Смоленцев Н. К.

Основы теории вейвлетов
Вейвлеты в MATLAB

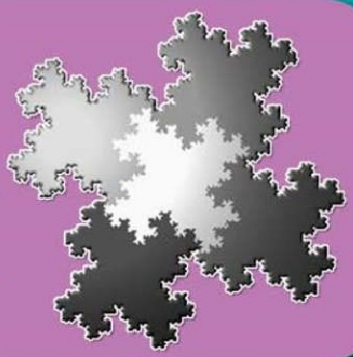
Издание 4-ое, дополненное и переработанное

Преобразование Фурье

Основы теории вейвлетов

Функции Wavelet Toolbox

Графический интерфейс вейвлет-анализа в Matlab



Допущено Министерством образования и науки Российской Федерации в качестве учебного пособия

ДМК издательство

Н. К. Смоленцев

Основы теории вейвлетов
Вейвлеты в MATLAB

Издание четвертое,
дополненное и переработанное

Допущено Министерством образования и науки
Российской Федерации в качестве учебного пособия
для студентов высших учебных заведений,
обучающихся по направлениям подготовки и специальностям
«Математика», «Математика. Прикладная математика»

ДМК Пресс
издательство
Москва, 2014

УДК 519.6
ББК В162я73
С51

Смоленцев Н. К.
С51 Основы теории вейвлетов. Вейвлеты в MATLAB. – М.: ДМК Пресс, 2014. – 628 с.

ISBN 978-5-94074-955-4

Предлагаемая читателю книга может служить учебником по теории вейвлетов и их применениям в системе MATLAB. Она предназначена для студентов высших учебных заведений, обучающихся по направлениям и специальностям, связанным с математикой, прикладной математикой и информационными технологиями, и будет полезна специалистам-практикам, использующим вейвлеты в своей работе. В книгу включены сведения по преобразованию Фурье, фильтрам и разложению сигналов. Впервые в учебной литературе представлено построение вейвлетов с произвольным натуральным коэффициентом масштабирования *N*. Рассматриваются вейвлеты с матричным коэффициентом масштабирования, гармонические вейвлеты и мультивейвлеты. Вторая часть книги посвящена описанию основных функций вейвлет-анализа в системе MATLAB и их использования для удаления шума и сжатия сигнала, для обработки изображений и трехмерных массивов. Отдельно в качестве примера рассмотрен вейвлет-анализ кардиосигналов.

УДК 519.6
ББК В162я73

Все права защищены. Любая часть этой книги не может быть воспроизведена в какой бы то ни было форме и какими бы то ни было средствами без письменного разрешения владельцев авторских прав.

Материал, изложенный в данной книге, многократно проверен. Но поскольку вероятность технических ошибок все равно существует, издательство не может гарантировать абсолютную точность и правильность приводимых сведений. В связи с этим издательство не несет ответственности за возможные ошибки, связанные с использованием книги.

ISBN 978-5-94074-955-4

© Смоленцев Н. К., 2014
© Оформление, ДМК Пресс, 2014

Содержание

Предисловие 9

▼ Часть I

Основы теории вейвлетов 12

▼ Глава 1

Преобразование Фурье и фильтры 13
1.1. Преобразование Фурье и фильтры 13
1.1.1. Предварительные замечания 13
1.2. Ряды Фурье 15
1.3. Преобразование Фурье 18
1.3.1. Преобразование Фурье в $L^1(R)$ 19
1.3.2. Преобразование Фурье в $L^2(R)$ 20
1.3.3. Свойства преобразований Фурье 22
1.3.4. Примеры 24
1.3.5. Теорема Пэли-Винера 25
1.3.6. Преобразование Фурье экспоненциально убывающей функции 26
1.3.7. Формула суммирования Пуассона 27
1.3.8. Оконное преобразование Фурье 28
1.4. Преобразование Фурье дискретных сигналов 29
1.4.1. Дискретизация 29
1.4.2. Дискретное преобразование Фурье сигнала длины N 32
1.4.3. Преобразование Фурье числовой последовательности 35
1.4.4. Z-преобразование 38

1.5. Фильтры ... 38
1.5.1. Фильтрация непрерывных сигналов ... 38
1.5.2. Примеры фильтров ... 41
1.5.3. Цифровые фильтры ... 44
1.5.4. Примеры цифровых фильтров ... 45
1.6. Разложение сигнала на низкочастотную и высокочастотную составляющие ... 47
1.6.1. Разложение идеальными фильтрами ... 48
1.6.2. Восстановление идеальными фильтрами ... 53
1.6.3. Общий случай ... 54
1.6.4. Примеры ... 60
1.6.5. Многоуровневый анализ сигналов ... 64

▼ Глава 2

Основы теории вейвлетов ... 66
2.1. Вейвлеты Хаара ... 66
2.1.1. Последовательность масштабированных подпространств ... 68
2.1.2. Операторы проектирования ... 69
2.1.3. Пространства вейвлетов ... 72
2.2. Масштабирующие функции ... 77
2.2.1. Примеры и общие свойства масштабирующих функций ... 77
2.2.2. Построение масштабирующей функции ... 81
2.3. Ортогональный кратномасштабный анализ ... 88
2.3.1. Ортогональное кратномасштабное разложение ... 89
2.3.2. Вейвлеты ... 93
2.3.3. О единственности порождающих функций ... 101
2.3.4. Неортогональный случай ... 103
2.3.5. Достаточные условия ортогональности ... 107
2.4. Вейвлет-преобразование ... 109
2.4.1. Вейвлет-разложение ... 110
2.4.2. Быстрое вейвлет-преобразование ... 114
2.4.3. Вопрос о начальных коэффициентах ... 116
2.4.4. Восстановление ... 117
2.4.5. Вейвлет-пакеты ... 119
2.5. Примеры кратномасштабного анализа и вейвлетов ... 124
2.5.1. Вейвлеты Шеннона–Котельникова ... 124

2.5.2. Вейвлеты Мейера 128
2.6. Вейвлеты Батла–Лемарье. B-сплайны 133
2.6.1. Вейвлеты на основе B-сплайна степени 1 133
2.6.2. B-сплайны 138
2.6.3. Сплайновые вейвлеты 141
2.7. Регулярность и нулевые моменты 145
2.8. Построение вейвлетов Добеши с компактным носителем 151
2.8.1. Построение функции $H_0(\omega)$ 153
2.8.2. Симлеты 159
2.9. Койфлеты 161
2.10. Биортогональные вейвлеты 164
2.10.1. Мотивировка и определение 165
2.10.2. Условия на функции $\varphi(x)$ и $\psi(x)$ 167
2.10.3. Построение функции $\tilde{\varphi}(x)$ 169
2.10.4. Построение функций $\psi(x)$ и $\tilde{\psi}(x)$ 172
2.10.5. Условия на коэффициенты 174
2.10.6. Симметричные биортогональные вейвлеты 174
2.10.7. Сплайны 176
2.11. Двумерные вейвлеты 179
2.11.1. Вейвлет-преобразование 184
2.12. Непрерывное вейвлет-преобразование 185
2.12.1. Непрерывное вейвлет-преобразование в одномерном случае 185
2.12.2. Многомерные обобщения непрерывного вейвлет-преобразования 189
2.13. Вейвлеты с коэффициентом масштабирования N 200
2.13.1. Масштабирующие функции 201
2.13.2. N-кратномасштабное разложение 203
2.13.3. Вейвлеты с коэффициентом масштабирования N 205
2.13.4. Вейвлет-преобразование 207
2.13.5. Разложение и восстановление в неортогональном случае 209
2.14. Примеры N-масштабирующих функций и вейвлетов 212
2.14.1. Вейвлеты Хаара с параметром сжатия N 212
2.14.2. Вейвлеты Шеннона с параметром сжатия N 217
2.14.3. Вырожденные масштабирующие функции и вейвлеты Кантора 220

2.14.4. Сплайновые масштабирующие функции 225
2.14.5. Вейвлеты на основе B-сплайнов 228
2.14.6. Кратные коэффициенты масштабирования 242
2.15. Построение ортогональных вейвлетов с компактным носителем для N > 2 247
2.15.1. Условия ортогональности 248
2.15.2. Построение матрицы частотных функций 250
2.15.3. Построение матрицы частотных функций в случае N = 3 254
2.15.4. Примеры масштабирующих функций и вейвлетов для N = 3 255
2.16. Многомерные вейвлеты с матричным коэффициентом масштабирования 262
2.16.1. Масштабирующие функции 263
2.16.2. A-кратномасштабное разложение 275
2.16.3. Вейвлеты с матрицей масштабирования 280
2.16.4. Вейвлет-преобразование 281
2.16.5. Разложение и восстановление 283
2.16.6. Построение вейвлетов с матрицей масштабирования 288
2.17. Гармонические вейвлеты 295
2.17.1. Гармонические вейвлеты на R 296
2.17.2. Периодические гармонические вейвлеты 299
2.17.3. Дискретное гармоническое вейвлет-разложение 301
2.18. Мультивейвлеты 304

▼ Часть II

Вейвлеты в MATLAB 314

▼ Глава 3

Функции вейвлет-анализа в MATLAB 315
3.1. Вейвлеты и банки фильтров в системе MATLAB 317
3.1.1. Вещественные и комплексные вейвлеты 317
3.1.2. Ортогональные и биортогнальные банки фильтров 332
3.1.3. Построение вейвлетов. Лифтинг 338
3.2. Непрерывное вейвлет-преобразование cwt 344
3.3. Дискретный вейвлет-анализ 359

3.3.1. Анализ одномерных сигналов 359
3.3.2. Анализ изображений 381
3.3.3. Трехмерный анализ 402
3.3.4. Анализ мультисигналов 407
3.4. Вейвлет-пакеты 416
3.5. Удаление шума и сжатие сигнала 431
3.5.1. Функции MATLAB для удаления шума и сжатия 433
3.5.2. Многовариантное удаление шума 446
3.5.3. Сжатие сигнала 457
3.6. Тестовые сигналы в MATLAB 466
3.6.1. Одномерные тестовые сигналы 466
3.6.2. Изображения 467
3.6.3. Генерирование сигналов 468
3.7. Вейвлет-анализ кардиосигнала 472
3.7.1. Многоуровневый анализ кардиосигнала 474
3.7.3. Удаление шума, компрессия и сглаживание кардиосигнала 495
3.7.4. Использование пакетных разложений 498

▼ Глава 4

Главное меню пакета Wavelet Toolbox 508
4.1. Просмотр вейвлетов 508
4.1.1. Просмотр вейвлетов 508
4.1.2. Просмотр пакетных вейвлетов 512
4.2. Продолжение сигналов и изображений (Extension) 513
4.3. Одномерный вейвлет-анализ 516
4.3.1. Одномерный вейвлет-анализ (Wavelet 1D) 517
4.3.2. Одномерный пакетный вейвлет-анализ 528
4.3.3. Непрерывный вейвлет-анализ (Continuous Wavelet 1D) 531
4.3.4. Комплексный непрерывный вейвлет-анализ (Complex Continuous Wavelet 1D) 536
4.3.5. Непрерывный вейвлет-анализ на основе FFT 538
4.4. Специализированные средства одномерного вейвлет-анализа 543
4.4.1. Удаление шума стационарного одномерного сигнала (SWT De-noising 1-D) 544
4.4.2. Оценка плотности (Density Estimation 1D) 550

4.4.3. Оценка регрессии (Regression Estimation 1D) 554
4.4.4. Выбор вейвлет-коэффициентов сигнала (Wavelet Coefficients Selection 1D) 558
4.4.5. Моделирование дробного броуновского движения (Fractional Brownian Generation 1D) 561
4.4.6. Выполнение подгонки (Matching Pursuit 1D) 563
4.5. Двумерный вейвлет-анализ 567
4.5.1. Двумерный дискретный вейвлет-анализ (Wavelet 2D) 568
4.5.2. Двумерный пакетный вейвлет-анализ 571
4.6. Специализированные средства двумерного вейвлет-анализа 572
4.6.1. Удаление шума изображения (SWT De-noising 2D) 572
4.6.2. Выбор вейвлет-коэффициентов изображения (Wavelet Coefficients Selection 2D) 574
4.6.3. Слияние двух изображений (Image Fusion) 576
4.6.4. Полное сжатие с использованием вейвлетов (True Compression 2D) 578
4.7. Трехмерный вейвлет-анализ (Wavelet 3D) 584
4 .8. Мультисигналы (Multiple 1D) 587
4.8.1. Вейвлет-анализ мультисигнала 588
4.8.2. Многовариантное удаление шума (Multivariate Denoising) 594
4.8.3. Многомасштабный анализ главных компонент 602
4.9. Проектирование вейвлетов для непрерывного вейвлет-преобразования (New Wavelet for CWT) 606

Приложение. Список функций Wavelet Toolbox 610

Список литературы 616

Предметный указатель 624

Предисловие

Функции типа маленькой волны (всплески, или вейвлеты) в математике возникли достаточно давно при изучении базисов функциональных пространств. Однако только в последние десятилетия они нашли широкие применения в обработке сигналов и изображений. Эти приложения стимулировали мощное развитие теории вейвлетов. Теория вейвлетов является альтернативой анализу Фурье и дает более гибкую технику обработки сигналов. Одно из основных преимуществ вейвлет-анализа заключается в том, что он позволяет заметить хорошо локализованные изменения сигнала, тогда как анализ Фурье этого не дает – в коэффициентах Фурье отражается поведение сигнала за все время его существования. Разработана глубокая и красивая математическая теория вейвлетов [Дб], [М], [НС], [Чуи], [J03].

Из имеющихся на русском языке книг по вейвлетам отметим фундаментальные монографии: И. Добеши [Дб], К. Чуи [Чуи] и Новикова И. Я., Протасова В. Ю., Скопиной М. А. [НПС], Яковлева А. Н. [Я] и Фаркова Ю. А. [Фар]. Издан перевод замечательной книги С. Малла [М] – это наиболее полное учебное пособие по обработке сигналов при помощи вейвлетов. В ней прекрасно сочетаются доступность и глубина изложения. Кроме того, издан еще ряд пособий, среди которых нужно отметить книгу Блаттера К. [Бл], учебные пособия Новикова Л. В. [Но], Петухова А. П. [Пе] и Захарова В. Г. [За]. В книге Дьяконова В. П. [Д] дано описание работы с вейвлетами в системах компьютерной математики Mathcad, MATLAB и Mathematica. Приложениям вейвлетов в компьютерной графике посвящены книги [СДС] и [У].

Данная книга предлагается как учебник по теории вейвлетов и их применениям для студентов по математическим специальностям и направлениям подготовки. Она возникла на основе курса лекций, читаемых автором в течение ряда лет. Чтобы сделать книгу более

независимой, в нее включены сведения по рядам и преобразованию Фурье, по фильтрам и разложению сигналов. Теоретический материал не должен быть самоцелью, нужно овладеть и практическими приемами работы с вейвлетами. Поэтому в книгу включено описание основных функций вейвлет-анализа в системе MATLAB и их использования для обработки сигналов. В соответствии с этим книга состоит из двух частей: «Основы теории вейвлетов» и «Вейвлеты в MATLAB». Первая часть книги содержит необходимый теоретический материал, а вторая часть прямо ориентирована на практические занятия по вейвлетам. Выбор системы компьютерной математики MATLAB объясняется тем, что она популярна среди инженеров и математиков, занимающимися прикладными разработками, а также потому, что именно в MATLAB вейвлеты представлены наиболее полно.

Данное издание книги является переработанным и дополненным. Многие вопросы изложены более доступно. Добавлен раздел о гармонических вейвлетах. Рассмотрены масштабирующие функции и вейвлеты в многомерном случае с матричным коэффициентом масштабирования. Добавлено описание новых возможностей пакета Wavelet Toolbox MATLAB.

Рассмотрим кратко содержание книги. Теория вейвлетов широко использует технику рядов Фурье и преобразования Фурье. Поэтому в первой главе излагаются основные факты из этих тем. Даже если сигнал представлен функцией, практически для его анализа берется достаточно плотная выборка значений (дискретизация). Рассмотрены вопросы, которые возникают при дискретизации сигнала, определено дискретное преобразование Фурье и изучаются его свойства. Вейвлет-преобразование сигнала сводится к действию на этот сигнал определенных фильтров. Поэтому в первой главе изложены также основные факты фильтрации сигналов. Рассмотрены вопросы разложения сигнала на сглаженную и высокочастотную компоненты и последующего восстановления сигнала. Хотя первая глава является вспомогательной, результаты последних параграфов существенны для понимания теории вейвлетов и их практических применений.

Вторая глава «Основы теории вейвлетов» является центральной в данной книге. Она содержит изложение основ теории вейвлетов и способов их построения. Начинается изложение с определения масштабирующих функций и построения известного вейвлет-базиса Хаара. На этом примере мы рассматриваем основные конструкции, которые затем последовательно развиваются в следующих параграфах. Рассмотрены примеры масштабирующих функций и соответствую-

щих вейвлетов: Шеннона–Котельникова, Мейера, ортогональных вейвлетов с компактным носителем, биортогональных вейвлетов. Кратко изложены вопросы о двумерных вейвлетах. Рассмотрены непрерывное вейвлет-преобразование и его многомерные обобщения. Достаточно подробно изучаются масштабирующие функции с произвольным натуральным коэффициентом сжатия N и приводятся примеры построения соответствующих вейвлетов. В конце главы рассмотрены масштабирующие функции и вейвлеты в многомерном случае с матричным коэффициентом масштабирования, гармонические вейвлеты и мультивейвлеты. Несмотря на то что в книге изложены не все темы теории вейвлетов, надеюсь, что ее содержание достаточно для первоначального изучения предмета студентами, прослушавшими курс функционального анализа.

Во второй части книги дается описание основных функций системы MATLAB, связанных с вейвлетами и их использованием. Показано, как можно получить значения и построить графики основных типов вейвлетов, как найти масштабирующие фильтры и фильтры разложения и восстановления вейвлетов. Рассмотрены возможности Wavelet Toolbox MATLAB для анализа сигналов, очистки от шума, сжатия. Дано описание применения пакетных вейвлетов и двумерных вейвлетов. Приведены примеры вейвлет-анализа кардиосигнала. Дано описание тестовых сигналов и приведен список всех команд Wavelet Toolbox MATLAB. Для облегчения работы с вейвлетами в MATLAB создан комплекс графических оболочек для вейвлет-анализа и визуализации исходных данных и результатов. Этот комплекс называется главным вейвлет-меню, или графическим интерфейсом пользователя (GUI). В последней главе достаточно подробно рассматривается работа с вейвлетами с использованием графического интерфейса пользователя MATLAB Wavelet Toolbox.

Ссылки на литературу даны, по возможности, на доступные издания.

Надеюсь, что данная книга будет доступна и полезна студентам вузов и специалистам, начинающим использовать вейвлеты в своей работе.

ОСНОВЫ ТЕОРИИ ВЕЙВЛЕТОВ



Теория вейвлетов является мощной альтернативой классическому анализу Фурье и дает более гибкую технику обработки сигналов. В то же время она широко использует технику рядов Фурье и преобразования Фурье. Поэтому в первой главе излагаются основные факты из указанных тем, включая дискретное преобразование Фурье и фильтрацию сигналов. Во второй главе рассматриваются основы теории вейвлетов и способов их построения. Рассмотрены примеры масштабирующих функций и соответствующих вейвлетов: Шеннона–Котельникова, Мейера, ортогональные вейвлеты с компактным носителем, биортогональные вейвлеты и вейвлеты с произвольным натуральным коэффициентом сжатия N. Рассмотрены непрерывное вейвлет-преобразование и его многомерные обобщения. В конце главы проанализированы масштабирующие функции и вейвлеты в многомерном случае с матричным коэффициентом масштабирования и гармонические вейвлеты.

Преобразование Фурье и фильтры



1

1.1. Преобразование Фурье и фильтры

В данной главе кратко представлены основные сведения по рядам Фурье и преобразованию Фурье, изложены вопросы дискретного преобразования Фурье, фильтров и разложения сигналов.

1.1.1. Предварительные замечания

Напомним понятия, которые далее часто встречаются, и примем некоторые обозначения.

Числовые последовательности $\{x_n\}$, которые мы будем рассматривать, являются «бесконечными в обе стороны», то есть номер n может принимать любые целые значения $n \in \mathbf{Z}$. Числовой ряд $\sum_{n=-\infty}^{n=\infty} x_n$ называется сходящимся, если существует предел $\lim_{N,K\to\infty} \sum_{n=-K}^{N} x_n$. Степенные ряды рассматриваются как формальные, они содержат отрицательные степени, и вопрос об их сходимости не рассматривается. Такие степенные ряды будут обозначаться следующими символами:

$$\sum_n a_n x^n = \sum_{n\in\mathbf{Z}} a_n x^n = \sum_{n=-\infty}^{\infty} a_n x^n.$$

Функции $f(x)$ являются, вообще говоря, комплексно-значными и определены на множестве $\mathbf{R}$ действительных чисел. *Носителем* функции $f(x)$ называется замыкание множества точек x, в которых $f(x) \neq 0$. Носитель обозначается символом $\mathrm{supp}(f)$. Если $\mathrm{supp}(f)$ находится на конечном промежутке $[a, b]$, то $f(x)$ называется функцией с *компактным носителем*.

Некоторое свойство функции $f(x)$ выполняется *почти всюду* (*п. в.*), если множество точек, в которых это свойство не выполнено, имеет нулевую меру.

Векторное пространство E называется евклидовым, если в нем задано скалярное произведение (u, v). В этом случае для любого элемента $v \in E$ определены норма $||v|| = \sqrt{(v, v)}$ и сходимость: $v_n \to v$, если $\| v_n - v \| \to 0$. Пространство E называется *гильбертовым,* если оно является полным относительно определенной выше сходимости. Система элементов $\{u_n, n \in \mathbf{Z}\}$ в гильбертовом пространстве E называется *ортонормированной,* если

$$(u_n, u_m) = \delta_{n,m} = \begin{cases} 1, & n = m \\ 0, & n \neq m \end{cases}, \quad \text{для любых } n, m \in \mathbf{Z}.$$

Ортонормированная система $\{u_n\}$ в гильбертовом пространстве E называется *полной*, если замыкание множества всех линейных комбинаций элементов из $\{u_n\}$ совпадает с пространством E. Другими словами, если E является наименьшим замкнутым пространством, содержащим $\{u_n\}$. Полная ортонормированная система $\{u_n\}$ называется *ортонормированным базисом* гильбертова пространства E. Примером ортогонального базиса может служить система функций $\{1, \cos nx, \sin nx, n = 1, 2, ...\}$ в гильбертовом пространстве $L^2[0, 2\pi]$ функций на $[0, 2\pi]$, интегрируемых с квадратом.

Элементарные гармоники – это наиболее простые сигналы вида

$$a \sin(\omega x + \varphi_0) \text{ и } a \cos(\omega x + \varphi_0),$$

где a – амплитуда, ω – *круговая частота*, φ_0 – начальная фаза. Число $T = 2\pi/\omega$ есть период времени, за который система делает одно полное колебание. Величина $\nu = 1/T = \omega/2\pi$ есть *частота*, число колебаний за единицу времени.

Аналоговым сигналом называется функция $f(x)$ действительного переменного $x \in \mathbf{R}$. Энергией сигнала $f(x)$ называется интеграл

$$\| f \|^2 = \int_{-\infty}^{\infty} |f(x)|^2 dx.$$

Другими интересными числовыми характеристиками сигнала $f(x)$ являются «моменты». *Начальным моментом* порядка n сигнала $f(x)$ называется интеграл

$$m_n(f) = \int_{-\infty}^{\infty} x^n f(x) dx.$$

Центральным моментом порядка n называется интеграл

$$\mu_n(f) = \int_{-\infty}^{\infty} (x - m_1)^n f(x) dx.$$

Момент первого порядка m_1 имеет смысл математического ожидания m. Центральный момент второго порядка имеет смысл дисперсии и характеризует разброс значений $f(x)$ относительно m_1.

Если аргумент x принимает дискретный ряд значений $x = x_n$, $n \in \mathbf{Z}$, то значения $y_n = f(x_n)$ называются отсчетами, а весь набор значений $\{y_n\}$ – *выборкой*. *Дискретизация* непрерывного сигнала $f(x)$ – это замена его выборкой $\{y_n\}$. Такая процедура называется *оцифровкой* аналогового сигнала.

1.2. Ряды Фурье

Напомним основные факты теории рядов Фурье. Подробности и доказательства можно найти в любом учебнике по математическому анализу, см., напр., [АСЧ], [КФ].

