(N - 2n - 1)}{4N} \right) = g_{k,n}, \quad n = 0, \ldots, N/2 - 1,$$
9.4. Cosine Modulated \( N \)-channel Filter Banks

Figure 9.8: Lapped orthogonal transform for \( N = 8 \), with a rectangular window. (a) The 8 basis sequences. Note the symmetric and antisymmetric tails. (b) Fourier transform of the basis sequences, showing the uniform split of the spectrum.

since \( \cos(\alpha) = \cos(-\alpha) \). To see the antisymmetry of the right tail, start with

\[
g_{k,N+n} = \frac{1}{2\sqrt{N}} \cos \left( \frac{(2k+1)(N+2n+1)\pi}{4N} \right), \quad n = 0, \ldots, N/2 - 1.
\]

Then

\[
g_{k,2N-n-1} = \frac{1}{2\sqrt{N}} \cos \left( \frac{(2k+1)(4N-N-2n-1)}{4N} \right)
= \frac{1}{2\sqrt{N}} \cos \left( \frac{(2k+1)(-N-2n-1)}{4N} + \pi \right) = -g_{k,N+n},
\]

since \( \cos(\alpha + \pi) = -\cos(\alpha) \). We thus have proven that the set

\[
\{g_{k,n-\ell N}\}_{k=0,1,\ldots,N-1,\ell\in\mathbb{Z}}
\]

is an orthonormal set. Because the filter bank is perfect reconstruction, the set is an orthonormal basis for \( \ell^2(\mathbb{Z}) \). An example for \( N = 8 \) is given in Figure 9.8.

9.4.3 Block Matrix Notation

It is worthwhile to look at the filter impulse responses and their shifts in block matrix notation, where the columns of \( G_0 \) and \( G_1 \) are the right and left tails respectively, or

\[
\begin{bmatrix}
G_0 \\
G_1
\end{bmatrix}
= \begin{bmatrix}
g_{0,0} & g_{1,0} & \cdots & g_{N-1,0} \\
g_{0,1} & g_{1,1} & \cdots & g_{N-1,1} \\
\vdots & \vdots & \ddots & \vdots \\
g_{0,N-1} & g_{1,N-1} & \cdots & g_{N-1,N-1} \\
g_{0,N} & g_{1,N} & \cdots & g_{N-1,N} \\
g_{0,N+1} & g_{1,N+1} & \cdots & g_{N-1,N+1} \\
\vdots & \vdots & \ddots & \vdots \\
g_{0,2N-1} & g_{1,2N-1} & \cdots & g_{N-1,2N-1}
\end{bmatrix}.
\]
Similarly, a

The synthesis matrix is a block Toeplitz matrix with two diagonal blocks,

\[
G = \begin{bmatrix}
G_0 & G_0 & \cdots & G_0 \\
G_1 & G_0 & \cdots & G_1 \\
& G_1 & \cdots & \\
& & \ddots & \\
& & & G_1
\end{bmatrix},
\]

and \(G^T G = I\), or

\[
G_0^T G_0 + G_1^T G_1 = I, \tag{9.13}
\]

\[
G_0^T G_1 + G_1^T G_0 = 0. \tag{9.14}
\]

Because \(G\) is full rank, the rows form also an orthonormal set, and the product

commutes, or \(GG^T = I\). Therefore, we have also

\[
G_0 G_0^T + G_1 G_1^T = I, \tag{9.15}
\]

\[
G_1 G_0^T + G_0 G_1^T = 0. \tag{9.16}
\]

Following the symmetry/antisymmetry of the tails, the matrices \(G_0\) and \(G_1\) have repeated rows. For example, for \(N = 4\),

\[
G_0 = \begin{bmatrix}
90.0 & 91.0 & 92.0 & 93.0 \\
90.1 & 91.1 & 92.1 & 93.1 \\
90.1 & 91.1 & 92.1 & 93.1 \\
90.0 & 91.0 & 92.0 & 93.0
\end{bmatrix}
\quad \text{and} \quad
G_1 = \begin{bmatrix}
90.4 & 91.4 & 92.4 & 93.4 \\
90.5 & 91.5 & 92.5 & 93.5 \\
-90.5 & -91.5 & -92.5 & -93.5 \\
-90.4 & -91.4 & -92.4 & -93.4
\end{bmatrix}.
\]

Denoting by \(\tilde{G}_0\) and \(\tilde{G}_1\) the upper halves of \(G_0\) and \(G_1\), respectively, we can express \(G_0\) and \(G_1\) as

\[
G_0 = \begin{bmatrix} I_{N/2} \\ J_{N/2} \end{bmatrix} \tilde{G}_0 \quad \text{and} \quad
G_1 = \begin{bmatrix} I_{N/2} \\ -J_{N/2} \end{bmatrix} \tilde{G}_1,
\]

where \(I_{N/2}\) is the identity matrix of size \(N/2\) and \(J_{N/2}\) is the antidiagonal (or reflection) matrix of size \(N/2\). Note that premultiplying by \(J_N\) reverses the order of the rows, while postmultiplication reverses the order of the columns. Also, \(J_N^2 = I_N\).

From the above, both \(G_0\) and \(G_1\) have rank \(N/2\). It can be verified that the rows of \(\tilde{G}_0\) and \(\tilde{G}_1\) form an orthogonal set, with norm \(1/\sqrt{2}\). From the structure and orthogonality, it follows that

\[
G_0 G_0^T = \begin{bmatrix} I_{N/2} \\ J_{N/2} \end{bmatrix} \tilde{G}_0 \tilde{G}_0^T \begin{bmatrix} I_{N/2} \\ J_{N/2} \end{bmatrix} = \begin{bmatrix} I_{N/2} \\ J_{N/2} \end{bmatrix} \frac{1}{2} \begin{bmatrix} I_{N/2} \end{bmatrix} \begin{bmatrix} I_{N/2} \\ J_{N/2} \end{bmatrix} = \frac{1}{2} (I_N + J_N). \tag{9.17}
\]

Similarly,

\[
G_1 G_1^T = \frac{1}{2} (I_{N/2} - J_{N/2}). \tag{9.18}
\]
9.4. Cosine Modulated \( N \)-channel Filter Banks

9.4.4 Windowing

So far, we have \( N \) filters of length \( 2N \), but their impulse responses are simply rectangular-windowed cosine functions. Such a rectangular window is discontinuous at the boundary, and thus not desirable. Instead, a smooth tapering off, as shown in Figure 9.7(a), is more appropriate. So the question is: Can we window our previous solution and still retain orthogonality?

For this, we choose a power complementary real and symmetric window function, namely a sequence \( w_n \) such that

\[
 w_n = w_{2N-n-1}, \quad n = 0, 1, \ldots, N-1
\]

and

\[
 |w_n|^2 + |w_{N-n-1}|^2 = 2, \quad n = 0, 1, \ldots, N-1.
\]

Let

\[
 W_0 = \text{diag}[w_0, w_1, \ldots, w_{N-1}],
\]

\[
 W_1 = \text{diag}[w_N, w_{N+1}, \ldots, w_{2N-1}].
\]

Then, the above relations can be written as

\[
 W_1 = J_N W_0 J_N,
\]

and

\[
 W_0^2 + W_1^2 = 2I. \tag{9.19}
\]

The windowed filter impulse responses are

\[
 \begin{bmatrix} G'_0 \\ G'_1 \end{bmatrix} = \begin{bmatrix} W_0 & J_N W_0 J_N \\ J_N W_0 J_N & G_1 \end{bmatrix} \begin{bmatrix} G_0 \\ G_1 \end{bmatrix}.
\]

Orthogonality of the tails can be checked as follows. Start with

\[
 G'_1^T \cdot G'_0 = G'_1^T [I_{N/2} - J_{N/2}] J_N W_0 J_N W_0 \begin{bmatrix} I_{N/2} \\ J_{N/2} \end{bmatrix} \tilde{G}_0.
\]

Notice that the product \( S = J_N W_0 J_N W_0 \) is diagonal and symmetric (the \( k \)th entry is \( w_k w_{N-k} \)), which then leads to

\[
 [I_{N/2} - J_{N/2}] S \begin{bmatrix} I_{N/2} \\ J_{N/2} \end{bmatrix} = 0,
\]

proving orthogonality of the tails. To complete the orthogonality proof, we need to verify (see (9.15))

\[
 G'_0 G'_0^T + G'_1 G'_1^T = I.
\]

Rewrite the windowed version as

\[
 W'_0 G'_0 G'_0^T W'_0 + W'_1 G'_1 G'_1^T W'_1,
\]
Figure 9.9: Lapped orthogonal transform for $N = 8$, with a smooth, power complementary window. (a) The window function. (b) The basis sequences. (c) The Fourier transform of the basis sequences. Note the improved frequency resolution with respect to Figure 9.8(b).

Figure 9.10: Transition from a size-$N$ LOT to a size-$2N$ LOT. On the interval $[0, \ldots, N]$, the window from the size-$N$ LOT is used, while on the interval $[N, \ldots, 2N]$, the window from the size-$2N$ LOT is used.

and use (9.17)–(9.18) to get

$$\frac{1}{2} W_0(I_N + J_N)W_0 + \frac{1}{2} W_1(I_N - J_N)W_1.$$  

Using $W_1 = J_N W_0 J_N$, the terms involving $+J_N$ and $-J_N$ cancel, and we are left with

$$\frac{1}{2} W_0^2 + \frac{1}{2} W_1^2 = I_N,$$

where the last identity follows from the power complementary property (9.19). An example of a windowed cosine modulated filter bank is shown in Figure 9.9 for an 8-channel filter bank.

Let us finally discuss a variation on the theme of windows, both for its importance in practice and because it shows the same basic principles at work. Assume one wants to process a signal 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, this can be achieved by having power complementarity of the two adjacent windows of the transition. This is shown schematically in Figure 9.10. Calling $w_n^{(L)}$ and $w_n^{(R)}$ the two windows involved, then

$$|w_n^{(L)}|^2 + |w_n^{(R)}|^2 = 2$$

leads again to orthogonality of the overlapping tails of the two filter banks.
9.5. Examples

9.5.1 Spectrograms

- Stationary and ergodic time-series.
- Equivalence of power spectrum and Fourier transform of the autocorrelation function

\[ r_n = E[x_k x_{k+n}] \]

\[ R(e^{j\omega}) = \sum_{n=-\infty}^{\infty} e^{-j\omega n} r_n \]

\[ |X(e^{j\omega})|^2 = |R(e^{j\omega})|^2 \]

- Idea: “local” power spectrum.
- Method: compute a “local” Fourier transform and take magnitude squared, this is the spectrogram.

- Key parameters:
  - number of channels
  - length of window
  - subsampling factor

- Key result: smoothing is necessary to reduce variance.
- Put an example, like AR process.
- Put an example of a “non-stationary” signal like speech or music, and plot the time-varying spectrogram for various choices of parameters.

9.5.2 Filter banks for audio coding

- Critical bands for hearing.
- Masking effects.
- Cosine modulated filter banks.
- Bit allocations.
- Time segmentation.
- An example of audio coding: MP3.
9.5.3 Orthogonal frequency division multiplexing

- Duality of analysis-synthesis and synthesis-analysis.
- Transmultiplexing.
- Overcomplete systems.
- Critically sampled systems.
- OFDM
- An example: modulation for wireless local area networks, or 802.11.

9.6 What have we accomplished in this chapter?

- Theory of $N$-channel filter banks, with key results.
- Fourier modulated filter banks, relation to local Fourier analysis, negative result on “good” local orthonormal Fourier bases.
- Cosine modulated filter banks.
- Good local orthonormal bases.
- Examples of applications.

Chapter at a Glance

TBD

Historical Remarks

TBD

Further Reading

TBD

Exercises with Solutions

9.1. TBD.
Exercises

9.1. NAME MISSING:
Consider the set of filters \( \{ h_k \}_{k=0,\ldots,N-1} \) given in equations (9.1-9.2).

(i) Prove that the impulse responses and their shifts by multiples of \( N \),
\[ \{h_{k,n-l}: k=0,\ldots,N-1, \quad l \in \mathbb{Z} \}, \]
form an orthonormal set.

(ii) Verify that all filters are modulates of the prototype filter \( h_0 \), following (9.6-9.7),
and this in both time and frequency domains.

9.2. Conditioning of Modulated Filter Banks:
Given is a modulated filter bank, subsampled by \( N \), with \( G_k(z) = G_0(W_k z) \), and polyphase components of \( G_0(z) \) given by \( \{ c_0^{(0)}(z), c_0^{(1)}(z), \ldots, c_0^{(N-1)}(z) \} \).

(i) Consider the simplest case, namely \( N = 2 \), and the two filters \( \{ G_0(z), G_1(z) = G_0(-z) \} \). Show that the conditioning of the filter bank is given by the conditioning of the matrix
\[
M(e^{j\omega}) = \begin{bmatrix}
|c_0^{(0)}(e^{j\omega})|^2 & 0 \\
0 & |c_0^{(1)}(e^{j\omega})|^2
\end{bmatrix}.
\]

By conditioning, we mean that we want to find bounds \( \alpha \) and \( \beta \) between the norms of the input signal \( x \) and its expansion in the filter bank domain, or
\[
\alpha \|x\|^2 \leq \sum_{i=0}^{N-1} \|y_i\|^2 \leq \beta \|x\|^2,
\]
where the \( y_i \)'s are the \( N \) channel signals.

Hint: take first a fixed frequency \( \omega_0 \) and find \( \alpha(\omega_0) \) and \( \beta(\omega_0) \). Then extend the argument to \( \omega \in [-\pi, \pi] \).

(ii) Compute the bounds \( \alpha \) and \( \beta \) for
\begin{itemize}
  \item[(i)] The Haar filter \( G_0(z) = \frac{1}{\sqrt{2}}(1 + z^{-1}) \)
  \item[(ii)] The ideal half band filter \( g_0, n = \frac{1}{\sqrt{2}} \frac{\sin(\pi/2 \cdot n)}{\pi/2 \cdot n} \).
  \item[(iii)] The 4 point average \( G_0(z) = \frac{1}{4}(1 + z^{-1} + z^{-2} + z^{-3}) \).
  \item[(iv)] The windowed average \( G_0(z) = \sqrt{\frac{1}{8}}(\frac{1}{2} + z^{-1} + z^{-2} + \frac{1}{2} z^{-3}) \).
\end{itemize}

(v) Extend the argument to general \( N \).

(vi) Compute numerically \( \alpha \) and \( \beta \) for
\begin{itemize}
  \item[(i)] \( N = 4 \) and the triangular window \( [1 \ 2 \ 3 \ 4 \ 4 \ 3 \ 2 \ 1] \).
  \item[(ii)] \( N = 3 \) and the triangular window \( [1 \ 2 \ 3 \ 4 \ 5 \ 4 \ 3 \ 2 \ 1] \).
\end{itemize}

9.3. Oversampled Modulated Filter Banks:
Consider Fourier modulated filter banks, but where the subsampling is not critical, resulting in a redundant representation. In particular, consider 2N channels, subsampling by \( N \) and filters of length \( L = K \cdot 2N \).

(i) Show that the outputs can be obtained from running two filter banks with subsampling by \( 2N \) on \( x_n \) and \( x_{n+N} \), respectively.

(ii) Estimate the computational complexity.

(iii) Generalize the above to \( MN \) channels, and verify (9.10).

9.4. NAME MISSING:
Consider a Fourier modulated filter bank with 4 channels and subsampled by 2.
(i) Write the polyphase matrix $G_p(z)$ as a function of the 2 polyphase components of the prototype filter $G(z) = G_0(z^2) + z^{-1}G_1(z^2)$.

(ii) Find conditions on $G_0(z)$ and $G_1(z)$ such that

$$G_p(z)G_p^T(z^{-1}) = 2 \cdot I,$$

that is, that the oversampled filter bank is a tight frame with frame bound equal to 2.
Chapter 10
Frames on Sequences

Contents

10.1 Introduction ........................................... 367
10.2 Frame Definitions and Properties ............... 372
10.3 Finite-Dimensional Frames ....................... 376
10.4 Infinite-Dimensional Frames Via Filter Banks . 379
10.5 Algorithms ............................................ 386
Chapter at a Glance ....................................... 387
Historical Remarks ....................................... 387
Further Reading ........................................... 388
Exercises with Solutions ............................... 390
Exercises ................................................... 392

This chapter is somewhat unusual in its scope. While most of the chapters in Part II deal either with “Fourier”-like 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.

Redundancy is a common tool in our daily lives, and the same idea of removing doubt is present in signal representations. Given a signal, we represent it in another system, typically a basis, where its characteristics are more readily apparent in the transform coefficients. However, these representations are typically nonredundant, and thus corruption or loss of transform coefficients can be serious. In comes redundancy; we build a safety net into our representation so that we can avoid those disasters. The redundant counterpart of a basis is called a frame, the topic of the present chapter.

10.1 Introduction

To start, we will concentrate on finite-dimensional frames only, as they are easier to present and grasp. We will try to convey the essential concepts through simple examples; more formal treatment will come in the next section.

367
General frames

What is the easiest way to add redundancy to a signal representation? Let us take an ONB, add a vector to it and see what happens. Suppose our system is as given in Fig. 10.1(a), with \( \Phi = \{ \varphi_1, \varphi_2, \varphi_3 \} \). The first two vectors \( \varphi_1, \varphi_2 \) are the ones forming the ONB and the third one \( \varphi_3 \) was added to the ONB. What can we say about such a system?

First, it is clear that by having three vectors in \( \mathbb{R}^2 \), those vectors must necessarily be linearly dependent; indeed, \( \varphi_3 = \varphi_1 - \varphi_2 \). It is also clear that these three vectors must be able to represent every vector in \( \mathbb{R}^2 \) since their subset is able to do so (which also means that we could have added any other vector \( \varphi_3 \) to our ONB with the same result.) In other words, since we know that the following is true:

\[
x = (\varphi_1, x)\varphi_1 + (\varphi_2, x)\varphi_2,
\]

nothing stops us from adding a zero to the above expression:

\[
x = (\varphi_1, x)\varphi_1 + (\varphi_2, x)\varphi_2 + ((\varphi_1, x) - (\varphi_1, x))\varphi_1 - \varphi_2).
\]

We now rearrange it slightly to read:

\[
x = (2\varphi_1, x)\varphi_1 + ((-\varphi_1 + \varphi_2), x)\varphi_2 + (-\varphi_1, x)(\varphi_1 - \varphi_2).
\]

We recognize here \( (-\varphi_1 + \varphi_2) \) as \( -\varphi_3 \), and the vectors inside the inner products we will call

\[
\tilde{\varphi}_1 = 2\varphi_1, \quad \tilde{\varphi}_2 = -\varphi_1 + \varphi_2, \quad \tilde{\varphi}_3 = -\varphi_1.
\]

With this notation, we can rewrite the expansion as

\[
x = (\tilde{\varphi}_1, x)\varphi_1 + (\tilde{\varphi}_2, x)\varphi_2 + (\tilde{\varphi}_3, x)\varphi_3 = \sum_{i=1}^{3}(\tilde{\varphi}_i, x)\varphi_i,
\]

or, if we introduce matrix notation as before:

\[
\Phi = \begin{bmatrix} 1 & 0 & 1 \\ 0 & 1 & -1 \end{bmatrix}, \quad \tilde{\Phi} = \begin{bmatrix} 2 & -1 & -1 \\ 0 & 1 & 0 \end{bmatrix}
\]
and
\[ x = \sum_{i=1}^{3} \langle \hat{\varphi}_i, x \rangle \varphi_i, = \Phi \hat{\Phi}^* x. \] (10.1)

The only difference between the above expression and the one for general bases is that matrices \( \Phi \) and \( \hat{\Phi} \) are now rectangular. Fig. 10.1 shows this example pictorially.

Therefore, we have shown that starting with an ONB and adding a vector, we obtained another expansion with 3 vectors. This expansion is reminiscent of the one for general biorthogonal bases we have seen earlier in Chapter 1, except that the vectors involved in the expansion are now linearly dependent. This redundant set of vectors \( \Phi = \{ \varphi_i \}_{i \in I} \) is called a frame while \( \hat{\Phi} = \{ \hat{\varphi}_i \}_{i \in I} \) is called the dual frame. As for biorthogonal bases, these two are interchangeable, and thus, \( x = \Phi \hat{\Phi}^* x = \hat{\Phi} \Phi^* x \).

While we have constructed a redundant representation, we would like to know how good/convenient/useful it is. What type of measure can we use to compare one frame to another? What comes to mind and what is often of concern to practitioners, is how stable such a representation is and how will it behave under perturbation. For example, ONBs are favored since their norm is preserved throughout the system and there are no problems when trying to reconstruct (invert). We could ask the same question here by asking ourselves what the energy of the transform coefficients is and whether we can put any bounds on that energy.

We can extract that information by looking at (10.1). The transform coefficients are given by \( \langle \hat{\varphi}_i, x \rangle \) and their energy is
\[ \sum_{i=1}^{3} |\langle \hat{\varphi}_i, x \rangle|^2 = x^* \hat{\Phi} \Phi^* x = x^* \begin{bmatrix} 6 & -1 \\ -1 & 1 \end{bmatrix} x. \] (10.2)

From matrix theory, we know that \( \lambda_{\min} I \leq \hat{\Phi} \Phi^* \leq \lambda_{\max} I \), where \( \lambda_{\min}, \lambda_{\max} \) are the smallest and largest eigenvalues of \( \hat{\Phi} \Phi^* \). Using this and computing \( \lambda_{\min} \approx 0.8, \lambda_{\max} \approx 6.2 \), we find that the energy of the transform coefficients is bounded from below and above as:
\[ 0.8 \|x\|^2 \leq \sum_{i=1}^{3} |\langle \hat{\varphi}_i, x \rangle|^2 \leq 6.2 \|x\|^2. \] (10.3)

The boundedness of the energy of the transform coefficients is closely related to the stability of reconstruction (we will make this precise in Section 10.2). The actual numbers bounding it are called frame bounds and the closer they are, the faster and numerically better behaved reconstruction we have. (From this, can you guess what the bounds would be on an ONB?)

**Tight frames**

Well, adding a vector worked but we ended up with an expansion that does not look very elegant. Is it possible to have frames which would somehow mimic ONBs? To do that, let us think for a moment what characterizes ONBs. It is not linear independence since that is true for biorthogonal bases as well. How about the following two facts:
Chapter 10. Frames on Sequences

Figure 10.2: Simplest unit-norm tight frame—Mercedes Benz frame (MB). This is also an example of a harmonic tight frame.

- ONBs are self dual, and
- ONBs preserve the norm?

What is the role of frame bounds in such a frame? We develop some intuition through an example.

Example 10.1 (Mercedes-Benz (MB) Frame). The Mercedes-Benz frame is arguably the most famous frame. It is a collection $\Phi$ of three vectors in $\mathbb{R}^2$, and is an excellent representative for many classes of frames. It is given by (see Fig. 10.2):

$$\Phi^* = \begin{bmatrix} 0 & 1 \\ -\sqrt{3}/2 & -1/2 \\ \sqrt{3}/2 & -1/2 \end{bmatrix} = \begin{bmatrix} \varphi_1^* \\ \varphi_2^* \\ \varphi_3^* \end{bmatrix}, \quad (10.4)$$

with the corresponding expansion:

$$x = \frac{2}{3} \sum_{i=1}^{3} \langle \varphi_i, x \rangle \varphi_i = \frac{2}{3} \Phi \Phi^*, \quad (10.5)$$

and the norm:

$$\|X\|^2 = \sum_{i=1}^{3} |\langle \varphi_i, x \rangle|^2 = \frac{3}{2} \|x\|^2. \quad (10.6)$$

Given (10.5), it is clear that the MB frame can represent any $x$ from $\mathbb{R}^2$ (real plane). Since the same set of vectors (within scaling) is used both for expansion and reconstruction (see (10.5)), $\Phi$ is self dual (again, within scaling). We can think of the expansion in (10.5) as a generalization of an ONB except that the vectors are
10.1. Introduction

not linearly independent anymore. The frame of this type is called a tight frame, and in particular, unit-norm tight, as all frame vectors are of norm 1. One can compare the expansion into an ONB with the expansion into the MB frame and see that the frame version has an extra scaling of 2/3 (see (10.5)). When the frame is tight and all the vectors have unit norm as in this case, the inverse of this scaling factor denotes the redundancy of the system: we have 3/2 or 50% more vectors than needed to represent any vector in \( \mathbb{R}^2 \).

This discussion took care of the first question, whether we can have a self-dual frame. To check the question about norms, we compute the sum of the squared transform coefficients as in (10.6). We see that, indeed, this frame preserves the norm, within the scaling factor of 3/2; this is fairly intuitive, as in the transform domain, where we have more coefficients than we started with, the energy is 3/2 times higher than in the original domain. This also answers our question about frame bounds. Comparing (10.6) to (10.3) we see that here, the bounds are identical and equal to 3/2. This will turn out to be a characteristic of all tight frames, ensuring stability and ease of reconstruction, similarly to ONBs.

Thus, the tight frame we constructed is very similar to an ONB, with a linearly dependent set of vectors. Actually, tight frames are redundant sets of vectors closest to ONBs (we will make this statement precise later).

One more interesting tidbit about this particular frame; note how all its vectors have the same norm. This is not necessary for tightness but if it is true, then the frame is called an equal-norm tight frame (ENTF). ■

Summary

To summarize what we have done until now, assume that we are dealing with a finite-dimensional space of dimension \( n \) and \( m > n \) linearly-dependent frame vectors:

- We represented our signal in another domain to more easily extract its salient characteristics. We did that in a redundant fashion.

- Given a pair of dual frames \((\Phi, \tilde{\Phi})\), the coordinates of our signal in the new domain (that is, with respect to the new frame) are given by

\[
X = \tilde{\Phi}^* x, \tag{10.7}
\]

where \( \tilde{\Phi} \) is a rectangular \( n \times m \) matrix describing the frame change and it contains the dual frame vectors as its columns, while \( X \) collects all the transform coefficients together. This is called the analysis or decomposition expression.

- The synthesis, or reconstruction is given by

\[
x = \Phi X, \tag{10.8}
\]

where \( \Phi \) is again a rectangular \( n \times m \) matrix, and it contains frame vectors as its columns.
If the expansion is into a tight frame, then 
\[ \tilde{\Phi} = \Phi, \quad \text{and} \quad \Phi \Phi^* = I_{n \times n}. \]

Note that, unlike for bases, \( \Phi^* \Phi \) is not identity (why?).

If the expansion is into a general frame, then 
\[ \Phi \tilde{\Phi}^* = I. \]

A note of caution: In frame theory, the frame change is usually denoted by \( \Phi \), not \( \tilde{\Phi}^* \). Given that \( \Phi \) and \( \tilde{\Phi} \) are interchangeable, we can use one or the other without risk of confusion.

Chapter Outline

We will explain how frames are obtained in Sections 10.1-10.4, Section 10.4 being the oversampled counterpart of filter banks (bases) from Chapters 6-7. Frames on functions are discussed in the next chapter.

10.2 Frame Definitions and Properties

In the last section, we introduced frames through examples and developed some intuition. We now discuss frames more generally and examine a few of their properties. We formally define frames as follows: A family \( \Phi = \{\phi_i\}_{i \in I} \) in a Hilbert space \( H \) is called a frame if there exist two constants \( 0 < A \leq B < \infty \), such that for all \( x \) in \( H \),

\[
A \|x\|^2 \leq \sum_{i \in I} |\langle \phi_i, x \rangle|^2 \leq B \|x\|^2. \tag{10.9}
\]

\( A, B \) are called frame bounds.

We hinted in Section 10.1 that the frame bounds are intimately related to the issues of stability. We make this more formal now. To have stable reconstruction, the operator mapping \( x \in l^2(\mathbb{Z}) \) into its transform coefficients \( \langle \phi_i, x \rangle \) has to be bounded, that is, \( \sum_{i \in I} |\langle \phi_i, x \rangle|^2 \) has to be finite, achieved by the bound from above. On the other hand, no \( x \) with \( \|x\| > 0 \) should be mapped to 0. In other words, \( \|x\|^2 \) and \( \sum_{i \in I} |\langle \phi_i, x \rangle|^2 \) should be close. This further means that if \( \sum_{i \in I} |\langle \phi_i, x \rangle|^2 < 1 \) there should exist an \( \alpha < \infty \) such that \( \|x\|^2 < \alpha \). Taking an arbitrary \( x \) and defining \( \hat{x} = (\sum_{i \in I} |\langle \phi_i, x \rangle|^2)^{-1/2} x \) ensures that \( \sum_{i \in I} |\langle \phi_i, \hat{x} \rangle|^2 < 1 \) and thus, \( \|\hat{x}\|^2 < \alpha \). This is equivalent to

\[
A \|x\|^2 \leq \sum_{i \in I} |\langle \phi_i, x \rangle|^2,
\]

for some \( A = 1/\alpha \). We can use the above to see that if \( x = x_1 - x_2 \), the norm of \( x \) cannot be arbitrarily large if \( \sum_{i \in I} |\langle \phi_i, x_1 \rangle - \langle \phi_i, x_2 \rangle|^2 \) is small, exactly what is needed for stable reconstruction. To summarize, the above discussion justifies why
10.2. Frame Definitions and Properties

Figure 10.3: Frames at a glance. ENF: Equal-norm frames TF: Tight frames. ENTF: Equal-norm tight frames. UNF: Unit-norm frames. PTF: Parseval tight frames. UNTF: Unit-norm tight frames. ENPTF: Equal-norm Parseval tight frames. ONB: Orthonormal bases.

A numerically stable reconstruction of \( x \) from its transform coefficients is possible only if (10.9) is satisfied. In a short while, when introducing a dual frame, we will see how the frame bounds influence the reconstruction.

We now define various classes of frames, depicted in Fig. 10.3. Their names, as well as alternate names under which they have been used in the literature, are given in Historical Remarks.

Tight frames (TF) are frames with equal frame bounds, that is, \( A = B \). Equal-norm frames (ENF) are those frames where all the elements have the same norm, \( \| \varphi_i \| = \| \varphi_j \| \), for \( i, j \in I \). Unit-norm frames (UNF) are those frames where all the elements have norm 1, \( \| \varphi_i \| = 1 \), for \( i \in I \). \( A \)-tight frames (\( A \)-TF) are tight frames with frame bound \( A \). Parseval tight frames (PTF) are tight frames with frame bound \( A = 1 \) and could also be denoted as 1-tight frames.

From (10.9), in a tight frame (that is, when \( A = B \)), we have

\[
\sum_{i \in I} |\langle \varphi_i, x \rangle|^2 = A \| x \|^2. \tag{10.10}
\]

By pulling \( 1/A \) into the sum, this is equivalent to:

\[
\sum_{i \in I} |\langle \frac{1}{\sqrt{A}} \varphi_i, x \rangle|^2 = \| x \|^2, \tag{10.11}
\]

that is, the family \( \Phi = \{ (1/\sqrt{A}) \varphi_i \}_{i \in I} \) is a 1-tight frame. In other words, any tight frame can be rescaled to be a tight frame with frame bound 1—a Parseval tight frame. With \( A = 1 \), the above looks similar to (1.46a), Parseval’s equality, thus the name Parseval tight frame.
In an A-tight frame, \( x \in H \) is expanded as follows:

\[
x = \frac{1}{A} \sum_{i \in I} \langle \varphi_i, x \rangle \varphi_i.
\]  

(10.12)

While this last equation resembles the expansion formula in the case of an ONB as in (1.41)-(1.42) (except for the factor \( 1/A \)), a frame does not constitute an orthonormal basis in general. In particular, vectors may be linearly dependent and thus not form a basis. If all the vectors in a tight frame have unit norm, then the constant \( A \) gives the redundancy ratio. For example, \( A = 2 \) means there are twice as many vectors than needed to cover the space. For the MB frame we discussed earlier, redundancy is \( 3/2 \), that is, we have \( 3/2 \) times more vectors than needed to represent vectors in a two-dimensional space. Note that if \( A = B = 1 \) (PTF), and \( \| \varphi_i \| = 1 \) for all \( i \) (UTF), then \( \Phi = \{ \varphi_i \}_{i \in I} \) is an ONB (see Fig. 10.3).

Because of the linear dependence which exists among the vectors used in the expansion, the expansion is not unique anymore. Consider \( \Phi = \{ \varphi_i \}_{i \in I} \) where \( \sum_{i \in I} \alpha_i \varphi_i = 0 \) (where not all \( \alpha_i \)'s are zero because of linear dependence). If \( x \) can be written as \( x = \sum_{i \in I} X_i \varphi_i \), then one can add \( \alpha_i \) to each \( X_i \) without changing the decomposition. The expansion (10.12) is unique in the sense that it minimizes the norm of the expansion among all valid expansions. Similarly, for general frames, there exists a unique canonical dual frame, which is discussed later in this section (in the tight frame case, the frame and its dual are equal).

### 10.2.1 Frame Operators

The analysis operator \( \Phi^* \) maps the Hilbert space \( H \) into \( \ell^2(I) \):

\[
X_i = (\Phi^* x)_i = \langle \varphi_i, x \rangle, \quad i \in I.
\]

As a matrix, the analysis operator \( \Phi^* \) has rows which are the Hermitian-transposed frame vectors \( \varphi_i^* \):

\[
\Phi^* = \begin{bmatrix}
\varphi_1^* & \cdots & \varphi_n^*
\varphi_2^* & \cdots & \varphi_n^*
\vdots & \ddots & \vdots
\varphi_m^* & \cdots & \varphi_m^*
\vdots & \vdots & \ddots
\end{bmatrix}.
\]

As an example, when \( H = \mathbb{R}^N, \mathbb{C}^N \), the above is an \( m \times n \) matrix. When \( H = \ell^2(\mathbb{Z}) \), it is an infinite matrix.

The frame operator, defined as \( S = \Phi \Phi^* \), plays an important role. (We have seen that when trying to compute the frame bounds earlier in this section.) The product \( G = \Phi^* \Phi \) is called the Grammian.

### 10.2.2 Useful Frame Facts

When manipulating frame expressions, the frame facts given below often come in handy. It is a useful exercise for you to try to derive some of these on your own.
10.2. Frame Definitions and Properties

Some of these we have encountered already, such as (10.14) when computing frame bounds in (10.2). In many of these expressions, we will be using the fact that for any matrix $\Phi^*$ with rows $\varphi_i^*$, $S = \Phi\Phi^* = \sum_{i\in I} \varphi_i^* \varphi_i$. If $S$ is a frame operator, then

$$Sx = \Phi\Phi^* x = \sum_{i\in I} \langle \varphi_i, x \rangle \varphi_i,$$

$$\langle Sx, x \rangle = \langle \Phi\Phi^* x, x \rangle = \langle \Phi^* x, \Phi^* x \rangle = \|\Phi^* x\|^2 = \sum_{i\in I} |\langle \varphi_i, x \rangle|^2,$$

(10.13)

$$\sum_{i\in I} \langle S\varphi_i, \varphi_i \rangle = \sum_{i\in I} \langle \Phi^* \varphi_i, \Phi^* \varphi_i \rangle = \sum_{i,j\in I} |\langle \varphi_i, \varphi_j \rangle|^2.$$

(10.14)

From (10.9), we have that

$$AI \leq S = \Phi\Phi^* \leq BI,$$

as well as

$$B^{-1}I \leq S^{-1} \leq A^{-1}I.$$

We say that two frames $\Phi$ and $\Psi$ for $H$ are equivalent if there is a bounded linear bijection $L$ on $H$ for which $L\varphi_i = \psi_i$ for $i \in I$. Two frames $\Phi$ and $\Psi$ are unitarily equivalent if $L$ can be chosen to be a unitary operator. Any $A$-TF is equivalent to a PTF as $\varphi_{\text{PTF}} = (1/\sqrt{A})\varphi_{A-\text{TF}}$. In other words, $\{S^{-1/2}\varphi_i\}_{i\in I}$ is a Parseval tight frame for any frame $\Phi$.

The nonzero eigenvalues $\{\lambda_i\}_{i\in I}$ of $S = \Phi\Phi^*$ and $G = \Phi^*\Phi$ are the same. Thus,

$$\text{tr}(\Phi\Phi^*) = \text{tr}(\Phi^*\Phi).$$

(10.16)

A matrix $\Phi^*$ of a TF has orthonormal columns. In finite dimensions, this is equivalent to the Naimark Theorem, which says that every TF is obtained by projecting an ONB from a larger space.

10.2.3 Dual Frame Operators

The canonical dual frame of $\Phi$ is a frame defined as $\check{\Phi} = \{\check{\varphi}_i\}_{i\in I} = \{S^{-1}\varphi_i\}_{i\in I}$, where

$$\check{\varphi}_i = S^{-1}\varphi_i, \quad i \in I.$$

(10.17)

Noting that $\check{\varphi}_i^* = \varphi_i^* S^{-1}$ and stacking $\check{\varphi}_1^*, \check{\varphi}_2^*, \ldots$, in a matrix, the analysis frame operator associated with $\check{\Phi}$ is

$$\check{\Phi}^* = \Phi^* S^{-1},$$

while its frame operator is $S^{-1}$, with $B^{-1}$ and $A^{-1}$ its frame bounds. Since

$$\Phi \check{\Phi}^* = \Phi \Phi^* S^{-1} = I,$$

then

$$x = \sum_{i\in I} \langle \check{\varphi}_i, x \rangle \varphi_i = \Phi \check{\Phi}^* x = \check{\Phi} \Phi^* x.$$
Continuing our earlier discussion on frame bounds and their influence on the
stability of reconstruction, we see from above that to reconstruct, one has to invert
the frame operator
\[ S = (\Phi \Phi^*) \text{ as (for a full proof, see [52])} \]
\[ S^{-1} = \frac{2}{A + B} \sum_{k=0}^{\infty} \left( I - \frac{2}{A + B} \Phi \Phi^* \right)^k. \]

Thus, when \( B/A \) is large, convergence is slow, while as \( B/A \) tends to 1, convergence
is faster. Specifically, when \( A = B = 1 \), we have an ONB and \( S^{-1} = I \).

### 10.3 Finite-Dimensional Frames

We now consider the finite-dimensional case, that is, when \( H = \mathbb{R}^N, \mathbb{C}^N \) and few of
the properties of such frames.