Будем рассматривать функции $f(x)$ на $[0, 2\pi]$, *интегрируемые с квадратом*, то есть, для которых существует интеграл (Лебега) $\int_0^{2\pi} |f(x)|^2 dx$. таких подобных функций образует гильбертово пространство $L^2[0, 2\pi]$. Скалярное произведение и норма в $L^2[0, 2\pi]$ определены следующим образом:

$$(f, g) = \int_0^{2\pi} f(x)\overline{g(x)}\, dx, \qquad \| f \|_2 = \sqrt{(f, f)},$$

где черта обозначает комплексное сопряжение. Полную ортогональную систему в $L^2[0, 2\pi]$ образуют функции

$\{1, \cos nx, \sin nx, n = 1, 2, ...\}$.

Если $f(x) \in L^2[0, 2\pi]$, то для нее можно определить *ряд Фурье*:

$$f(x) \;\to\; \frac{a_0}{2} + \sum_{n=1}^{\infty} a_n \cos nx \;+\; b_n \sin nx, \tag{1}$$

где

$$a_0 = \frac{1}{\pi}\int_0^{2\pi} f(x)\, dx, \quad a_n = \frac{1}{\pi}\int_0^{2\pi} f(x)\cos nx\, dx, \quad b_n = \frac{1}{\pi}\int_0^{2\pi} f(x)\sin nx\, dx. \tag{2}$$

Из полноты тригонометрической системы следует, что ряд Фурье функции $f(x) \in L^2[0, 2\pi]$ сходится к $f(x)$ в пространстве $L^2[0, 2\pi]$. Это означает, что $\|f(x) - S_n(x)\|_{L^2} \to 0$, где $S_n(x)$ – частичные суммы ряда. Имеет место равенство Парсеваля

$$\frac{1}{\pi}\int_0^{2\pi}|f(x)|^2dx=\frac{a_0^2}{2}+\sum_{n=1}^{\infty}a_n^2+b_n^2.$$

С другой стороны, если сходится числовой ряд $a_0^2/2+\sum_{n=1}^{\infty}a_n^2+b_n^2$, то ряд (1) сходится в L^2 к функции, у которой коэффициентами Фурье являются числа a_n и b_n.

Отметим также, что если сходится числовой ряд $\sum_{n=1}^{\infty}|a_n|+|b_n|$, то тригонометрический ряд (1) сходится равномерно, его сумма является непрерывной функцией и ряд (1) является рядом Фурье этой функции.

Комплексная форма ряда Фурье. Пусть

$$c_n=\frac{a_n+i\,b_n}{2},\quad c_{-n}=\frac{a_n-i\,b_n}{2},\quad n=0,1,2,...$$

и $e^{inx}=\cos nx+i\sin nx$. Тогда

$$\frac{a_0}{2}+\sum_{n=1}^{\infty}a_n\cos nx+b_n\sin nx=c_0+\sum_{n=1}^{\infty}c_ne^{-inx}+c_{-n}e^{inx}=\sum_{n=-\infty}^{\infty}c_ne^{-inx}.$$

Таким образом,

$$f(x)\to\sum_{n=-\infty}^{\infty}c_ne^{-inx},\ \text{где}\ c_n=\frac{1}{2\pi}\int_0^{2\pi}f(x)e^{inx}dx. \tag{3}$$

Равенство Парсеваля. Если функция $f(x)\in L^2[0,2\pi]$, то

$$\sum_{n=-\infty}^{\infty}|c_n|^2=\frac{1}{2\pi}\int_0^{2\pi}|f(x)|^2dx.$$

Случай промежутка [0, 2*N*]. Пусть функция $f(x)$ определена на промежутке $[0,2N]$, где N – некоторое положительное число. Полную ортогональную систему в $L^2[0,2N]$ образуют функции:

$$\left\{1,\ \cos\frac{\pi}{N}nx,\ \sin\frac{\pi}{N}nx,\quad n=1,2,...\right\}.$$

Ряд Фурье функции $f(x)$ имеет вид:

$$f(x)\to\frac{a_0}{2}+\sum_{n=1}^{\infty}a_n\cos\frac{\pi}{N}nx+b_n\sin\frac{\pi}{N}nx,$$

где

$$a_n=\frac{1}{N}\int_0^{2N}f(x)\cos\left(\frac{\pi}{N}nx\right)dx,\qquad b_n=\frac{1}{N}\int_0^{2N}f(x)\sin\left(\frac{\pi}{N}nx\right)dx.$$

В комплексной форме:

$$f(x) \to \sum_{n=-\infty}^{\infty} c_n e^{-i\frac{\pi}{N}nx}, \quad \text{где} \quad c_n = \frac{1}{2N}\int_0^{2N} f(x) e^{i\frac{\pi}{N}nx}\,dx.$$

В приложениях функцию $f(x)$ часто называют сигналом. При разложении $f(x)$ в ряд Фурье величина каждого коэффициента a_n, b_n показывает, насколько значителен вклад гармоники $\cos nx$ или $\sin nx$ в формирование сигнала $f(x)$. Эти коэффициенты характеризуют интенсивности элементарных гармоник. Набор коэффициентов Фурье $\{a_0, a_1, b_1, a_2, b_2, \ldots\}$ называется спектром сигнала $f(x)$. Спектром считают также набор комплексных коэффициентов Фурье $\{c_n, n \in \mathbf{Z}\}$.

Замечание. Пусть $f(x) = \sum_{k=-\infty}^{\infty} c_k e^{-ikx}$ и $g(x) = \sum_{m=-\infty}^{\infty} d_m e^{-imx}$. Тогда коэффициенты Фурье произведения $f(x)g(x)$ получаются в виде свёртки коэффициентов c_k и d_m:

$$c_n(fg) = \sum_{k=-\infty}^{\infty} c_k d_{n-k}.$$

Действительно,

$$f(x)g(x) = \left(\sum_k c_n e^{-ikx}\right)\left(\sum_m d_m e^{-imx}\right) = \sum_{k,m} c_k\, d_m e^{-i(k+m)x} =$$

$$= \sum_n \left(\sum_{k+m=n} c_k d_m\right) e^{-inx} = \sum_n \left(\sum_k c_k d_{n-k}\right) e^{-inx} = \sum_n c_n(fg)\, e^{-inx}.$$

Приведем несколько примеров вычисления коэффициентов Фурье функций, которые будут в дальнейшем использоваться.

Пример 1. Коэффициенты Фурье функции

$$f(x) = \begin{cases} 1, & |\,x\,| \le a \le \pi \\ 0, & \text{при остальных } x \end{cases}$$

на промежутке $[-\pi, \pi]$. Функция четная, поэтому все коэффициенты b_n равны нулю и соответствующий ряд Фурье содержит только косинусы. Коэффициенты Фурье a_n легко вычисляются:

$$a_0 = \frac{2a}{\pi}, \quad a_n = \frac{2\sin na}{\pi n}, \quad c_n = \frac{\sin na}{\pi n}, \quad f(x) = \frac{a}{\pi} + \sum_{n=1}^{\infty} \frac{2\sin na}{\pi n}\cos nx.$$

В частности, если $a = \pi/2$, то

$$a_n = \begin{cases} 1, & n = 0, \\ 0, & n = 2k, k \ne 0, \\ (-1)^k \dfrac{2}{n\pi}, & n = 2k+1, \end{cases} \qquad c_n = c_{-n} = \frac{a_n}{2}.$$

Пример 2. Коэффициенты Фурье функции $f(x) = \cos^2(x/2)$ на промежутке $[-\pi, \pi]$. Используя известную формулу $\cos^2(x/2) = (\cos x + 1)/2$, получаем два ненулевых коэффициента $a_0 = 1$, $a_1 = 1/2$. Для функции $f(x) = \sin^2(x/2)$ аналогично получаем: $a_0 = 1$, $a_1 = -1/2$.

Пример 3. Коэффициенты Фурье функции $f(x) = e^{ix/2}$ на промежутке $[-\pi, \pi]$ находятся прямым вычислением:

$$c_n = \frac{1}{2\pi}\int_{-\pi}^{\pi} e^{ix/2}e^{inx}dx = \frac{2}{\pi}\frac{(-1)^n}{2n+1}, \quad \forall n \in \mathbf{Z}.$$

Коэффициенты Фурье функции $f(x) = e^{-ix/2}$ на промежутке $[-\pi, \pi]$ следующие:

$$c_n = \frac{1}{2\pi}\int_{-\pi}^{\pi} e^{-ix/2}e^{inx}dx = -\frac{2}{\pi}\frac{(-1)^n}{2n-1}, \quad \forall n \in \mathbf{Z}.$$

В общем случае для функции $f(x) = e^{ikx/2}$ на промежутке $[-\pi, \pi]$ имеем:

В случае четного $k = 2m$:

$$c_n = \begin{cases} 1, & n = -m, \\ 0 & n \neq -m \end{cases}.$$

В случае нечетного $k = 2m + 1$:

$$c_n = \frac{2}{\pi}\frac{(-1)^{n+m}}{2n+2m+1}.$$

Пример 4. Коэффициенты Фурье функции $f(x) = \cos(x/2)$ на промежутке $[-\pi, \pi]$. Можно использовать пример 3 для нахождения комплексных коэффициентов либо непосредственным интегрированием, используя известную формулу $\cos\alpha \cdot \cos\beta = {}^1/_2(\cos(\alpha+\beta) + \cos(\alpha-\beta))$. В любом случае $\forall\, n \in \mathbf{N}$,

$$a_n = (-1)^{n+1}\frac{4}{\pi(4n^2-1)}, \quad c_n = c_{-n} = (-1)^{n+1}\frac{2}{\pi(4n^2-1)}.$$

1.3. Преобразование Фурье

В данном параграфе мы напомним основные факты, связанные с преобразованием Фурье. Все доказательства можно найти в учебниках по математическому анализу, см., напр., [АСЧ], [КФ].

Пусть $L^1(\mathbf{R})$ – пространство, состоящее из абсолютно интегрируемых функций на $\mathbf{R}$. Оно является банаховым пространством с нормой

$$\|f\|_1 = \int_{-\infty}^{\infty} |f(x)|\,dx.$$

Произведение $f(x)g(x)$ двух функций из пространства $L^1(\mathbf{R})$ может не принадлежать пространству $L^1(\mathrm{R})$. Тем не менее для функций класса $L^1(\mathbf{R})$ определена операция *свертки* $f * g$, превращающая пространство $L^1(\mathbf{R})$ в коммутативную банахову алгебру без единицы:

$$(f * g)(x) = \int_{-\infty}^{\infty} f(x-y)g(y)\,dy = \int_{-\infty}^{\infty} f(y)g(x-y)\,dy. \qquad (1)$$

Корреляция функций $f(x)$ и $g(x)$, интегрируемых с квадратом на $\mathbf{R}$, определяется как следующий интеграл:

$$K_{fg}(\tau) = \int_{-\infty}^{\infty} f(x+\tau)\overline{g(x)}\,dx = \int_{-\infty}^{\infty} f(x)\overline{g(x-\tau)}\,dx. \qquad (2)$$

Корреляция показывает степень сходства функций $f(x)$ и $g(x)$ при различных значениях сдвигов по оси Ox. Хотя корреляция функций $f(x)$ и $g(x)$ по форме напоминает свертку, по существу имеются серьезные различия. Нет коммутативности относительно $f(x)$ и $g(x)$. Корреляция определена для функций из $L^2(\mathrm{R})$. При интегрировании переменная интегрирования x в функциях $f(x)$ и $g(x)$ ориентирована одинаково. Функции $f(x)$ и $g(x)$ называются некоррелированными, если $K_{fg}(\tau) = 0$. В случае $g(x) = f(x)$ функция $K_f(\tau) = K_{ff}(\tau)$ называется автокорреляционной функцией.

1.3.1. Преобразование Фурье в $L^1(\mathbf{R})$

Определение 1. *Преобразованием Фурье функции* $f(x) \in L^1(\mathbf{R})$ *называется интеграл*

$$F[f] = \hat{f}(\omega) = \int_{-\infty}^{\infty} f(x)e^{-ix\omega}\,dx, \qquad \omega \in \mathbf{R}. \qquad (3)$$

Переменная ω имеет смысл частоты. Поэтому переход от $f(x)$ к $\hat{f}(\omega)$ часто называют переходом из пространственной области в частотную.

Рассмотрим вопрос о восстановлении функции $f(x)$ по ее преобразованию Фурье $\hat{f}(\omega)$. Естественно ожидать, что $f(x)$ будет выражаться интегралом

$$F^{-1}[\hat{f}] = \frac{1}{2\pi}\int_{-\infty}^{\infty} \hat{f}(\omega)e^{ix\omega}\,d\omega. \qquad (4)$$

К сожалению, $\hat{f}(\omega)$ может оказаться неинтегрируемой. Например, для характеристической функции $f(x) = \chi_{[-1,1]}(x)$ промежутка $[-1, 1]$ преобразованием Фурье будет функция $\hat{f}(\omega) = 2(\sin\omega)/\omega$. Хорошо известно, что она не является абсолютно интегрируемой на $\mathbf{R}$ (для нее существует только условно сходящийся несобственный интеграл Римана). Из свойства 1 преобразования Фурье (см. ниже) следует, что причина неинтегрируемости $\hat{f}(\omega)$ заключается в разрывности исходной функции $f(x) = \chi_{[-1,1]}(x)$.

Для того чтобы обойти эту трудность с неинтегрируемостью $\hat{f}(\omega)$, обычно используют несобственные интегралы, то есть полагают

$$\int_{-\infty}^{\infty} \hat{f}(\omega) e^{ix\omega}\, d\omega = \lim_{a,b\to+\infty} \int_{-a}^{b} \hat{f}(\omega) e^{ix\omega}\, d\omega.$$

Главное значение $V.p.\int_{-\infty}^{\infty} \hat{f}(\omega) e^{ix\omega}\, d\omega$ несобственного интеграла получается, когда $a = b \to \infty$.

Обратным преобразованием Фурье называется выражение (4), в котором интеграл понимается как несобственный в смысле главного значения.

Хорошо известно, что равенство

$$f(x) = \frac{1}{2\pi} \int_{-\infty}^{\infty} \hat{f}(\omega) e^{ix\omega}\, d\omega \tag{5}$$

имеет место в точке $x \in \mathbf{R}$, если $f(x)$ удовлетворяет в этой точке условию Дини, а интеграл справа понимается в смысле главного значения. Приведем некоторые достаточные условия для обращения преобразования Фурье:

- если функции $f(x)$ и $\hat{f}(\omega)$ абсолютно интегрируемы, то равенство (5) имеет место в любой точке x, где $f(x)$ непрерывна;
- если $f(x) \in L^1(\mathbf{R})$ непрерывна и кусочно-дифференцируема на $\mathbf{R}$, то имеет место формула (5) обращения преобразования Фурье.

Функция $f(x)$ называется кусочно-дифференцируемой на $\mathbf{R}$, если она дифференцируема всюду, кроме дискретного множества точек, в каждой из которых она имеет односторонние производные.

1.3.2. Преобразование Фурье в $L^2(\mathbf{R})$

Как уже отмечалось, преобразование Фурье F определено формулой (3) для функций из $L^1(\mathbf{R})$, но функции $F[f]$ могут не принадлежать

$L^1(\mathbf{R})$. Можно решить эту проблему, обратившись к гильбертову пространству $L^2(\mathbf{R})$ функций, интегрируемых с квадратом на $\mathbf{R}$. При этом мы получаем все преимущества, связанные с существованием на $L^2(\mathbf{R})$ скалярного произведения.

Для функций, интегрируемых с квадратом, интеграл (3) может не существовать. Однако можно определить преобразование Фурье как оператор $F: L^2(\mathbf{R}) \to L^2(\mathbf{R})$. Это делается по следующей схеме. Сначала определяется обычным образом преобразование Фурье на некотором всюду плотном подпространстве S из $L^2(\mathbf{R})$. Затем показывается, что оператор F на этом подпространстве S является непрерывным. Наконец, преобразование Фурье $F: L^2(\mathbf{R}) \to L^2(\mathbf{R})$ определяется как продолжение по непрерывности с подпространства S на все пространство $L^2(\mathbf{R})$. В качестве подпространства $S \subset L^2(\mathbf{R})$ обычно берут пространство быстро убывающих функций. Нам пока достаточно в качестве S взять пересечение $L^1(\mathbf{R}) \cap L^2(\mathbf{R})$ пространств $L^1(\mathbf{R})$ и $L^2(\mathbf{R})$.

Возможность продолжения оператора $F: L^1(\mathbf{R}) \cap L^2(\mathbf{R}) \to L^2(\mathbf{R})$ на $L^2(\mathbf{R})$ основывается на следующей теореме.

Теорема 1. (Формула Планшереля). *Пусть* $f, g \in L^1(\mathbf{R}) \cap L^2(\mathbf{R})$. *Тогда* $\hat{f}, \hat{g} \in L^2(\mathbf{R})$ *и имеет место равенство*

$$\int_{-\infty}^{\infty} f(x)\overline{g(x)}dx = \frac{1}{2\pi}\int_{-\infty}^{\infty} \hat{f}(\omega)\overline{\hat{g}(\omega)}d\omega, \tag{6}$$

то есть $(f, g) = (\hat{f}, \hat{g})/2\pi$.

Из данной теоремы следует, что оператор $F: L^1(\mathbf{R}) \cap L^2(\mathbf{R}) \to L^2(\mathbf{R})$ является ограниченным и

$$\|F[f]\|_2 = \sqrt{2\pi}\,\|f\|_2. \tag{7}$$

Поскольку пространство $L^1(\mathbf{R}) \cap L^2(\mathbf{R})$ всюду плотно в $L^2(\mathbf{R})$, то F единственным образом продолжается на все пространство $L^2(\mathbf{R})$ с сохранением нормы. Продолжение оператора F на $L^2(\mathbf{R})$ строится следующим образом. Пусть $f(x) \in L^2(\mathbf{R})$. Тогда функция $f_n(x)$, равная $f(x)$ при $|x| < n$ и равная нулю при остальных x, лежит в пространстве $L^1(\mathbf{R})$. Поэтому определено ее преобразование Фурье $\hat{f}_n(\omega)$. Последовательность $\{\hat{f}_n(\omega)\}$ фундаментальна в $L^2(\mathbf{R})$, и, следовательно, существует единственный предел $F[f] = \lim \hat{f}_n(\omega)$ в пространстве $L^2(\mathbf{R})$. Он и называется преобразованием Фурье функции $f(x) \in L^2(\mathbf{R})$.

Из теоремы 3 следует, что преобразование Фурье взаимно однозначно отображает $L^2(\mathbf{R})$ на $L^2(\mathbf{R})$. В частности, для любой функции $g \in L^2(\mathbf{R})$ определено обратное преобразование Фурье $F^{-1}[g]$.

Теорема 2 (Планшереля). *Преобразование Фурье*

$$\frac{1}{\sqrt{2\pi}}F : L^1(\mathbf{R}) \cap L^2(\mathbf{R}) \to L^2(\mathbf{R})$$

продолжается до унитарного оператора пространства $L^2(\mathbf{R})$. *Обращение* F^{-1} *преобразования Фурье на* $L^2(\mathbf{R})$ *можно получить, полагая*

$$F^{-1}[\hat{f}] = \frac{1}{2\pi}F[\hat{f}](-x).$$

Интересно, что в пространстве $L^2(\mathbf{R})$ существует ортогональный базис, в котором оператор F: $L^2(\mathbf{R}) \to L^2(\mathbf{R})$ диагонализируется. Это хорошо известный базис функций Эрмита. Напомним его построение. Известно, что функция Гаусса $f(x) = e^{-x^2/2}$ является собственной функцией для преобразования Фурье:

$$F[e^{-x^2/2}] = \sqrt{2\pi}e^{-\omega^2/2}. \tag{8}$$

Искомый базис функций Эрмита можно построить из системы функций

$$e^{-x^2/2},\ xe^{-x^2/2},\ x^2e^{-x^2/2},\ \ldots,\ x^ne^{-x^2/2},\ \ldots$$

обычным процессом ортогонализации. В результате получаются функции вида $\varphi_n(x) = H_n(x)e^{-x^2/2}$, где $H_n(x)$ – многочлены Эрмита степени n. Они могут быть найдены по формуле

$$H_n(x) = (-1)^n e^{x^2}\frac{d^n}{dx^n}(e^{-x^2}).$$

Приведем в качестве примеров несколько первых многочленов Эрмита, $H_0(x) = 1$, $H_1(x) = 2x$, $H_2(x) = 4x^2 - 2$, $H_3(x) = 8x^3 - 12x$.

Хорошо известно (см., напр., [КФ]), что система функций Эрмита $\{\varphi_n(x), n = 0, 1, 2, ...\}$ образует ортогональный базис в $L^2(\mathbf{R})$. Кроме того, эти функции являются собственными для преобразования Фурье:

$$F[\varphi_n] = c_n\varphi_n(\omega).$$

Коэффициенты c_n удовлетворяют условию $c_n^4 = (2\pi)^2$, то есть $c_n = \pm\sqrt{2\pi}, \pm i\sqrt{2\pi}$. Можно показать, что эти коэффициенты будут чередоваться в следующем порядке: $\sqrt{2\pi},\ -i\sqrt{2\pi},\ -\sqrt{2\pi},\ i\sqrt{2\pi}$.

1.3.3. Свойства преобразований Фурье

Свойство 1. *Если* $f(x) \in L^1(\mathbf{R})$, *то* $\hat{f}(\omega)$ *является ограниченной непрерывной функцией на* $\mathbf{R}$, *причем*

$$\left|\hat{f}(\omega)\right| \le \| f \|_1 \qquad \hat{f}(\omega) \to 0 \quad \text{при} \quad \omega \to \infty. \tag{9}$$

Свойство 2. *Если функция $f(x)$ дифференцируема n раз и $f, f', \ldots, f^{(n)} \in L^1(\mathbf{R})$, то*

$$F[f^{(n)}] = (i\omega)^n F[f]. \tag{10}$$

Следствие. *Если функция $f(x)$ дифференцируема n раз и $f, f', \ldots, f^{(n)} \in L^1(\mathbf{R})$, то существует число $M \ge 0$, такое что*

$$\left|\hat{f}(\omega)\right| \le \frac{M}{(1+|\omega|)^n} \qquad \text{и} \qquad |\omega|^n \left|\hat{f}(\omega)\right| \to 0 \quad \text{при} \quad \omega \to \infty.$$

Свойство 3. *Если $f(x), xf(x), \ldots, x^n f(x) \in L^1(\mathbf{R})$, то $\hat{f}(\omega)$ дифференцируема n раз и*

$$(F[f])^{(n)} = (-i)^n F[x^n f(x)]. \tag{11}$$

Вывод. Чем выше степень гладкости функции $f(x)$, тем быстрее убывает ее преобразование Фурье $\hat{f}(\omega)$. Чем быстрее убывает на бесконечности функция $f(x)$, тем выше порядок дифференцируемости $\hat{f}(\omega)$.

Свойство 4. Изменение масштаба. *Если $f(x) \in L^1(\mathbf{R})$, то для любого числа $a \ne 0$*

$$F[f(ax)] = \frac{1}{|a|} \hat{f}\left(\frac{\omega}{a}\right). \tag{12}$$

Свойство 5. Сдвиг в пространстве и в области частот. *Если $f(x) \in L^1(\mathbf{R})$, то для любых чисел b и τ имеют место равенства*

$$F[f(x-b)] = e^{-i\omega b} \hat{f}(\omega), \quad F[e^{ix\tau} f(x)] = \hat{f}(\omega - \tau). \tag{13}$$

Свойство 6. Преобразование Фурье и свертка.

$$F[f * g] = F[f] \cdot F[g], \quad F[f \cdot g] = \frac{1}{2\pi} F[f] * F[g]. \tag{14}$$

Свойство 7. Преобразование Фурье корреляции. *Если $f(x), g(x) \in L^1(\mathbf{R}) \cap L^2(\mathbf{R})$, то*

$$F[K_{fg}](\omega) = \hat{f}(\omega)\overline{\hat{g}(\omega)}. \tag{15}$$

Замечание 1. Функции $f(x)$ и $g(x)$ называются некоррелированными, если $K_{fg}(\tau) = 0$. В этом случае и $F[K_{fg}](\omega) = \hat{f}(\omega)\overline{\hat{g}(\omega)} = 0$. Последнее равенство означает, что частотные спектры (носители функ-

ций $\hat{f}(\omega)$ и $\hat{g}(\omega)$) не пересекаются. Верно и обратное: функции с непересекающимися частотными спектрами являются некоррелированными. В случае $g(x) = f(x)$ функция $K_f(\tau) = K_{ff}(\tau)$ называется *автокорреляционной* функцией. Из свойства 7 следует, что преобразование Фурье автокорреляционной функции $K_f(\tau)$ есть энергетический спектр $|\hat{f}(\omega)|^2$ сигнала $f(x)$.