For example, for an ENTF with norm-\( a \) vectors, since \( S = \Phi \Phi^* = A I_{n \times n} \),
\[ tr(S) = \sum_{j=1}^{n} \lambda_j = nA, \quad (10.18) \]
where \( \lambda_j \) are the eigenvalues of \( S = \Phi \Phi^* \). On the other hand, because of (10.16)
\[ tr(S) = tr(G) = \sum_{i=1}^{m} \|\varphi_i\|^2 = ma^2. \quad (10.19) \]
Combining (10.18) and (10.19), we get
\[ A = \frac{m}{n} a^2. \quad (10.20) \]

Then, for a UNTF, that is, when \( a = 1 \), (10.20) yields the redundancy ratio:
\[ A = \frac{m}{n}. \]

Recall that for the MB frame, \( A = 3/2 \). These, and other trace identities for all
frame classes are given in Table 10.2.

An important result tells us that every Parseval tight frame can be realized
as a projection of an orthonormal basis from a larger space. It is known under
the name Naimark Theorem and has been rediscovered by several people in the
past decade, Han and Larson, for example [79]. Its finite-dimensional instantiation
specifies how all tight frames are obtained by projecting from a larger-dimensional
space. The same is true in general, that is, any frame can be obtained by project-
ing a biorthogonal basis from a larger space [79] (we are talking here about finite
dimensions only).

We call this process seeding and will say that a frame \( \Phi \) is obtained by seeding
from a basis \( \Psi \) by deleting a suitable set of columns of \( \Psi \) [126]. We denote this as
\[ \Phi^* = \Psi[J], \]
10.3. Finite-Dimensional Frames

where $J \subset \{1, \ldots, m\}$ is the index set of the retained columns.

We can now reinterpret the Parseval tight frame identity $\Phi \Phi^* = I$ from Chapter 1: It says that the columns of $\Phi^*$ are orthonormal. In view of the theorem, this makes sense as that frame was obtained by deleting columns from an ONB from a larger space.

Example 10.2 (MB Frame (cont’d)). Let us look again at the MB frame we discussed earlier. We can see how it is obtained from the following ONB:

$$
\Psi = \begin{bmatrix}
0 & \sqrt{2}/3 & 1/\sqrt{3} \\
-1/\sqrt{2} & -1/\sqrt{6} & 1/\sqrt{3} \\
1/\sqrt{2} & -1/\sqrt{6} & 1/\sqrt{3}
\end{bmatrix},
$$

(10.21)

using the following projection operator $P$:

$$
P = \frac{1}{\sqrt{3}} \begin{bmatrix}
2/3 & -1/3 & -1/3 \\
-1/3 & 2/3 & -1/3 \\
-1/3 & -1/3 & 2/3
\end{bmatrix},
$$

(10.22)

that is, the MB frame seen as a collection of vectors in the three-dimensional space is $\Phi_{3D} = P\Psi$. The projection operator essentially “deletes” the last column of $\Psi$ to create the frame operator $\Phi^*$. The MB frame obtained is in its PTF version (all the vectors are rescaled so that the frame bound $A = 1$).

As the ONBs have specific characteristics highly prized among bases, the same distinction belongs to tight frames among all frames. As such, they have been studied extensively, but only recently have Benedetto and Fickus [9] formally shown why tight frames and ONBs indeed belong together. They found that both the ONBs as well as TFs are solutions to the same minimization problem, just with the different parameters (nonredundant versus redundant cases).

10.3.1 Harmonic Tight Frames

As it turns out, the MB frame is a representative of a large class of well known tight frames called Harmonic tight frames (HTF). The HTFs are frame counterparts of the DFT, and in fact, they are obtained from the DFT by seeding.

Specifically, HTFs are obtained by seeding from $\Psi = DFT_m$ given in (2.108), by deleting the last $(m - n)$ columns:

$$
\varphi_i = \sqrt{\frac{m}{n}} (W^0_m, W^1_m, \ldots, W^{m(n-1)}_m),
$$

(10.23)

for $i = 0, \ldots, m - 1$. HTFs have a number of interesting properties: (a) For $m = n + 1$, all ENTFs are unitarily equivalent to it; in other words, since we have HTFs for all $n, m$, we have all ENTFs for $m = n + 1$. (b) It is the only ENPTF with frame bound 1 such that its elements are generated by a group of unitary operators with one generator. (c) HTFs are maximally robust (MR) to erasures (one can erase up to $m - n$ frame elements and the remaining representation will still be a frame).
10.3.2 Other Finite-Dimensional Frame Families

Variations on Harmonic Tight Frames

HTFs have been generalized in an exhaustive work by Vale and Waldron [156], where the authors look at frames with symmetries. Some of these they term HTFs (their definition is more general than what is given in (10.23)), and are the result of the operation of a unitary \( U \) on a finite Abelian group \( G \). When \( G \) is cyclic, the resulting frames are cyclic. In [31], the HTFs we showed above are with \( U = I \) and generalized HTFs are with \( U = D \) diagonal. These are cyclic in the parlance of [156]. An example of a cyclic frame are \((n + 1)\) vertices of a regular simplex in \( \mathbb{R}^n \). There exist HTFs which are not cyclic.

Grassmannian Packings and Equiangular Frames

Equiangular (referring to \(|\langle \varphi_i, \varphi_j \rangle| = \text{const.}\)), frame families have become popular recently due to their use in quantum computing, where rank-1 measurements are represented by positive operator valued measures (POVMs). Each rank-1 POVM is a tight frame.

The first family is symmetric informationally complete POVMs (SIC-POVMs) [129]. A SIC-POVM is a family \( \Phi \) of \( m = n^2 \) vectors in \( \mathbb{C}^n \) such that

\[
|\langle \varphi_i, \varphi_j \rangle|^2 = \frac{1}{n + 1}
\]  

holds for all \( i, j, i \neq j \).

The second family are mutually unbiased bases (MUBs). A MUB is a family \( \Phi \) of \( m = n(n + 1) \) vectors in \( \mathbb{C}^n \) such that

\[
|\langle \varphi_i, \varphi_j \rangle|^2 = \frac{1}{n},
\]  

or 0 (these are \((n + 1)\) ONBs).

Both harmonic tight frames and equiangular frames have strong connections to Grassmannian frames. In a comprehensive paper [147], Strohmer and Heath discuss those frames and their connection to Grassmannian packings, spherical codes, graph theory, Welch Bound sequences (see also [88]). These frames are of unit norm (not a necessary restriction) and minimize the maximum correlation \(|\langle \varphi_i, \varphi_j \rangle|\) among all frames. The problem arises from looking at overcomplete systems closest to orthonormal bases (which have minimum correlation). A simple example is an HTF in \( \mathbb{H}^n \). Theorem 2.3 in [147] states that, given a frame \( \Phi \):

\[
\min_{\Phi} \left( \max_{(\varphi_i, \varphi_j)} |\langle \varphi_i, \varphi_j \rangle| \right) \geq \sqrt{\frac{m - n}{n(m - 1)}}.
\]  

The equality in (10.26) is achieved if and only if \( \Phi \) is equiangular and tight. In particular, for \( \mathbb{H} = \mathbb{R} \), equality is possible only for \( m \leq n(n + 1)/2 \), while for \( \mathbb{H} = \mathbb{C} \), equality is possible only for \( m \leq n^2 \). Note that the above inequality is exactly the one Welch proved in [166] and which later lead to what is today commonly referred to as the Welch’s Bound by minimizing interuser interference in a CDMA system [116].
10.4 Infinite-Dimensional Frames Via Filter Banks

Here we plan on discussing those frames most often used in practice—filter bank frames. They operate on signals in $\ell^2(\mathbb{Z})$, and subsume many well known frame families. This will be the redundant counterpart of Chapter 6 and, as in that chapter, we will heavily use the matrix representation of filter banks implementing frames.

Example 10.3 (MB Frame (cont’d)). Let us look at the simplest case using our favorite example: the MB frame we introduced in Section 10.1. Our $\Phi^*$ is now block-diagonal, with $\Phi_0^* = \Phi^*$ from (10.4) on the diagonal. In contrast to finite-dimensional bases implemented by filter banks (see Haar in (6.43)), the block $\Phi_0^*$ is now rectangular of size $3 \times 2$. This finite-dimensional frame is equivalent to the filter bank shown in Fig. 10.4, with $\{\tilde{\phi}_i\} = \{\phi_i\}$, given in (10.4).

As we could for finite-dimensional bases, we can investigate finite-dimensional frames within the filter bank framework (see Fig. 10.5). In other words, most of the interesting cases we encounter in practice, both finite dimensional and infinite dimensional, we can look at as filter banks.

Similarly to bases, if the support of the frame vectors is larger than the sampling factor, we resort to the polyphase-domain analysis. Assume that the filter length is $l = kn$ (if not, we can always pad with zeros), and write the frame as (causal filters)

$$
\Phi^* = \begin{bmatrix}
\Phi_0^* & 0 & \cdots & 0 & 0 & \cdots \\
\Phi_1^* & \Phi_0^* & \cdots & 0 & 0 & \cdots \\
\vdots & \vdots & \ddots & \vdots & \vdots & \ddots \\
\Phi_{k-1}^* & \Phi_{k-2}^* & \cdots & \Phi_0^* & 0 & \cdots \\
\cdots & 0 & \Phi_{k-1}^* & \cdots & \Phi_1^* & \Phi_0^* & \cdots \\
\vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \ddots
\end{bmatrix}
$$

(10.27)
Chapter 10. Frames on Sequences

Figure 10.5: A filter bank implementation of a frame expansion: It is an $m$-channel filter bank with sampling by $n$.

where each block $\Phi_i$ is of size $n \times m$:

$$
\Phi_0 = \begin{bmatrix}
\varphi_{00} & \cdots & \varphi_{0,m-1} \\
\vdots & \ddots & \vdots \\
\varphi_{n-1,0} & \cdots & \varphi_{n-1,m-1}
\end{bmatrix}.
$$

In the above, we enumerate template frame vectors from $0, \ldots, m-1$.

To contrast filter-bank frames with what we already know, filter-bank bases, we can see that the class of multiresolution transforms obtained using a filter bank depends on three parameters: the number of vectors $m$, the shift or sampling factor $n$ and the length $l$ of the nonzero support of the vectors.

For bases, $m = n$, the filter bank is critically sampled and implements a nonredundant expansion—basis. The basis $\Phi$ has a dual basis associated with it, $\tilde{\Phi}$, leading to biorthogonal filter banks. The associated matrices $\Phi, \tilde{\Phi}$ are invertible. We have already seen that in the $z$-domain, this is equivalent to (6.81). When the basis is orthonormal, $\tilde{\Phi} = \Phi$, the associated matrix $\Phi$ is unitary, and its polyphase matrix is paraunitary as in (6.42). When $l = m$, we have a block transform, a famous example being the Discrete Fourier Transform (DFT), which we discussed earlier. When $m = 2$, we get two-channel filter banks, discussed in Chapter 6. By iterating on the lowpass channel, we get the Discrete Wavelet Transform (DWT), introduced in Chapter 7. When $l = 2m$, we get Lapped Orthogonal Transforms (LOT).

For frames, $m > n$. The filter bank in this case implements a redundant expansion—frame. The frame $\Phi$ has a dual frame associated with it, $\tilde{\Phi}$. The associated matrices $\Phi, \tilde{\Phi}$ are rectangular and left/right invertible. This has been formalized in $z$-domain in [48], as the following result: A filter bank implements a frame decomposition in $\ell^2(Z)$ if and only if its polyphase matrix is of full rank on the unit circle.

An important subcase is when the frame $\Phi$ is tight, in which case it is self-dual, that is, $\tilde{\Phi} = \Phi$, and the associated matrix $\Phi$ satisfies $\Phi \Phi^* = I$. This has been formalized in $z$-domain in [48], as the following result: A filter bank implements a tight frame expansion in $\ell^2(Z)$ if and only if its polyphase matrix is paraunitary. A well-known subcase of tight frames is the following: When $l = n$, we have a block transform. Then, in (10.27), only $\Phi_0$ is nonzero, making $\Phi$ block-diagonal. In
10.4. Infinite-Dimensional Frames Via Filter Banks

effect, since there is no overlap between processed blocks, this can be analyzed as a finite-dimensional case, where both the input and the output are \( n \)-dimensional vectors. A thorough analysis of oversampled filter banks seen as frames is given in [48, 22, 49].

### 10.4.1 Gabor-Like Frames

The same way LOTs were constructed to avoid problems of block transforms, Gabor frames are counterparts in frames. They are overlapping with complex basis functions.

The idea behind this class of frames, consisting of many families, dates back to Gabor [71] and the insight of constructing bases by modulation of a single prototype function. Gabor originally used complex modulation, and thus, all those families with complex modulation are termed 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. 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 Gabor frame (with the same generator and twice the redundancy) and an additional operator, which vanishes if the generator satisfies certain symmetry properties. While this decomposition has first been used by Auscher in the context of Wilson bases [5], it is valid more generally. Both of these classes can be seen as general oversampled filter banks with \( m \) channels and sampling by \( n \) (see Fig. 10.5).

#### Gabor Frames

A Gabor frame is \( \Phi = \{\varphi_i\}^{m-1}_{i=0} \), with

\[
\varphi_{i,k} = W^{-ik} \varphi_{0,k},
\]

where index \( i = 0, \ldots, m - 1 \) refers to the number of frame elements, \( k \in \mathbb{Z} \) is the discrete-time index, \( W_m \) is the \( m \)th root of unity and \( \varphi_0 \) is the template frame function. Comparing (10.28) with (10.23), we see that for filter length \( l = n \) and \( \varphi_{0,k} = 1, k = 0 \) and 0 otherwise, the Gabor system is equivalent to a HTF frame. Thus, it is sometimes called the oversampled DFT frame.

For the critically-sampled case it is known that one cannot have Gabor bases with good time and frequency localization at the same time (this is similar in spirit to the Balian-Low theorem which holds for \( L^2(\mathbb{R}) \) [52]); this prompted the development of oversampled (redundant) Gabor systems (frames). They are known under various names: oversampled DFT FBs, complex-modulated FBs, short-time Fourier FBs and Gabor FBs and have been studied in [46, 22, 21, 20, 69] (see also [146] and references within).

#### Cosine-Modulated Frames

The other kind of modulation, cosine, was used with great success within critically-sampled filter banks due to efficient implementation algorithms. Its oversampled
version was introduced in [21], where the authors define the frame elements as:

$$\varphi_{i,k} = \sqrt{2} \cos\left(\frac{(i+1/2)\pi}{m} + \alpha_i\right)\varphi_0, k,$$

(10.29)

where index $i = 0, \ldots, m - 1$ refers to the number of frame elements, $k \in \mathbb{Z}$ is the discrete-time index and $\varphi_0$ is the template frame function. Equation (10.29) defines the so-called odd-stacked cosine modulated FBs; even-stacked ones exist as well.

Cosine-modulated filter banks do not suffer from time-frequency localization problems, given by a general result stating that the generating window of an orthogonal cosine modulated FB can be obtained by constructing a tight complex filter bank with oversampling factor 2 while making sure the window function satisfies a certain symmetry property (for more details, see [21]). Since we can get well-localized Gabor frames for redundancy 2, this also shows that we can get well-localized cosine-modulated filter banks.

### 10.4.2 Oversampled Discrete Wavelet Transform

The same way we could argue for going from just the plain filter bank to the DWT, where some time-frequency constraints were added (see Table 10.1), we can argue for going from overlapping frames such as Gabor ones to oversampled DWT. How do we achieve such redundancy? One questions is whether we can get finer in time, and it is clear that this is possible if we remove some downsamplers. Consider Fig. 10.8(a), showing the sampling grid for the DWT. 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 (see Fig. 7.2 with three levels) with the samplers removed only at the first level (see Fig. 10.6). This transform is also known as the partial DWT (see Section 10.4.3). The redundancy of this scheme is clearly $A_j = 2$ at each level for a total redundancy of $A = 2$. The sampling grid is depicted in Fig. 10.8(c).

By collapsing the tree into three equivalent branches, we obtain the filter bank as in Fig. 10.7 (similarly to what we have done in Fig. 7.6). Using the notation as in Fig. 10.7, the frame can be expressed as

$$\Phi = \{\tau_{2i}(g * g), \tau_{2i}(g * h), \tau_i h\}_{i \in \mathbb{Z}},$$

(10.30)

where $\tau_i$ denotes shift by $i$. The template vector $h$ moves by 1, while the other two, $g * g$ and $g * h$ move by 2. Thus, the basic block of the infinite matrix is of size $8 \times 4$ (the smallest period after which it starts repeating itself). Therefore, one could look at this example as having 8 vectors where 4 would have been enough (thus redundancy of 2). One can thus redefine the template frame vectors as:

$$\varphi_0 = (g * g), \quad \varphi_1 = \tau_2(g * g),$$
$$\varphi_2 = (g * h), \quad \varphi_3 = \tau_2(g * h),$$
$$\varphi_4 = h, \quad \varphi_5 = \tau_2 h, \quad \varphi_6 = \tau_4 h, \quad \varphi_7 = \tau_6 h.$$

Then,

$$\Phi = \{\tau_{4i}\varphi_j\}_{i \in \mathbb{Z}, j=0,\ldots,7}.$$
10.4. Infinite-Dimensional Frames Via Filter Banks

The synthesis part of the filter bank implementing the oversampled DWT. The samplers are omitted at the first level but exist at all other levels. The analysis part is analogous.

![Figure 10.6](image)

**Figure 10.6:** The synthesis part of the filter bank implementing the oversampled DWT. The samplers are omitted at the first level but exist at all other levels. The analysis part is analogous.

The synthesis part of the equivalent three-channel filter bank implementing the oversampled DWT with 2 levels. The analysis part is analogous.

![Figure 10.7](image)

**Figure 10.7:** The synthesis part of the equivalent three-channel filter bank implementing the oversampled DWT with 2 levels. The analysis part is analogous.

10.4.3 Other Infinite-Dimensional Frame Families

We now give a brief account of other known infinite-dimensional frame families.

The Algorithme à Trous

The algorithme à trous is a fast implementation of the dyadic (continuous) wavelet transform. It was first introduced by Holschneider, Kronland-Martinet, Morlet, and Tchamitchian in 1989 [89]. The transform is implemented via a biorthogonal, nondownsampled filter bank. An example would be the 2-level DWT with samplers removed (see Fig. 10.10).

Let $g$ and $h$ be the filters used in this filter bank. At level $i$ we will have equivalent upsampling by $2^i$ which means that the filter moved across the upsampler will be upsampling by $2^i$, inserting $(2^i - 1)$ zeros between every two samples and thus creating holes (“trous” means “hole” in French).

Fig. 10.8(d) shows the sampling grid for the à trous algorithm. It is clear from the figure, that this scheme is completely redundant, as all the points exist. This is in contrast to a completely nonredundant scheme such as the DWT, given part (a) of the figure. In fact, the redundancy (or frame bound) of this algorithm grows exponentially since $A_1 = 2, A_2 = 4, \ldots, A_j = 2^j, \ldots$ (note that here we use a
two-channel filter bank and that $A_j$ is the frame bound when we use $j$ levels). This growing redundancy is the price we pay for the simplicity of the algorithm. The 2D version of the algorithm is obtained by extending the 1D version in a separable manner.

In Section 10.5, we give pseudo-code for the à trous algorithm.

**Pyramid Frames**

Pyramid frames for coding were introduced in 1983 by Burt and Adelson [26]. 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 is derived (an example of how this could be done is in Fig. 10.9). While in
10.4. Infinite-Dimensional Frames Via Filter Banks

The analysis part of the pyramid filter bank [26] with orthonormal filters $g$ and $h$, corresponding to a tight frame.

Fig. 10.9 the intensity of the coarse approximation $X_0$ 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. 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 $X_1$. If the prediction is good (which will be the case for most natural images which 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. In the absence of quantization of $X_1$, the original is obtained by simply adding back the prediction at the synthesis side.

The redundancy for pyramid frames is $A_1 = \frac{3}{2}, A_2 = \frac{7}{4}, \ldots A_\infty = 2$ (see Fig. 10.8(b)), far less than the à trous construction, for example. Thanks to the above, pyramid coding has been recently used together with directional coding to form the basis for nonseparable MD frames called contourlets (see Section 10.4.3).

The Dual-Tree Complex Wavelet Transform

The dual-tree complex wavelet transform (DT-CWT) was first introduced by Kingsbury in 1998 [94, 95, 96]. The basic idea is to have two DWT trees working in parallel. One tree represents the real part of the complex transform while the second tree represents the imaginary part. That is, when the DT-CWT is applied to a real signal, the output of the first tree is the real part of the complex transform whereas the output of the second tree is its imaginary part. When the two DWTs used are orthonormal, the DT-CWT is a tight frame.

Because the two DWT trees used in the DT-CWT are fully downsampled, the redundancy is only 2 for the 1D case (it is $2^d$ for the d-dimensional case). We can see that in Fig. 10.8(c), where the redundancy at each level is twice that of the DWT, that is $A_1 = A_2 = \ldots A_j = 2$. Unlike the à trous algorithm, however, here the redundancy is independent of the number of levels used in the transform.

Double-Density Frames

Selesnick in [133] introduces the double-density DWT (DD-DWT), which can approximately be implemented using a three-channel FB with sampling by 2 as in
Fig. 10.4. The filters in the analysis bank are time-reversed versions of those in the synthesis bank. The redundancy of this FB tends towards 2 when iterated on the channel with $\phi_1$. Actually, we have that $A_1 = \frac{3}{2}, A_2 = \frac{7}{4}, \ldots A_\infty = 2$ (see Fig. 10.8(b)). Like the DT-CWT, the DD-DWT is nearly shift invariant when compared to the à trous construction. In [134], Selesnick introduces the combination of the DD-DWT and the DT-CWT which he calls double-density, DT-CWT (DD-DT-CWT). This transform can be seen as the DT-CWT, with individual filter banks being overcomplete ones given in Fig. 10.4 (DD-DWT).

### Multidimensional Frames

Apart from obvious, tensor-like, constructions (separate application of 1D methods in each dimension) of multidimensional (mD) frames, we are interested in true mD solutions. The oldest mD frame seems to be the steerable pyramid introduced by Simoncelli, Freeman, Adelson and Heeger in 1992 [137], following on the previous work by Burt and Adelson on pyramid coding [26]. The steerable pyramid possesses many nice properties, such as joint space-frequency localization, approximate shift invariance, approximate tightness, oriented kernels, approximate rotation invariance and a redundancy factor of $4j/3$ where $j$ is the number of orientation subbands. The transform is implemented by a first stage of lowpass/highpass filtering followed by oriented bandpass filters in the lowpass branch plus another lowpass filter in the same branch followed by downsampling. An excellent overview of the steerable pyramid and its applications is given on Simoncelli’s web page [132].

Another beautiful example is the recent work of Do and Vetterli on contourlets [58, 45]. This work was 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 coding and pyramid filter banks [57] with directional processing, to obtain contourlets, expansions capturing contour segments. The transform is a frame composed of a pyramid FB and a directional FB. Thus, first a wavelet-like method is used for edge detection (pyramid) followed by local directional transform for contour segment detection. It is almost critically sampled, with redundancy of 1.33. It draws on the ideas of a pyramidal directional filter banks (PDFB) which is a Parseval tight frame when all the filters used are orthogonal.

### 10.5 Algorithms

#### 10.5.1 The Algorithme à Trous

This algorithm is depicted in Fig. 10.10. The figure shows that first, the equivalent filters in each branch are computed, 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.
Algorithm 10.1 [Algorithm à trous] Input: $x = \alpha_0$, the input signal. Output: $\alpha_j, \beta_i, i = 1, \ldots, j$, transform coefficients.

\textbf{ATROUS}($\alpha_0$)

\begin{algorithm}
\hspace{1cm} \textbf{initialize} \\
\hspace{1cm} \textbf{for} \hspace{0.5cm} i = 1 \hspace{0.5cm} \textbf{to} \hspace{0.5cm} j \hspace{0.5cm} \textbf{do} \\
\hspace{1.8cm} \alpha_j = \alpha_{j-1} \ast (\uparrow 2^{j-1})g \\
\hspace{1.8cm} \beta_j = \alpha_{j-1} \ast (\uparrow 2^{j-1})h \\
\hspace{1cm} \textbf{end for} \\
\hspace{1cm} \textbf{return} \hspace{0.5cm} \alpha_j, \beta_i, i = 1, \ldots, j \\
\end{algorithm}

\begin{figure}[h]
\centering
\includegraphics[width=0.5\textwidth]{figure10.10.png}
\caption{The synthesis part of the equivalent filter bank with 2 levels corresponding the à trous algorithm. The analysis part is analogous. This is similar to Fig. 7.6 (three levels) with sampling removed.}
\end{figure}

Chapter at a Glance

Table 10.1 depicts relationships existing between various classes of bases and frames. For example, the block-transform counterpart of the DFT will be the HTF (harmonic tight frames), while the same for the LOT will be Gabor frames. By increasing the support of basis functions we can go from the DFT to the LOT, and similarly, from HTF to Gabor frames. Imposing time-frequency constraints leads to new classes of representations, such as the DWT, whose frame counterpart is the oversampled DWT.

Historical Remarks

It is generally acknowledged\(^{58}\) that frames were born in 1952 in the paper by Duffin and Schaeffer [64]. Despite being over half a century old, frames gained popularity only in the last decade, due mostly to the work of the three wavelet pioneers—Daubechies, Grossman and Meyer [54]. Frame-like ideas, that is, building redundancy into a signal expansion, can be found in pyramid coding [26], source coding [52, 118, 13, 76, 47, 53, 77, 12], denoising [169, 39, 61, 91, 70], robust transmission [75, 88, 147, 14, 15, 16, 157, 31, 19, 126], CDMA systems [116, 163, 164, 149], multiantenna code design [80, 87], segmentation [150, 106, 56], classification [150, 106, 35], prediction of epileptic seizures [10, 11], restoration and enhancement [94], motion estimation [112], signal reconstruction [6], coding theory [81, 131],

\(^{58}\)At least in the signal processing and harmonic analysis communities.
Chapter 10. Frames on Sequences

\[ \beta_j \rightarrow (\uparrow 2^j)h \]

\[ \alpha_j \rightarrow (\uparrow 2^j)g \]

\[ \beta_1 \]

\[ h \]

\[ \alpha_1 \]

\[ g \]

\[ \begin{array}{c}
\downarrow \quad \uparrow \quad \uparrow \quad \uparrow \\
Bases \quad \text{DFT} \quad \rightarrow \quad \text{LOT} \quad \rightarrow \quad \text{DWT} \\
Frames \quad \text{HTF} \quad \rightarrow \quad \text{Gabor} \quad \rightarrow \quad \text{Oversampled DWT}
\end{array} \]

**Figure 10.11:** The synthesis part of the filter bank implementing the à trous algorithm. This version is actually used in implementation. The analysis part is analogous.

**Table 10.1:** Bases versus frames.

Further Reading

The sources on frames are the beautiful book by Daubechies (our wavelet Bible) [52], a recent book by Christensen [36] as well as a number of classic papers [30, 83, 51, 79] as well as an introductory tutorial on frames [100].

**Frame nomenclature.** Frame nomenclature is far from uniform and can result in confusion. For example, frames with unit-norm frame vectors have been called normalized frames (normalized as in all vectors normalized to norm 1, similarly to the meaning of normalized in orthonormal bases), uniform, as well as uniform frames with norm 1. We now mentioned a few of these, following our nomenclature: Equal-norm frames have been called uniform frames [101]. Unit-norm frames have been called uniform frame with norm 1 [101], uniform frames [75], normalized frames [9]. Parseval tight frames have been called normalized frames [7]. Unit-norm tight frames have been called uniform tight frames with norm 1 [101], uniform tight frames [75], normalized tight frames [9].

**Naimark Theorem.** We mentioned the theorem in Section 10.3. The theorem has been rediscovered by several people in the past decade: We heard it from Daubechies in the mid-90’s. Han and Larson rediscovered it in [79]; they came up with the idea that a frame could be obtained by compressing a basis in a larger space and that the process is reversible.
### Further Reading

<table>
<thead>
<tr>
<th>Frame Constraints</th>
<th>Properties</th>
</tr>
</thead>
<tbody>
<tr>
<td>General</td>
<td>${\varphi_i}<em>{i \in I}$ is a Riesz basis for $H$&lt;br&gt;$A|x|^2 \leq \sum</em>{i \in I}</td>
</tr>
<tr>
<td>ENF</td>
<td>$|\varphi_i| = |\varphi_j| = a$&lt;br&gt;for all $i$ and $j$&lt;br&gt;$A|x|^2 \leq \sum_{i \in I}</td>
</tr>
<tr>
<td>TF</td>
<td>$A = B$&lt;br&gt;$\sum_{i \in I}</td>
</tr>
<tr>
<td>PTF</td>
<td>$A = B = 1$&lt;br&gt;$\sum_{i \in I}</td>
</tr>
<tr>
<td>ENTF</td>
<td>$A = B$&lt;br&gt;$\sum_{i \in I}</td>
</tr>
<tr>
<td>UNTF</td>
<td>$A = B$&lt;br&gt;$|\varphi_i| = 1$&lt;br&gt;for all $i$&lt;br&gt;$\sum_{i \in I}</td>
</tr>
<tr>
<td>ENPTF</td>
<td>$A = B = 1$&lt;br&gt;$|\varphi_i| = |\varphi_j| = a$&lt;br&gt;for all $i$ and $j$&lt;br&gt;$\sum_{i \in I}</td>
</tr>
<tr>
<td>UNPTF</td>
<td>$A = B = 1$&lt;br&gt;$|\varphi_i| = 1$&lt;br&gt;$\sum_{i \in I}</td>
</tr>
</tbody>
</table>

Table 10.2: Summary of properties for various classes of frames. All trace identities are given for $H = \mathbb{R}^N, \mathbb{C}^N$.

Finally, it was Šoljanin [141] who pointed out to us that this is, in fact, Naimark’s theorem, which has been widely known in operator algebra and used in quantum information theory.

**Harmonic Tight Frames.** Similar ideas to those of harmonic tight frames have appeared in the work by Eldar and Bölcskei [66] under the name *geometrically uniform frames (GU)*, frames defined over a finite Abelian group of unitary matrices both with a
single generator as well as multiple generators. The authors also consider constructions of such frames from given frames, closest in the least-squares sense, a sort of a “Gram-Schmidt” procedure for GU frames.

**Grassmannian frames.** Coming more from the wireless setting, in a more recent work, Xia, Zhou and Giannakis [168] constructed some new frames meeting the original Welch’s Bound.

Grassmannian frames coincide with some optimal packings in Grassmannian spaces [40], spherical codes [41], equiangular lines [109], and many others. The equiangular lines are equivalent to the SIC-POVMs we discussed above.

**Gabor frames.** More recent work on Gabor frames includes [107], where the authors study finite-dimensional Gabor systems and show a family in $\mathbb{C}^n$, with $m = n^2$ vectors, which allows for $n^2 - n$ erasures, where $n$ is prime. In [103], new classes of Gabor ENTFs are shown, which are also MR.

**Double-density frames and variations.** In [1], Abdelnour and Selesnick introduce symmetric, nearly shift-invariant FBs implementing tight frames. These filter banks have 4 filters in two couples, obtained from each other by modulation. Sampling is by 2 and thus the total redundancy is 2.

Another variation on a theme is the power-shiftable DWT (PSDWT) [137] or partial DWT (PDWT) [142], which removes samplers at the first level but leaves them at all other levels. The sampling grid of the PSDWT/PDWT is shown in Fig. 10.8(c). We see that it has redundancy $A_j = 2$ at each level (similarly to the CWT). The PSDWT/PDWT achieves near shift invariance.

Bradley in [24] introduces overcomplete DWT (OC-DWT), the DWT with critical sampling for the first $k$ levels followed by à trous for the last $j - k$ levels. The OC-DWT becomes the à trous algorithm when $k = 0$ or the DWT when $k = j$.

**Multidimensional frames.** Some other examples of multidimensional frames include [110] where the authors build both critically-sampled and nonsampled (à trous like) 2D DWT. It is obtained by a separable 2D DWT producing 4 subbands. The lowest subband is left as is, while the three higher ones are split into two subbands each using a quincunx FB (checkerboard sampling). The resulting FB possesses good directionality with low redundancy. Many “-lets” are also multidimensional frames, such as curvelets [29, 28] and shearlets [105]. As the name implies, curvelets are used to approximate curved singularities in an efficient manner [29, 28]. 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 [105].

---

**Exercises with Solutions**

10.1. **Unit-Norm Tight Frames:**

Prove that the following two statements are equivalent:

(i) $\{\varphi_i = (\cos \alpha_i, \sin \alpha_i)^T\}_{i=1}^m$, is a unit-norm tight frame.

(ii) $\sum_{i=1}^m z_i = 0$ where $z_i = e^{j2\pi \alpha_i}$ for $i = 1, 2, \ldots, m$. 
Exercises with Solutions

Solution:
Form the frame operator matrix

$$\Phi^* = \begin{pmatrix}
\cos \alpha_1 & \sin \alpha_1 \\
\cos \alpha_2 & \sin \alpha_2 \\
\vdots & \vdots \\
\cos \alpha_m & \sin \alpha_m \\
\end{pmatrix}. \quad (E10.1-1)$$

For the frame to be tight is to have \( \Phi \Phi^* = \frac{m}{2} I_2 \), which leads to

$$\sum_{i=1}^m \cos^2 \alpha_i = \frac{m}{2}, \quad (E10.1-2)$$
$$\sum_{i=1}^m \sin^2 \alpha_i = \frac{m}{2}, \quad (E10.1-3)$$
$$\sum_{i=1}^m \cos \alpha_i \sin \alpha_i = 0. \quad (E10.1-4)$$

Subtracting (E10.1-3) from (E10.1-2) gives

$$\sum_{i=1}^m \cos 2\alpha_i = 0, \quad (E10.1-5)$$

while multiplying (E10.1-4) by 2 yields

$$\sum_{i=1}^m \sin 2\alpha_i = 0. \quad (E10.1-6)$$

Finally, adding (E10.1-5) to \( j \) times (E10.1-6) gives

$$\sum_{i=1}^m z_i = 0, \text{ where } z_i = e^{j2\alpha_i}. \quad (E10.1-7)$$

10.2. Four-Dimensional Frames:
You are given a unit-norm tight frame with four vectors in \( \mathbb{R}^2 \), that is, \( \{ \varphi_i = (\cos \alpha_i, \sin \alpha_i)^T \}_{i=1}^4 \).
Prove that all unit-norm tight frames with \( n = 2, m = 4 \), are unions of two orthonormal bases, parameterized by the angle between them (within the equivalence class of all frames obtained from the original frame by rigid rotations, reflections around an axis and negation of individual vectors).

Solution:

Using the result of Problem 10.1, we know that for our frame:

$$z_1 + z_2 + z_3 + z_4 = 0.$$

This means that we have to construct a closed path starting at 0, using four vectors of length 1. The solution is a parallelogram. Without loss of generality, assume that \( z_1 = 1 \). Then, \( z_3 = -z_1 \) and \( z_4 = -z_2 \). Denoting by \( 2\alpha \) the angle between \( z_1 \) and \( z_2 \), we get that

$$2\alpha_1 = 2k\pi, \quad 2\alpha_3 = 2n\pi + \pi, \quad 2\alpha_2 = 2m\pi + 2\alpha, \quad 2\alpha_4 = 2l\pi + \pi + 2\alpha,$$

from where

$$\alpha_1 = k\pi, \quad \alpha_3 = n\pi + \pi/2, \quad \alpha_2 = m\pi + \alpha, \quad \alpha_4 = l\pi + \pi/2 + \alpha.$$ 

Choosing \( k = 0, n = 0, m = 0, l = 0 \), we obtain

$$\alpha_1 = 0, \quad \alpha_3 = \pi/2, \quad \alpha_2 = \alpha, \quad \alpha_4 = \pi/2 + \alpha.$$

It is now obvious that \( \varphi_1 \) and \( \varphi_3 \) form an orthonormal basis, as do \( \varphi_2 \) and \( \varphi_4 \) and the angle between the two ONBs is \( \alpha \).
Exercises

10.1. Parseval’s Formula for Tight Frames:
You are given a two-channel orthogonal filter bank with lowpass/highpass filters as in Theorem 6.1: Remove samplers from such a filter bank and prove the corresponding Parseval’s formula:
\[ \|x\|^2 = 2(\|\alpha\|^2 + \|\beta\|^2), \]
where \(x\) is the input signal and \(\alpha_k, \beta_k\) are the sequences of transform coefficients at the outputs of the lowpass/highpass channels, respectively.

10.2. Relation Between the MSE in the Original and Transform Domains:
Consider a signal \(x \in \mathbb{R}^N\) and its transform coefficients \(X \in \mathbb{R}^{M \times n}\) defined as \(X = \Phi^T x\), where \(\Phi^T \in \mathbb{R}^{m \times n}\) is a tight frame or an orthogonal basis, and \(m \geq n\). The main goal of approximation is to represent a signal \(x\) by a portion of \(N\) transform coefficients, whereas the rest is set to zero. Equivalently, the \(N\)-term approximation \(\hat{x}_N\) is obtained by the inverse of \(\hat{X}_N\) that is the truncated version of \(X\). Owing to the truncation of the coefficients, the approximating signal \(\hat{x}_N\) does not match exactly the original signal \(x\). Show that the mean-square error (MSE) in the original and transform domains are related as
\[ \|x - \hat{x}_N\|^2 \leq \frac{1}{A}\|X - \hat{X}_N\|^2, \]
where \(A\) is the frame bound of \(\Phi^T\). When does the equality hold?

10.3. Frame Filter Bank:
Consider a 3-channel analysis/synthesis filter bank downsampled by 2 as in Fig. 10.4, with filtering of the channels (channel filters \(C_i, i = 1, 2, 3\) are located between the down- and upsamplers). The filters are given by
\[ \begin{align*}
\Phi_1(z) &= z^{-1}, & \Phi_2(z) = 1 + z^{-1}, & \Phi_3(z) = 1 \\
\Phi_1(z) &= 1 - z^{-1}, & \Phi_2(z) = z^{-1}, & \Phi_3(z) = z^{-2} - z^{-1} \\
C_1(z) &= F_0(z), & C_2(z) &= F_0(z) + F_1(z), & C_3(z) &= F_1(z).
\end{align*} \]
Verify that the overall system is shift-invariant and performs a convolution with a filter having the \(z\)-transform \(F(z) = (F_0(z^2) + z^{-1}F_1(z^2))z^{-1}\).