Свойство 8. *Если* $f(x), g(x) \in L^1(\mathbf{R})$, *то* $\hat{f}, \hat{g} \in L^2(\mathbf{R})$ *и имеет место равенство*

$$\int_{-\infty}^{\infty} f(x)\overline{g(x)}dx = \frac{1}{2\pi}\int_{-\infty}^{\infty} \hat{f}(\omega)\overline{\hat{g}(\omega)}\,d\omega, \tag{16}$$

то есть $(f, g) = (\hat{f}, \hat{g}) / 2\pi$, *в частности* $\|\hat{f}(\omega)\|_2 = \sqrt{2\pi}\,\|f\|_2$.

Свойство 9. Прямое и обратное преобразования Фурье. *Если* $f, \hat{f} \in L^1(\mathbf{R})$, *то*

$$F^{-1}[f](x) = \frac{1}{2\pi}F[f(-x)], \quad F[F[f]](x) = 2\pi f(-x). \tag{17}$$

1.3.4. Примеры

Таблица 1.3.1. Преобразование Фурье некоторых функций

Тип функции	Функция	Образ Фурье
Характеристическая функция	$f(x) = \chi_{[a,b]}(x) = \begin{cases} 1, & x \in [a,b], \\ 0, & x \notin [a,b] \end{cases}$	$\hat{f}(\omega) = \dfrac{e^{-ia\omega} - e^{-ib\omega}}{i\omega}$
	$\varphi(x) = \chi_{[0,1]}(x) = \begin{cases} 1, & x \in [0,1], \\ 0, & x \notin [0,1] \end{cases}$	$\hat{\varphi}(\omega) = e^{-i\omega/2}\,\dfrac{\sin(\omega/2)}{\omega/2}$
	$f(x) = \chi_{[-a,a]}(x) = \begin{cases} 1, & \lvert x\rvert \le a \\ 0, & \lvert x\rvert > a \end{cases}$	$\hat{f}(\omega) = \dfrac{2\sin a\omega}{\omega}$
	$\psi(x) = \begin{cases} -1, & x \in [-a,0) \\ 1, & x \in [0,a) \\ 0, & x \notin [-a,a) \end{cases}$	$\hat{\psi}(\omega) = 2\dfrac{\sin^2(a\omega/2)}{i\omega/2}$
sinc	$f(x) = \dfrac{\sin a\,x}{\pi x}$	$\hat{f}(\omega) = \chi_{[-a,a]}(\omega)$
	$f(x) = \dfrac{\sin a(x-n)}{\pi(x-n)}$	$\hat{f}(\omega) = e^{-i\omega n}\chi_{[-a,a]}(\omega)$

Сплайн степени 1	$f(x)=\begin{cases}1-\lvert x\rvert, & x\in[-1,1]\\ 0, & x\notin[-1,1]\end{cases}$	$\hat{f}(\omega)=\left(\frac{\sin\omega/2}{\omega/2}\right)^2=2\frac{1-\cos\omega}{\omega^2}$
B-сплайн степени 2	$f(x)=\begin{cases}\frac{1}{2}(x+1)^2, & x\in[-1,0]\\ \frac{3}{4}-\left(x-\frac{1}{2}\right)^2, & x\in[0,1]\\ \frac{1}{2}(x-2)^2, & x\in[1,2]\\ 0, & x\notin[-1,2]\end{cases}$	$\hat{f}(\omega)=e^{-i\omega/2}\left(\frac{\sin\omega/2}{\omega/2}\right)^3$
Экспонента	$f(x)=\begin{cases}e^{-x}, & x\geq 0\\ 0, & x<0\end{cases}$	$\hat{f}(\omega)=\frac{1}{1+i\omega}$
	$f(x)=e^{-a\lvert x\rvert}$	$\hat{f}(\omega)=\frac{2a}{a^2+\omega^2}$
Функция Гаусса	$f(x)=e^{-x^2/2}$	$\hat{f}(\omega)=\sqrt{2\pi}\,e^{-\omega^2/2}$
Гауссов вейвлет	$\varphi(x)=-f''(x)=(1-x^2)e^{-x^2/2}$	$\hat{\varphi}(\omega)=\sqrt{2\pi}\omega^2e^{-\omega^2/2}$

1.3.5. Теорема Пэли–Винера

Рассмотрим вопрос о свойствах преобразования Фурье функции $f(x) \in L^2(\mathbf{R})$, имеющей носитель на конечном промежутке $[-\tau, \tau]$. В этом случае интеграл в преобразовании Фурье определен не только для вещественных значений ω, но и для комплексных:

$$F[f](z)=\int_{-\infty}^{\infty}f(x)e^{-ixz}\,dx=\int_{-\tau}^{\tau}f(x)e^{-ixz}\,dx,\qquad z\in\mathbf{C}.$$

Поскольку интеграл существует при всех значениях комплексного переменного $z \in \mathbf{C}$ и является дифференцируемым по параметру z, то функция $F[f](z)$ является целой функцией.

Напомним, что функция $F(z)$ комплексного переменного $z \in \mathbf{C}$ называется *целой*, если она является аналитической функцией, не имеющей конечных особых точек. Если $z = \infty$ – устранимая особая точка, то $F(z) = \text{const}$. Если $z = \infty$ – полюс, то $F(z)$ – многочлен. Если $z = \infty$ – существенно особая точка, то $F(z)$ – трансцендентная, как, например, $\sin(z)$, $\cos(z)$, $\exp(z)$.

Для функции $F[f](z)$, $z = u + iv$ имеет место следующая оценка:

$$|F[f](z)|=|F[f](u+iv)|\leq\sqrt{2\tau}\left(\int_{-\tau}^{\tau}|f(x)|^2\,dx\right)^{1/2}e^{\tau|v|}\leq Ae^{\tau|z|}.$$

Определение 2. *Пусть* $F(z)$ *– целая функция комплексного переменного* $z \in \mathbf{C}$. *Будем говорить, что* $F(z)$ *– функция экспоненциального типа* $\tau > 0$, *если для любого* $\varepsilon > 0$ *существует постоянная* A_ε, *такая что*

$$|F(z)| \le A_\varepsilon e^{(\tau+\varepsilon)|z|},\ z \in \mathbf{C}. \tag{18}$$

Если $f(x)$ имеет носитель на промежутке $[-\tau, \tau]$, то функция $F[f](z)$ является экспоненциального типа $\tau > 0$. Теорема *Пэли–Винера* утверждает, что верно и обратное при условии, что сужение функции $F(z)$ на вещественную ось принадлежит пространству $L^2(\mathbf{R})$.

Теорема 5. *Функция* $F(u) \in L^2(\mathbf{R})$ *является преобразованием Фурье некоторой функции* $f(x)$, *имеющей носитель на промежутке* $[-\tau, \tau]$, *тогда и только тогда, когда* $F(u)$ *является сужением на вещественную ось целой функции* $F(z)$ *экспоненциального типа* $\tau > 0$.

Доказательство см. в книге [СВ].

Из теоремы Пэли–Винера и из формулы $F^{-1}[f] = \frac{1}{2\pi}F[f(-x)]$ следует, что если $f(x) \in L^2(\mathbf{R})$ является сужением на вещественную ось целой функции $f(z)$ экспоненциального типа $\tau > 0$, то ее преобразование Фурье $\hat{f}(\omega)$ имеет носитель на промежутке $[-\tau, \tau]$.

Определение 3. *Функция* $f(x) \in L^2(\mathbf{R})$ *называется функцией с ограниченной шириной полосы* τ, *если ее преобразование Фурье* $\hat{f}(\omega)$ *равно нулю вне полосы частот* $[-\tau, \tau]$.

Другими словами, функция с ограниченной шириной полосы τ имеет преобразование Фурье $\hat{f}(\omega)$ с носителем на промежутке $[-\tau, \tau]$. По теореме Пэли–Винера функция с ограниченной шириной полосы τ продолжается с вещественной оси на всю комплексную плоскость как целая функция $f(z)$ экспоненциального типа $\tau > 0$.

1.3.6. Преобразование Фурье экспоненциально убывающей функции

Функция $\varphi(x)$ на $\mathbf{R}$ является экспоненциально убывающей, если она допускает оценку

$$|\varphi(x)| \le Ae^{-\lambda|x|},\ \lambda > 0. \tag{19}$$

Преобразование Фурье экспоненциально убывающей функции является, очевидно, бесконечно дифференцируемой функцией. Более того, оно определено не только для вещественных ω, но и для комплексных значений $z = u + iv$ параметра. Действительно, для

$$F[f] = \hat{f}(z) = \int_{-\infty}^{\infty} f(x) e^{-ixz}\, dx$$

оценим подынтегральную функцию:

$$|f(x)e^{-ixz}| \le Ae^{-\lambda|x|}e^{|xv|} = Ae^{(-\lambda+|v|)|x|}.$$

Поэтому интеграл существует при $-\lambda + |v| < 0$, то есть когда $\mathrm{Im}(z) = v \in (-\lambda, \lambda)$. Мы получили, что преобразование Фурье экспоненциально убывающей функции определено в полосе $\mathrm{Im}(z) \in (-\lambda, \lambda)$ и является в этой полосе аналитической функцией комплексного переменного.

1.3.7. Формула суммирования Пуассона

Теорема 6. *Пусть* $f(x) \in L^1(\mathbf{R})$. *Тогда ряд* $\sum_{k\in\mathbf{Z}} f(x + 2k\pi)$ *сходится абсолютно почти всюду к* 2π*-периодической функции* $S(x) \in L^1(0, 2\pi)$. *Кроме того, коэффициенты Фурье суммы* $S(x)$ *задаются формулой*

$$c_n = \frac{1}{2\pi}\hat{f}(n). \tag{20}$$

Доказательство. Первая часть теоремы следует из теоремы Леви [КФ] и оценки

$$\int_0^{2\pi} \sum_{k\in\mathbf{Z}} |f(x + 2k\pi)|\, dx = \int_{-\infty}^{\infty} |f(x)|\, dx < \infty.$$

Вычислим коэффициенты Фурье:

$$c_n = \frac{1}{2\pi}\int_0^{2\pi} e^{-inx}\left(\sum_{k\in\mathbf{Z}} f(x + 2k\pi)\right) dx = \frac{1}{2\pi}\sum_{k\in\mathbf{Z}}\int_0^{2\pi} e^{-inx} f(x + 2k\pi)\, dx =$$

$$= \frac{1}{2\pi}\sum_{k\in\mathbf{Z}}\int_{2\pi k}^{2\pi(k+1)} e^{-iny} f(y)\, dy = \frac{1}{2\pi}\int_{-\infty}^{\infty} e^{-iny} f(y)\, dy = \frac{1}{2\pi}\hat{f}(n).$$

Замечание 2. Из теоремы 6 вытекает необходимое и достаточное условие, чтобы сумма $S(x) = \sum_{k\in\mathbf{Z}} f(x + 2k\pi)$ была равна 1 п. в.:

$$\hat{f}(0) = 2\pi \quad \text{и} \quad \hat{f}(n) = 0 \quad \text{для} \quad n \in \mathbf{Z} \setminus \{0\}.$$

Поскольку мы вычислили коэффициенты Фурье, то можно составить ряд Фурье функции $S(x)$: $\sum_{n=-\infty}^{\infty} c_n e^{inx} = \frac{1}{2\pi}\sum_{n=-\infty}^{\infty} \hat{f}(n) e^{inx}$. Функция $S(x)$ принадлежит пространству $L^1(0, 2\pi)$, поэтому ряд Фурье не обязан сходиться. Но в случае его сходимости естественно ожидать равенства $S(x)$ сумме ряда Фурье.

Теорема 7 [Чуи]. *Пусть* $f(x) \in L^1(\mathbf{R})$ *обладает свойствами:*

- *ряд* $\sum_{k\in\mathbf{Z}} f(x+2k\pi)$ *сходится всюду к непрерывной функции* $S(x)$,
- *ряд* $\frac{1}{2\pi}\sum_{n=-\infty}^{\infty} \hat{f}(n)e^{inx}$ *сходится всюду.*

Тогда имеет место следующая формула суммирования Пуассона:

$$\sum_{k=-\infty}^{\infty} f(x+2k\pi) = \frac{1}{2\pi}\sum_{n=-\infty}^{\infty} \hat{f}(n)e^{inx}, \quad x \in \mathbf{R}, \tag{21}$$

в частности

$$\sum_{k=-\infty}^{\infty} f(2k\pi) = \frac{1}{2\pi}\sum_{n=-\infty}^{\infty} \hat{f}(n).$$

Замечание 3. Установленные выше теоремы верны и для сдвигов, кратных некоторому числу $a > 0$. В частности, имеют место формулы

$$\sum_{k=-\infty}^{\infty} f(x+k) = \sum_{n=-\infty}^{\infty} \hat{f}(2\pi n)e^{i2\pi nx}, \quad x \in \mathbf{R}, \tag{22}$$

$$\sum_{k=-\infty}^{\infty} f(k) = \sum_{n=-\infty}^{\infty} \hat{f}(2\pi n). \tag{23}$$

1.3.8. Оконное преобразование Фурье

Преобразование Фурье является эффективным средством исследования сигналов. Однако оно имеет недостаток, который заключается в слабой пространственной локализации. Это проявляется, во-первых, в том, что для нахождения преобразования Фурье $\hat{f}(\omega)$ необходимо знать сигнал $f(x)$ полностью на всей числовой прямой $\mathbf{R}$ (в прошлом и в будущем). Во-вторых, положение и величина «пиков» функции $f(x)$ не получаются легко из $\hat{f}(\omega)$, поскольку данное значение $\hat{f}(\omega)$ формируется с учетом всех значений функции $f(x)$. Улучшить пространственную локализацию можно за счет искусственной локализации сигнала. Для этого можно умножить сигнал $f(x)$ на подходящую функцию $g(x - x_0)$, которая имеет носитель вблизи точки x_0. В результате получаем преобразование

$$(T^{ok}f)(\omega, x_0) = \int_{-\infty}^{\infty} f(x)g(x-x_0)e^{-ix\omega}\,dx, \tag{24}$$

которое называется *оконным* преобразованием Фурье. Размер носителя функции $g(x)$ определяет ширину окна. В качестве функции $g(x)$ можно взять, например, такую:

$$g_{a,b}(x)=\begin{cases}1, & |x|\le a,\\ \cos^2(\pi b(|x|-a)), & a\le|x|\le a+1/2b\\ 0, & \text{для остальных } x\end{cases}$$

Более распространенным является выбор *функции Гаусса*

$$g_\alpha(x)=\frac{1}{2\sqrt{\pi\alpha}}e^{-\frac{x^2}{4\alpha}},$$

где $\alpha>0$ – некоторое фиксированное число. Хотя эта функция имеет некомпактный носитель, она экспоненциально убывает на бесконечности. Ширину окна можно охарактеризовать стандартным отклонением $\sigma=\sqrt{\alpha}$, считая α дисперсией. Таким образом, ширина окна равна $2\sqrt{\alpha}$. Формула

$$(G^\alpha_{x_0}f)(\omega)=\int_{-\infty}^{\infty}f(x)g_\alpha(x-x_0)e^{-ix\omega}\,dx \qquad (5)$$

называется *преобразованием Габора* функции $f(x)$. Более подробные сведения об оконном преобразовании Фурье и о преобразовании Габора можно найти в [НС] и в книге [Чуи], § 3.1 и 3.2.

1.4. Преобразование Фурье дискретных сигналов

Пусть $f(t)$ – непрерывная на $\mathbf{R}$ функция, принадлежащая пространству $L^1(\mathbf{R})$. Будем называть ее аналоговым сигналом. В случае практических применений обычно имеют дело с набором последовательных значений, получаемых в результате дискретизации непрерывного во времени сигнала (эта процедура называется оцифровкой аналогового сигнала). В этом параграфе мы определим аналог преобразования Фурье для дискретных сигналов.

1.4.1. Дискретизация

Пусть Δt – некоторый достаточно малый промежуток времени. Рассмотрим последовательность точек $t_n = n\Delta t$ и соответствующую последовательность

$$x_n = f(n\Delta t),\ n\in\mathbf{Z}$$

значений сигнала $f(t)$. Набор значений $\{x_n\}$ называется *выборкой* (sampling), или *оцифровкой*, сигнала $f(t)$.

Основной вопрос, который возникает при оцифровке, заключается в том, насколько хорошо выборочные данные представляют исходную функцию $f(t)$. Оказывается, что при использовании выборки $\{x_n\}$ могут быть потеряны некоторые гармонические составляющие функции $f(t)$ с достаточно большими частотами. Например, функция $f(t) = \sin(\omega t)$ при частоте $\omega = k\pi/\Delta t$, $k \in \mathbf{Z}$, имеет выборку

$$x_n = \sin(\omega n \Delta t)$$

такую же, что и постоянная функция $f(t) = 0$: $x_n = \sin(\pi k n) = 0$. Поэтому при оцифровке $\{x_n\}$ будут незамеченными гармоники $f(t) = \sin(\omega t)$ с частотой $\omega = k\pi/\Delta t$.

Определение 1. *Частотой Найквиста выборки* $\{x_n\}$ *с шагом* Δt *называется частота*

$$\omega_N = \pi/\Delta t.$$

Замечание 1. Это частота синусоиды $\sin(\omega_N t)$, которая принимает нулевые значения именно в точках отсчета времени $n\Delta t$:

$$\sin(\omega_N n\Delta t) = \sin(\pi n) = 0.$$

Это предельная частота, при которой может получиться нулевая оцифровка ненулевого сигнала.

Таким образом, если мы хотим восстановить сигнал $f(t)$ по выборочным данным $\{x_n\}$, то у нас возникнут проблемы с частотами выше, чем $\pi/\Delta t$. Можно надеяться на точное восстановление $f(t)$, если в частотном представлении функции $f(t)$ будут отсутствовать высокие частоты. Это предположение означает, что преобразование Фурье $\hat{f}(\omega)$ должно иметь компактный носитель, например лежащий на промежутке $[-\Omega, \Omega]$.

Теорема 1 (Котельникова–Шеннона)**.** *Предположим, что непрерывная абсолютно интегрируемая функция* $f(t)$ *на* $\mathbf{R}$ *имеет преобразование Фурье* $\hat{f}(\omega)$ *с компактным носителем, лежащим на промежутке* $[-\Omega, \Omega]$ *и имеющее кусочно-непрерывную производную* $\hat{f}(\omega)$. *Если шаг выборки* Δt *удовлетворяет условию* $\Delta t \le \pi/\Omega$, *тогда функция* $f(t)$ *может быть восстановлена по выборочным данным* $x_n = f(n\Delta t)$, $n \in \mathbf{Z}$, *по следующей формуле:*

$$f(t) = \sum_{n \in \mathbf{Z}} f(n\Delta t) \frac{\sin\left(\dfrac{\pi}{\Delta t} t - \pi n\right)}{\dfrac{\pi}{\Delta t} t - \pi n}. \qquad (1)$$

Доказательство. Из условий теоремы следует, что функция $f(t)$ может быть восстановлена по своему образу Фурье:

$$f(t)=\frac{1}{2\pi}\int_{-\infty}^{\infty}\hat{f}(\omega)e^{i\omega t}d\omega=\frac{1}{2\pi}\int_{-\Omega}^{\Omega}\hat{f}(\omega)e^{i\omega t}d\omega.$$

Разложим функцию $\hat{f}(\omega)$ в ряд Фурье на промежутке $[-\Omega, \Omega]$ и затем подставим ряд в интеграл:

$$\hat{f}(\omega)=\sum_{n=-\infty}^{\infty}c_n e^{-i\frac{\pi}{\Omega}\omega n},$$

где c_n – коэффициенты Фурье. Поскольку $\hat{f}(\omega)$ имеет носитель на конечном промежутке $[-\Omega, \Omega]$, то коэффициенты Фурье c_n могут быть выражены при помощи обратного преобразования Фурье следующим образом:

$$c_n=\frac{1}{2\Omega}\int_{-\Omega}^{\Omega}\hat{f}(\omega)e^{i\frac{\pi}{\Omega}\omega n}d\omega=\frac{2\pi}{2\Omega}\frac{1}{2\pi}\int_{-\infty}^{\infty}\hat{f}(\omega)e^{i\frac{\pi}{\Omega}\omega n}d\omega=\frac{\pi}{\Omega}f\left(\frac{\pi}{\Omega}n\right).$$

Ряд Фурье функции $\hat{f}(\omega)$ сходится равномерно, поэтому его можно проинтегрировать почленно. Получаем, с учетом полученных выражений для коэффициентов c_n:

$$f(t)=\frac{1}{2\pi}\int_{-\Omega}^{\Omega}\hat{f}(\omega)e^{i\omega t}d\omega=\frac{1}{2\pi}\int_{-\Omega}^{\Omega}\sum_{n=-\infty}^{\infty}\frac{\pi}{\Omega}f\left(\frac{\pi}{\Omega}n\right)e^{-i\frac{\pi}{\Omega}\omega n}e^{i\omega t}d\omega=$$

$$=\frac{1}{2\Omega}\sum_{n=-\infty}^{\infty}f\left(\frac{\pi}{\Omega}n\right)\int_{-\Omega}^{\Omega}e^{i\omega\left(t-\frac{\pi}{\Omega}n\right)}d\omega=\frac{1}{2\Omega}\sum_{n\in Z}f\left(\frac{\pi}{\Omega}n\right)\frac{2i\sin\left(t-\frac{\pi}{\Omega}n\right)\Omega}{i(t-\frac{\pi}{\Omega}n)}=$$

$$=\sum_{n\in Z}f\left(\frac{\pi}{\Omega}n\right)\frac{\sin(\Omega t-\pi n)}{\Omega t-\pi n}.$$

Таким образом, мы получили

$$f(t)=\sum_{n\in Z}f\left(\frac{\pi}{\Omega}n\right)\frac{\sin(\Omega t-\pi n)}{\Omega t-\pi n}, \qquad (2)$$

где предполагается, что $(\sin t)/t = 1$ при $t = 0$. Формула (1) получается при $\Delta t = \pi/\Omega$.

Полученная формула (2) для восстановления $f(t)$ называется разложением Котельникова–Шеннона. В частности, если преобразова-

ние Фурье $\hat{f}(\omega)$ имеет носитель на промежутке $[-\pi, \pi]$, то $\Delta t = 1$ и формула принимает вид

$$f(t) = \sum_{n \in Z} f(n) \frac{\sin \pi(t-n)}{\pi(t-n)}.$$

В дальнейшем нам потребуется формула

$$f(t) = \sum_{n \in Z} f(2n) \frac{\sin(t-2n)\pi/2}{(t-2n)\pi/2} \tag{3}$$

для случая, когда $\Delta t = 2$ и преобразование Фурье $\hat{f}(\omega)$ имеет носитель на промежутке $[-\pi/2, \pi/2]$. Из доказательства теоремы вытекает, что в этом случае $c_n = 2f(2n)$, поэтому на промежутке $[-\pi/2, \pi/2]$ имеем также

$$\hat{f}(\omega) = 2\sum_{n=-\infty}^{\infty} f(2n) e^{-i2n\omega}. \tag{4}$$

1.4.2. Дискретное преобразование Фурье сигнала длины *N*

Рассмотрим преобразование Фурье сигнала $f(t)$:

$$\hat{f}(\omega) = \int_{-\infty}^{\infty} f(t) e^{-i\omega t} dt.$$

Из определения интеграла следует, что при малых Δt мы будем иметь приближенное равенство

$$\hat{f}(\omega) = \int_{-\infty}^{\infty} f(t) e^{-i\omega t} dt \approx \sum_{n=-\infty}^{\infty} f(n\Delta t) e^{-i\omega n \Delta t} \Delta t.$$

Последняя формула дает основание для следующего определения.

Определение 2. *Преобразованием Фурье оцифрованного сигнала* $f(t)$ *называется ряд*

$$\hat{f}_d(\omega) = \Delta t \sum_{n=-\infty}^{\infty} x_n e^{-i\omega n \Delta t}, \tag{5}$$

где $x_n = f(n\Delta t)$ *– выборочные данные функции* $f(t)$.

Замечание 2. Достаточным условием сходимости ряда (5) является абсолютная сходимость ряда $\sum_n x_n$. Последнее условие не следует из абсолютной интегрируемости функции $f(t)$.

Замечание 3. В отличие от обычного преобразования Фурье, функция $\hat{f}_d(\omega)$ является периодической с периодом $T = 2\pi/\Delta t$, поэтому обращение преобразования (5) сводится к вычислению коэффициентов Фурье по формуле

$$x_n = \frac{1}{2\pi}\int_0^{\frac{2\pi}{\Delta t}} \hat{f}_d(\omega) e^{i\omega n\Delta t} d\omega. \qquad (6)$$

В практических применениях обычно имеют дело с конечными выборками значений

$x_n = f(n\Delta t)$, $n = 0, 1, ..., N - 1$.

Существует естественный способ сделать из конечного набора значений бесконечную последовательность. Для этого достаточно продолжить $\{x_n\}$ нулевыми значениями при остальных $n \in \mathbf{Z}$. Тогда формула (5) будет конечной:

$$\hat{f}_d(\omega) = \Delta t \sum_{n=0}^{N-1} x_n e^{-i\omega n\Delta t}.$$

Некоторая нелогичность данной формулы остается в том, что параметр ω является непрерывным, в отличие от конечного числа значений аргумента $t = n\Delta t$, в то время как в исходной формуле преобразования Фурье переменные ω и t были равноправными. Было бы правильнее, если бы при дискретном преобразовании Фурье дискретный набор значений $x_0, ..., x_{N-1}$ преобразовывался бы также в дискретный набор $y_0, ..., y_{N-1}$. Добиться этого несложно. Поскольку функция $\hat{f}_d(\omega)$ периодическая с периодом $2\pi/\Delta t$, то естественно взять дискретную серию частот

$$\omega_k = \frac{2\pi}{N\Delta t}k, \;\; k = 0, ..., N-1,$$

которые проходят промежуток $[0, 2\pi/\Delta t)$ с шагом $2\pi/(N\Delta t)$. Тогда получаем

$$y_k = \hat{f}_d(\omega_k) = \Delta t\sum_{n=0}^{N-1} x_n e^{-i\omega_k n\Delta t} = \Delta t\sum_{n=0}^{N-1} x_n e^{-i\frac{2\pi}{N\Delta t}kn\Delta t} = \Delta t\sum_{n=0}^{N-1} x_n e^{-i\frac{2\pi}{N}kn}.$$

В последней формуле удобно опустить Δt.

Определение 3. *Дискретным преобразованием Фурье массива $\{x_n\}$ конечной длины N называется массив $\{y_n\}$, элементы которого находятся по следующей формуле:*

$$y_k = \sum_{n=0}^{N-1} x_n e^{-i\frac{2\pi}{N}kn}, \quad k = 0, ..., N-1. \tag{7}$$

Аналогично определяется обратное преобразование Фурье:

$$x_n = \frac{1}{N}\sum_{k=0}^{N-1} y_k e^{i\frac{2\pi}{N}kn}. \tag{8}$$

Покажем обратимость введенных преобразований (7) и (8).

$$\frac{1}{N}\sum_{k=0}^{N-1} y_k e^{i\frac{2\pi}{N}kn} = \frac{1}{N}\sum_{k=0}^{N-1}\left(\sum_{p=0}^{N-1} x_p e^{-i\frac{2\pi}{N}kp}\right) e^{i\frac{2\pi}{N}kn} =$$

$$= \frac{1}{N}\sum_{k=0}^{N-1}\sum_{p=0}^{N-1} x_p e^{-i\frac{2\pi}{N}k(p-n)} = \frac{1}{N}\sum_{p=0}^{N-1} x_p \sum_{k=0}^{N-1} e^{-i\frac{2\pi}{N}k(p-n)} = x_n.$$

Последнее равенство следует из того, что внутренняя сумма принимает значения

$$\sum_{k=0}^{N-1} e^{-i\frac{2\pi}{N}k(p-n)} = \begin{cases} N, \text{ если } p = n \\ 0, \text{ если } p \neq n \end{cases}.$$

Первое значение очевидно, а второе находится по формуле суммы членов геометрической прогрессии со знаменателем $q = \exp(-i2\pi(p - n)/N)$.

Для дискретного преобразования Фурье легко устанавливается следующее равенство Парсеваля.

$$\sum_{k=0}^{N-1} |x_n|^2 = \frac{1}{N}\sum_{k=0}^{N-1} |y_k|^2. \tag{9}$$

Замечание 4. В определении 3 считается, что $\Delta t = 1$, тогда частота Найквиста $\omega_N = \pi$. Дискретное преобразование Фурье частоте $\omega_k = 2k\pi/N$ ставит в соответствие значение y_k. При этом частоте Найквиста $\omega_N = \pi$ соответствует номер $k = N/2$, то есть середина последовательности $\{y_k\}$, $k = 0, 1, ..., N$.

Замечание 5. Определенный выше конечный ряд значений y_k, $k = 0, 1, ..., N$, можно продолжить на все остальные целые k по периодичности, а именно считая, что $y_{k+Np} = y_k$, $k = 0, 1, ..., N-1$, $p \in \mathbf{Z}$. В частности, $y_{N-1} = y_{-1}$, $y_N = y_0$, $y_{N+1} = y_1$.

Замечание 6. Если числа $\{x_n\}$ вещественные, то из формулы (7) сразу следует, что $\overline{y}_k = y_{-k}$. Поэтому массив $\{y_k\}$ имеет следующую структуру. Очевидно, что $y_0 = \sum_{n=0}^{N-1} x_n$. Далее, до серединного значения $k = N/2$, соответствующего частоте Найквиста, идут некоторые значе-

ния y_k. После $k = N/2$ значения y_k будут повторяться симметрично относительно $N/2$ и с сопряжением. Например, если N четное, то $y_{N/2}$ – вещественное и

$$y_{(N/2)+1} = \overline{y_{(N/2)-1}}, \quad y_{(N/2)+2} = \overline{y_{(N/2)-2}}, \quad \ldots, \quad y_N = \overline{y_0} = \sum_{n=0}^{N-1} x_n.$$

Первые значения $y_0, y_1, ..., y_{N/2}$ содержат полную информацию относительно всего массива $\{y_k\}$.

1.4.3. Преобразование Фурье числовой последовательности

Обратимся снова к бесконечным выборкам. Пусть $x = \{x_n\}$ – числовая последовательность, $n \in \mathbf{Z}$, $x_n \in \mathbf{C}$. Предположим, что

$$\| x \|_1 = \sum_{n=-\infty}^{\infty} |x_n| < \infty. \tag{10}$$

Можно считать, что $\{x_n\}$ является оцифровкой кусочно-постоянной функции $f(t)$, $f(t) = x_n$, при $t \in [n, n + 1)$, $n \in \mathbf{Z}$. Тогда преобразование Фурье последовательности $\{x_n\}$ можно определить в соответствии с формулой (5) при $\Delta t = 1$.

Определение 4. *Преобразованием Фурье последовательности* $\{x_n\}$ *называется следующий ряд:*

$$X(\omega) = \sum_{n=-\infty}^{\infty} x_n e^{-i\omega n}. \tag{11}$$

Из сделанных предположений относительно $\{x_n\}$ следует, что ряд (11) сходится равномерно и его сумма $X(\omega)$ является непрерывной 2π-периодической функцией. Из (11) следует, что числа x_n – это коэффициенты Фурье функции $X(\omega)$. Поэтому формула обращения выглядит следующим образом:

$$x_n = \frac{1}{2\pi} \int_{-\pi}^{\pi} X(\omega) e^{i\omega n} d\omega. \tag{12}$$

Для преобразования Фурье числовой последовательности имеют место свойства, аналогичные свойствам преобразования Фурье функции. Отметим некоторые из них.

Свертка последовательностей $\{x_n\}$ и $\{y_n\}$ есть последовательность $\{z_n\}$, определенная равенством

$$z_n = \sum_{k=-\infty}^{\infty} x_k y_{n-k}.$$

Свойство 1 (Преобразование Фурье и свертка). *Пусть* $X(\omega)$ *– преобразование Фурье последовательности* $\{x_n\}$, *а* $Y(\omega)$ *– преобразование Фурье последовательности* $\{y_n\}$. *Тогда преобразование Фурье* $Z(\omega)$ *свертки последовательностей* $\{x_n\}$ *и* $\{y_n\}$ *равно произведению функций* $X(\omega)$ *и* $Y(\omega)$,

$$Z(\omega) = X(\omega)Y(\omega). \tag{13}$$

Для последовательностей $x = \{x_n\}$ и $y = \{y_n\}$ определим их *корреляцию* формулой

$$z_k = \sum_{n=-\infty}^{\infty} x_n \overline{y_{n-k}}.$$

Свойство 2 (Преобразование Фурье и корреляция). *Пусть* $X(\omega)$ *– преобразование Фурье последовательности* $\{x_n\}$, *а* $Y(\omega)$ *– преобразование Фурье последовательности* $\{y_n\}$. *Тогда преобразование Фурье* $Z(\omega)$ *корреляции последовательностей* $\{x_n\}$ *и* $\{y_n\}$ *выражается следующим образом:*

$$Z(\omega) = X(\omega)\,\overline{Y(\omega)}.$$

Доказательство.

$$X(\omega)\overline{Y(\omega)} = \left(\sum_{n=-\infty}^{\infty} x_n e^{-in\omega}\right)\left(\sum_{n=-\infty}^{\infty} \overline{y_m} e^{im\omega}\right) = \sum_{n,m} x_n \overline{y_m} e^{-i(n-m)\omega} =$$

$$[k = n - m] = \sum_k \sum_n x_n \overline{y_{n-k}}\, e^{-ik\omega} = \sum_k z_k\, e^{-ik\omega} = Z(\omega).$$

Следствие 1. *Преобразование Фурье автокорреляционной последовательности* $z_k = \sum_n x_n \overline{x_{n-k}}$ *равно квадрату модуля функции* $X(\omega)$:

$$Z(\omega) = |X(\omega)|^2.$$

Следствие 2. *Если преобразование Фурье* $X(\omega)$ *последовательности* $\{x_n\}$ *имеет постоянный модуль,* $|X(\omega)|^2 = c$, *то*

$$\sum_n x_n \overline{x_{n-k}} = c\delta_{k0},$$

где c *– положительная константа и* $\delta_{k,0}$ *– символ Кронекера.*

Доказательство. Пусть $c = |X(\omega)|^2$. Тогда из следствия 1 и из условия получаем

$$\sum_k z_k\, e^{-ik\omega} = Z(\omega) = |X(\omega)|^2 = c.$$

Из полноты тригонометрической системы следует, что $z_0 = c$, а при остальных k все $z_k = 0$. Поэтому

$$\sum_n x_n \overline{x_{n-k}} = z_k = c\delta_{k0}.$$

Следствие 3. *Если преобразование Фурье $X(\omega)$ последовательности $\{x_n\}$ имеет постоянный модуль, то последовательность является абсолютно некоррелированной.*

Свойство 3. *Равенство Парсеваля.*

$$\sum_{n=-\infty}^{\infty} |x_n|^2 = \frac{1}{2\pi}\int_{-\pi}^{\pi} |X(\omega)|^2 \, d\omega.$$

Замечание 7. Отметим, что числовые последовательности, используемые в приложениях, обычно имеют конечное число ненулевых членов, поэтому условие (10) всегда выполнено, если считать остальные члены нулями.

Замечание 8. Из теоремы Котельникова–Шеннона следует способ построения непрерывной функции $g(t)$, значения которой в целочисленных точках равны x_n. А именно:

$$g(t) = \sum_{n=-\infty}^{\infty} x_n \frac{\sin\pi(t-n)}{\pi(t-n)}. \tag{14}$$

В силу наших предположений ряд (14) сходится равномерно, и его сумма $g(t)$ является непрерывной функцией на **R**. Очевидно, что $g(n) = x_n$.

Функция $g(t)$ является обратным преобразованием Фурье функции $X(\omega)$, если считать, что последняя равна нулю вне промежутка $[-\pi, \pi]$. Действительно, преобразование Фурье функции $(\sin\pi t)/\pi t$ равно характеристической функции $\chi_{[-\pi,\pi]}(\omega)$ промежутка $[-\pi, \pi]$, тогда

$$F\left[\frac{\sin\pi(t-n)}{\pi(t-n)}\right] = \chi_{[-\pi,\pi]}(\omega)e^{-i\omega n}.$$

Поэтому преобразование Фурье функции $g(t)$ легко вычисляется:

$$\hat{g}(\omega) = \begin{cases} \sum_{n=-\infty}^{\infty} x_n e^{-i\omega n}, & \omega \in [-\pi, \pi], \\ 0, & \text{для остальных } \omega \end{cases}, \tag{15}$$

что дает нам на отрезке $[-\pi, \pi]$ ту же формулу (11) преобразования Фурье $X(\omega)$ числовой последовательности. Отличие заключается в том, что в формуле (11) функция $X(\omega)$ является 2π-периодической, а в формуле (15) функция $\hat{g}(\omega)$ обращается в нуль вне отрезка $[-\pi, \pi]$.

1.4.4. Z-преобразование

Рассмотрим преобразование Фурье числовой последовательности $X(\omega)=\sum_{n=-\infty}^{\infty} x_n e^{-i\omega n}$. Сделаем в этой формуле замену $z = e^{-i\omega}$, а затем будем считать, что это произвольная комплексная переменная $z \in \mathbf{C}$. Полученный формальный степенной ряд обозначим символом $X(z)$ и будем называть z-преобразованием последовательности $\{x_n\}$:

$$X(z)=\sum_{n=-\infty}^{\infty} x_n z^n. \qquad (16)$$

Переход к $X(z)$ удобен в том случае, когда последовательность $\{x_n\}$ имеет конечное число ненулевых элементов (в прикладных вопросах это обычная ситуация). В этом случае $X(z)$ представляет собой рациональную функцию. Нули и полюсы функции $X(z)$ являются тогда важными новыми характеристиками последовательности $\{x_n\}$.

Часто делают другую замену $z = e^{i\omega}$. Тогда z-преобразование последовательности $\{x_n\}$ имеет вид

$$X(z)=\sum_{n=-\infty}^{\infty} x_n z^{-n}.$$

1.5. Фильтры

Сигналом будем называть функцию $x(t)$, определенную на всей числовой прямой $\mathbf{R}$, аргумент t будем называть временем. Будем предполагать, что сигналы $x(t)$ абсолютно интегрируемы на $\mathbf{R}$ и имеют конечную L^2-норму $||x(t)||_2$. Квадрат этой нормы обычно называют энергией сигнала $x(t)$.

Поскольку мы будем постоянно использовать преобразование Фурье, то будет удобно в этом параграфе обозначать преобразование Фурье сигнала $x(t)$ соответствующей заглавной буквой:

$$X(\omega)=\int_{-\infty}^{\infty} x(t)e^{-i\omega t}\,dt.$$

1.5.1. Фильтрация непрерывных сигналов

Рассмотрим некоторое преобразование сигналов L: $x(t) \to y(t)$. Исходный сигнал $x(t)$ называется *сигналом на входе*, преобразованный сигнал $y(t)$ называется *сигналом на выходе*.

Оператор преобразования L: $x(t) \to y(t)$ называется *фильтром*, если он обладает двумя свойствами:

- линейность: $L(\alpha x_1 + \beta x_2) = \alpha L(x_1) + \beta L(x_2)$, $\forall\ \alpha, \beta \in \mathbf{C}$;
- инвариантность относительно сдвигов во времени: если $y(t) = L(x(t))$, то $\forall\ u \in \mathbf{R}$,

$$L(x(t+u)) = y(t+u).$$

Преобразование Фурье переводит сдвиг по времени в умножение на экспоненту $e^{i\omega u}$. Поэтому если мы перейдем к Фурье-образам $X(\omega) = F[x]$ и $Y(\omega) = F[y]$, то получаем:

$$\begin{array}{ccc} L\colon x(t+u) & \to & y(t+u) \\ \downarrow F & & \downarrow F \end{array}$$

$$\hat{L}\colon e^{i\omega u}X(\omega) \ \to\ e^{i\omega u}Y(\omega).$$

Поэтому соответствующее преобразование $\hat{L}$ Фурье-образов $\hat{L}\colon X(\omega) \to Y(\omega)$ будет обладать свойствами линейности и инвариантности относительно умножения на функции

$$\hat{L}(e^{i\omega u}X(\omega)) = e^{i\omega u}Y(\omega) = e^{i\omega u}\hat{L}(X(\omega)).$$

Отсюда следует [На], что действие оператора $\hat{L}\colon X(\omega) \to Y(\omega)$ заключается в умножении $X(\omega)$ на некоторую функцию $H_L(\omega)$:

$$Y(\omega) = H_L(\omega)\, X(\omega). \qquad (1)$$

Основные понятия, относящиеся к фильтрам

1. *Передаточная функция* $H_L(\omega)$. Это функция, определенная выше равенством (1) и осуществляющая фильтрацию сигналов на уровне Фурье-образов.
2. *Коэффициент усиления* фильтра. Это модуль $|H_L(\omega)|$ передаточной функции $H_L(\omega)$.
3. *Фаза* $\varphi(\omega) = \arg H_L(\omega)$.
4. *Пиковая частота* фильтра. Это частота ω_0, в которой коэффициент усиления имеет максимальное значение.
5. *Транспонированный* (*сопряженный*) *фильтр*. Это фильтр L^*, у которого передаточная функция является комплексно-сопряженной к $H_L(\omega)$.
6. *Импульсная функция отклика*. Это функция $h_L(t)$, такая что ее преобразование Фурье равно передаточной функции $H_L(\omega)$.

Действие фильтра L на сигнал $x(t)$ может быть представлено в виде свертки с импульсной функцией отклика $h_L(t)$:

$$y(t) = (h_L * x)(t) = \int_{-\infty}^{\infty} h_L(u)x(t-u)du. \qquad (2)$$

Это сразу следует из (1) и из того факта, что при преобразовании Фурье свертка переходит в произведение. Указанное название для $h_L(t)$ происходит из того факта, что функцию $h_L(t)$ можно считать образом единичного импульса, описываемого δ-функцией Дирака, $L: \delta(t) \to h_L(t)$. Действительно,

$$L(\delta) = (h_L * \delta)(t) = \int_{-\infty}^{\infty} h_L(u)\delta(t-u)du = h_L(t).$$

Напомним, что δ-функция определяется как функционал, который функции ставит в соответствие ее значение в точке $t = 0$.

Легко видеть, что сопряженный фильтр L^* имеет импульсную функцию отклика вида $\overline{h_L(-t)}$.

Используя передаточную функцию, можно выделить следующие типы фильтров.

Полосно-пропускающие фильтры. Фильтр, имеющий передаточную функцию вида

$$H(\omega) = \begin{cases} 1, & \text{если } |\omega - \omega_0| \le \Delta \\ 0, & \text{в противном случае} \end{cases}$$

называется идеальным фильтром с полосой пропускания ширины 2Δ, центрированной по частоте ω_0.

Полосно-заграждающие (режекторные). Фильтры задаются передаточной функцией вида

$$H(\omega) = \begin{cases} 0, & \text{если } |\omega - \omega_0| \le \Delta \\ 1, & \text{в противном случае} \end{cases}$$

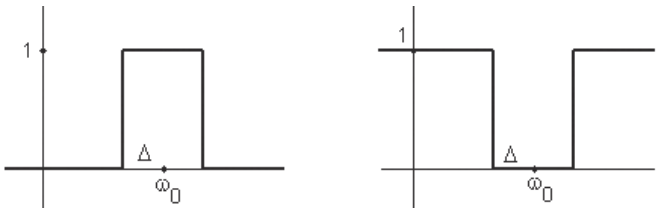


Рис. 1.5.1. Графики идеальных фильтров

1.5.2. Примеры фильтров

1. Сдвиг времени. Для любого числа $u \in \mathbf{R}$ пусть $y(t) = x_u(t) = x(t - u)$. Поскольку $X_u(\omega) = e^{-i\omega u}X(\omega)$, то передаточная функция $H(\omega) = e^{-i\omega u}$. Импульсная функция отклика является δ-функцией Дирака, $h(t) = \delta(t - u)$. Транспонированный фильтр имеет передаточную функцию $\overline{H}(\omega) = e^{i\omega u}$ и действует следующим образом:

$y(t) = x_{-u}(t) = x(t + u)$.

2. Свертка. Пусть $h(t)$ – некоторая абсолютно интегрируемая функция на $\mathbf{R}$. Определим преобразование сигнала $x(t)$ как свертку его с функцией $h(t)$:

$$y(t) = (h * x)(t) = \int_{-\infty}^{\infty} h(u)x(t-u)du. \tag{3}$$

Передаточная функция $H(\omega)$ есть преобразование Фурье функции $h(t)$, и, следовательно, $h(t)$ и есть импульсная функция отклика. Коэффициент усиления фильтра имеет следующую оценку:

$$\left|H(\omega)\right| \le \| h \|_1 = \int_{-\infty}^{\infty} \left|h(t)\right| dt.$$

Транспонированный фильтр:

$$y(t) = \int_{-\infty}^{\infty} \overline{h(u)}\, x(t+u)du.$$

3. Конечное усреднение. Пусть $h(t) = 1$ при $|\, t\,| \le a$ и $h(t) = 0$ при остальных t, где $a > 0$ – некоторое число. Пусть

$$y(t) = \frac{1}{2a}(h * x)(t) = \frac{1}{2a}\int_{-\infty}^{\infty} h(u)x(t-u)du = \frac{1}{2a}\int_{-a}^{a} x(t-u)du = \frac{1}{2a}\int_{t-a}^{t+a} x(\tau)d\tau.$$

Последний интеграл представляет собой среднее значение функции $x(t)$ на промежутке $[t - a, t + a]$. Передаточная функция в данном случае легко вычисляется как преобразование Фурье функции $h(t)/2a$:

$$H(\omega) = \frac{\sin \omega a}{\omega a}, \qquad \left|H(\omega)\right| \le \frac{1}{|\,\omega\,|\, a}.$$

Пиковая частота: $\omega_0 = 0$. Действие этого фильтра $Y(\omega) = H(\omega)X(\omega)$ в частотной области приводит к тому, что в области частот, близких к нулю, сигнал $X(\omega)$ меняется мало, а в области высоких частот, когда ω удалено от нуля, происходит подавление сигнала $X(\omega)$ за счет того,

что $|H(\omega)| \le (|\omega|a)^{-1} \to 0$ при $\omega \to \infty$. Поэтому такой фильтр называется низкочастотным: низкие частоты он пропускает, а высокие – подавляет. Во временно́й области локальное усреднение функции приводит к сглаживанию сигнала $x(t)$.

4. Усредненная разность. Пусть $h(t) = \mathrm{sign}(t)$, если $|t| \le a$ и $h(t) = 0$ при остальных t, где $a > 0$ – некоторое число. Пусть

$$y(t) = \frac{1}{2a}(h * x)(t) = \frac{1}{2a}\int_{-\infty}^{\infty} h(u)x(t-u)du =$$

$$= \frac{1}{2a}\int_{-a}^{0} -x(t-u)du + \frac{1}{2a}\int_{0}^{a} x(t-u)du = \frac{1}{2a}\int_{0}^{a}(x(t-u) - x(t+u))du.$$

Последний интеграл представляет собой среднее разности $x(t-u) - x(t+u)$, когда параметр u меняется на промежутке $[0, a]$. Передаточная функция имеет вид:

$$H(\omega) = \frac{\sin^2(a\omega/2)}{ia\omega/2}, \quad |H(\omega)| \le \frac{2}{|\omega|a}.$$

Действие этого фильтра в частотной области, $Y(\omega) = H(\omega)X(\omega)$, приводит к тому, что в области частот, близких к нулю, и для больших частот сигнал $X(\omega)$ подавляется. Фильтр пропускает частоты, близкие к пиковой частоте $\omega 0 = \pm 2t_0/a$, где $t_0 = 1{,}1656$ – приближенное решение уравнения $\mathrm{tg}\, t = 2t$.

5. Конечно-разностная производная. Пусть $a > 0$ – некоторое число. Определим фильтр следующей формулой:

$$y(t) = \frac{x(t) - x(t-a)}{a}.$$

Находим передаточную функцию:

$$Y(\omega) = \frac{1}{a}(X(\omega) - e^{-i\omega a}X(\omega)) = \frac{1}{a}(1 - e^{-i\omega a})X(\omega);$$

$$H(\omega) = \frac{1}{a}(1 - e^{-i\omega a}) = \frac{2i\sin\dfrac{a\omega}{2}}{a}e^{-i\omega\frac{a}{2}}, \quad |H(\omega)| = \frac{2}{a}\left|\sin\frac{a\omega}{2}\right|.$$

Эта передаточная функция сосредоточена в окрестностях частот $\omega = (2k+1)\pi/a$, $k \in \mathbf{Z}$. Сопряженный фильтр имеет передаточную функцию $\overline{H}(\omega) = (1 - e^{i\omega a})/a$ и действует по формуле

$$y(t) = \frac{x(t) - x(t+a)}{a}.$$

6. Дифференциальный оператор. Предположим, что сигнал на входе фильтра представлен дифференцируемой функцией $x(t)$. Определим фильтр равенством

$$y(t) = \frac{d}{dt}x(t) + bx(t), \tag{4}$$

где b – некоторое число. Чтобы найти передаточную функцию, сделаем в (4) преобразование Фурье $Y(\omega) = i\omega X(\omega) + bX(\omega) = (i\omega + b)X(\omega)$. Поэтому

$H(\omega) = i\omega + b$.