10.4. Orthogonal Pyramid:
Consider a pyramid decomposition as discussed in Section 10.4.3 and shown in Fig. 10.9. Now assume that \(g_n\) is an “orthogonal” filter, that is, \((g_n, g_{n-2l}) = \delta_l\). Perfect reconstruction is achieved by upsampling the coarse version, filtering it by \(h\), and adding it to the difference signal.
(i) Analyze the above system in time domain and in \(z\)-transform domain, and show perfect reconstruction.
(ii) Take \(g_n = (1/\sqrt{2})(1 + \delta_n)\). Show that \(X_1\) can be filtered by \(h = (1/\sqrt{2})(1 - \delta_n)\) and downsampled by 2 while still allowing perfect reconstruction.
(iii) Show that (ii) is equivalent to a two-channel perfect reconstruction filter bank with filters \(g_n = (1/\sqrt{2})(1 + \delta_n)\) and \(h_n = (1/\sqrt{2})(1 - \delta_n)\).
(iv) Show that (ii) and (iii) are true for general orthogonal lowpass filters, that is, \(X_1\) can be filtered by \(h_n = (-1)^ng_{-n+L-1}\) and downsamples by 2, and reconstruction is still perfect using an appropriate filter bank.
Chapter 11

Continuous Wavelet and Windowed Fourier Transforms

Contents

11.1 The Continuous Wavelet Transform . . . . . . . . 394
11.2 The Windowed Fourier Transform . . . . . . . . . 408
11.3 Sampling and Computing the CWT and WFT . . 416
11.4 Discussion and Summary . . . . . . . . . . . . . . . 416
Chapter at a Glance ................................. 416
Historical Remarks ................................ 416
Further Reading .................................... 417
Exercises with Solutions ............................ 417
Exercises ........................................... 417

Through Chapter 9, we were concerned primarily with bases for sequences and functions. Then in the preceding chapter, we considered frames, which are intuitively “overcomplete bases.” What happens when this overcompleteness goes to infinity, for example, we have all scales and shifts in the wavelet case or all modulations and shifts in the local Fourier case? Clearly, this is very different than the basis or frame expansions seen so far, since a function of one variable is mapped to a function of two continuous variables. Yet lots of intuition can be carried over from the previous cases, and thus, this is a natural continuation of our exploration of time-scale and time-frequency representations. The major difference is that we are moving from expansions, which are countable, to transforms, which are not. For example, instead of a local Fourier series as in Chapter 9, we have a local Fourier transform.

The main questions for these transforms parallel the questions asked in earlier chapters: (i) Is the transform invertible? (ii) Is it energy preserving? (iii) What are the characteristics of the transform in terms of time-frequency-scale analysis? We consider these questions for the continuous wavelet transform in Section 11.1 and for the windowed Fourier transform in Section 11.2. Section 11.3 concludes the chapter with a discussion of how the sampling of these continuous transforms leads to frames and how they can be computed efficiently.
### 11.1 The Continuous Wavelet Transform

#### 11.1.1 Definition

The continuous wavelet transform uses a function \( \psi(t) \) and all its shifted and scaled versions to analyze signals. Here we consider only real wavelets; this can be extended to complex wavelets without too much difficulty.

Consider a real wavelet \( \psi(t) \in L^2(\mathbb{R}) \) 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

\[
\psi_{a,b}(t) = \frac{1}{\sqrt{a}} \psi\left(\frac{t - b}{a}\right), \quad a \in \mathbb{R}^+, \quad b \in \mathbb{R},
\]

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 \( L^2 \) 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 \( \Psi(\omega) \) must satisfy

\[
C_\psi = \int_0^\infty \frac{|\Psi(\omega)|^2}{|\omega|} \, d\omega < \infty. \tag{11.2}
\]

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, (11.2) is a very mild requirement that is satisfied by all wavelets of interest (see, for example, Exercise 11.1). Now, given a function \( f \) in \( L^2(\mathbb{R}) \), we can define its continuous wavelet transform as

\[
\text{CWT}_f(a,b) = \frac{1}{\sqrt{a}} \int_{-\infty}^\infty \psi\left(\frac{t - b}{a}\right) f(t) \, dt = \int_{-\infty}^\infty \psi_{a,b}(t) f(t) \, dt = \langle \psi_{a,b}, f \rangle. \tag{11.3}
\]

In words, we take the inner product of the function \( f \) with a wavelet centered at location \( b \), and rescaled by a factor \( a \), which is shown in Figure 11.1. A numerical example is given in Figure 11.2, which displays the magnitude \( |\text{CWT}_f(a,b)| \) as an image. It is already clear that the CWT 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,

\[
\psi_a(t) = \frac{1}{\sqrt{a}} \psi\left(\frac{t}{a}\right) \quad \text{CTFT} \quad \Psi_a(\omega) = \sqrt{a} \Psi(a\omega), \tag{11.4}
\]

as well as the notation \( \tilde{\psi}(t) = \psi(-t) \). Then

\[
\text{CWT}_f(a,b) = \int_{-\infty}^\infty \frac{1}{\sqrt{a}} \psi\left(\frac{t - b}{a}\right) f(t) \, dt = \int_{-\infty}^\infty \psi_a(t - b) f(t) \, dt = (f * \tilde{\psi}_a)(b). \tag{11.5}
\]
11.1. The Continuous Wavelet Transform

Figure 11.1: Continuous wavelet transform. (a) The prototype wavelet, which is here the usual Haar wavelet, but centered at 0. (b) A function $f(t)$, and below a few examples of wavelets $\psi_{a,b}(t)$ positioned in scale space $(a, b)$.

Figure 11.2: The continuous wavelet transform. (a) An example signal. (b) The magnitude of continuous wavelet transform $|\text{CWT}_f(a,b)|$.

Now the Fourier transform of $\text{CWT}_f(a,b)$ over the “time” variable $b$ is

$$\text{CWT}_f(a, \omega) = F(\omega)\Psi^*(\omega) = F(\omega)\sqrt{a}\Psi^*(a\omega), \quad (11.6)$$

where we used $\psi(-t) \xrightarrow{\text{CTFT}} \Psi^*(\omega)$ since $\psi(t)$ is real.

11.1.2 Invertibility

The invertibility of the continuous wavelet transform is of course a key result: not only can we compute the CWT, but we are actually able to come back! This inversion formula was first proposed by J. Morlet.\footnote{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.”}

**Proposition 11.1 (Inversion of the Continuous Wavelet Transform).** Consider a real wavelet $\psi$ satisfying the admissibility condition (11.2). A function $f \in L^2(\mathbb{R})$...
can be recovered from its continuous wavelet transform $CWT_f(a,b)$ by the inversion formula

$$f(t) = \frac{1}{C_\psi} \int_0^\infty \int_{-\infty}^\infty CWT_f(a,b) \psi_{a,b}(t) \frac{db \, da}{a^2}, \quad (11.7)$$

where equality is in the $L^2$ sense.

**Proof.** Denote the right hand side of (11.7) by $x(t)$. In that expression, we replace $CWT_f(a,b)$ by (11.5) and $\psi_{a,b}(t)$ by $\psi_a(t-b)$ to obtain

$$x(t) = \frac{1}{C_\psi} \int_0^\infty \int_{-\infty}^\infty (f * \bar{\psi}_a)(b) \psi_a(t-b) \frac{db \, da}{a^2}$$

$$= \frac{1}{C_\psi} \int_0^\infty (f * \bar{\psi}_a * \psi_a)(t) \frac{da}{a^2},$$

where the integral over $b$ was recognized as a convolution. We will show the $L^2$ equality of $x(t)$ and $f(t)$ through the equality of their Fourier transforms. The Fourier transform of $x(t)$ is

$$X(\omega) = \frac{1}{C_\psi} \int_0^\infty \int_{-\infty}^\infty (f * \bar{\psi}_a * \psi_a)(t) e^{-j\omega t} \frac{da \, dt}{a^2}$$

$$\overset{(a)}{=} \frac{1}{C_\psi} \int_0^\infty F(\omega) \Psi^*_a(\omega) \Psi_a(\omega) \frac{da}{a^2}$$

$$\overset{(b)}{=} \frac{1}{C_\psi} F(\omega) \int_0^\infty a |\Psi(a\omega)|^2 \frac{da}{a^2}, \quad (11.8)$$

where (a) we integrated first over $t$, and transformed the two convolutions into products; and (b) we used (11.4). In the remaining integral above, apply a change of variable $\Omega = a\omega$ to compute:

$$\int_0^\infty |\Psi(a\omega)|^2 \frac{da}{a} = \int_0^\infty |\Psi(\Omega)|^2 \frac{d\Omega}{\Omega} = C_\psi,$$

which together with (11.8), shows that $X(\omega) = F(\omega)$. By Fourier inversion, we have proven that $x(t) = f(t)$ in the $L^2$ sense. \(\square\)

The formula (11.7) is also sometimes called the *resolution of the identity* and goes back to Calderon in the 1960’s in a context other than wavelets.

### 11.1.3 Energy Conservation

Closely related to the resolution of the identity is an energy conservation formula, an analogue to Parseval’s formula.

**Proposition 11.2** (Energy Conservation of the Continuous Wavelet Transform). Consider a function $f \in L^2(\mathbb{R})$ and its continuous wavelet transform $CWT_f(a,b)$
with respect to a real wavelet $\psi$ satisfying the admissibility condition (11.2). Then, the following energy conservation holds:

$$\int_{-\infty}^{\infty} |f(t)|^2 dt = \frac{1}{C_\psi} \int_{0}^{\infty} \int_{-\infty}^{\infty} |CWT_f(a,b)|^2 \frac{db \, da}{a^2}. \tag{11.10}$$

Proof. Expand the right hand side (without the leading constant) as

$$\int_{0}^{\infty} \int_{-\infty}^{\infty} |CWT_f(a,b)|^2 \frac{db \, da}{a^2} \overset{(a)}{=} \int_{0}^{\infty} \int_{-\infty}^{\infty} |(f \ast \bar{\psi}_a)(b)|^2 \frac{db \, da}{a^2}$$

$$\overset{(b)}{=} \int_{0}^{\infty} \frac{1}{2\pi} \int_{-\infty}^{\infty} |F(\omega)\sqrt{a}\Psi^*(a\omega)|^2 d\omega \frac{da}{a^2}$$

$$= \int_{0}^{\infty} \frac{1}{2\pi} \int_{-\infty}^{\infty} |F(\omega)|^2 |\Psi(a\omega)|^2 d\omega \frac{da}{a^2},$$

where (a) uses (11.5); and (b) uses Parseval’s relation 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

$$\int_{0}^{\infty} \int_{-\infty}^{\infty} |CWT_f(a,b)|^2 \frac{db \, da}{a^2} = \int_{-\infty}^{\infty} \frac{1}{2\pi} |F(\omega)|^2 \int_{0}^{\infty} |\Psi(a\omega)|^2 \frac{da}{a^2} d\omega \overset{(c)}{=} \int_{-\infty}^{\infty} \frac{1}{2\pi} |F(\omega)|^2 \int_{0}^{\infty} |\Psi(\Omega)|^2 \frac{d\Omega}{\Omega} d\omega.$$

Therefore

$$\frac{1}{C_\psi} \int_{0}^{\infty} \int_{-\infty}^{\infty} |CWT_f(a,b)|^2 \frac{db \, da}{a^2} = \frac{1}{2\pi} \int_{-\infty}^{\infty} |F(\omega)|^2 d\omega,$$

and applying Parseval’s formula to the right side proves (11.10). \qed

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.

### 11.1.4 Shift and Scale Invariance

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 $g(t) = f(t - \tau)$, or a delayed version of $f(t)$. Then

$$CWT_g(a,b) = \frac{1}{\sqrt{a}} \int_{-\infty}^{\infty} \psi \left( \frac{t-b}{a} \right) f(t-\tau) \, dt = \frac{1}{\sqrt{a}} \int_{-\infty}^{\infty} \psi \left( \frac{t' + \tau - b}{a} \right) f(t') \, dt'$$

$$= CWT_f(a,b-\tau) \tag{11.11}$$
by using the change of variables $t' = t - \tau$. That is, the CWT of $g$ is simply a delayed version of the CWT of $f$, as shown in Figure 11.3.

For the scaling property, consider a scaled and normalized version of $f(t)$,

$$g(t) = \frac{1}{\sqrt{s}} f\left(\frac{t}{s}\right),$$

where the renormalization ensures that $\|g\| = \|f\|$. Computing the CWT of $g$, using the change of variables $t' = t/s$, gives

$$\text{CWT}_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'\]

$$= \sqrt{\frac{s}{a}} \int_{-\infty}^{\infty} \psi\left(\frac{t' - b/s}{a/s}\right) f(t') dt' = \text{CWT}_f\left(\frac{a}{s}, \frac{b}{s}\right).$$

(11.12)

In words: if $g(t)$ is a version of $f(t)$ scaled by a factor $s$ and normalized to maintain its energy, then its CWT is a scaled by $s$ both in $a$ and $b$. A graphical representation of the scaling property is shown in Figure 11.4.

Consider now a function $f(t)$ with unit energy and having its CWT concentrated mostly in a unit square, say $[a_0, a_0 + 1] \times [b_0, b_0 + 1]$. The CWT 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 CWT now covers a
11.1. The Continuous Wavelet Transform

Surface increased by \( s^2 \). Therefore, when evaluating an energy measure in the CWT domain, we need to renormalize by a factor \( a^2 \), as was seen in both the inversion formula (11.7) and the energy conservation formula (11.10).

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.

11.1.5 Singularity Characterization

The continuous wavelet transform has an interesting localization property which is related to the fact that as \( a \to 0 \), the wavelet \( \psi_{a,b}(t) \) becomes arbitrarily narrow, performing a zoom in the vicinity of \( b \). This is easiest to see for \( f(t) = \delta(t - \tau) \).

Then

\[
CWT_f(a, b) = \frac{1}{\sqrt{a}} \int_{-\infty}^{\infty} \psi(t) \delta(t - \tau) \, dt = \frac{1}{\sqrt{a}} \delta(t) \left( \frac{\tau - b}{a} \right). \tag{11.13}
\]

This is the wavelet scaled by \( a \) and centered at \( b \). As \( a \to 0 \), the CWT 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. To start, consider the most elementary case, namely the Haar wavelet (with a single zero moment), and a signal \( f(t) \) being a Heaviside or step function at location \( \tau \),

\[
f(t) = \begin{cases} 
0, & \text{for } t < \tau; \\
1, & \text{for } t \geq \tau.
\end{cases}
\]

The idea is to relate this to the case studied above, using integration by parts.

We need the primitive of the wavelet, which we call \( \theta(t) \):

\[
\theta(t) = \int_{-\infty}^{t} \psi(\tau) \, d\tau.
\]

Because \( \psi(t) \) has at least one zero moment, \( \theta(t) \) is also compactly supported, namely on the same interval as \( \psi(t) \). For example, the primitive of the Haar wavelet is the hat function, or

\[
\theta(t) = \begin{cases} 
1/2 - |t|, & -1/2 \leq t \leq 1/2; \\
0, & \text{otherwise.}
\end{cases} \tag{11.14}
\]

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.

We also need the derivative of \( f(t) \). This derivative exists only in a generalized sense (using distributions) and can be shown to be a Dirac function at \( \tau \):

\[
f'(t) = \delta(t - \tau).
\]
Now, the continuous wavelet transform of a step function follows as

\[
\text{CWT}_f(a,b) = \int_{-\infty}^{\infty} \frac{1}{\sqrt{a}} \psi \left( \frac{t - b}{a} \right) f(t) \, dt \\
= (a) \left[ \sqrt{a} \theta \left( \frac{t - b}{a} \right) f'(t) \right]_{-\infty}^{\infty} - \int_{-\infty}^{\infty} \sqrt{a} \theta \left( \frac{t - b}{a} \right) f'(t) \, dt \\
= (b) - \int_{-\infty}^{\infty} \sqrt{a} \theta \left( \frac{t - b}{a} \right) \delta(t - \tau) \, dt \\
= (c) \sqrt{a} \theta \left( \frac{\tau - b}{a} \right),
\]

where (a) we used integration by parts; (b) the left part is zero because of compact support and we replaced \( f'(t) \) by a Dirac; and (c) we used the sifting property of the Dirac. Thus, as \( a \to 0 \), the CWT zooms towards the singularity and scales as \( a^{1/2} \), with a shape given by the primitive of the wavelet.

We consider one more example, namely the ramp function starting at \( \tau \):

\[
f(t) = \begin{cases} 
0, & t \leq \tau; \\
\tau, & t > \tau.
\end{cases}
\]

This function is continuous, but its derivative is not. Actually, its second derivative is a Dirac 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 \( \text{CWT}_f(a,b) \), we can apply integration by parts just like in (11.15) to obtain

\[
\text{CWT}_f(a,b) = - \int_{-\infty}^{\infty} \sqrt{a} \theta \left( \frac{t - b}{a} \right) f'(t) \, dt,
\]

where \( f'(t) \) is now a step function. We apply integration by parts one more time to get

\[
\text{CWT}_f(a,b) = - \left[ a^{3/2} \theta^{(1)} \left( \frac{t - b}{a} \right) f'(t) \right]_{-\infty}^{\infty} + a^{3/2} \int_{-\infty}^{\infty} \theta^{(1)} \left( \frac{t - b}{a} \right) f''(t) \, dt \\
= a^{3/2} \int_{-\infty}^{\infty} \theta^{(1)} \left( \frac{t - b}{a} \right) \delta(t - \tau) \, dt = a^{3/2} \theta^{(1)} \left( \frac{\tau - b}{a} \right),
\]

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 \( a \to 0 \), the CWT zooms towards the singularity and has a behavior of the order \( a^{3/2} \). These are examples of the following general result.

**Proposition 11.3 (Localization Property of the CWT).** Consider a wavelet \( \psi \) of compact support having \( N \) zero moments and a function \( f \) with a singularity of order \( n \leq N \) (meaning the \( n \)th derivative is a Dirac; for example, Dirac = 0, step
11.1. The Continuous Wavelet Transform

Figure 11.5: A signal with singularities of order 0, 1 and 2, and its continuous wavelet transform.

= 1, ramp = 2, etc.). Then, the continuous wavelet transform in the vicinity of the singularity at \( \tau \) is of the form

\[
CWT_f(a,b) = (-1)^n a^{n-1/2} \psi^{(n)} \left( \frac{\tau - b}{a} \right),
\]

(11.18)

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, \( f(t) \) has been differentiated \( n \) times, is thus a Dirac, 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 CWT also indicates the singularity type. Figure 11.5 sketches the CWT of a signal 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 \( f(t) \) is polynomial of order \( N - 1 \) or less, all inner products with the wavelet will be exactly zero due to the zero moment property. If the function \( f(t) \) is piecewise polynomial,\(^{60}\) 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. Let us calculate an example which makes the above explicit, while also pointing out what happens when the wavelet does not have enough zero moments.