Мы видим, что передаточная функция дифференциального оператора $D = d/dt + b$ есть его полный символ. Сопряженный фильтр определяется сопряженным дифференциальным оператором $D^* = -d/dt + \bar{b}$. Импульсная функция отклика выражается через δ-функцию и ее производную, $h(t) = \delta' + b\delta$.

7. Решение дифференциального уравнения. Предположим теперь, что сигнал на входе $x(t)$ – это правая часть дифференциального уравнения

$$\frac{dy(t)}{dt} + by(t) = x(t), \tag{5}$$

а сигнал на выходе $y(t)$ – это решение уравнения (5) (предполагаются условия, обеспечивающие существование и единственность решения). С учетом предыдущего примера легко видеть, что передаточная функция имеет вид

$$H(\omega) = \frac{1}{i\omega + b}.$$

Функция Грина (фундаментальное решение) уравнения (5) является импульсной функцией отклика.

Замечание 1. Если в качестве фильтра рассматривается дифференциальный оператор степени n с постоянными коэффициентами, то его передаточная функция $H(\omega)$ является многочленом от ω степени n с комплексными коэффициентами. Наоборот, если $y(t)$ – это решение дифференциального уравнения степени n с постоянными коэффициентами, то передаточная функция $H(\omega)$ является рациональной дробью, числитель которой равен единице, а знаменатель – многочлен степени n с постоянными коэффициентами.

Во всех рассмотренных примерах мы сначала задаем фильтр, а затем находим его передаточную функцию. В теории фильтрации ши-

роко распространен и другой подход, когда сначала задается передаточная функция, обеспечивающая необходимые свойства фильтрации в области частот, а затем по ней восстанавливается фильтр. С этой точки зрения, учитывая примеры 6 и 7 и замечание к ним, большой класс фильтров задается рациональными передаточными функциями вида

$$H(\omega) = \frac{P_N(\omega)}{Q_K(\omega)} = \frac{a_0\omega^N + a_1\omega^{N-1} + \cdots + a_{N-1}\omega + a_N}{b_0\omega^K + b_1\omega^{K-1} + \cdots + b_{K-1}\omega + b_K}. \qquad (6)$$

1.5.3. Цифровые фильтры

В этом случае сигнал на входе $\{x_n\}$ и сигнал на выходе $\{y_n\}$ являются дискретными. Можно считать, что сигнал $\{x_n\}$ является оцифровкой аналогового сигнала $x(t)$, $x_n = x(n\Delta t)$, $n \in \mathbf{Z}$. С теоретической точки зрения, шаг выборки Δt не играет большого значения. Поэтому в дальнейшем будем считать, что $\Delta t = 1$.

Понятие фильтра и другие, связанные с ним основные понятия полностью применимы и в случаях дискретного сигнала $\{x_n\}$. Отличие заключается в том, что преобразование Фурье дискретного сигнала является 2π-периодической функцией

$$X(\omega) = \sum_{n=-\infty}^{\infty} x_n e^{-i\,n\,\omega}.$$

Передаточная функция $H(\omega) = Y(\omega)/X(\omega)$ также является 2π-периодической, а ее импульсная функция отклика представлена числовой последовательностью $\{h_n\}$, такой что

$$H(\omega) = \sum_{n=-\infty}^{\infty} h_n e^{-i\,n\,\omega}.$$

Последовательность $\{h_n\}$ называется *импульсной характеристикой* фильтра. Термин «импульсная характеристика» отражает тот факт, что коэффициенты $\{h_n\}$ составляют сигнал на выходе, когда сигнал на входе есть единичный импульс: $x_0 = 1$ и $x_n = 0$ при $n \neq 0$.

Отметим, что сопряженный фильтр имеет передаточную функцию $\overline{H(\omega)}$ и импульсную характеристику вида $h_n^* = \overline{h_{-n}}$.

В частотной области действие фильтра заключается в умножении на передаточную функцию $H(\omega)$:

$$Y(\omega) = H(\omega)\,X(\omega). \qquad (7)$$

Поскольку преобразование Фурье свертки последовательностей $\{h_n\}$ и $\{x_n\}$ является произведением $H(\omega)\,X(\omega)$ их преобразований

Фурье, то для значений $\{x_n\}$ сигнала действие фильтра выражается сверткой

$$y_n = \sum_{k=-\infty}^{\infty} h_k x_{n-k}. \qquad (8)$$

Поэтому в дальнейшем под *цифровым фильтром* мы будем понимать либо функцию $H(\omega)$, преобразующую образ Фурье $X(\omega)$ сигнала $\{x_n\}$ по формуле (7), либо последовательность $\{h_n\}$, $n \in \mathbf{Z}$, преобразующую сигнал $\{x_n\}$ по формуле (8).

Говорят, что фильтр имеет *конечную импульсную характеристику*, если он имеет конечное число ненулевых коэффициентов $\{h_n\}$. В противном случае фильтр имеет бесконечную импульсную характеристику. Такие фильтры для краткости называют КИХ- и БИХ-фильтами соответственно.

1.5.4. Примеры цифровых фильтров

1. Сдвиг. Задается формулой $y_n = x_{n-k}$, где k – некоторое целое число. Фильтр имеет единственный ненулевой коэффициент $h_k = 1$ и передаточную функцию $H(\omega) = e^{-i\omega u}$. Транспонированный фильтр действует по формуле $y_n = x_{n+k}$ и имеет единственный ненулевой коэффициент $h^*_{-k} = 1$.

2. Скользящее среднее. Результатом действия фильтра является вычисление среднего арифметического значения N значений:

$$y_n = \frac{1}{N}\sum_{k=0}^{N-1} x_{n-k} = \frac{1}{N}(x_n + x_{n-1} + ... + x_{n-N+1}).$$

Ненулевые коэффициенты фильтра: $h_k = 1/N$, $k = 0, 1, ..., N-1$. Передаточная функция фильтра:

$$H(\omega) = \frac{1}{N}(1 + e^{-i\omega} + ... + e^{-i(N-1)\omega}).$$

В z-представлении, заменяя $e^{-i\omega}$ комплексной переменной z, получаем многочлен $H(\omega) = (1 + z + ... + z^{N-1}) / N$. Транспонированный фильтр действует по формуле $y_n = N^{-1}\sum_{k=0}^{N-1} x_{n+k}$ и имеет коэффициенты $h^*_k = 1 / N$, $k = 0, -1, ..., -N + 1$.

3. Конечная разность. Пусть $y_n = x_n - x_{n-1}$. Ненулевые коэффициенты фильтра $h_0 = 1$ и $h_1 = -1$. Передаточная функция:

$$H(\omega) = 1 - e^{-i\omega}, \quad |H(\omega)| = 2|\sin(\omega/2)|.$$

Транспонированный фильтр $\overline{H}(\omega)$ имеет ненулевые коэффициенты $h^*_0 = 1$, $h^*_{-1} = -1$ и действует по формуле $y_n = x_n - x_{n+1}$.

4. Рекурсивный фильтр первого порядка. Определяется формулой

$$y_n = \sum_{k=0}^{N-1} a_k x_{n-k} - b y_{n-1},$$

где a_k и b – некоторые числа. Название *рекурсивный* отражает тот факт, что значение y_n вычисляется через предыдущее значение y_{n-1}. Данный фильтр может быть представлен в виде композиции следующих двух фильтров. Сначала действует фильтр $z_n = \sum_{k=0}^{N-1} a_k x_{n-k}$, а затем сигнал на выходе y_n находится как сигнал, удовлетворяющий соотношению $y_n + by_{n-1} = z_n$. Последнее условие является дискретным аналогом дифференциального уравнения 1-го порядка (см. пример 7 из предыдущего пункта). Найдем передаточную функцию

$$Y(\omega) = \sum_{n=-\infty}^{\infty} y_n e^{-in\omega} = \sum_{n\in Z}\left(\sum_{k=0}^{N-1} a_k x_{n-k} - b y_{n-1}\right) e^{-i\omega n} =$$

$$= \sum_{n\in Z}\left(\sum_{k=0}^{N-1} a_k e^{-i\omega k} x_{n-k} e^{-i\omega(n-k)} - b e^{-i\omega} y_{n-1} e^{-i\omega(n-1)}\right) =$$

$$= \sum_{k=0}^{N-1} a_k e^{-i\omega k} X(\omega) - b e^{-i\omega} Y(\omega).$$

$$Y(\omega)(1 + be^{-i\omega}) = \sum_{k=0}^{N-1} a_k e^{-i\omega k} X(\omega).$$

Поэтому

$$H(\omega) = \frac{1}{1 + be^{-i\omega}} \sum_{k=0}^{N-1} a_k e^{-i\omega k}, \quad H(z) = \frac{a_0 + a_1 z + \ldots + a_{N-1} z^{N-1}}{1 + bz}.$$

5. Общий рекурсивный фильтр порядка M определяется формулой

$$y_n = \sum_{k=0}^{N-1} a_k x_{n-k} - \sum_{p=1}^{M} b_p y_{n-p},$$

где a_k, b_k – некоторые числа. Передаточная функция $H(\omega)$, в том числе и в z-представлении $H(z)$, имеет вид

$$H(\omega) = \frac{\sum_{k=0}^{N-1} a_k e^{-i\omega k}}{1 + \sum_{p=1}^{M} b_p e^{-i\omega p}}, \qquad H(z) = \frac{a_0 + a_1 z + \ldots + a_{N-1} z^{N-1}}{1 + b_1 z + \ldots + b_M z^M}.$$

Последнее выражение считается наиболее общим видом передаточной функции.

1.6. Разложение сигнала на низкочастотную и высокочастотную составляющие

Преобразование Фурье числовой последовательности является 2π-периодической функцией. Поэтому когда говорят о низких или высоких частотах сигнала, имеют в виду их близость к нулю или к $\pm\pi$, соответственно. Дело еще и в том, что когда мы рассматриваем преобразование Фурье последовательности $\{x_n\}$, то мы считаем шаг выборки единичным, $\Delta t = 1$. В этом случае частота Найквиста (предельная частота) равна π.

Фильтр $\{h_n\}$, $n \in \mathbf{Z}$, называется *низкочастотным*, если модуль его передаточной функции $H(\omega)$ принимает большие значения в основном на отрезке $[-\pi/2, \pi/2]$. Фильтр $\{g_n\}$, $n \in \mathbf{Z}$, называется *высокочастотным*, если модуль его передаточной функции $G(\omega)$ принимает большие значения в основном на $[-\pi, -\pi/2]\cup[\pi/2, \pi]$. Здесь мы имеем в виду значения 2π-периодических функций $H(\omega)$ и $G(\omega)$ только на промежутке $[-\pi, \pi]$.

Пример 1. Фильтр с передаточной функцией $H(\omega) = \cos^2(\omega/2)$ является низкочастотным, а фильтр $G(\omega) = \sin^2(\omega/2)$ – высокочастотным. Эти фильтры имеют следующие ненулевые коэффициенты: $h_{-1} = 1/4$, $h_0 = 1/2$, $h_1 = 1/4$ и $g_{-1} = -1/4$, $g_0 = 1/2$, $g_1 = -1/4$.

Пример 2. Идеальные низкочастотный $H_0(\omega)$ и высокочастотный $G_0(\omega)$ фильтры задаются (на основном промежутке $[-\pi, \pi]$) передаточными функциями вида

$$H_0(\omega)=\begin{cases}1, & \omega\in[-\pi/2,\pi/2],\\ 0, & \omega\in[-\pi,-\pi/2)\cup(\pi/2,\pi]\end{cases}, \qquad G_0(\omega)=\begin{cases}0, & \omega\in[-\pi/2,\pi/2],\\ 1, & \omega\in[-\pi,-\pi/2)\cup(\pi/2,\pi]\end{cases}. \tag{1}$$

Раскладывая функции $H_0(\omega)$ и $G_0(\omega)$ в ряды Фурье, $H_0(\omega) = \sum_{n\in Z} h_n^0 e^{-i\omega n}$ и $G_0(\omega) = \sum_{n\in Z} g_n^0 e^{-i\omega n}$, мы находим коэффициенты идеальных фильтров:

$$h_n^0=\begin{cases}\dfrac{1}{2}, & n=0,\\ 0, & n=2k, k\neq 0,\\ \dfrac{(-1)^k}{(2k+1)\pi}, & n=2k+1\end{cases}, \qquad g_n^0=\begin{cases}\dfrac{1}{2}, & n=0,\\ 0, & n=2k, k\neq 0,\\ \dfrac{(-1)^{k+1}}{(2k+1)\pi}, & n=2k+1.\end{cases} \tag{2}$$

Отметим, что достаточно найти $\{h_n^0\}$, тогда коэффициенты $\{g_n^0\}$ определяются из соотношения $G_0(\omega) = 1 - H_0(\omega)$.

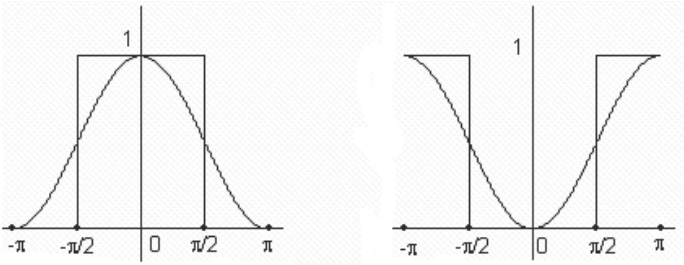


Рис. 1.6.1. Графики низкочастотных и высокочастотных фильтров

Пусть $\{x_n\}_{n\in\mathbf{Z}}$ – некоторый сигнал. В целях его анализа часто бывает целесообразно выделить низкочастотную $\{x_n^H\}$ и высокочастотную $\{x_n^G\}$ части сигнала $\{x_n\}$. Понятно, что кроме процедуры разложения должна быть еще и процедура восстановления сигнала $\{x_n\}$ по его компонентам. Рассмотрим сначала разложение и восстановление сигнала идеальными фильтрами.

1.6.1. Разложение идеальными фильтрами

Разложим сигнал $\{x_n\}_{n\in\mathbf{Z}}$ на низкочастотную и высокочастотную составляющие $\{x_n^H\}$ и $\{x_n^G\}$ идеальными фильтрами. С этой целью рассмотрим преобразование Фурье сигнала $\{x_n\}$:

$$X(\omega)=\sum_{n\in Z} x_n e^{-i\omega n}.$$

Если $H_0(\omega)$ и $G_0(\omega)$ – идеальные низкочастотный и высокочастотный фильтры, то $H_0(\omega) + G_0(\omega) = 1$, и тогда

$$X(\omega) = H_0(\omega)X(\omega) + G_0(\omega)X(\omega) = X^H(\omega) + X^G(\omega).$$

Таким образом, мы разложили сигнал в области частот. Для того чтобы выделить, например, низкочастотную часть $\{x_n^H\}$ сигнала $\{x_n\}$, нужно у полученного выражения

$$X^H(\omega) = H_0(\omega)\, X(\omega) \qquad (3)$$

найти коэффициенты Фурье $\{x_n^H\}$, то есть такие коэффициенты, что

$$X^H(\omega)=\sum_{n\in Z} x_n^H e^{-i\,\omega n}.$$

Из (3) следует, что эти коэффициенты находятся в виде свертки

$$x_n^H=\sum_{k\in Z} h_k^0 x_{n-k}, \qquad (4)$$

где h_k^0 – коэффициенты идеального низкочастотного фильтра $H_0(\omega)$.

Для того чтобы выделить высокочастотную часть $\{x_n^G\}$ сигнала $\{x_n\}$, нужно у полученного выражения

$$X^G(\omega) = G_0(\omega)\, X(\omega) \tag{5}$$

найти коэффициенты Фурье $\{x_n^G\}$, то есть такие коэффициенты, что

$$X^G(\omega) = \sum_{n\in Z} x_n^G e^{-i\,\omega n}.$$

Из (5) следует, что эти коэффициенты находятся в виде свертки

$$x_n^G = \sum_{k\in Z} g_k^0 x_{n-k}, \tag{6}$$

где g_k^0 – коэффициенты идеального высокочастотного фильтра $G_0(\omega)$.

При разложении сигнала на две компоненты $\{x_n\} \to \{x_n^H\}, \{x_n^G\}$ получается, что общее количество полученных данных $\{x_n^H, x_n^G\}$ в два раза больше, чем у первоначального сигнала $\{x_n\}$. Однако оказывается, что элементы с нечетными номерами из массивов $\{x_n^H\}$ и $\{x_n^G\}$ можно удалить без потери информации.

Такая процедура удаления из массивов элементов с нечетными номерами называется диадической *децимацией* и обозначается символом $2\downarrow$.

Обоснование децимации. По теореме Котельникова–Шеннона по массиву данных $\{x_n\}$ однозначно восстанавливается непрерывная функция $x(t)$ формулой

$$x(t) = \sum_{n\in Z} x_n \frac{\sin\pi(t-n)}{\pi(t-n)}, \tag{7}$$

значения которой в целочисленных точках равны x_n, то есть $x(n) = x_n$, и для которой преобразование Фурье $\hat{x}(\omega)$ имеет носитель на $[-\pi, \pi]$. Оказывается, что на этом промежутке преобразование Фурье $x^G(n) = x_n^G$ функции $x(t)$ совпадает с преобразованием Фурье $X(\omega)$ исходного дискретного сигнала $\{x_n\}$:

$$\hat{x}(\omega) = F\left[\sum_{n\in Z} x_n \frac{\sin\pi(t-n)}{\pi(t-n)}\right] = \sum_{n\in Z} x_n F\left[\frac{\sin\pi(t-n)}{\pi(t-n)}\right] = \sum_{n\in Z} x_n e^{-in\omega} F\left[\frac{\sin\pi t}{\pi t}\right] =$$

$$= \sum_{n\in Z} x_n e^{-in\omega} \chi_{[-\pi,\pi]}(\omega) = \begin{cases} X(\omega), & \omega\in[-\pi,\pi] \\ 0, & \text{для других } \omega \end{cases}.$$

В последней формуле символом $\chi_{[-\pi,\,\pi]}(\omega)$ обозначена характеристическая функция промежутка $[-\pi, \pi]$.

Совершенно аналогично, по массивам данных $\{x_n^H\}$ и $\{x_n^G\}$, однозначно восстанавливаются непрерывные функции $x^H(t)$ и $x^G(t)$ формулами

$$x^H(t)=\sum_{n\in Z}x_n^H\frac{\sin\pi(t-n)}{\pi(t-n)}\quad\text{и}\quad x^G(t)=\sum_{n\in Z}x_n^G\frac{\sin\pi(t-n)}{\pi(t-n)},\tag{8}$$

такие что $x^H(n)=x_n^H$ и $x^G(n)=x_n^G$. При этом их преобразования Фурье $\hat{x}^H(\omega)$ и $\hat{x}^G(\omega)$ совпадают на промежутке $[-\pi,\pi]$ с компонентами разложения $X^H(\omega)$ и $X^G(\omega)$:

$$\hat{x}^H(\omega)=\begin{cases}X^H(\omega), & \omega\in[-\pi,\pi]\\ 0, & \omega\notin[-\pi,\pi]\end{cases}=\begin{cases}X(\omega), & \omega\in[-\pi/2,\pi/2]\\ 0, & \text{для других }\omega\end{cases},\tag{9}$$

$$\hat{x}^G(\omega)=\begin{cases}X^G(\omega), & \omega\in[-\pi,\pi]\\ 0, & \omega\notin[-\pi,\pi]\end{cases}=\begin{cases}X(\omega), & \omega\in[-\pi,-\pi/2]\cup[\pi/2,\pi]\\ 0, & \text{для других }\omega\end{cases}.\tag{10}$$

Отметим, что разница между $X^H(\omega)$, $X^G(\omega)$ и $\hat{x}^H(\omega)$, $\hat{x}^G(\omega)$ заключается в том, что первые функции $X^H(\omega)$, $X^G(\omega)$ являются 2π-периодическими, а две другие, $\hat{x}^H(\omega)$ и $\hat{x}^G(\omega)$, имеют носитель на $[-\pi,\pi]$ (см. рис. 1.6.2).

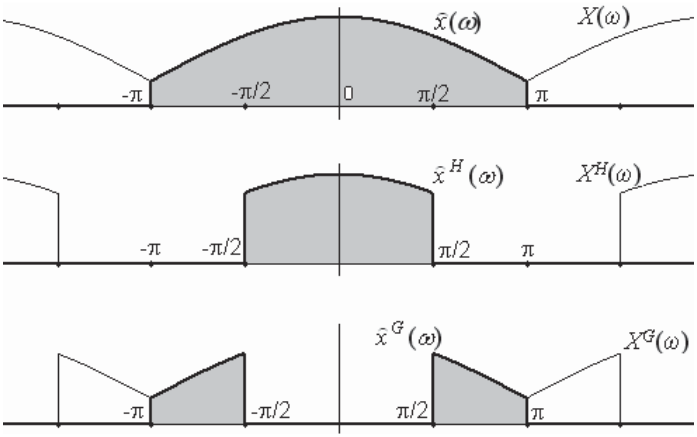


Рис. 1.6.2. Графики функций $X(\omega)$, $\hat{x}(\omega)$, $\hat{x}^H(\omega)$ и $\hat{x}^G(\omega)$

Мы видим, что функция $x^H(t)$ имеет преобразование Фурье $\hat{x}^H(\omega)$ с носителем на промежутке $[-\pi/2,\pi/2]$ – в два раза меньше, чем $[-\pi,\pi]$. Тогда, по теореме Котельникова, функция $x^H(t)$ может быть восстановлена по значениям $x^H(2n)=x_{2n}^H$, по формуле (3) параграфа 1.4:

$$x^H(t) = \sum_{n\in Z} x^H_{2n} \frac{\sin(t-2n)\pi/2}{(t-2n)\pi/2}. \qquad (11)$$

Поэтому все элементы с нечетными номерами x^H_{2n+1} могут быть без ущерба удалены: они могут быть найдены как значения функции $x^H(t)$ при $t = 2n + 1$, $x^H_{2n+1} = x^H(2n + 1)$.

Совершенно аналогично функция $x^G(t)$ имеет преобразование Фурье $\widehat{x}^G(\omega)$ с носителем на промежутке $[-\pi,-\pi/2]\cup[\pi/2,\pi]$ – в два раза меньше, чем $[-\pi, \pi]$. По теореме Котельникова $x^G(t)$ также может быть восстановлена по значениям $x^G(2n) = x^G_{2n}$, но уже по другой формуле:

$$x^G(t) = \sum_{n\in Z} x^G_{2n} \frac{\sin(t-2n)\pi/2}{(t-2n)\pi/2}(2\cos(t-2n)\frac{\pi}{2} - 1). \qquad (12)$$

Поэтому все элементы с нечетными номерами x^G_{2n+1} могут быть без ущерба удалены, поскольку они могут быть найдены как значения функции $x^G(t)$ при $t = 2n + 1$, $x^G_{2n+1} = x^G(2n + 1)$.

Установим формулу (12). Поскольку идеальный высокочастотный фильтр $G_0(\omega)$ обращается в нуль на промежутке $[-\pi/2, \pi/2]$, то преобразование Фурье $\widehat{x}^G(\omega)$ высокочастотной компоненты $x^G(t)$ имеет носитель на промежутке $[-\pi, -\pi/2]\cup[\pi/2, \pi]$. Поэтому функция $x^G(t)$ может быть найдена обратным преобразованием Фурье функции $\widehat{x}^G(\omega)$ в виде:

$$x^G(t) = \frac{1}{2\pi} \int\limits_{\frac{\pi}{2}\le|\omega|\le\pi} \hat{x}^G(\omega) e^{i\omega t} d\omega.$$

Для вычисления последнего интеграла используем разложение функции $\widehat{x}^G(\omega)$ в ряд Фурье на промежутке $[-\pi, -\pi/2]\cup[\pi/2, \pi]$. Для этого функцию $\widehat{x}^G(\omega)$ продолжим с промежутка $[-\pi, -\pi/2]\cup[\pi/2, \pi]$ на **R** как π-периодическую функцию.

Затем разложим ее в ряд Фурье на промежутке $[-\pi/2, \pi/2]$. Из π-периодичности следует, что это же разложение будет иметь место и на промежутке $[-\pi, -\pi/2]\cup[\pi/2, \pi]$.