Example 11.1 (Haar CWT of a piecewise polynomial function). Consider

\(^{60}\)That is, the function is a polynomial over intervals \( (t_i,t_{i+1}) \), with singularities at the interval boundaries.
a function \( f(t) \) defined as
\[
f(t) = \begin{cases} 
  t, & 1 \leq t < 2; \\
  1, & 2 \leq t < 3; \\
  0, & \text{otherwise},
\end{cases}
\]
and the Haar wavelet centered at the origin,
\[
\psi(t) = \begin{cases} 
  1, & -1/2 < t \leq 0; \\
  -1, & 0 < t \leq 1/2; \\
  0, & \text{otherwise},
\end{cases}
\]
as was shown in Figure 11.1(a). The function \( f(t) \) has 3 singularities, namely discontinuities at \( t = 1 \) and \( t = 3 \), and a discontinuous derivative at \( t = 2 \). The wavelet has 1 zero moment, so it will have zero inner product inside the interval \([2, 3]\), where \( f(t) \) is constant. What happens in the interval \([1, 2]\), where \( f(t) \) is linear?

Calculate the inner product at some shift \( \tau \in [1, 2] \) for a sufficiently small (so that the support of the shifted wavelet \([\tau - a/2, \tau + a/2]\) is contained in \([1, 2]\)):
\[
CWT_f(a, \tau) = \frac{1}{\sqrt{a}} \left( \int_{\tau-a/2}^{\tau} t \, dt - \int_{\tau}^{\tau+a/2} t \, dt \right) = \frac{1}{4} a^{3/2}. \quad (11.19)
\]
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 \( a \to 0 \).

To study qualitatively the overall behavior, apply the integration by part method once. The function \( f'(t) \) contains 2 Diracs \((\delta(t-1) - \delta(t-3))\) as well as the constant 1 on the interval \((1, 2)\). The primitive of the wavelet is \( \sqrt{a} \theta(t/a) \), where \( \theta(t) \) is the integral of the Haar wavelet, or the hat function in (11.14). Therefore, there are two cones of influence due to the Diracs, with an order \( a^{1/2} \) behavior around \( b = 1 \) and \( b = 3 \), a constant of order \( a^{3/2} \) in the \((1, 2)\) interval due to (11.19), which spills over into the \((2, 3)\) interval, and zero elsewhere. This is sketched in Figure 11.6.

### 11.1.6 Decay in Smooth Regions

Beyond polynomial and piecewise polynomial functions, let us consider more general smooth functions. Among the many possible classes 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.
11.1. The Continuous Wavelet Transform

Figure 11.6: A piecewise polynomial function and its continuous wavelet transform. (a) The function. (b) Its continuous wavelet transform.

For example, the primitive of the Haar wavelet is the hat function in (11.14), 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 as \( a \to 0 \) in that, for a continuous function \( f(t) \),

\[
\lim_{a \to 0} \int_{-\infty}^{\infty} \frac{1}{a} \psi^{(N)} \left( \frac{t - b}{a} \right) f(t) \, dt = Cf(b). \tag{11.20}
\]

Again, the Haar wavelet with its primitive is a typical example, since a limit of scaled hat functions is a classic way to obtain the Dirac function. We are now ready to prove the decay behavior of the CWT as \( a \to 0 \).

**Proposition 11.4 (Decay of the CWT for a Function in \( C^N \)).** Consider a compactly supported wavelet \( \psi \) with \( N \) zero moments, \( N \geq 1 \), and primitives \( \psi^{(1)}, \ldots, \psi^{(N)} \), where \( \int \psi^{(N)}(t) \, dt = C \). Given a function \( f \) having \( N \) continuous and bounded derivatives \( f^{(1)}, \ldots, f^{(N)} \), or \( f \in C^N \), then the continuous wavelet transform of \( f \) with respect to \( \psi \) behaves as

\[
|\text{CWT}_f(a,b)| \leq C' a^{N+1/2} \tag{11.21}
\]

for \( a \to 0 \).

**Proof.** (sketch)

The proof closely follows the method of integration by parts as used in Proposition 11.3. That is, we take the \( N \)th derivative of \( f(t) \), \( f^{(N)}(t) \), 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

$$CWT_f(a,b) = \int_{-\infty}^{\infty} \frac{1}{\sqrt{a}} \psi\left(\frac{t - b}{a}\right) f(t) \, dt$$

$$(a) \quad (-1)^N a^N \int_{-\infty}^{\infty} \psi^{(N)}\left(\frac{t - b}{a}\right) f^{(N)}(t) \, dt$$

$$(b) \quad (-1)^N a^{N+1/2} \int_{-\infty}^{\infty} \frac{1}{a} \psi^{(N)}\left(\frac{t - b}{a}\right) f^{(N)}(t) \, dt,$$

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 as $a \to 0$. Therefore, for small $a$, the integral above tends towards $Cf^{(N)}(b)$, which is finite, and the decay of the CWT is thus of order $a^{N+1/2}$.

While we used a global smoothness, it is clear that it is sufficient for $f(t)$ to be $C^N$ in the vicinity of $b$ for the decay to hold. The converse result, namely the necessary decay of the CWT for $f(t)$ to be in $C^N$, is a technical result which is more difficult to prove; it requires non-integer, Lipschitz, regularity. Note that if $f(t)$ 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.

### 11.1.7 Redundancy of the Continuous Wavelet Transform

The CWT 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 continuous wavelet transforms. We are thus interested in characterizing the image of one-dimensional functions in the CWT domain.

A simple analogue is in order. Consider an $M$ by $N$ matrix $T$ having orthonormal columns (i.e., $TT^T = I$) 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 $K$,

$$K = TT^T, \quad (11.22)$$

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

$$Kz = z. \quad (11.23)$$

Indeed, if $z$ is in $S$, then it can be written as $z = Tx$ for some $x$. Substituting this into the left side of (11.23) leads to

$$Kz = TT^Tx = Tx = z.$$
Conversely, if (11.23) holds then \( z = Kz = T^Tz = Tx \), 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 11.2 for a discussion of this, as well as the case of non-orthonormal columns in \( T \).

We now extend the test given in (11.23) to the case of the CWT. For this, let us introduce the reproducing kernel of the wavelet \( \psi \), defined as

\[
K(a_0, b_0, a, b) = \langle \psi_{a_0, b_0}, \psi_{a, b} \rangle.
\]

This is the cross correlation of two wavelets at scale and shifts \((a_0, b_0)\) and \((a, b)\), respectively, and is the equivalent of the matrix \( K \) in (11.22).

Call \( V \) the space of functions \( F(a, b) \) that are square integrable with respect to the measure \((db\, da)/a^2\) (see also Proposition 11.2). In this space, there exists a subspace \( S \) that corresponds to bona fide CWTs. Similarly to what we just did in finite dimensions, we give a test to check whether a function \( F(a, b) \) in \( V \) actually belongs to \( S \), that is, if it is the CWT of some one-dimensional function \( f(t) \).

**Proposition 11.5 (Reproducing kernel property of the CWT).** A function \( F(a, b) \) is the wavelet transform of a function \( f(t) \) if and only if it satisfies

\[
F(a_0, b_0) = \frac{1}{C_{\psi}} \int_0^{\infty} \int_{-\infty}^{\infty} K(a_0, b_0, a, b) F(a, b) \frac{db\, da}{a^2}.
\]  

(11.25)

**Proof.** We show that if \( F(a, b) \) is a CWT of some function \( f(t) \), then (11.25) holds. Completing the proof by showing that the converse is also true is left as Exercise 11.3.

By assumption,

\[
F(a_0, b_0) = \int_{-\infty}^{\infty} \psi_{a_0, b_0}(t) f(t) \, dt.
\]

Replace \( f(t) \) by its inversion formula (11.7), or

\[
F(a_0, b_0) = \int_{-\infty}^{\infty} \psi_{a_0, b_0}(t) \frac{1}{C_{\psi}} \int_0^{\infty} \int_{-\infty}^{\infty} \psi_{a, b}(t) F(a, b) \frac{db\, da}{a^2} \, dt
\]

\[
= \frac{1}{C_{\psi}} \int_0^{\infty} \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} \psi_{a_0, b_0}(t) \psi_{a, b}(t) F(a, b) \, dt \, \frac{db\, da}{a^2}
\]

\[
= \frac{1}{C_{\psi}} \int_0^{\infty} \int_{-\infty}^{\infty} K(a_0, b_0, a, b) F(a, b) \frac{db\, da}{a^2},
\]

where (a) we interchanged the order of integration; and (b) we integrated over \( t \) to get the reproducing kernel (11.24).

Figure 11.7 shows the reproducing kernel of the Haar wavelet, namely \( K(1, 0, a, b) \).
11.1.8 Examples

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 CWT. 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 CWT 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 (11.2) and having a sufficient number of zero moments will do. In the next subsection, scalograms will highlight differences between CWTs using different wavelets.

The 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

\[ \psi(t) = \frac{1}{\sqrt{2\pi}} e^{-j\omega_0 t} e^{-t^2/2}, \]  

(11.26)

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
11.1. The Continuous Wavelet Transform

Figure 11.8: Morlet wavelet. (a) Time domain function, with real and imaginary parts in solid and dotted lines, respectively. (b) Magnitude spectrum of the Fourier transform.

Figure 11.9: A signal and its scalogram. (a) Signal with various modes. (b) Scalogram with a Daubechies wavelet. (c) Scalogram with a symmetric wavelet. (d) Scalogram with a Morlet wavelet.

small (of order $10^{-7}$) and has numerically no consequence (and can be corrected by removing it from the wavelet). Figure 11.8 shows the Morlet wavelet in time and frequency domains.

11.1.9 Scalograms

So far, we have only sketched continuous wavelet transforms, to point out general behavior like localization and other relevant properties. For “real” signals, a usual way of displaying the CWT is to plot the magnitude $|CTW_{f}(a,b)|$ like an image. This is done in Figure 11.9 for a particular signal 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 11.9 (c) and (d)) have no drift across scales, which helps identify singularities. The zooming property at small scales is quite evident from the scalogram.
11.1.10 Remarks

The continuous-time 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 CWT. 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 CWT, as was seen in the proofs of Propositions 11.3 and 11.4. Together with the fact that all scales are considered, this shows that the CWT is a multiscale differential operator.

11.2 The Windowed Fourier Transform

11.2.1 Introduction and Definition

The idea of a localized Fourier transform is natural, and dates back to D. Gabor, in honor of whom it is sometimes called Gabor transform. If some frequency events appear and disappear (think of notes in music) then a local frequency analysis is necessary. Thus, the global Fourier transform has to be localized using a window. Then, this window is shifted over the function to perform a local Fourier analysis where the window is located. More formally, start with the usual Fourier transform which maps \( f(t) \) into \( F(\omega) \),

\[
F(\omega) = \int_{-\infty}^{\infty} f(t)e^{-j\omega t} \, dt = \langle e^{j\omega t}, f(t) \rangle.
\]

Introduce a window function \( g(t) \) that is symmetric \( g(t) = g(-t) \) and real. The window should be smooth, in particular, it should be smoother than the function to be analyzed.\(^6\) The windowed function, for a shift \( \tau \), is simply

\[
f_\tau(t) = g(t-\tau)f(t)
\]

and thus, the windowed Fourier transform at shift \( \tau \) is the Fourier transform of \( f_\tau(t) \),

\[
WFT_f(\omega, \tau) = \int_{-\infty}^{\infty} g(t-\tau)f(t)e^{-j\omega t} \, dt.
\]

From the construction, it is clear why it is called windowed Fourier transform, as shown in Figure 11.10, another common name (besides Gabor transform) is continuous short-time Fourier transform (STFT).

The expression (11.28) can be rewritten as an inner product by introducing

\[
g_{\omega,\tau}(t) = g(t-\tau)e^{j\omega t},
\]

leading to

\[
WFT_f(\omega, \tau) = \langle g_{\omega,\tau}(t), f(t) \rangle
\]

\(^6\) Otherwise, it will interfere with the smoothness of the function to be analyzed; see Section 11.2.6.
11.2. The Windowed Fourier Transform

Figure 11.10: Windowed Fourier transform. The window $g(t)$ is centered in $\tau$, and thus, the subsequent Fourier transform only “sees” the neighborhood around $\tau$. For simplicity, a triangular window is shown, in practice, smoother ones are used.

Figure 11.11: Time-frequency atom used in windowed Fourier transform. (a) Time-domain waveform $g_{\omega,\tau}(t)$. The window is triangular, and the real and imaginary parts of the complex modulated window are shown. (b) Schematic time-frequency footprint of $g_{\omega,\tau}(t)$.

where we used the fact that $g(t)$ is real. For the analysis of a function $f(t)$, the $WFT_f(\omega, \tau)$ uses time-frequency atoms $g_{\omega,\tau}(t)$ that are centered around $\omega$ and $\tau$, as shown schematically in Figure 11.11.

From the above discussion, two facts emerge clearly. First, the WFT is highly redundant, mapping a one-dimensional function $f(t)$ into a two-dimensional transform $WFT_f(\omega, \tau)$. This is reminiscent of the CWT, except that (scale, shift) is now (frequency, shift). Second, just like the underlying wavelet was key in the CWT, the choice of the window function $g(t)$ is critical in the WFT. The “classical” choice for $g(t)$ is the Gaussian bell, or

$$g(t) = \frac{1}{\sqrt{2\pi a}} e^{-t^2/(2a^2)} \quad (11.31)$$

where $a$ is a scale parameter and the normalization insures $\|g\| = 1$. The scale parameter is used to make the window of adequate size to “see” interesting features
Another classic choice is the sinc function, or
\[ g(t) = \sqrt{a} \frac{\sin(\pi t/a)}{\pi t}, \]  
that is, a perfect lowpass of bandwidth \([-\pi/a, \pi/a]\], with normalization such that \(\|g(t)\| = 1\). Again, the scale parameter allows to “tune” the frequency resolution of the WFT.

Other windows of choice include rectangular, triangular or higher order spline windows, as well as other classic windows from spectral analysis. An example is the Hanning or raised cosine window, defined as

\[
g(t) = \begin{cases} \sqrt{\frac{2}{3a}}(1 + \cos(2\pi t/a)) & t \in [-a/2, a/2] \\ 0 & \text{else} \end{cases}
\]

where \(a\) is a scale parameter and the normalization is chosen such that \(\|g\| = 1\).\(^{62}\)

Below, we derive the main properties of the WFT, including inversion and energy conservation, followed by basic characteristics like localization properties. This is followed by some specific examples, including spectrograms, which are displays of the magnitude squared of the WFT.

### 11.2.2 Inversion

Given a function \(f(t)\) and its windowed Fourier transform \(WFT_f(\omega, \tau)\), the first question that comes to mind is the inversion of the WFT. Given the redundancy present in the WFT, we expect such an inversion to be possible. This is given by the following proposition.

**Proposition 11.6 (Inversion of the WFT).** Consider a function \(f(t)\) in \(L^1 \cap L^2\) and its windowed Fourier transform \(WFT_f(\omega, \tau)\) with respect to a unit-norm window \(g(t)\).\(^{63}\) Then, \(f(t)\) can be recovered by the following inversion formula:

\[
f(t) = \frac{1}{2\pi} \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} WFT_f(\omega, \tau) g(\omega, \tau)(t) \, d\omega \, d\tau.
\]  

\[(11.33)\]

**Proof.** We are going to apply Parseval’s formula to (11.33), and for this we need the Fourier transform with respect to \(\tau\) for \(WFT_f(\omega, \tau)\). Note that

\[
WFT_f(\omega, \tau) = \int_{-\infty}^{\infty} g(t-\tau)f(t)e^{-j\omega t} \, dt = g * f_\omega(\tau),
\]

where we introduced \(f_\omega(t) = f(t)e^{-j\omega t}\) and used the fact that \(g(t) = g(-t)\). The Fourier transform of \(f_\omega(t)\) is \(F(\Omega + \omega)\) using the modulation property. Thus, the transform of \(WFT_f(\omega, \tau)\) with respect to \(\tau\) becomes, using the convolution property,

\[
WFT_f(\omega, \Omega) = F(\Omega + \omega)G(\Omega).
\]  

\[(11.34)\]

---

\(^{62}\)In the signal processing literature, the normalization factor is usually 1/2, such that \(g(0) = 1\).

\(^{63}\)The restriction to \(L^1 \cap L^2\) is not fundamental; it allows us to not worry about orders of integration thanks to Fubini.
11.2. The Windowed Fourier Transform

In (11.33), the other term involving $\tau$ comes from $g_{\omega,\tau}(t) = g(t-\tau)e^{j\omega t}$. The transform of $g(t-\tau)$ with respect to $\tau$ is

$$g(t-\tau) \xrightarrow{\text{CTFT}} e^{-j\Omega t}G(\Omega). \quad (11.35)$$

Consider now the right side of (11.33) and use Parseval’s formula:

$$\frac{1}{2\pi} \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} |WFT_f(\omega, \tau)g(t-\tau)e^{j\omega t} \, d\tau\, d\omega$$

$$= (a) \frac{1}{2\pi} \int_{-\infty}^{\infty} \left( \frac{1}{2\pi} \int_{-\infty}^{\infty} F(\Omega + \omega)G(\Omega)G^*(\Omega)e^{j\Omega t} \, d\Omega \right) \, d\omega$$

$$= (b) \frac{1}{2\pi} \int_{-\infty}^{\infty} |G(\Omega)|^2 \left( \frac{1}{2\pi} \int_{-\infty}^{\infty} F(\Omega + \omega)e^{j(\Omega + \omega) t} \, d\omega \right) \, d\Omega,$$

where (a) we used (11.34), (11.35) and Parseval; and (b) we changed integration order thanks to Fubini. The term inside the parentheses is recognized as the inverse Fourier transform of $F(\omega)$ and equals $f(t)$. Thus, since $\|g\| = 1$ implies $\frac{1}{2\pi} \int |G(\Omega)|^2 \, d\Omega = 1$, the expression above is

$$\frac{1}{2\pi} \int_{-\infty}^{\infty} |G(\Omega)|^2 f(t) \, d\Omega = f(t),$$

and (11.33) is proven. \qed

Note the similarity of the expression (11.33) with the resolution of the identity (11.7) for the CWT.

11.2.3 Energy Conservation

The next obvious question is that of energy conservation between the time domain and the WFT domain, or a Parseval-like formula.

**Proposition 11.7** (Energy Conservation of the WFT). Consider a function $f(t)$ in $L^1 \cap L^2$ and its windowed Fourier transform $WFT_f(\omega, \tau)$ with respect to a unit-norm window $g(t)$. Then

$$\|f\|^2 = \int_{-\infty}^{\infty} |f(t)|^2 \, dt = \frac{1}{2\pi} \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} |WFT_f(\omega, \tau)|^2 \, d\omega \, d\tau. \quad (11.36)$$

**Proof.** It should come as no surprise that to derive Parseval’s formula for the WFT, we use Parseval’s formula for the Fourier transform. Start with the right side of (11.36) to get

$$\frac{1}{2\pi} \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} |WFT_f(\omega, \tau)|^2 \, d\omega \, d\tau$$

$$= (a) \frac{1}{2\pi} \int_{-\infty}^{\infty} \left( \frac{1}{2\pi} \int_{-\infty}^{\infty} |F(\Omega + \omega)G(\Omega)|^2 \, d\Omega \right) \, d\omega$$

$$= (b) \frac{1}{2\pi} \int_{-\infty}^{\infty} |G(\Omega)|^2 \left( \frac{1}{2\pi} \int_{-\infty}^{\infty} |F(\Omega + \omega)|^2 \, d\omega \right) \, d\Omega,$$
where (a) we used Parseval and (11.34); and (b) we reversed the order of integration. Now, the integral over \( \omega \) in parentheses is simply \( \|f\|^2 \) and then the integral over \( \Omega \) is \( \|g\|^2 \) (both by Parseval). Since \( \|g\|^2 = 1 \) by assumption, (11.36) is proven. \( \square \)

### 11.2.4 Reproducing Kernel and Redundancy

The windowed Fourier transform, just like the continuous wavelet transform, maps a function of one variable into a two-dimensional function. It is thus highly redundant, and this redundancy is expressed by the reproducing kernel. This function is given by

\[
K(\omega_0, \tau_0, \omega, \tau) = \langle g_{\omega_0, \tau_0}(t), g_{\omega, \tau}(t) \rangle = \int_{-\infty}^{\infty} g(t - \tau_0)g(t - \tau)e^{j(\omega - \omega_0)t} \, dt. \tag{11.37}
\]

While this is a four-dimensional object, its magnitude depends only on the two differences \( \omega - \omega_0 \) and \( \tau - \tau_0 \). This is expressed in a closely related function called the \textit{ambiguity function}; see Exercise 11.4. Similarly to Proposition 11.5 we have:

**Proposition 11.8 (Reproducing Kernel Formula for the WFT).** A function \( F(\omega, \tau) \) is the WFT of some function \( f(t) \) if and only if it satisfies

\[
F(\omega_0, \tau_0) = \frac{1}{2\pi} \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} F(\omega, \tau)K(\omega_0, \tau_0, \omega, \tau) \, d\omega \, d\tau. \tag{11.38}
\]

**Proof.** If \( F(\omega, \tau) \) is a WFT, then there is a function \( f(t) \) such that \( F(\omega, \tau) = \text{WFT}_f(\omega, \tau) \), or

\[
F(\omega_0, \tau_0) = \int_{-\infty}^{\infty} g_{\omega_0, \tau_0}^*(t)f(t) \, dt.
\]

Now, write \( f(t) \) in terms of the inversion formula (11.33), to get

\[
F(\omega_0, \tau_0) = \frac{1}{2\pi} \int_{-\infty}^{\infty} g_{\omega_0, \tau_0}^*(t) \left( \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} F(\omega, \tau)g_{\omega, \tau}(t) \, d\omega \, d\tau \right) dt
\]

\[
= \frac{1}{2\pi} \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} F(\omega, \tau) \int_{-\infty}^{\infty} g_{\omega_0, \tau_0}(t)g_{\omega_0, \tau_0}(t) dt \, d\omega \, d\tau,
\]

where the integral over \( t \) leads to \( K(\omega_0, \tau_0, \omega, \tau) \), showing that \( F(\omega_0, \tau_0) \) indeed satisfies (11.38).

For the converse, write (11.38) by making \( K(\omega_0, \tau_0, \omega, \tau) \) explicit as an integral over \( t \) (see (11.37)):

\[
F(\omega_0, \tau_0) = \frac{1}{2\pi} \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} F(\omega, \tau)g_{\omega_0, \tau_0}(t)g_{\omega_0, \tau_0}(t) \, dt \, d\omega \, d\tau
\]

\[
= \int_{-\infty}^{\infty} g_{\omega_0, \tau_0}(t) \frac{1}{2\pi} \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} F(\omega, \tau)g_{\omega_0, \tau_0}(t) \, d\omega \, d\tau \, dt
\]

\[
= \int_{-\infty}^{\infty} g_{\omega_0, \tau_0}(t) f(t) \, dt,
\]
11.2. The Windowed Fourier Transform

where (a) we interchanged the order of integration; and (b) we used \( f(t) \) as the inversion of \( F(\omega, \tau) \). Therefore, \( F(\omega, \tau) \) is indeed a WFT, namely the WFT of \( f(t) \).

The redundancy present in the WFT allows sampling and interpolation, and the interpolation kernel depends on the reproducing kernel.

11.2.5 Shift and Modulation Properties

As is to be expected, the shift and modulation properties of the Fourier transform lead to equivalent relations for the windowed version. Consider a function \( h(t) = f(t - T) \). Its WFT is

\[
WFT_h(\omega, \tau) = \int_{-\infty}^{\infty} g(t - \tau) f(t - T)e^{-j\omega t} dt
\]

\[= e^{-j\omega T} \int_{-\infty}^{\infty} g(t' - (\tau - T)) f(t')e^{-j\omega t'} dt'\]

\[= e^{-j\omega T} WFT_f(\omega, \tau - T),
\]

where (a) we used \( t' = t - T \); and (b) the definition of the WFT. Thus, a shift by \( T \) simply shifts the WFT and adds a phase factor. The former is specific to the locality of the WFT, while the latter follows from the equivalent Fourier transform property. Similarly, consider a modulated version of \( f(t) \), or \( h(t) = e^{j\Omega t} f(t) \), then

\[
WFT_h(\omega, \tau) = \int_{-\infty}^{\infty} g(t - \tau) e^{j\Omega t} f(t)e^{-j\omega t} dt
\]

\[= \int_{-\infty}^{\infty} g(t - \tau) f(t)e^{-j(\omega - \Omega)t} dt = WFT_f(\tau, \omega - \Omega),
\]

which is the same property as for the Fourier transform.

11.2.6 Characterization of Singularities and Smoothness

To discuss singularities, recall that we are considering the Fourier transform of \( f(t) \), see (11.27). Since this involves a product between the function and the window, the Fourier transform involves the respective convolution. That is, singularities are “smoothed” by the window. Take a simple example, namely \( f(t) = \delta(t - T) \). Then

\[
WFT_f(\omega, \tau) = \int_{-\infty}^{\infty} g(t - \tau) \delta(t - T)e^{-j\omega t} dt
\]

\[= g(T - \tau)e^{-j\omega T}.
\]

This points to the localization of the WFT. An event at location \( T \) will spread around \( T \) according to the window function, and this at all frequencies. If \( g(t) \) has compact support \([-L/2, L/2]\), then \( WFT_f(\omega, \tau) \) has support \([-\infty, \infty] \times [T - T / 2, T + T / 2]\).
Figure 11.12: Localization properties of the WFT. (a) A Dirac pulse at $\tau$, with a compactly supported window $[-L/2, L/2]$. (b) A pure sinusoid at $\Omega$, with a window having a compactly supported Fourier transform $[-B/2, B/2]$. The dual situation, namely of a pure sinusoid $f(t) = e^{j\Omega t}$, leads to

$$WFT_f(\omega, \tau) = \int_{-\infty}^{\infty} g(t-\tau) e^{-j(\omega-\Omega)t} \, dt$$

(a) $= e^{-j(\omega-\Omega)\tau} \int_{-\infty}^{\infty} g(t') e^{-j(\omega-\Omega)t'} \, dt'$

(b) $= e^{-j(\omega-\Omega)\tau} G(\omega - \Omega)$, (11.41)

where (a) uses the change of variables $t' = t-\tau$; and (b) uses the Fourier transform of $g(t)$. Thus, the WFT spreads around $\Omega$ according to $G(\omega)$, and if the window $g(t)$ has a compactly supported Fourier transform $[-B/2, B/2]$ then $WFT_f(\omega, \tau)$ has support $[\Omega-B/2, \Omega+B/2] \times [-\infty, \infty]$. Both of the above situations are schematically displayed in Figure 11.12.

The important point is that if singularities appear together within a window, they appear mixed in the WFT domain. This is unlike the CWT and time-domain singularities, where arbitrary time resolution is possible for the scale factor going to 0.

If the window is smoother than the signals, then the type of singularity (assuming there is a single one inside the window) is determined by the decay of the Fourier transform.

Example 11.2 (Singularity Characterization of the WFT). Let us consider, as an illustrative example, a triangular window

$$g(t) = \begin{cases} \sqrt{3/2}(1-|t|), & |t| < 1; \\ 0, & \text{otherwise,} \end{cases}$$

which has a Fourier transform decaying as $|\omega|^{-2}$ for large $\omega$.

Consider a function $f(t)$ that is $C^1$ (continuous and with at least one continuous derivative) except for a discontinuity at $t = T$. If it were not for the discontinuity, the Fourier transform of $f(t)$ would decay faster than $|\omega|^{-2}$ (that is,
faster than \(|G(\omega)|\) does). But, because of the singularity at \(t = T\), \(|F(\omega)|\) decays only like \(|\omega|^{-1}\).

Now the locality of the WFT comes into play. There are two modes, given by the regularity of the windowed function \(f_\tau(t)\). When \(\tau\) is far from \(T\) (namely \(\tau < T - 1\) or \(\tau > T + 1\)), \(f_\tau(t)\) is continuous (but its derivative is not, because of the triangular window). In that region, \(|\text{WFT}_f(\omega, \tau)|\) will thus decay as \(|\omega|^{-2}\). But around the discontinuity (or \(\tau \in [T - 1, T + 1]\)), \(f_\tau(t)\) is discontinuous, and \(|\text{WFT}_f(\omega, \tau)|\) decays only as \(|\omega|^{-1}\).

The above example indicates that there is a subtle interplay between the smoothness and support of the window, and the singularities or smoothness of the analyzed function. This is formalized in the next two propositions.

**Proposition 11.9 (Singularity Characterization in the WFT domain).** Assume a window \(g(t)\) with compact support \([-L/2, L/2]\) and sufficient smoothness. Consider a function \(f(t)\) which is smooth except for a singularity of order \(n\) at \(t = T\), that is its \(n\)th derivative at \(T\) is a Dirac. Then its WFT decays as

\[
|\text{WFT}_f(\omega, \tau)| \sim \frac{1}{1 + |\omega|^2}
\]

in the region \(\tau \in [T - L/2, T + L/2]\).

The proof follows by using the decay property of the Fourier transform applied to the windowed function and is left as Exercise 11.5.

Conversely, a sufficiently decaying WFT indicates a smooth function in the region of interest.

**Proposition 11.10 (Smoothness from Decay of the WFT).** Consider a sufficiently smooth window \(g(t)\) of compact support \([-L/2, L/2]\). If the WFT at \(T\) decays sufficiently fast, or for some \(C\) and \(\epsilon > 0\),

\[
|\text{WFT}_f(\omega, T)| \leq \frac{C}{1 + |\omega|^{p+1+\epsilon}}
\]

then \(f(t)\) is \(C^p\) on the interval \([T - L/2, T + L/2]\).

### 11.2.7 Spectrograms

The standard way to display the WFT is to show its magnitude \(|\text{WFT}_f(\omega, \tau)|\) like an image. This is called the spectrogram and is very popular, for example, for speech and music signals. Figure 11.13 shows a standard signal with various modes and two spectrograms.

As can be seen, the sinusoid is picked out, and the singularities are identified but not exactly localized due to the size of the window. For the short window (Figure 11.13(b)), the various singularities are still isolated, but the sinusoid is not well localized. The reverse is true for the long window (Figure 11.13(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.
11.2.8 Remarks

It is interesting to note that the Morlet wavelet and the Gabor function are related. From (11.26) 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 (11.29) and (11.31), 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 CWT (with a Morlet wavelet) and a WFT (with a Gabor function) coincide.

11.3 Sampling and Computing the CWT and WFT

11.3.1 Sampling Grids

11.3.2 Frames from Sampled CWT and WFT

11.3.3 Fast computation of Sampled CWT

11.3.4 Fast computation of Sampled WFT

11.4 Discussion and Summary
Exercises with Solutions

11.1. TBD

Exercises

11.1. Admissibility of Daubechies wavelets:
Show that all orthonormal and compactly supported wavelets from the Daubechies family satisfy the admissibility condition (11.1).

11.2. Finite dimensional reproducing kernels:
Consider an \( M \times N \) matrix \( T \), \( M > N \), that maps vectors \( x \) from \( \mathbb{R}^N \) into vectors \( y \) living on a subspace \( S \) of \( \mathbb{R}^M \).

(i) For \( T \) having orthonormal columns, or \( TT^T = I \), and \( K = TT^T \) (see (11.22)), what can you say about the vector \( \hat{y} = Ky \), where \( y \) is an arbitrary vector from \( \mathbb{R}^M \)?

(ii) For \( T \) having \( N \) linearly independent (but not necessary orthonormal) columns, give a simple test to check whether a vector \( y \) in \( \mathbb{R}^M \) belongs to \( S \) (see (11.23)).

(iii) In case (ii) above, indicate how to compute the orthogonal projection of an arbitrary vector \( y \) in \( \mathbb{R}^M \) onto \( S \).

11.3. Reproducing kernel formula for the CWT:
Show the converse part of Proposition 11.5. That is, show that if a function \( F(a, b) \) satisfies (11.25), then there exists a function \( f(t) \) with CWT equal to \( F(a, b) \).

11.4. Ambiguity function and reproducing kernel of the WFT:

11.5. Smoothness and decay of the WFT:
Using the appropriate relations from the Fourier transform, prove Propositions 11.9 and 11.10. Make sure the smoothness of the window function is properly taken into account.
Chapter 11. Continuous Wavelet and Windowed Fourier Transforms
Chapter 12

Approximation, Estimation, and Compression

Contents

12.1 Introduction ................................................. 419
12.2 Modeling .................................................. 419
12.3 Approximation ............................................. 420
12.4 Estimation .................................................. 420
12.5 Denoising .................................................. 420
12.6 Compression ............................................... 420
12.7 Algorithms .................................................. 420
Chapter at a Glance .............................................. 420
Historical Remarks .............................................. 421
Further Reading ............................................... 421
Exercises with Solutions ....................................... 421
Exercises ......................................................... 421

12.1 Introduction

- from bases and signals to applications
- need for models
- need for approximation, estimation and compression
- the combinatoric FCWS (see Notes)

Chapter Outline

12.2 Modeling

* Modeling for approximation and compression is different than modeling for estimation/denoising
- ℓ₀, ℓᵖ, 0 < p < 1, etc.
Chapter 12. Approximation, Estimation, and Compression

- Besov, (Sobolev?)
- Others (BV)
- Stochastic modeling

12.3 Approximation
- fixed subspaces
- adaptive subspaces
- deterministic
- stochastic
- best bases

12.4 Estimation
- signal models
- noise models
- linear estimation
- adaptive estimation

12.5 Denoising
- Wiener
- wavelet-based
- frames

12.6 Compression
- from coefficients to bits
- the rate-distortion primer
- transform coding fundamentals
- Fourier-based coding
- wavelet-based coding
- best basis algorithm with Lagrange

12.7 Algorithms
- MP, OMP
- BP

Chapter at a Glance

TBD
Chapter 12. Approximation, Estimation, and Compression
Bibliography


Bibliography


The bibliography contains a list of academic references, each formatted with the following structure: 


[59] D. L. Donoho and X. Huo. Uncertainty principles and ideal atomic decom-


directional wavelet bases and frames for image compression and denoising. In

[62] E. Dubois. The sampling and reconstruction of time-varying imagery with


[67] Y. Eldar and G. D. Forney, Jr. Optimal tight frames and quantum measure-

[68] D. Esteban and C. Galand. Applications of quadrature mirror filters to split
band voice coding schemes. In *Proc. IEEE Int. Conf. Acoust., Speech and


[70] A. K. Fletcher, S. Rangan, V. K Goyal, and K. Ramchandran. Denois-
ing by sparse approximation: Error bounds based on rate-distortion theory.


[73] C. Gasquet and P. Witomski. *Analyse de Fourier et Applications*. Masson,

[74] I. Gohberg and S. Goldberg. *Basic Operator Theory*. Birkhäuser, Boston,
MA, 1981.

[75] V. K Goyal, J. Kovačević, and J. A. Kelner. Quantized frame expansions with


<table>
<thead>
<tr>
<th>Reference Number</th>
<th>Author(s)</th>
<th>Title</th>
<th>Publisher, Location</th>
<th>Edition</th>
<th>Year</th>
</tr>
</thead>
<tbody>
<tr>
<td>130</td>
<td>H. L. Royden</td>
<td>Real Analysis</td>
<td>MacMillan, New York</td>
<td></td>
<td>1968</td>
</tr>
<tr>
<td>132</td>
<td><a href="http://www.cns.nyu.edu/~eero/steerpyr/">http://www.cns.nyu.edu/~eero/steerpyr/</a></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>133</td>
<td>I. W. Selesnick</td>
<td>Wavelets in Signal and Image Analysis, chapter The double density DWT</td>
<td>Kluwer, 2001</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>


Index

$C^p([a,b])$, 26
- inner product, 26
- norm, 26
$\mathbb{C}^N$, 17, 23, 28, 53, 66–68
- basis, 39, 75
- inner product, 20, 21, 23, 33, 46, 55
- norm, 21–23, 46, 66–68, 74
$\ell^p(\mathbb{Z})$, 24, 66–68, 74
- completeness, 28
- norm, 24
$\ell^2(\mathbb{Z})$, 25, 28, 75
- basis, 38
- inner product, 25
- linear operator on, 32, 33
- norm, 25
- projection operator on, 37
$L^p(\mathbb{R})$, 25
- norm, 25
$L^2(\mathbb{R})$, 26, 28
- basis, 38
- inner product, 26
- linear operator on, 32, 33
- norm, 26
- projection operator on, 37, 38
$\mathbb{R}^N$, 17, 20, 23, 28, 53, 55
- basis, 39, 75
- inner product, 20, 21, 23, 46, 55
- norm, 21–23, 46
$\mathbb{R}^2$, 10, 14, 70, 76
- basis, 12, 15, 47, 48
- frame, 13
- inner product, 10–13
- norm, 10
$z$-transform, $x_i$

Absolute convergence, see Convergence, absolute
Adjugate matrix, see Matrix, adjugate

Algorithms
- Gaussian elimination, 46, 70–71
- Gram-Schmidt orthogonalization, 46–48

Antidiagonal matrix, see Matrix, antidiagonal

Approximation, 4
- in a Fourier basis, 4
- in a Haar basis, 4
- linear, 4
- nonlinear, 4
- quality, 4