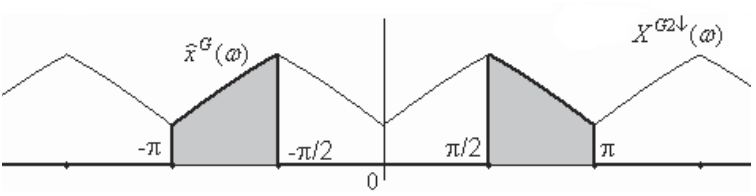


Рис. 1.6.3. Графики функций $\widehat{x}^G(\omega)$ и $X^{G2\downarrow}(\omega)$

Пусть $X^{G2\downarrow}(\omega)$ есть π-периодическое продолжение функции $\hat{x}^G(\omega)$ с промежутка $[-\pi, -\pi/2]\cup[\pi/2, \pi]$ на $\mathbf{R}$. Тогда на $[-\pi, -\pi/2]\cup[\pi/2, \pi]$ имеет место равенство $\hat{x}^G(\omega) = X^{G2\downarrow}(\omega)$. Разложим функцию $X^{G2\downarrow}(\omega)$ в ряд Фурье на $[-\pi/2, \pi/2]$:

$$X^{G2\downarrow}(\omega) = \sum_{n\in Z} c_{2n} e^{-i2n\omega}.$$

Для вычисления коэффициентов Фурье применим (как и в теореме Котельникова) обратное преобразование Фурье:

$$c_{2n} = \frac{1}{\pi}\int_{-\pi/2}^{\pi/2} X^{G2\downarrow}(\omega) e^{i2n\omega} d\omega = \left[\begin{array}{c}\text{в силу}\\ \pi\text{-периодичности}\end{array}\right] =$$

$$= \frac{1}{\pi}\left(\int_{\pi/2}^{\pi} \hat{x}^G(\omega) e^{i2n\omega} d\omega + \int_{-\pi}^{-\pi/2} \hat{x}^G(\omega) e^{i2n\omega} d\omega\right) = 2\frac{1}{2\pi}\int_{-\infty}^{\infty} \hat{x}^G(\omega) e^{i2n\omega} d\omega = 2x^G(2n).$$

Таким образом, на промежутке $[-\pi, -\pi/2]\cup[\pi/2, \pi]$ имеем:

$$\hat{x}^G(\omega) = X^{G2\downarrow}(\omega) = 2\sum_{n\in Z} x^G(2n) e^{-i2n\omega} = 2\sum_{n\in Z} x^G_{2n} e^{-i2n\omega}.$$

Значит, высокочастотная компонента также определяется только четными элементами. Теперь функция $x^G(t)$ может быть найдена обратным преобразованием Фурье функции $\hat{x}^G(\omega)$ с учетом ее разложения в ряд на носителе $[-\pi, -\pi/2]\cup[\pi/2, \pi]$:

$$x^G(t) = \frac{1}{2\pi}\int_{\frac{\pi}{2}\le|\omega|\le\pi} \hat{x}^G(\omega) e^{i\omega t} d\omega = \frac{1}{2\pi}\int_{\frac{\pi}{2}\le|\omega|\le\pi} 2\sum_{n\in Z} x^G_{2n} e^{-i2n\omega} e^{i\omega t} d\omega =$$

$$= \frac{1}{\pi}\sum_{n\in Z} x^G_{2n} \int_{\frac{\pi}{2}\le|\omega|\le\pi} e^{i\omega(t-2n)} d\omega = \sum_{n\in Z} x^G_{2n} \frac{\sin(t-2n)\pi/2}{(t-2n)\pi/2}(2\cos((t-2n)\pi/2)-1).$$

Таким образом, при разложении сигнала $\{x_n\}$ мы получаем два массива $\{x^H_{2n}\}$ и $\{x^G_{2n}\}$, каждый из которых имеет длину в два раза меньше, чем исходный массив $\{x_n\}$. Обозначим полученные данные $A_n = x^H_{2n}$ и $D_n = x^G_{2n}$. В соответствии с формулами (4) и (6) имеем:

$$A_n = x^H_{2n} = \sum_{k\in Z} h^0_k x_{2n-k} \quad \text{и} \quad D_n = x^G_{2n} = \sum_{k\in Z} g^0_k x_{2n-k}, \tag{13}$$

где h^0_k – коэффициенты идеального низкочастотного фильтра $H_0(\omega)$ и g^0_k – коэффициенты идеального высокочастотного фильтра $G_0(\omega)$.

1.6.2. Восстановление идеальными фильтрами

Мы разложили сигнал $\{x_n\}$ на низкочастотную $\{A_n\}$ и высокочастотную $\{D_n\}$ составляющие по формуле (13). Восстановление сигнала $\{x_n\}$ по массивам $\{A_n\}$ и $\{D_n\}$ не представляет труда. Для этого мы найдем по массивам $\{A_n\}$ и $\{D_n\}$ непрерывную функцию $x(t)$, значения которой при целых значениях t дают массив $\{x_n\}$, $x_n = x(n)$.

Пусть $\hat{x}(\omega)$ – ее преобразование Фурье. Как было показано выше, функция $\hat{x}(\omega)$ имеет носитель на $[-\pi, \pi]$ и на этом промежутке $\hat{x}(\omega)$ совпадает с преобразованием Фурье $X(\omega)$ исходного дискретного сигнала $\{x_n\}$. Поскольку $H_0(\omega) + G_0(\omega) \equiv 1$, то

$$X(\omega) = X^H(\omega) + X^G(\omega).$$

Поэтому из формул (9) и (10) получаем:

$$\hat{x}(\omega) = \hat{x}^H(\omega) + \hat{x}^G(\omega),$$

где $\hat{x}^H(\omega)$ и $\hat{x}^G(\omega)$ – функции, которые имеют носители, соответственно, на промежутках $[-\pi/2, \pi/2]$ и $[-\pi, -\pi/2] \cup [\pi/2, \pi]$ и совпадают на них с компонентами разложения $X^H(\omega)$ и $X^G(\omega)$. Следовательно,

$$x(t) = x^H(t) + x^G(t),$$

где функции $x^H(t)$ и $x^G(t)$ выражаются формулами (11) и (12), а с учетом $A_n = x^H_{2n}$ и $D_n = x^G_{2n}$ имеем:

$$x^H(t) = \sum_{n \in Z} A_n \frac{\sin(t-2n)\pi/2}{(t-2n)\pi/2} \quad \text{и}$$

$$x^G(t) = \sum_{n \in Z} D_n \frac{\sin(t-2n)\pi/2}{(t-2n)\pi/2} \left(2\cos(t-2n)\frac{\pi}{2} - 1\right).$$

Теперь находим значения x_n:

$$x_n = x(n) = x^H(n) + x^G(n) =$$

$$= \sum_{k \in Z} (A_k + D_k(2\cos((n-2k)\pi/2) - 1)) \frac{\sin(n-2k)\pi/2}{(n-2k)\pi/2}.$$

Отсюда после элементарных вычислений получаем:

$$\begin{aligned} x_{2n} &= A_n + D_n, \\ x_{2n+1} &= \sum_{k \in Z} \frac{2(-1)^k}{(2k+1)\pi} (A_{n-k} - D_{n-k}). \end{aligned} \tag{14}$$

Учитывая значения (2) коэффициентов идеальных фильтров H_0 и G_0, последние соотношения (14) можно записать в виде:

$$x_n = 2\sum_{k\in Z}(h^0_{n-2k}A_k + g^0_{n-2k}D_k). \qquad (15)$$

Проанализируем последнюю формулу. Она отличается от свертки тем, что в суммировании участвуют не все коэффициенты h^0_{n-p} и g^0_{n-p}, а только с четными значениями $p = 2k$. Расширим массивы $\{A_n\}$ и $\{D_n\}$, добавив к ним на нечетные места нулевые значения:

$$\tilde{A}_n = \begin{cases} A_k, \text{ если } n = 2k, \\ 0, \text{ если } n = 2k+1 \end{cases}, \quad \tilde{D}_n = \begin{cases} D_k, \text{ если } n = 2k, \\ 0, \text{ если } n = 2k+1 \end{cases}.$$

Тогда формула (15) принимает вид свертки расширенных массивов $\tilde{A}$ и $\tilde{D}$ с коэффициентами идеальных фильтров:

$$x_n = 2\sum_{k\in Z}(h^0_{n-k}\tilde{A}_k + g^0_{n-k}\tilde{D}_k), \qquad (16)$$

$$x = 2(h^0 * \tilde{A} + g^0 * \tilde{D}).$$

Таким образом, восстановление сигнала $\{x_n\}$ по низко- и высокочастотным компонентам $\{A_n\}$ и $\{D_n\}$ производится в обратном порядке:

- добавление нулевых значений в массивы $\{A_n\}$ и $\{D_n\}$ на нечетные места (обратная децимация);
- применение к полученным массивам $\{\tilde{A}_n\}$ и $\{\tilde{D}_n\}$ цифровых фильтров $2H_0$ и $2G_0$.

Операцию расширения массивов добавлением нулей на нечетные места называют обратной децимацией и обозначают символом $2\uparrow$.

1.6.3. Общий случай

В предыдущем разделе мы показали, что если $\{x_n\}_{n\in \mathbf{Z}}$ – некоторый сигнал и $H_0(\omega)$ и $G_0(\omega)$ – идеальные низкочастотный и высокочастотный фильтры, то разложение сигнала на высокочастотную и низкочастотную части производится в два этапа:

- разложение сигнала по формулам свертки сигнала с коэффициентами фильтров, $x^H_n = \sum_{k\in Z} h^0_k x_{n-k}$ и $x^G_n = \sum_{k\in Z} g^0_k x_{n-k}$;
- децимация, $A_n = x^H_{2n}$ и $D_n = x^G_{2n}$.

Восстановление сигнала $\{x_n\}$ по низко- и высокочастотным компонентам $\{A_n\}$ и $\{D_n\}$ производится в обратном порядке. Однако использование идеальных фильтров для анализа сигнала неудобно тем, что фильтры H_0 и G_0 имеют бесконечную импульсную характеристику (бесконечное число ненулевых коэффициентов h^0_n и g^0_n). Кроме того, коэффициенты h^0_n и g^0_n убывают достаточно медленно.

С «физической» точки зрения, может быть целесообразнее использовать фильтры, у которых полоса пропускания плавно переходит в полосу заграждения, как, например, для $H(\omega) = \cos^2(\omega/2)$ и $G(\omega) = \sin^2(\omega/2)$. Их преимущество в том, что такие фильтры могут иметь небольшое число ненулевых коэффициентов h_n и g_n. Основная неприятность заключается в том, что при использовании подобных фильтров нельзя будет использовать теорему Котельникова–Шеннона для обоснования децимации и получения выражений компонент $x^H(t)$ и $x^G(t)$ через x^H_{2n} и x^G_{2n}. Однако ничто не мешает нам провести разложение с децимацией, если заданы какие-нибудь низкочастотный и высокочастотный фильтры $H(\omega)$ и $G(\omega)$:

$$\{x_n\} \;\to\; \{x^H_n\} = h * x \;\to\; 2\downarrow \;\to\; A_n = x^H_{2n},$$
$$\{x_n\} \;\to\; \{x^G_n\} = g * x \;\to\; 2\downarrow \;\to\; D_n = x^G_{2n}.$$

Проблемы возникают на этапе восстановления. Формальное применение обратной процедуры не дает точного восстановления по формуле (16). Можно попробовать решить эту проблему, используя для восстановления другие фильтры.

Выясним, для каких фильтров $H(\omega)$ и $G(\omega)$ возможно точное восстановление сигнала $\{x_n\}$ по данным разложения $\{A_n\}$ и $\{D_n\}$, и найдем новые фильтры $\tilde{H}(\omega)$ и $\tilde{G}(\omega)$, обеспечивающее это восстановление.

Итак, пусть

$$H(\omega) = \sum_{n\in Z} h_n e^{-in\omega}, \quad G(z) = \sum_{n\in Z} g_n e^{-in\omega} \tag{17}$$

– низкочастотный фильтр с коэффициентами $\{h_n\}$ и, соответственно, высокочастотный фильтр с коэффициентами $\{g_n\}$. Пусть $\{x_n\}$ – некоторый сигнал и $X(\omega)$ – его преобразование Фурье. В дальнейшем будет удобно использовать z-преобразования:

$$H(z) = \sum_{n\in Z} h_n z^n, \quad G(z) = \sum_{n\in Z} g_n z^n, \quad X(z) = \sum_{n\in Z} x_n z^n,$$

где z – комплексная переменная. Если $z = e^{-i\omega}$, получаем функции $H(\omega)$, $G(\omega)$ и $X(\omega)$.

1. Разложение. Производим разложение сигнала $\{x_n\}$ по формулам свертки

$$x^H_n = \sum_{k\in Z} h_k x_{n-k}, \quad x^G_n = \sum_{k\in Z} g_k x_{n-k}.$$

На уровне преобразований Фурье (и z-преобразований) разложение сигнала $X(z) \to \{X^H(z), X^G(z)\}$ заключается в применении к функции $X(z)$ фильтров $H(z)$ и $G(z)$ по формулам:

$X^H(z) = H(z)X(z), \quad X^G(z) = G(z)X(z).$

2. Децимация. Для степенных рядов децимация производится совершенно просто: нужно обратить в нуль коэффициенты при нечетных степенях, а для этого достаточно взять $(X^H(z) + X^H(-z))/2$. Затем в полученном выражении нужно сделать замену $z^2 = w$. Получается степенной ряд $A(w)$:

$$^1/_2(X^H(z) + X^H(-z)) = A(z^2) \to A(w). \qquad (18)$$

Коэффициенты ряда $A(w)$ образуют массив $\{A_n\}$:

$$A_n = x^H_{2n} = \sum\nolimits_{k\in Z} h_k x_{2n-k}.$$

Пример 1. Пусть $X^H(z) = x^H_0 + x^H_1 z + x^H_2 z^2 + x^H_3 z^3 + x^H_4 z^4$. Тогда $X^H(-z) = x^H_0 - x^H_1 z + x^H_2 z^2 - x^H_3 z^3 + x^H_4 z^4$. Поэтому

$$\frac{1}{2}(X^H(z) + X^H(-z)) = x^H_0 + x^H_2 z^2 + x^H_4 z^4 = A(z^2).$$

Многочлен $A(z^2)$ содержит, вообще говоря, пять членов, два из которых – нулевые. Для их удаления сделаем замену $z^2 = w$. Тогда получается многочлен $A(w)$, содержащий только три слагаемых:

$$A(z^2) = x^H_0 + x^H_2 z^2 + x^H_4 z^4 = x^H_0 + x^H_2 w + x^H_4 w^2 = A_0 + A_1 w + A_2 w^2 = A(w).$$

Массив $\{A_n\}$ состоит из трех элементов $\{A_0, A_1, A_2\}$.

Пусть $X(z) \to \{X^H(z), X^G(z)\}$ – разложение сигнала фильтрами и пусть $A(w)$ и $D(w)$ – многочлены, которые получаются из компонент $X^H(z)$ и $X^G(z)$ после децимации и замены $w = z^2$. Тогда мы получили следующую схему разложения:

$$X(z) \to \{X^H(z), X^G(z)\} \to \{A(w), D(w)\},$$

где $A(w)$ определяется формулой (18), а $D(w)$ – формулой

$$D(z^2) = {}^1/_2(X^G(z) + X^G(-z)). \qquad (19)$$

Коэффициенты ряда $D(w)$ образуют массив $\{D_n\}$:

$$D_n = x^G_{2n} = \sum\nolimits_{k\in Z} g_k x_{2n-k}.$$

3. Обратная децимация. Если у нас имеются массивы $\{A_n\}$ и $\{D_n\}$, то им соответствуют ряды

$$A(w) = \sum_{n\in Z} A_n w^n \quad \text{и} \quad D(w) = \sum_{n\in Z} D_n w^n.$$

На уровне степенных рядов обратная децимация – это замена $w = z^2$. Тогда ряды, соответствующие массивам $\tilde{A}_n$ и $\tilde{D}_n$, есть $A(z^2)$ и $D(z^2)$.

Пример 2. Пусть массив $\{A_n\}$ состоит из трех элементов $\{A_0, A_1, A_2\}$. Ему соответствует ряд $A(w) = A_0 + A_1w + A_2w^2$. Сделаем замену $w = z^2$. Тогда $A(z^2) = A_0 + A_1z^2 + A_2z^4 = A_0 + 0z + A_1z^2 + 0z^3 + A_2z^4$. Получили массив $\tilde{A}_n$, состоящий из пяти элементов $\{A_0, 0, A_1, 0, A_2\}$.

4. Восстановление. Найдем новые фильтры

$$\tilde{H}(z) = \sum_{n\in Z} \tilde{h}_n z^n, \quad \tilde{G}(z) = \sum_{n\in Z} \tilde{g}_n z^n \tag{20}$$

с коэффициентами $\{\tilde{h}_n\}$ и $\{\tilde{g}_n\}$, которые дадут точную реконструкцию по формуле (аналогичной формуле (15)):

$$x_n = 2\sum_{k\in Z} (\tilde{h}_{n-2k} A_k + \tilde{g}_{n-2k} D_k). \tag{21}$$

Последнюю формулу запишем в виде свертки, используя обратную децимацию массивов $\{A_n\}$ и $\{D_n\}$:

$$\tilde{A}_n = \begin{cases} A_k, \text{ если } n = 2k, \\ 0, \text{ если } n = 2k+1 \end{cases}, \quad \tilde{D}_n = \begin{cases} D_k, \text{ если } n = 2k, \\ 0, \text{ если } n = 2k+1 \end{cases}.$$

Тогда формула (21) принимает вид свертки:

$$x_n = 2\sum_{k\in Z} (\tilde{h}_{n-k} \tilde{A}_k + \tilde{g}_{n-k} \tilde{D}_k). \tag{22}$$

При z-преобразовании свертке соответствует умножение степенных рядов. Поэтому запись формулы восстановления в виде свертки позволяет достаточно просто решить задачу нахождения фильтров реконструкции $\tilde{H}(z)$ и $\tilde{G}(z)$.

Как уже отмечалось, ряды, соответствующие массивам $\tilde{A}_n$ и $\tilde{D}_n$, есть $A(z^2)$ и $D(z^2)$. Тогда последняя формула (22) после z-преобразования принимает вид:

$$X(z) = 2(\tilde{H}(z)A(z^2) + \tilde{G}(z)D(z^2)). \tag{23}$$

Поэтому восстановление на уровне степенных рядов делается по схеме:

$$\{A(w), D(w)\} \to 2(\tilde{H}(z)\,A(z^2) + \tilde{G}(z)D(z^2)) = X(z).$$

Теперь наша цель заключается в нахождении фильтров $\tilde{H}(z)$ и $\tilde{G}(z)$, таких чтобы выполнялось последнее равенство.

Подставляя в (23) выражения для $A(z^2)$ и $D(z^2)$ из формул (18), (19), получаем:

$$X(z) = (\tilde{H}(z)H(z) + \tilde{G}(z)G(z))X(z) + (\tilde{H}(z)H(-z) + \tilde{G}(z)G(-z))X(-z).$$

В силу произвольности $X(z)$ из последнего равенства мы видим, что искомые фильтры $\tilde{H}(z)$ и $\tilde{G}(z)$ должны удовлетворять системе уравнений

$$\left.\begin{aligned}\tilde{H}(z)H(z)+\tilde{G}(z)G(z)&=1\\ \tilde{H}(z)H(-z)+\tilde{G}(z)G(-z)&=0\end{aligned}\right\} \tag{24}$$

В матричном виде:

$$\begin{pmatrix} H(z) & G(z)\\ H(-z) & G(-z)\end{pmatrix}\begin{pmatrix}\tilde{H}(z)\\ \tilde{G}(z)\end{pmatrix}=\begin{pmatrix}1\\0\end{pmatrix} \tag{25}$$

Решая систему (24), мы находим фильтры $\tilde{H}(z)$ и $\tilde{G}(z)$. Условия существования решения вполне очевидны – матрица системы (24) должна быть невырожденной, для этого определитель системы должен быть отличен от нуля:

$$\Delta_{HG} = H(z)G(-z) - G(z)H(-z) \neq 0 \tag{26}$$

всюду на единичной окружности $z = e^{-i\omega}$. Решение находится просто:

$$\tilde{H}(z)=\frac{1}{\Delta_{HG}}G(-z) \quad \text{и} \quad \tilde{G}(z)=-\frac{1}{\Delta_{HG}}H(-z). \tag{27}$$

Замечание 1. Из системы (25) следует, что

$$\begin{pmatrix} H(z) & G(z)\\ H(-z) & G(-z)\end{pmatrix}\begin{pmatrix}\tilde{H}(z) & \tilde{H}(-z)\\ \tilde{G}(z) & \tilde{G}(-z)\end{pmatrix}=\begin{pmatrix}1 & 0\\0 & 1\end{pmatrix}$$

Поэтому задача о нахождении фильтров восстановления $\tilde{H}(z)$ и $\tilde{G}(z)$ сводится к нахождению матрицы, обратной к матрице системы (25) на единичной окружности $z = e^{-i\omega}$. Матрица системы (25) в дальнейшем будет нам встречаться достаточно часто, и ее принято записывать в транспонированном виде:

$$\mathbf{H}(z)=\begin{pmatrix} H(z) & H(-z)\\ G(z) & G(-z)\end{pmatrix}. \tag{28}$$

Таким образом, если фильтры $H(z)$ и $G(z)$ выбрать удовлетворяющими условию (26), то фильтры восстановления $\tilde{H}(z)$ и $\tilde{G}(z)$ находятся единственным образом из системы (25) по формулам (27). Проблема заключается в том, что нам нужно найти не только функции $\tilde{H}(z)$ и $\tilde{G}(z)$, не менее важно, чтобы были получены хорошие выражения для их коэффициентов через коэффициенты фильтров $H(z)$ и $G(z)$. Следующий способ решает эту задачу.

1. Сопряженные квадратурные фильтры. Пусть $H(z)$ – некоторый фильтр с вещественными коэффициентами $\{h_n\}$. Найдем сначала такой фильтр $G(z)$, чтобы на единичной окружности $z = e^{-i\omega}$ была бы обратимой матрица (28). Напомним, что $H(\omega)$ – это функция $H(z)$ при $z = e^{-i\omega}$. На единичной окружности имеем $-z = -e^{-i\omega} = e^{-i(\omega+\pi)}$, поэтому $H(-z) = H(\omega + \pi)$. Тогда искомая матрица $\mathbf{H}(z)$ принимает вид:

$$\begin{pmatrix} H(z) & H(-z) \\ G(z) & G(-z) \end{pmatrix} = \begin{pmatrix} H(\omega) & H(\omega+\pi) \\ G(\omega) & G(\omega+\pi) \end{pmatrix}$$

Проще всего попробовать сделать ее унитарной матрицей. Общий вид унимодулярной унитарной матрицы следующий:

$$\begin{pmatrix} w_1 & w_2 \\ -\overline{w}_2 & \overline{w}_1 \end{pmatrix},$$

где $|w_1|^2 + |w_2|^2 = 1$. Поэтому следует рассмотреть матрицу

$$\begin{pmatrix} H(\omega) & \overline{H(\omega+\pi)} \\ -\overline{H(\omega+\pi)} & \overline{H(\omega)} \end{pmatrix}.$$

Требование $|w_1|^2 + |w_2|^2 = 1$ принимает вид

$$|H(\omega)|^2 + |H(\omega + \pi)|^2 = 1. \qquad (29)$$

Если мы положим $G(\omega) = -\overline{H(\omega+\pi)}$, тогда $G(\omega+\pi) = -\overline{H(\omega)}$, и мы видим, что появился «лишний» коэффициент -1 перед $\overline{H(\omega)}$. Это можно исправить добавлением коэффициента $e^{i\omega}$ в выражение для $G(\omega)$. А именно: $G(\omega) = -e^{i\omega}\,\overline{H(\omega+\pi)}$, тогда $G(\omega+\pi) = -e^{i(\omega+\pi)}\overline{H(\omega)} = e^{i\omega}H(\omega)$.

Мы получили унитарную матрицу

$$\mathbf{H}(\omega) = \begin{pmatrix} H(\omega) & H(\omega+\pi) \\ G(\omega) & G(\omega+\pi) \end{pmatrix} = \begin{pmatrix} H(\omega) & H(\omega+\pi) \\ -e^{i\omega}\,\overline{H(\omega+\pi)} & e^{i\omega}\overline{H(\omega)} \end{pmatrix}.$$

Поэтому ее обратная матрица является сопряженной к ней, то есть является транспонированной к матрице

$$\tilde{\mathbf{H}}(\omega) = \begin{pmatrix} \tilde{H}(\omega) & \tilde{H}(\omega+\pi) \\ \tilde{G}(\omega) & \tilde{G}(\omega+\pi) \end{pmatrix} = \begin{pmatrix} \overline{H(\omega)} & \overline{H(\omega+\pi)} \\ -e^{-i\omega}H(\omega+\pi) & e^{-i\omega}H(\omega) \end{pmatrix}.$$

Таким образом, все искомые фильтры будут следующие:

$$G(\omega) = -e^{i\omega}\,\overline{H(\omega+\pi)}, \qquad \tilde{H}(\omega) = \overline{H(\omega)}, \qquad \tilde{G}(\omega) = \overline{G(\omega)} = -e^{-i\omega}H(\omega+\pi), \quad (30)$$

при условии что выполняется соотношение (29) – оно служит для выбора фильтра $H(\omega)$.

2. Фильтры, основанные на теореме Безу. В случае фильтров конечной импульсной характеристики $H(z)$ и $G(z)$ будут полиномами (могут быть с отрицательными степенями). В соответствии с формулами (27) можно считать, что $\tilde{H}(z) = G(-z)$ и $\tilde{G}(z) = -H(-z)$, при условии что $\Delta_{HG} = H(z)G(-z) - G(z)H(-z) = 1$. Для большей общности положим

$$\tilde{H}(z) = p(z)G(-z), \quad \tilde{G}(z) = -p(z)H(-z), \tag{31}$$

где $p(z)$ – произвольный многочлен. Они будут решением системы (24) при условии

$$p(z)(\, H(z)G(-z) - H(-z)G(z)) = 1. \tag{32}$$

Поскольку $p(z)$ – многочлен и $H(z)G(-z) - H(-z)G(z)$ также является многочленом, то $p(z) = \alpha z^k$ (произведение двух многочленов равно константе только тогда, когда каждый из сомножителей является одночленом). Тогда уравнение $H(z)G(-z) - H(-z)G(z) = p(z)^{-1}$ будет полиномиальным. Обычно полином $H(z)$ известен, и задача заключается в нахождении неизвестного многочлена $G(z)$. При фиксированном $p(z) = \alpha z^k$ она решается следующей теоремой.

Теорема Безу. *Пусть $P_1(z)$, $P_2(z)$ и $R(z)$ – полиномы степеней $n \geq 1$, $m \geq 1$ и $k < n + m$ соответственно, причем $P_1(z)$, $P_2(z)$ не имеют общих нулей. Тогда существуют единственные полиномы $Q_1(z)$ и $Q_2(z)$ степени $m - 1$, $n - 1$, такие что*

$$P_1(z)Q_1(z) + P_2(z)Q_2(z) = R(z).$$

Доказательство. Пусть $\{z_i,\ i = 1, ..., n\}$ – нули многочлена $P_1(z)$. Поскольку степень многочлена $Q_2(z)$ – не более, чем $n - 1$, то он может быть найден из n условий $Q_2(z_i) = R(z_i)/P_2(z_i)$, где в случае корня z_i кратности s равенство выполняется вместе с производными до порядка $s - 1$. Многочлен $Q_1(z)$ находится аналогично. Теперь равенство теоремы Безу выполняется с учетом кратности в $n + m$ точках. Поскольку левая и правая части – полиномы порядка не выше $n + m - 1$, то оно выполняется тождественно.

Замечание 2. В уравнении $H(-z)G(z) - H(z)G(-z) = p(z)^{-1}$ полиномы $H(z)$, $G(-z)$ и $p(z)^{-1}$ могут иметь отрицательные степени. Для того чтобы получить многочлены, достаточно умножить обе части уравнения на подходящую степень z^n.

1.6.4. Примеры

Пример 1. Фильтры Хаара. Рассмотрим два фильтра h и g с ненулевыми элементами

$h_0 = {}^1\!/_2$, $h_1 = {}^1\!/_2$, $g_0 = {}^1\!/_2$, $g_1 = -{}^1\!/_2$.

Передаточные функции:

$$H(\omega) = \frac{1}{2} + \frac{1}{2}e^{-i\omega} = \frac{1}{2}(1 + \cos\omega - i\sin\omega) = \cos\frac{\omega}{2}e^{-i\frac{\omega}{2}},$$

$$G(\omega) = \frac{1}{2} - \frac{1}{2}e^{-i\omega} = \frac{1}{2}(1 - \cos\omega + i\sin\omega) = i\sin\frac{\omega}{2}e^{-i\frac{\omega}{2}}.$$

Отметим некоторые свойства:

$$H(0) = 1, \qquad \left|H(\omega)\right| = \cos\frac{\omega}{2}, \qquad G(0) = 0, \qquad \left|G(\omega)\right| = \left|\sin\frac{\omega}{2}\right|,$$

$G(\omega) = H(\omega + \pi)$, $H(\omega) + G(\omega) = 1$.

Мы видим, что фильтр h – низкочастотный, а фильтр g – высокочастотный. Пусть последовательность $\{x_n\}$ представляет некоторый сигнал и $X(\omega)$ – его преобразование Фурье. При умножении $X(\omega)$ на $H(\omega)$ частоты сигнала $\{x_n\}$, близкие к нулю, остаются почти без изменения, тогда как частоты ω, по модулю близкие к π, подавляются. Другими словами, низкие частоты сигнала $\{x_n\}$ пропускаются, а высокие – задерживаются. Функция $A(\omega) = H(\omega)X(\omega)$ описывает новый сигнал $\{a_n\}$, значения которого находятся в виде свертки

$$a_n = \sum_{k\in Z} h_k x_{n-k} = \frac{1}{2}x_n + \frac{1}{2}x_{n-1}.$$

Таким образом, фильтр h из последовательности $\{x_n\}$ производит последовательность $\{a_n\}$ средних арифметических, сглаживая колебания $\{x_n\}$.

Совершенно аналогично фильтр g имеет передаточную функцию $G(\omega)$, которая подавляет низкие частоты (близкие к нулю) и пропускает высокие. В области значений действие этого фильтра $D(\omega) = G(\omega)X(\omega)$ дает новый сигнал

$$d_n = \sum_{k=0}^{1} g_k x_{n-k} = \frac{x_n - x_{n-1}}{2},$$

который имеет смысл первой конечно-разностной производной сигнала x (точнее, ее половины). Новый сигнал $\{d_n\}$ отражает колебания последовательности $\{x_n\}$.

Отметим также, что $x_n = a_n + d_n$. Последнее равенство означает, что сигнал $\{x_n\}$ мы разложили на сглаженную часть $\{a_n\}$ и высокочастотную компоненту $\{d_n\}$. Мы видим, что сигнал восстанавливается непосредственно. Тем не менее построим фильтры восстановления $\tilde{H}(z)$ и $\tilde{G}(z)$ по предложенной выше схеме.

Легко видеть, что функции $H(\omega)$ и $G(\omega)$ связаны соотношением

$$G(\omega) = -e^{-i\omega}\,\overline{H(\omega+\pi)}.$$

Кроме того, выполняется равенство $|H(\omega)|^2 + |H(\omega + \pi)|^2 = 1$. Поэтому можно построить фильтры восстановления.

$$\tilde{H}(\omega) = \overline{H(\omega)} = \frac{1}{2}(1+e^{i\omega}), \qquad \tilde{G}(\omega) = \overline{G(\omega)} = -e^{i\omega}H(\omega+\pi) = \frac{1}{2}(1-e^{i\omega}).$$

Ненулевые коэффициенты этих фильтров:

$$\tilde{h}_{-1} = 1/2,\ \tilde{h}_0 = 1/2, \quad \tilde{g}_{-1} = -1/2,\ \tilde{g}_0 = 1/2.$$

Пример 2. Рассмотрим два фильтра h и g с ненулевыми элементами

$$h_{-1} = \frac{1}{4},\ h_0 = \frac{1}{2},\ h_1 = \frac{1}{4},\ g_{-1} = -\frac{1}{4},\ g_0 = \frac{1}{2},\ g_1 = -\frac{1}{4}.$$

Передаточные функции $H(\omega)$ и $G(\omega)$ этих последовательностей легко вычисляются:

$$H(\omega) = \frac{1}{2}(1+\cos\omega) = \cos^2\frac{\omega}{2}, \quad G(\omega) = \frac{1}{2}(1-\cos\omega) = \sin^2\frac{\omega}{2}.$$

Отметим очевидные соотношения: $G(\omega) = H(\omega + \pi)$ и $H(\omega) + G(\omega) = 1$. По сравнению с предыдущим примером, полоса пропускания низких частот у фильтра $H(\omega)$ стала более узкой. То же самое можно сказать о фильтре $G(\omega)$ высоких частот. Переходя из частотной области в область значений, мы получаем два фильтра

$$a_n = \sum_{k=-1}^{1} h_k x_{n-k} = \frac{1}{4}x_{n+1} + \frac{1}{2}x_n + \frac{1}{4}x_{n-1} = \frac{1}{2}\left(\frac{x_n + x_{n-1}}{2} + \frac{x_n + x_{n+1}}{2}\right),$$

$$d_n = \sum_{k=-1}^{1} g_k x_{n-k} = -\frac{1}{4}x_{n+1} + \frac{1}{2}x_n - \frac{1}{4}x_{n-1} = \frac{1}{4}((x_n - x_{n-1}) - (x_{n+1} - x_n)).$$

Первый фильтр производит двойное сглаживание значений $\{x_n\}$ по среднему арифметическому, а второй фильтр выражается через вторую конечно-разностную производную сигнала $\{x_n\}$. Отметим также, что $x_n = a_n + d_n$. Последнее равенство означает, что сигнал $\{x_n\}$ мы непосредственно разложили на сглаженную часть $\{a_n\}$ и высокочастотную компоненту $\{d_n\}$.

Для данных фильтров не выполнено соотношение $|H(\omega)|^2 + |H(\omega + \pi)|^2 = 1$, однако и проблема восстановления сигнала также не возникает, поскольку $x_n = a_n + d_n$.

Пример 3. Рассмотрим два фильтра h и g с ненулевыми элементами:

$$h_0 = \frac{1}{8}(1+\sqrt{3}), \quad h_1 = \frac{1}{8}(3+\sqrt{3}), \quad h_2 = \frac{1}{8}(3-\sqrt{3}), \quad h_3 = \frac{1}{8}(1-\sqrt{3}),$$

$$g_0 = \frac{1}{8}(1-\sqrt{3}), \quad g_1 = -\frac{1}{8}(3-\sqrt{3}), \quad g_2 = \frac{1}{8}(3+\sqrt{3}), \quad g_3 = -\frac{1}{8}(1+\sqrt{3}).$$

Отметим соотношения между коэффициентами этих фильтров.

$h_0 + h_1 + h_2 + h_3 = 1, \quad g_0 + g_1 + g_2 + g_3 = 0,$
$g_0 = h_3, \quad g_1 = -h_2, \quad g_2 = h_1, \quad g_3 = -h_0,$
$h_0 + h_2 = h_1 + h_3 = {}^1/_2, \quad 2h_2 = h_1 + 3h_3.$

Найдем передаточные функции $H(z)$ и $G(z)$ в z-представлении.

$H(z) = h_0 + h_1 z + h_2 z^2 + h_3 z^3,$
$G(z) = g_0 + g_1 z + g_2 z^2 + g_3 z^3 = h_3 - h_2 z + h_1 z^2 - h_0 z^3 =$
$= -z^3(-h_3 z^{-3} + h_2 z^{-2} - h_1 z^{-1} + h_0) = -z^3 H(-z^{-1}).$

Таким образом, мы получили

$G(z) = -z^3 H(-z^{-1}).$

Для восстановления сигнала требуются дополнительные фильтры $\tilde{H}(z)$ и $\tilde{G}(z)$. Определим их как сопряженные квадратурные фильтры по формулам:

$$\tilde{H}(z) = H(z^{-1}), \quad \tilde{G}(z) = G(z^{-1}) = -z^{-3} H(-z). \tag{33}$$

Тогда второе соотношение $\tilde{H}(z)H(-z) + \tilde{G}(z)G(-z) = 0$ из (24) выполняется. Первое соотношение $\tilde{H}(z)H(z) + \tilde{G}(z)G(z) = 1$ принимает вид

$H(z^{-1})H(z) + G(z^{-1})G(z) = 1.$

Легко проверить, что оно выполняется.

Вернемся к частотной переменной $z = e^{-i\omega}$. Поскольку коэффициенты $\{h_n\}$ – вещественные, то $H(z^{-1}) = \overline{H(\omega)}$. Поэтому последнее соотношение принимает вид

$|H(\omega)|^2 + |G(\omega)|^2 = 1.$

Найдем коэффициенты фильтров восстановления $\tilde{H}(z)$ и $\tilde{G}(z)$ из их определения, $\tilde{H}(z) = H(z^{-1})$, $\tilde{G}(z) = G(z^{-1})$:

$$\tilde{h}_{-3} = \frac{1}{8}(1-\sqrt{3}), \quad \tilde{h}_{-2} = \frac{1}{8}(3-\sqrt{3}), \quad \tilde{h}_{-1} = \frac{1}{8}(3+\sqrt{3}), \quad \tilde{h}_0 = \frac{1}{8}(1+\sqrt{3}),$$

$$\tilde{g}_{-3} = -\frac{1}{8}(1+\sqrt{3}), \quad \tilde{g}_{-2} = \frac{1}{8}(3+\sqrt{3}), \quad \tilde{g}_{-1} = -\frac{1}{8}(3-\sqrt{3}), \quad \tilde{g}_0 = \frac{1}{8}(1-\sqrt{3}).$$

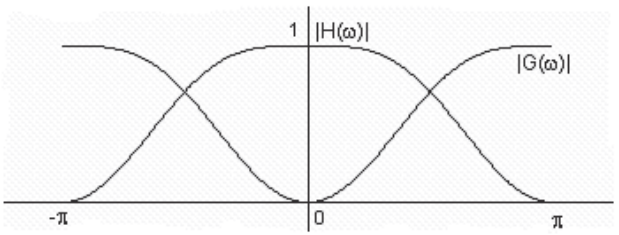


Рис. 1.6.4. Графики модулей передаточных функций примера 3

1.6.5. Многоуровневый анализ сигналов

Изложенная выше идея разложения сигнала $\{x_n\}$ на низкочастотную (сглаженную) и высокочастотную составляющие лежит в основе вейвлет-анализа сигнала. Можно даже не знать природы вейвлетов, важно только, что каждому ортогональному вейвлету соответствуют 4 фильтра:

- $\{h_n\}$ – низкочастотный фильтр разложения;
- $\{g_n\}$ – высокочастотный фильтр разложения;
- $\{\tilde{h}_n\}$ – низкочастотный фильтр восстановления;
- $\{\tilde{g}_n\}$ – высокочастотный фильтр восстановления.

Вейвлет-анализ сигнала производится по той схеме, которую мы изложили выше, но многократно. Берем сигнал $x = \{x_n\}$ и применяем к нему низкочастотный фильтр разложения $\{h_n\}$ с децимацией для получения сглаженной составляющей $A = \{A_k\}$, затем применяем к $\{x_n\}$ высокочастотный фильтр разложения $\{g_n\}$ с децимацией для получения деталей $D = \{D_k\}$, которые мы потеряли при сглаживании. Будем называть $A = \{A_k\}$ коэффициентами аппроксимации первого уровня разложения и обозначать их $A_1 = \{A_{1,k}\}$. Массив $D = \{D_k\}$ будем называть детализирующими коэффициентами первого уровня разложения и обозначать его $D_1 = \{D_{1,k}\}$.

Далее процедура разложения применяется к набору коэффициентов A_1, и мы получаем коэффициенты второго уровня разложения $A_2 = \{A_{2,m}\}$ и $D_2 = \{D_{2,m}\}$. Заметим, что после каждого шага число аппроксимирующих коэффициентов A и отдельно детализирующих D уменьшается в два раза. Повторяя процедуру разложений необходимое число j раз, мы получаем вместо сигнала $x = \{x_n\}$ серию коэффициентов

$\{A_j, D_j, D_{j-1}, ..., D_1\}$.

Вейвлет-анализ сигнала заключается теперь в изучении и обработке полученных коэффициентов разных уровней. После изменения коэффициентов разложения сигнал восстанавливается в обратном порядке использованием фильтров восстановления $\{\tilde{h}_n\}$ и $\{\tilde{g}_n\}$.

Схема разложения сигнала может быть изображена следующим образом:

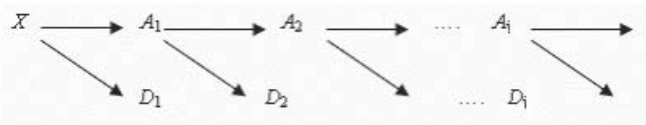


Рис. 1.6.5. Схема вейвлет-разложения сигнала

Можно применять процедуру разложения не только к аппроксимирующим коэффициентам A_j, но и к детализирующим D_j. Тогда коэффициенты разложения изображаются в виде дерева (рис. 1.6.6), наверху которого находится анализируемый сигнал $x = \{x_n\}$. Ветви влево указывают на аппроксимирующие коэффициенты, а правые ветви идут к детализирующим коэффициентам предыдущего узла. Узлы, в которых находятся коэффициенты разложения, удобно нумеровать так, как указано на рисунке ниже.

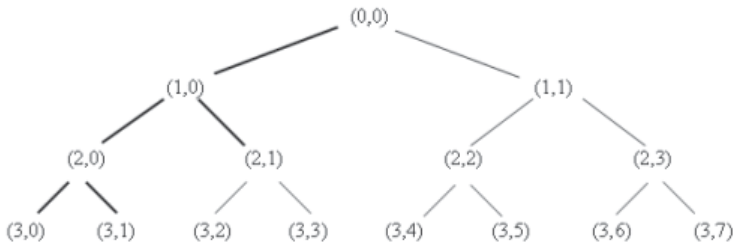
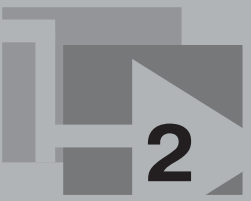


Рис. 1.6.6. Схема пакетного разложения сигнала

Основы теории вейвлетов



Вейвлеты – это функции типа маленькой волны (всплески), которые порождают базисы пространства $L^2(\mathbf{R})$, удобные для обработки сигналов. Пример подобного базиса известен с начала прошлого столетия – это базис Хаара. Однако теория таких функций и базисов значительно развита только в последние десятилетия. Мы начинаем с рассмотрения вейвлетов Хаара как наиболее простых. Основные конструкции будут в дальнейшем обобщены, что даст возможность строить другие функции, которые порождают другие базисы в $L^2(\mathbf{R})$.

2.1. Вейвлеты Хаара

Для того чтобы функции $\{\varphi_k(x)\}$ типа маленькой волны образовывали хороший базис $L^2(\mathbf{R})$, желательно, чтобы они обладали следующими свойствами: имели компактный носитель, то есть были бы сосредоточены на ограниченном промежутке; «присутствовали» бы в любой точке пространства $\mathbf{R}$; могли замечать быстрые колебания функций. Более формально от семейства $\{\varphi_k(x)\}$ требуются следующие свойства:

1. Компактный носитель каждой функции $\varphi_k(x)$.
2. Для любой точки $x_0 \in \mathbf{R}$ существует функция семейства $\varphi_k(x)$, носитель которой содержит точку x_0.
3. Среди функций семейства $\{\varphi_k(x)\}$ имеются такие, которые имеют сколь угодно большую частоту колебаний.

Первое свойство удобно во многих отношениях, в частности оно обеспечивает существование интегралов, которые необходимы для вычисления коэффициентов разложения.

Второе свойство будет выполнено, если мы потребуем, чтобы наряду с функциями $\varphi_k(x)$ в базис входили бы и их сдвиги по оси Ox, то есть функции вида $\varphi_k(x - n)$, для любых целочисленных $n \in \mathbf{Z}$.

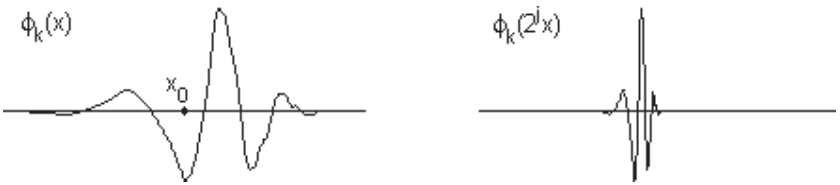


Рис. 2.1.1. Функции семейства $\varphi_k(x)$

Третье свойство будет выполнено, если наряду с функциями $\varphi_k(x)$ в базис будут входить их сжатия и растяжения, например функции вида $\varphi_k(2^j x)$, для любых целочисленных $j \in \mathbf{Z}$.

Наконец, было бы совсем хорошо, если бы базис в $L^2(\mathbf{R})$ порождался одной функцией $\psi(x)$ при помощи сдвигов и растяжений, то есть чтобы он состоял из функций вида

$\psi_{jn}(x) = \psi(2^j x - n), j, n \in \mathbf{Z}$.

Оказывается, это возможно! Более того, существуют даже две функции $\varphi(x)$ и $\psi(x)$, сдвиги и растяжения $\varphi(2^j x - n)$ первой порождают расширяющуюся последовательность подпространств $L^2(\mathbf{R})$, а вторая функция порождает базис $\{\psi_{jn}(x) = \psi(2^j x - n), j, n \in \mathbf{Z}\}$ пространства $L^2(\mathbf{R})$. Кроме того, эти функции обладают еще дополнительными свойствами, которые существенно облегчают вычисление коэффициентов разложения.

Пример подобного базиса известен с начала прошлого столетия – это базис Хаара. Однако теория подобных функций и базисов всплесков значительно развилась только в последние десятилетия.

Для того чтобы понять основные идеи в самом простом случае, построим в пространстве $L^2(\mathbf{R})$ ортонормированный базис Хаара. Он определяется на основе функции прямоугольной волны

$$\varphi(x) = \begin{cases} 1, & x \in [0,1) \\ 0, & \text{для других } x \end{cases}.$$

Процедуру построения базиса Хаара проведем в несколько этапов. Сначала определим возрастающую последовательность подпространств $L^2(\mathbf{R})$, в пределе дающую все пространство $L^2(\mathbf{R})$. На основе этой последовательности будут естественным образом введены пространства вейвлетов и сами вейвлеты Хаара.

2.1.1. Последовательность масштабированных подпространств

Рассмотрим систему функций, полученную из определенной выше функции $\varphi(x)$ целочисленными сдвигами:

$$\varphi_{0,n}(x) = \varphi(x - n),\ n \in \mathbf{Z}. \qquad (1)$$

Обозначим V_0 – пространство в $L^2(\mathbf{R})$, порожденное линейными комбинациями таких сдвигов (V_0 – замыкание в $L^2(\mathbf{R})$ линейной оболочки системы $\{\varphi_{0,n}(x),\ n \in \mathbf{Z}\}$). Пространство V_0 состоит из кусочно-постоянных функций из $L^2(\mathbf{R})$ с промежутками постоянства вида $[n, n + 1)$. Легко видеть, что система $\{\varphi_{0,n}(x)\}$ образует ортонормированный базис пространства V_0.

Теперь рассмотрим масштабированные сдвиги $\varphi(2x - n)$. Они получаются из $\varphi(2x)$ сдвигами на $n/2$, $\varphi(2x - n) = \varphi(2(x - n/2))$. Носитель функции стал в два раза меньше: $0 \le 2x - n < 1$, $n/2 \le x < (n + 1)/2$, $\mathrm{supp}(\varphi(2x - n)) = [n/2, (n + 1)/2]$. Поэтому

$$\| \varphi \|^2 = \int_{-\infty}^{\infty} \varphi^2(2x - n)dx = \frac{1}{2}.$$

Если умножить такие функции на $\sqrt{2}$, тогда все они будут единичной нормы. Рассмотрим систему функций

$$\varphi_{1,n}(x) = \sqrt{2}\,\varphi(2x - n), \quad n \in \mathbf{Z} \qquad (2)$$

и пространство V_1, порожденное ими. Система $\{\varphi_{1,n}(x)\}$ образует ортонормированный базис пространства V_1. Пространство V_1 состоит из кусочно-постоянных функций с промежутками постоянства длины $1/2$. По построению пространство V_1 является масштабированной версией пространства V_0, другими словами, $f(x) \in V_0 \Leftrightarrow f(2x) \in V_1$.

Нетрудно заметить, что $V_0 \subset V_1$. Действительно, порождающая функция $\varphi(x)$ пространства V_0 выражается в виде линейной комбинации элементов пространства V_1:

$$\varphi(x) = \varphi(2x) + \varphi(2x - 1) = \sqrt{2}\sum_n h_n \varphi(2x - n),$$

где ненулевые h_n только такие: $h_0 = h_1 = 1/\sqrt{2}$. Далее рассмотрим пространство V_2, порожденное функциями:

$$\varphi_{2,n}(x) = 2\varphi(2^2x - n),\ n \in \mathbf{Z},$$

полученными из функции $2\varphi(4x)$ сдвигами на $n/4$ по оси Ox. Носитель, $\mathrm{supp}(\varphi_{2,n}) = [n/2^2, (n+1)/2^2]$, есть отрезок длины $1/4$. Система $\{\varphi_{2,n}(x)\}$ образует ортонормированный базис пространства V_2. Нетрудно заметить, что $V_1 \subset V_2$.