Argument, see Real analysis, argument

Banach space, ix, 27
Band matrix, see Matrix, band
Basis, ix, 12, 19, 70
- biorthogonal, see Basis, general
dimension of, 19
dual, 13, 64
for $\mathbb{C}^N$, 39, 75
for $L^2(\mathbb{R})$, 38
for $\ell^2(\mathbb{Z})$, 38
for $\mathbb{R}^N$, 39, 75
for $\mathbb{R}^2$, 12, 15, 47, 48
Fourier series, 1
general, 13, 15, 44, 45, 64
Haar, 2
in a Hilbert space, 28, 38
linear (in)dependence, 64
matrix view, 15, 16, 44, 64
orthogonal, 13, 64
<table>
<thead>
<tr>
<th>Term</th>
<th>Page Range</th>
</tr>
</thead>
<tbody>
<tr>
<td>orthonormal</td>
<td>1, 12, 13, 15, 16, 38, 39, 43, 45, 53, 64, 75</td>
</tr>
<tr>
<td>Riesz</td>
<td>42</td>
</tr>
<tr>
<td>self dual, see Basis, orthonormal</td>
<td>39</td>
</tr>
<tr>
<td>standard, ix, 10–12</td>
<td></td>
</tr>
<tr>
<td>Bessel's inequality, 12, 39</td>
<td></td>
</tr>
<tr>
<td>Bijection, see Real analysis, bijection</td>
<td></td>
</tr>
<tr>
<td>Biorthogonal bases, basis, see Basis,</td>
<td></td>
</tr>
<tr>
<td>general</td>
<td></td>
</tr>
<tr>
<td>Block Toeplitz matrix, see Matrix, Toeplitz</td>
<td></td>
</tr>
<tr>
<td>Cauchy sequence, 27, 35, 74, 75</td>
<td></td>
</tr>
<tr>
<td>Cauchy-Schwarz inequality, see Inner</td>
<td></td>
</tr>
<tr>
<td>product space, Cauchy Schwarz inequality</td>
<td></td>
</tr>
<tr>
<td>Characteristic polynomial, see Matrix,</td>
<td></td>
</tr>
<tr>
<td>characteristic polynomial</td>
<td></td>
</tr>
<tr>
<td>Circulant matrix, see Matrix, circulant</td>
<td></td>
</tr>
<tr>
<td>Closed set, see Set, closed</td>
<td></td>
</tr>
<tr>
<td>Closure of a set, see Set, closure</td>
<td></td>
</tr>
<tr>
<td>Codomain, see Real analysis, codomain</td>
<td></td>
</tr>
<tr>
<td>Coefficients</td>
<td></td>
</tr>
<tr>
<td>expansion, 12, 14, 39, 41, 64</td>
<td></td>
</tr>
<tr>
<td>Fourier, 39, 64</td>
<td></td>
</tr>
<tr>
<td>Gabor, 64</td>
<td></td>
</tr>
<tr>
<td>generalized Fourier, 39</td>
<td></td>
</tr>
<tr>
<td>matrix, 62</td>
<td></td>
</tr>
<tr>
<td>polynomial, 18</td>
<td></td>
</tr>
<tr>
<td>transform, 39, 64</td>
<td></td>
</tr>
<tr>
<td>wavelet, 64</td>
<td></td>
</tr>
<tr>
<td>Complete inner product space, see Inner</td>
<td></td>
</tr>
<tr>
<td>product space, completeness</td>
<td></td>
</tr>
<tr>
<td>Completeness, see Hilbert space,</td>
<td></td>
</tr>
<tr>
<td>completeness</td>
<td></td>
</tr>
<tr>
<td>Composition, see Real analysis,</td>
<td></td>
</tr>
<tr>
<td>composition of functions</td>
<td></td>
</tr>
<tr>
<td>Conditional convergence, see Convergence</td>
<td></td>
</tr>
<tr>
<td>convergence, conditional</td>
<td></td>
</tr>
<tr>
<td>Convergence, 19, 26, 27</td>
<td></td>
</tr>
<tr>
<td>absolute, 49</td>
<td></td>
</tr>
<tr>
<td>conditional, 49</td>
<td></td>
</tr>
<tr>
<td>in normed vector spaces, 23, 27</td>
<td></td>
</tr>
<tr>
<td>of scalar series, 23, 24, 49</td>
<td></td>
</tr>
<tr>
<td>pointwise, 50</td>
<td></td>
</tr>
<tr>
<td>tests, 50, 76</td>
<td></td>
</tr>
<tr>
<td>uniform, 50</td>
<td></td>
</tr>
<tr>
<td>Cramer's formula, see Matrix, Cramer's formula</td>
<td></td>
</tr>
<tr>
<td>CTFT, see Fourier, continuous-time</td>
<td></td>
</tr>
<tr>
<td>Fourier transform</td>
<td></td>
</tr>
<tr>
<td>Decomposition, 1, 64</td>
<td></td>
</tr>
<tr>
<td>Determinant, see Matrix, determinant</td>
<td></td>
</tr>
<tr>
<td>DFT, see Fourier, discrete Fourier</td>
<td></td>
</tr>
<tr>
<td>transform</td>
<td></td>
</tr>
<tr>
<td>Diagonal matrix, see Matrix, diagonal</td>
<td></td>
</tr>
<tr>
<td>Dirac</td>
<td></td>
</tr>
<tr>
<td>function, ix, 50</td>
<td></td>
</tr>
<tr>
<td>Paul, see People, Dirac, Paul</td>
<td></td>
</tr>
<tr>
<td>sequence, ix, 50</td>
<td></td>
</tr>
<tr>
<td>Discrete metric, 74</td>
<td></td>
</tr>
<tr>
<td>Distance, see Normed vector space,</td>
<td></td>
</tr>
<tr>
<td>distance</td>
<td></td>
</tr>
<tr>
<td>Domain, see Real analysis, domain</td>
<td></td>
</tr>
<tr>
<td>Dot product, see Vector space, inner</td>
<td></td>
</tr>
<tr>
<td>product</td>
<td></td>
</tr>
<tr>
<td>DTFT, see Fourier, discrete-time Fourier transform</td>
<td></td>
</tr>
<tr>
<td>Eigensequence(s), see Linear operator,</td>
<td></td>
</tr>
<tr>
<td>eigenvectors</td>
<td></td>
</tr>
<tr>
<td>Eigenvalues, see Matrix, eigenvalues</td>
<td></td>
</tr>
<tr>
<td>Eigenvector(s), see Linear operator,</td>
<td></td>
</tr>
<tr>
<td>eigenvectors</td>
<td></td>
</tr>
<tr>
<td>Eigenvectors, see Matrix, eigenvectors</td>
<td></td>
</tr>
<tr>
<td>Ess sup, essential supremum, see Real</td>
<td></td>
</tr>
<tr>
<td>analysis, essential supremum</td>
<td></td>
</tr>
<tr>
<td>Euclid, see People, Euclid</td>
<td></td>
</tr>
<tr>
<td>Euclidean</td>
<td></td>
</tr>
<tr>
<td>basis, see Basis, standard</td>
<td></td>
</tr>
<tr>
<td>geometry, 30</td>
<td></td>
</tr>
<tr>
<td>norm, see Norm, Euclidean space, 9, 17</td>
<td></td>
</tr>
<tr>
<td>Expansion</td>
<td></td>
</tr>
<tr>
<td>coefficients, see Coefficients,</td>
<td></td>
</tr>
<tr>
<td>expansion</td>
<td></td>
</tr>
<tr>
<td>nonredundant, see Basis</td>
<td></td>
</tr>
<tr>
<td>redundant, see Frame</td>
<td></td>
</tr>
</tbody>
</table>

Filter banks
### Index

**Infinite matrix, see Matrix, infinite**

**Injection, see Real analysis, injection**

**Inner product, 19, 26, 28, 29, 34**
- for $C^N$, 20, 21, 23, 33, 46, 55, 73
- for $L^p(\mathbb{R})$, 25
- for $\ell^p(\mathbb{Z})$, 26
- for $\mathbb{R}^N$, 20, 21, 23, 46, 55
- for $\mathbb{R}^2$, 10–13
- for $C^p([a,b])$, 26

**Inner product space, 20, 21, 26–28, 30, 35**

**Cauchy-Schwarz inequality, 21, 74**

**Complete (Hilbert), 27**

**Parallelogram law, 22, 35, 74**

**Pythagorean theorem, 30**

**Inverse of a matrix, see Matrix, inverse**

**Kernel, see Linear operator, kernel**

**Kronecker delta, ix**

**Lagrange interpolation, see Polynomial(s), Lagrange interpolation**

**Least squares solution, 56, 69, 78**

**Legendre polynomials, see Polynomials, Legendre**

**Length, see Vector space, norm**

**Linear (in)dependence, linearly (in)dependent, see Vector space, linear (in)dependence**

**Linear algebra, 53**
- eigenvalues, 31
- eigenvectors, 31
- Gaussian elimination, 46, 56, 70–71
- Hermitian transpose, 54
- kernel, 30, 56
- linear system of equations, 56, 78
- normal equations, 57
- normal rank, 62
- range, 30, 34, 37, 56, 59
- singular value decomposition, 60
- singular values, 60, 75

**Linear operator**
- bounded, 31, 32, 33, 75
- eigenvalues, 31
- eigenvectors, 31
- kernel, 30, 56
- range, 30, 34, 37, 56, 59

**Linear system of equations, see Linear algebra, linear system of equations**

**Manhattan norm, see Norm, 1-norm**

**Map, see Real analysis, map**

**Matrix**
- adjugate, 56
- antidiagonal, 54
- band, 61
- characteristic polynomial, 58
- circulant, 61
- column vector, 54
- Cramer’s formula, 56
- determinant, 16, 55, 72–73
- diagonal, 54, 60
- diagonalization, 58
- eigenvalues, 58, 72–73, 75, 77
- eigenvectors, 58, 72–73, 77
- from a linear operator, 53
- Hermitian, 33, 59, 60
- identity, 54
- infinite, 53
- inner product, 55
- inverse, 56, 72–73
- least squares solution, 56, 69, 78
- left/right inverse, 56
- minor, 55
- multiplication, 30, 31, 55
- normal, 59, 75
- outer product, 55
- polynomial, 62
- positive definite, 60
- power, 77
- rank, 45, 56, 62, 64
- rational, 62
- rectangular, 54, 56, 64
- reflection, 16, 63
- rotation, 16, 63
<table>
<thead>
<tr>
<th><strong>Index</strong></th>
<th>439</th>
</tr>
</thead>
<tbody>
<tr>
<td>row vector, 54</td>
<td>Normal operator, see Operator, normal or Matrix, normal</td>
</tr>
<tr>
<td>scalar, 54</td>
<td>Normal rank, see Matrix, rank</td>
</tr>
<tr>
<td>self-adjoint, see Matrix, Hermitian</td>
<td>Normed vector space, 21, 27, 28</td>
</tr>
<tr>
<td>singular, 44, 55</td>
<td>Cauchy sequence, 27, 35, 74, 75</td>
</tr>
<tr>
<td>square, 54, 64</td>
<td>convergence in, 23, 27, 49</td>
</tr>
<tr>
<td>submatrix, 55</td>
<td>distance, 11, 22</td>
</tr>
<tr>
<td>symmetric, 57</td>
<td>Numbers</td>
</tr>
<tr>
<td>Toeplitz, 61</td>
<td>complex, ix</td>
</tr>
<tr>
<td>triangular, 54</td>
<td>integer, ix</td>
</tr>
<tr>
<td>unimodular, 62</td>
<td>natural, ix</td>
</tr>
<tr>
<td>unitary, 16, 62, 64</td>
<td>real, ix</td>
</tr>
<tr>
<td>MB frame, see Frame, Mercedes-Benz</td>
<td>ONB, see Basis, orthonormal</td>
</tr>
<tr>
<td>Mercedes-Benz frame, see Frame, Mercedes-Benz</td>
<td>Operator</td>
</tr>
<tr>
<td>Minkowski inequality, see Vector space, Minkowski inequality</td>
<td>adjoint, 32, 75</td>
</tr>
<tr>
<td>Norm, 10, 15, 21, 24–26, 28, 29, 31, 34, 35, 38, 44, 63, 64</td>
<td>bounded, 31, 32</td>
</tr>
<tr>
<td>1-norm, 22</td>
<td>bounded linear, 31, 32, 33, 53, 75</td>
</tr>
<tr>
<td>2-norm, see Norm, Euclidean</td>
<td>Hermitian, 33, 34, 75</td>
</tr>
<tr>
<td>$L^\infty$-norm, 25</td>
<td>idempotent, 34</td>
</tr>
<tr>
<td>$L^p$-norm, 25</td>
<td>inverse, 31</td>
</tr>
<tr>
<td>$L^2$-norm, 38, 75</td>
<td>invertible, 31</td>
</tr>
<tr>
<td>$\infty$-norm, 22, 74</td>
<td>linear, 26, 30, 31, 32, 45, 53, 56, 60</td>
</tr>
<tr>
<td>$\ell^\infty$-norm, 24</td>
<td>norm, 31, 75</td>
</tr>
<tr>
<td>$\ell^p$-norm, 24</td>
<td>normal, 75</td>
</tr>
<tr>
<td>$p$-norm, ix, 22, 74</td>
<td>projection, 26, 34, 36, 37</td>
</tr>
<tr>
<td>equivalence, 74</td>
<td>self-adjoint, see Operator, Hermitian</td>
</tr>
<tr>
<td>Euclidean, 21</td>
<td>Orthogonal complement, see Orthogonality, orthogonal, complement</td>
</tr>
<tr>
<td>for $\mathbb{C}^N$, 21, 23, 46, 66–68, 74</td>
<td>Orthogonality, orthogonal, 19</td>
</tr>
<tr>
<td>for $\mathbb{R}^N$, 21, 23, 46, 74</td>
<td>basis, 13, 42, 64</td>
</tr>
<tr>
<td>for $C^p([a,b])$, 26</td>
<td>complement, 30</td>
</tr>
<tr>
<td>induced, 21, 23</td>
<td>eigenvectors, 59, 63</td>
</tr>
<tr>
<td>Manhattan, see Norm, 1-norm</td>
<td>expansion, 42</td>
</tr>
<tr>
<td>nonnegativity, 21, 35</td>
<td>Gram-Schmidt orthogonalization, 46–48, 63, 71–72, 76</td>
</tr>
<tr>
<td>normed vector space, 21, 22, 23, 27, 28, 49</td>
<td>matrix, 63</td>
</tr>
<tr>
<td>operator, 31, 75</td>
<td>projection, 11, 12, 26, 34, 35–37, 39, 40, 46, 57, 58, 71–72</td>
</tr>
<tr>
<td>preservation of, 15, 43, 63, 64, 70</td>
<td>relations, 38, 64</td>
</tr>
<tr>
<td>Taxicab, see Norm, 1-norm</td>
<td>set(s), 30</td>
</tr>
<tr>
<td>unit, 22, 23, 34, 38, 43, 46, 59</td>
<td>successive approximation, 41, 64</td>
</tr>
<tr>
<td>Normal equations, see Linear algebra, normal equations</td>
<td>vector to a set, 30</td>
</tr>
<tr>
<td>Normal matrix, see Matrix, normal</td>
<td></td>
</tr>
</tbody>
</table>

**The World of Fourier and Wavelets**

Copyright 2008 M. Vetterli, J. Kovacevic, and V. K. Goyal

v3.0 [April 15, 2008] Do not distribute

Comments to book-errata@FourierAndWavelets.org
vectors, 11, 13, 29, 38
Orthonormal basis, see Basis, orthonormal
Orthonormality, orthonormal
basis, 12, 13, 16, 38, 39, 43, 45, 53, 64, 75
Bessel's inequality, 39
generalized Parseval's equality, 40
matrix, 63
Parseval's equality, 12, 39, 40, 75
relations, 38
set, 30, 39, 46, 71–72, 75
vectors, 30, 71–72

Parallelogram law, see Inner product
space, parallelogram law, 35
Parseval tight frame, see Frame, Parseval tight
Parseval's
equality, 12, 39, 40, 75
generalized equality, 40
relations, 42, 63, 75
tight frame, 43
Partial fraction expansion, see Polynomial(s), partial fraction expansion

People
Dirac, Paul, ix
Euclid, 65
Fourier, Joseph, 1
Haar, Alfred, 2
Hilbert, David, 65

Perpendicularity, perpendicular, see Orthogonality, orthogonal
Planar rotation, see Matrix, rotation
Pointwise convergence, see Convergence, pointwise
Polynomial(s)
definition, 52
degree of, 52
fundamental theorem of algebra, 52
interpolation, 52
irreducible, 52
Lagrange interpolation, 53
Legendre, 47, 76

matrix, see Matrix, polynomial
Neville's algorithm, 53
partial fraction expansion, 53
quadratic, 52

Positive definite matrix, see Matrix, positive definite

Probability theory
texts on, 9

Projection
oblique, 34
operator, 34–37
orthogonal, 11, 12, 19, 26, 34, 35–37, 39, 40, 46, 57, 58, 69, 71–72
theorem, 34, 35, 36

PTF, see Frame, Parseval tight

Pythagorean theorem, 30

Range, see Linear operator, range and
Real analysis, range

Rank, see Matrix, rank

Rational matrix, see Matrix, rational

Real analysis
argument, 49
bijection, 49
codomain, 49
composition of functions, 49
domain, 49
essential supremum, 25, 49
function, 49
graph, 49
infimum, 24, 49
injection, 49
map, 49
range, 49
supremum, 24, 49
surjection, 49

Reconstruction, 64

Reflection matrix, see Matrix, reflection

Representation, see Expansion

Riesz bases, see Basis, Riesz

Rotation matrix, see Matrix, rotation

Scalar matrix, see Matrix, scalar
Index

Scalar product, see Vector space, inner product
Self-adjoint matrix, see Matrix, Hermitian
Self-adjoint operator, see Operator, Hermitian
Separability, separable Hilbert space, see Hilbert space, separable
Sequences, ix
Series
  absolute convergence, 49
  conditional convergence, 49
  convergence, 49
  geometric, 77
  infinite, 49
  MacLaurin, 77
  partial sum, 49
  power, 77
  scalar, 49
  Taylor, 77
Set
  closed, 49
  closure, ix, 49
Signals and systems
texts on, 9
Sinc
  function, 51
  sequence, 51
Singular matrix, see Matrix, singular
Singular value decomposition, see Linear algebra, singular value decomposition
Singular values, see Linear algebra, singular values
Space
  of finite-energy functions, see $L^2(\mathbb{R})$
  of finite-energy sequences, see $\ell^2(\mathbb{Z})$
  of square-integrable functions, see $L^2(\mathbb{R})$
  of square-summable sequences, see $\ell^2(\mathbb{Z})$
Span, see Vector space, span
Square-integrable functions, see $L^2(\mathbb{R})$
Square-summable sequences, see $\ell^2(\mathbb{Z})$
Submatrix, see Matrix, submatrix
Subspace, see Vector space, subspace
Successive approximation, see Orthogonality, orthogonal, successive approximation
Sup, supremum, see Real analysis, supremum
Surjection, see Real analysis, surjection
SVD, see Linear algebra, singular value decomposition
Symmetric matrix, see Matrix, symmetric
Taxicab norm, see Norm, 1-norm
TF, see Frame, tight
Tight frame, see Frame, tight, see Frame, tight
Toeplitz matrix, see Matrix, Toeplitz Transform
  coefficients, 39, 64
  domain, 64
Triangle inequality, see Vector space, triangle inequality
Triangular (upper/lower) matrix, see Matrix, triangular
Uniform convergence, see Convergence, uniform
Unimodular matrix, see Matrix, unimodular
Unitary matrix, see Matrix, unitary
Vector space, ix, 10, 17
  Banach space, 27
  distance, 12, 19, 69, 74
  Hilbert space, 15, 26–30, 35, 36, 38, 40, 43, 45, 53
  inner product, 10–13, 19, 21–23, 25, 26, 28, 29, 33, 34, 46, 55, 73
  inner product space, 20, 21, 26–28, 30, 35
  linear (in)dependence, 19
  linear (in)dependence, 12, 15, 19, 38, 43, 46, 56–59, 64, 69, 77
  Minkowski inequality, 66, 67, 68
  norm, 21, 22–26, 28, 29, 31, 34, 35, 38, 44, 46, 63, 64, 74
normed vector space, 17, 21, 22, 23, 27, 28, 49
orthogonality, 11–13, 19, 26, 29, 34–36, 38–40, 46
properties
  additive identity, 17
  additive inverse, 17
  associativity, 17
  commutativity, 17
  distributivity, 17
  multiplicative identity, 17
  scalar multiplication, 17
  vector addition, 17
span, 12, 14, 18, 19, 40, 41, 46–48, 56–58, 69, 71–72
subspace, 11, 18, 28
texts on, 9
triangle inequality, 21, 74

Wavelet
  coefficients, 64
  theory, 9