Продолжая эту процедуру, для любого $j \in \mathbf{N}$ рассмотрим систему функций

$$\varphi_{j,n}(x) = \sqrt{2^j}\varphi(2^j x - n). \qquad (3)$$

Это ортонормированная система функций, $\mathrm{supp}(\varphi_{j,n}) = [n/2^j, (n+1)/2^j]$, все функции системы получаются из $\sqrt{2^j}\varphi(2^j x)$ сдвигами на $n/2^j$ по оси Ox.

Пусть V_j – пространство, порожденное системой функций (3). Имеют место следующие включения: $V_0 \subset V_1 \subset V_2 \subset \ldots \subset V_j$. Продолжим этот процесс до бесконечности. Тогда мы получим бесконечную систему вложенных подпространств $V_j \subset L^2(\mathbf{R})$:

$$V_0 \subset V_1 \subset V_2 \subset \ldots \subset V_j \subset \ldots$$

В каждом пространстве V_j выделен ортонормированный базис $\{\varphi_{j,n}(x)\}$. Линейные комбинации функций вида $\varphi_{j,n}(x), j, n \in \mathbf{Z}$ являются кусочно-постоянными функциями. Поскольку они образуют плотное множество в $L^2(\mathbf{R})$, то

$$\overline{\bigcup_{j=0}^{\infty} V_j} = L^2(\mathbf{R}),$$

где черта сверху обозначает замыкание.

Аналогичным образом можно ввести пространства V_j с отрицательным j. Тогда мы получаем систему вложенных подпространств, бесконечную в обе стороны:

$$\ldots \subset V_{-1} \subset V_0 \subset V_1 \subset V_2 \subset \ldots \subset V_j \subset \ldots$$

Очевидно, что

$$\bigcap_{j=-\infty}^{\infty} V_j = \{0\} \quad \text{и} \quad \overline{\bigcup_{j=-\infty}^{\infty} V_j} = L^2(\mathbf{R}).$$

2.1.2. Операторы проектирования

Построенная последовательность расширяющихся подпространств $\ldots \subset V_{-1} \subset V_0 \subset V_1 \subset V_2 \subset \ldots \subset V_j \subset \ldots$ может быть использована для того, чтобы от произвольной функции $f(x)$ из $L^2(\mathbf{R})$ перейти к ее достаточно точному приближению в пространстве V_j, $j \in \mathbf{Z}$. Это делается

ортогональным проектированием пространства $L^2(\mathbf{R})$ на подпространство V_j:

$$P_j : L^2(\mathbf{R}) \to V_j, \quad P_j(f) = \sum_{n\in\mathbf{Z}} (f, \varphi_{j,n})\varphi_{j,n}(x).$$

Отметим, что здесь не возникает вопроса о сходимости ряда, так как в любой точке $x \in \mathbf{R}$ сумма состоит из одного ненулевого слагаемого, поскольку промежутки, на которых функции $\varphi_{j,n}(x)$ отличны от нуля, при различных n не пересекаются. Операторы $P_j : L^2(\mathbf{R}) \to V_j$ линейны и непрерывны. Отметим также, что для любого $j \in \mathbf{Z}$ имеет место равенство $P_j \circ P_{j+1} = P_j$. Рассмотрим подробнее действие этих операторов. Пусть $f(x)$ – некоторая функция из $L^2(\mathbf{R})$. Тогда

$$(f, \varphi_{0,n}) = \int_{-\infty}^{\infty} f(x)\varphi(x-n)dx = \int_{n}^{n+1} f(x)dx = f_{0n}$$

– это среднее значение функции $f(x)$ на промежутке $[n, n+1)$. Учитывая, что

$$P_0(f) = \sum_{n\in\mathbf{Z}} (f, \varphi_{0,n})\varphi_{0,n}(x) = \sum_{n\in\mathbf{Z}} f_{0n}\,\varphi(x-n)$$

и $\varphi_{0,n}(x)$ есть характеристическая функция промежутка $[n, n+1)$, получаем, что $P_0(f)$ – это кусочно-постоянная функция, принимающая среднее значение $f_{0,n}$ на каждом промежутке $[n, n+1)$, см. рис. 2.1.2 и 2.1.3.

Для $P_1(f)$ имеем аналогично:

$$(f, \varphi_{1,n}) = \sqrt{2}\int_{-\infty}^{\infty} f(x)\,\varphi(2x-n)\,dx = \sqrt{2}\int_{\frac{n}{2}}^{\frac{n+1}{2}} f(x)\,dx =$$

$$= \frac{1}{\sqrt{2}} 2\int_{\frac{n}{2}}^{\frac{n+1}{2}} f(x)\,dx = \frac{1}{\sqrt{2}} f_{1n}(n),$$

где f_{1n} – среднее значение функции $f(x)$ на промежутке $[n/2, (n+1)/2)$ длины $1/2$. Тогда

$$P_1(f) = \sum_{n\in\mathbf{Z}} (f, \varphi_{1,n})\varphi_{1,n}(x) = \sum_{n\in\mathbf{Z}} \frac{1}{\sqrt{2}} f_{1n}(n)\,\sqrt{2}\,\varphi(2x-n) =$$

$$= \sum_{n\in\mathbf{Z}} f_{1n}(n)\varphi(2x-n)$$

есть ступенчатая функция, принимающая среднее значение f_{1n} на каждом промежутке $[n/2, (n+1)/2)$.

На рис. 2.1.2 показаны графики результатов действия операторов проектирования P_0, P_1 и P_2 на функцию $f(x) = \sin x$ на промежутке $[-1, 7]$. Так выглядит функция $\sin x$, если ее спроектировать из $L^2(\mathbf{R})$ на подпространства V_0, V_1 и V_2.

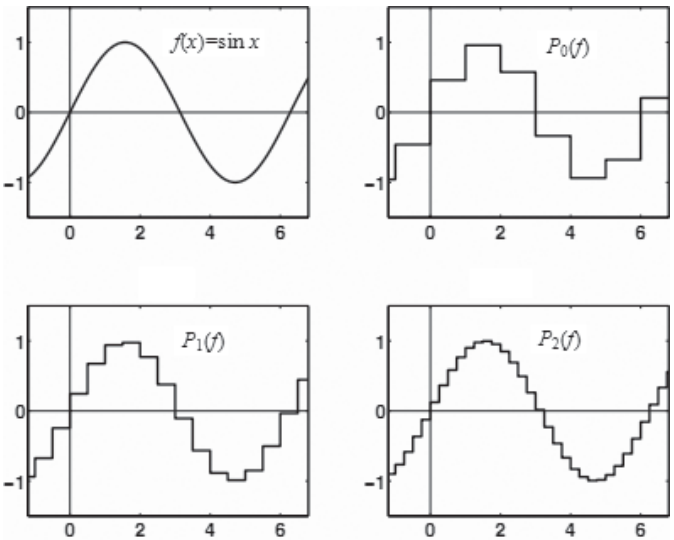


Рис. 2.1.2. Последовательные приближения к функции $f(x) = \sin x$

Проекции $P_j(f)$ являются приближениями функции $f(x)$, все более точными при увеличении j. Интересно сравнить два последовательных приближения. С этой целью рассмотрим разность $P_j^W(f) = P_{j+1}(f) - P_j(f)$. Эта разность лежит в V_{j+1} и ортогональна к $V_j \in V_{j+1}$, то есть она принадлежит ортогональному дополнению W_j к пространству V_j до V_{j+1}. Тогда формула

$$P_{j+1}(f) = P_j(f) + P_j^W(f)$$

соответствует ортогональному разложению

$$V_{j+1} = V_j \oplus W_j.$$

Более точное приближение $P_{j+1}(f)$ в пространстве V_{j+1} получается как сумма приближения $P_j(f)$ в пространстве V_j и дополнительного слагаемого $P_j^W(f)$, отражающего детали, которыми $P_{j+1}(f)$ отличается от $P_j(f)$.

Таким образом, пространства V_j содержат приближения $P_j(f)$ функций, а пространства W_j содержат детали $P_j^W(f)$, которыми более точное приближение отличается от более грубого.

На следующем рис. 2.1.3 для функции $f(x) = \sin x$ на промежутке $[-1, 7]$ изображены последовательно: приближение $P_0(f)$, более точное приближение $P_1(f)$ и разность $P_1^W(f) = P_1(f) - P_0(f)$.

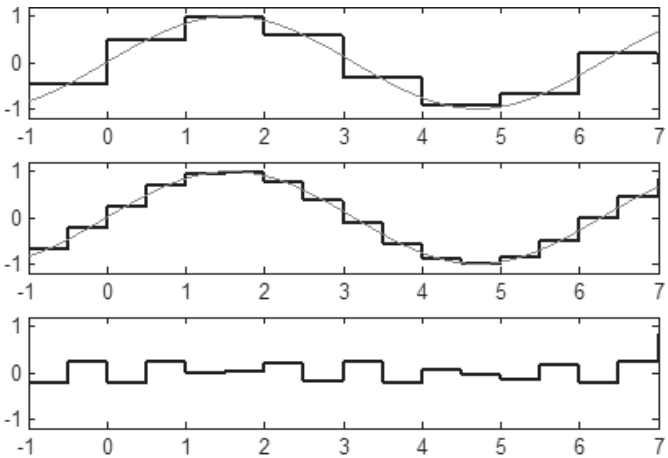


Рис. 2.1.3. Последовательные приближения и детали для $f(x) = \sin x$

2.1.3. Пространства вейвлетов

Используем введенные в предыдущем параграфе пространства деталей W_j для построения базиса всего пространства $L^2(\mathbf{R})$. Каждое из введенных выше подпространств V_j имеет свой базис, состоящий из функций $\varphi_{j,n}(x) = \sqrt{2^j}\varphi(2^j x - n)$. При этом базис следующего пространства V_{j+1} не содержит элементов базиса пространства V_j. Поэтому мы не можем пока из базисов пространств V_j получить базис всего пространства $L^2(\mathbf{R})$. Однако этого можно было бы достигнуть, если бы базис следующего пространства получался бы из базиса предыдущего пространства добавлением новых элементов.

Рассмотрим для простоты пространства V_0 и V_1. Поскольку V_0 есть замкнутое подпространство V_1, то существует ортогональное допол-

нение к V_0 в пространстве V_1, обозначим его W_0. Тогда мы имеем ортогональное разложение пространства V_1:

$V_1 = V_0 \oplus W_0$.

Поэтому к базису пространства V_0 можно добавить базис дополнительного пространства W_0 и в результате получить базис более широкого пространства V_1. Для того чтобы реализовать эту процедуру, выясним, из каких функций состоит W_0.

Пусть функция $\psi(x) \in W_0$. Поскольку $W_0 \subset V_1$, тогда $\psi(x)$ раскладывается по базису $\{\varphi_{1,n}(x)\}$ пространства V_1: $\psi = \sum_n c_n \varphi_{1,n}$. Так как $\psi \perp V_0$, то для любого n имеем: $(\psi, \varphi_{0,n}) = 0$. Пространство V_0 входит в V_1, следовательно, функция $\varphi_{0,n}(x)$ также раскладывается по базису $\{\varphi_{1,n}(x)\}$ пространства V_1. Коэффициенты этого разложения ранее были найдены:

$$\varphi_{0,n} = \frac{1}{\sqrt{2}}\varphi_{1,2n} + \frac{1}{\sqrt{2}}\varphi_{1,2n+1}. \tag{4}$$

Тогда условие ортогональности $\psi \perp V_0$ принимает вид:

$$(\psi, \varphi_{0,n}) = \left(\sum_k c_k \varphi_{1,k}, \frac{1}{\sqrt{2}}\varphi_{1,2n} + \frac{1}{\sqrt{2}}\varphi_{1,2n+1} \right) = 0, \quad \forall\, n.$$

Так как $\{\varphi_{1,n}(x)\}$ – ортонормированный базис, то из последнего равенства имеем:

$$c_{2n}\frac{1}{\sqrt{2}} + c_{2n+1}\frac{1}{\sqrt{2}} = 0, \quad \forall\, n.$$

Полученная система уравнений имеет множество решений (их совокупность порождает пространство W_0). Возьмем наиболее простое решение, состоящее из двух ненулевых значений:

$$c_0 = \frac{1}{\sqrt{2}}, \quad c_1 = -\frac{1}{\sqrt{2}}.$$

Ему соответствует функция

$$\psi(x) = \frac{1}{\sqrt{2}}\varphi_{1,0}(x) - \frac{1}{\sqrt{2}}\varphi_{1,1}(x) = \varphi(2x) - \varphi(2x-1),$$

$$\psi(x) = \begin{cases} 1, & x \in [0,1/2) \\ -1, & x \in [1/2,1), \\ 0, & x \notin [0,1) \end{cases} \tag{5}$$

называемая *вейвлетом Хаара*. Она замечательна тем, что ее сдвиги $\psi(x-n)$ образуют базис пространства W_0. Это покажем чуть позже, а сейчас приведем графики вейвлета Хаара $y=\psi(x)$ и его масштабированных сдвигов $y=\sqrt{2}\psi(2x-3)$, $y=2\psi(4x-11)$.

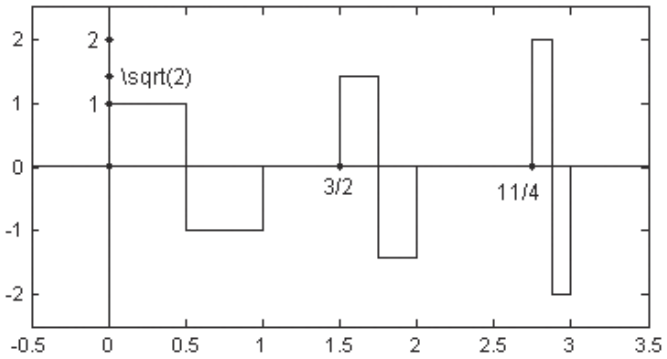


Рис. 2.1.4. Сдвиги
масштабированных вейвлетов Хаара

Покажем, что функции $\psi_{0,n}=\psi(x-n)=\frac{1}{\sqrt{2}}\varphi_{1,2n}-\frac{1}{\sqrt{2}}\varphi_{1,2n+1}$ образуют ортонормированную систему в W_0. Система функций $\{\varphi_{0,n}(x),\ \psi_{0,n}(x),\ n\in\mathbf{Z}\}$ образует (новый) ортонормированный базис пространства V_1. Это следует из того, что любой базисный элемент $\varphi_{1,n}$ пространства V_1 выражается через $\psi_{0,n}(x)$ и $\varphi_{0,n}(x)$. Действительно,

$$\varphi_{1,0}(x)=\frac{\varphi_{0,0}(x)+\psi_{0,0}(x)}{\sqrt{2}},\qquad \varphi_{1,1}(x)=\frac{\varphi_{0,0}(x)-\psi_{0,0}(x)}{\sqrt{2}}.$$

Выражения для остальных $\varphi_{1,n}(x)$ получаются сдвигами на n.

Таким образом, мы получили ортонормированный базис $\{\varphi_{0,n}(x),\ \psi_{0,n}(x), n\in\mathbf{Z}\}$ пространства V_1, где первый набор функций $\{\varphi_{0,n}(x)\}$ образует ортонормированный базис пространства V_0, и, следовательно, второй набор $\{\psi_{0,n}(x)\}$ – это базис дополнительного пространства W_0 в соответствии с разложением $V_1=V_0\oplus W_0$. Новый базис пространства V_1 получился из базиса V_0 добавлением элементов из W_0.

Аналогичным образом можно получить базис пространства V_2. Поскольку V_1 есть замкнутое подпространство V_2, то существует ортогональное дополнение к V_1 в пространстве V_2, обозначим его W_1, тогда

$V_2=V_1\oplus W_1$.

Учитывая, что $V_1 = V_0 \oplus W_0$, получаем:

$V_2 = V_0 \oplus W_0 \oplus W_1$.

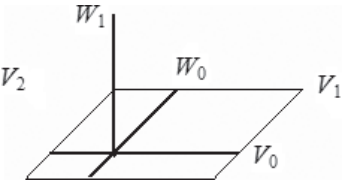


Рис. 2.1.5. Ортогональное разложение пространства V_2

По построению пространство V_2 является масштабированной версией пространства V_1, другими словами, $f(x) \in V_1 \Leftrightarrow f(2x) \in V_2$. Легко видеть, что при соответствии $f(x) \to f(2x)$ ортогональность функций сохраняется. Поэтому и пространство W_1 является масштабированной версией пространства W_0. Следовательно, полуцелые сдвиги $\psi(2x-n)$ функции $\psi(2x)$ образуют базис пространства W_1. Обозначим $\psi_{1,n}(x) = \sqrt{2}\psi(2x-n)$ нормированные базисные функции. Поскольку базис пространства V_1 образован функциями $\{\varphi_{0,n}(x), \psi_{0,n}(x), n \in \mathbf{Z}\}$, то базис пространства $V_2 = V_1 \oplus W_1$ состоит из элементов $\{\varphi_{0,n}(x), \psi_{0,n}(x), \psi_{1,n}(x), n \in \mathbf{Z}\}$. Он получен из базиса пространства V_0 добавлением новых элементов $\psi_{0,n}(x)$ и $\psi_{1,n}(x)$. Ясно, что процесс можно продолжить до бесконечности, используя разложение $V_{j+1} = V_j \oplus W_j$ для любого $j \in \mathbf{Z}$. Тогда на $(j+1)$-ом этапе имеем:

$V_{j+1} = V_0 \oplus W_0 \oplus W_1 \oplus \ldots \oplus W_j$,

где ортонормированный базис пространства W_j образуют функции вида

$$\psi_{j,n}(x) = \sqrt{2^j}\psi(2^j x - n). \qquad (6)$$

Поскольку $L^2(\mathbf{R}) = \overline{\bigcup_{j=-\infty}^{\infty} V_j}$, то

$L^2(\mathbf{R}) = V_0 \oplus W_0 \oplus W_1 \oplus \ldots \oplus W_j \oplus \ldots$

и ортонормированный базис пространства $L^2(\mathbf{R})$ состоит из функций

$\{\varphi_{0,n}(x), \psi_{j,n}(x), j = 0, 1, 2, \ldots, n \in \mathbf{Z}\}$.

Мы можем также продолжить разложение «в отрицательную сторону», $V_0 = V_{-1} \oplus W_{-1} = V_{-2} \oplus W_{-2} \oplus W_{-1} = \ldots$ Поскольку $\bigcap_{j=-\infty}^{\infty} V_j = \{0\}$, то в пределе при $j \to -\infty$ мы получаем

$V_0 = \oplus_{j=-\infty}^{-1} W_j$.

Следовательно,

$$L^2(\mathbf{R}) = \oplus_{j=-\infty}^{\infty} W_j. \quad (7)$$

Поэтому ортонормированный базис пространства $L^2(\mathbf{R})$ будут образовывать функции

$$\psi_{j,n}(x) = \sqrt{2^j}\,\psi(2^j x - n), \quad j,n \in \mathbf{Z}. \quad (8)$$

Это хорошо известный *базис Хаара*. Он называется также базисом вейвлетов Хаара.

Определение 1. *Элементы пространства W_j называются вейвлетами Хаара. Функции $\psi_{j,n}(x) = \sqrt{2^j}\,\psi(2^j x - n)$ называются базисными вейвлетами. Функция $\varphi(x)$ называется масштабирующей функцией Хаара. Функция $\psi(x)$ называется материнским вейвлетом Хаара.*

Замечание. На самом деле мы построили нечто большее, чем просто ортонормированный базис $\psi_{j,n}(x) = \sqrt{2^j}\,\psi(2^j x - n), \quad j,n \in \mathbf{Z}$ пространства $L^2(\mathbf{R})$. Мы получили систему подпространств

$$\ldots \subset V_{-1} \subset V_0 \subset V_1 \subset V_2 \subset \ldots \subset V_j \subset \ldots,$$

проекции на которые дают приближения элементов $f \in L^2(\mathbf{R})$. Указанная система обладает свойствами:

$$\overline{\bigcup_{j\in\mathbf{Z}} V_j} = L^2(\mathbf{R}) \quad \text{и} \quad \bigcap_{j\in\mathbf{Z}} V_j = \{0\}.$$

Пространства V_j являются масштабированными версиями пространства V_0, $f(x) \in V_j \Leftrightarrow f(2x) \in V_{j+1}$. Существует функция $\varphi(x)$, такая что для каждого $j \in \mathbf{Z}$ функции $\varphi_{j,n}(x) = \sqrt{2^j}\,\varphi(2^j x - n)$ образуют ортонормированный базис пространства V_j. Для каждого $j \in \mathbf{Z}$ имеет место разложение $V_{j+1} = V_j \oplus W_j$, где дополнительное пространство вейвлетов W_j имеет ортонормированный базис $\psi_{j,n}(x) = \sqrt{2^j}\,\psi(2^j x - n)$, порожденный одной функцией $\psi(x)$.

Оказывается, эта конструкция является достаточно общей. Существуют и другие функции, которые также порождают аналогичное разложение пространства $L^2(\mathbf{R})$. В рассмотренном случае приближения $P_j(f)$ функции $f(x)$ являются разрывными (ступенчатыми) функциями. Естественно ожидать лучшей аппроксимации гладкой функции, если вместо разрывных функций Хаара использовать гладкие функции $\varphi(x)$ и $\psi(x)$. Вопрос о построении таких функций составляет основное содержание данной главы.

2.2. Масштабирующие функции

В этом параграфе рассмотрим функции $\varphi(x)$, которые порождают, подобно функции Хаара, возрастающую последовательность подпространств $... \subset V_{-1} \subset V_0 \subset V_1 \subset ...$ Если пространство V_0 образовано сдвигами функции $\varphi(x)$, а пространство V_1 – сдвигами $\varphi(2x - n)$, то для включения $V_0 \subset V_1$ необходимо, чтобы $\varphi(x) \in V_1$, то есть чтобы $\varphi(x)$ могла быть представлена в виде линейной комбинации функций $\varphi(2x - n)$. Далеко не каждая функция обладает таким свойством.

Определение 1. *Функция* $\varphi(x) \in L^2(\mathbf{R})$ *называется масштабирующей, если она может быть представлена в виде*

$$\varphi(x) = \sqrt{2}\sum_{n\in\mathbf{Z}} h_n \varphi(2x - n), \tag{1}$$

где числа $\{h_n, n \in \mathbf{Z}\}$ *удовлетворяют условию*

$$\sum_{n\in\mathbf{Z}} |h_n|^2 < \infty. \tag{2}$$

Равенство (1) *называется масштабирующим уравнением. Набор* $\{h_n\}$ *коэффициентов разложения в уравнении* (1) *называется масштабирующим фильтром.*

2.2.1. Примеры и общие свойства масштабирующих функций

Приведем несколько простых примеров масштабирующих функций.

Пример 1. Функция Хаара $\varphi(x)$, рассмотренная в §1. Отметим, что функция, равная единице на промежутке $[-1/2, 1/2)$ и равная нулю вне промежутка $[-1/2, 1/2)$, не является масштабирующей.

Пример 2. Функция

$$\varphi(x) = \begin{cases} 1, & x \in [-1,1) \\ 0, & \text{для других } x \end{cases}$$

является масштабирующей с масштабирующим соотношением:

$$\varphi(x) = \varphi(2x - 1) + \varphi(2x + 1).$$

Пример 3. Функция «шляпа». Это B-сплайн $\varphi(x)$ степени 1,

$$\varphi(x) = \begin{cases} 1 - |x|, & x \in [-1,1], \\ 0 & \text{при остальных } x \end{cases}. \tag{3}$$

Функция имеет компактный носитель $[-1, 1]$ и удовлетворяет масштабирующему уравнению

$$\varphi(x)=\frac{1}{2}\varphi(2x+1)+\varphi(2x)+\frac{1}{2}\varphi(2x-1). \qquad (4)$$

Фильтр $\{h_n\}$ состоит из трех ненулевых чисел $h_{-1}=1/2\sqrt{2}$, $h_0=1/\sqrt{2}$, $h_1=1/2\sqrt{2}$. Приведенный ниже рисунок иллюстрирует равенство (4). Справа на рисунке изображены графики функций $0{,}5\varphi(2x+1)$, $\varphi(2x)$, $0{,}5\varphi(2x-1)$.

Отметим, что сдвиги $\{\varphi(x-n), n\in\mathbf{Z}\}$ образуют базис пространства V_0, но этот базис не ортогональный.

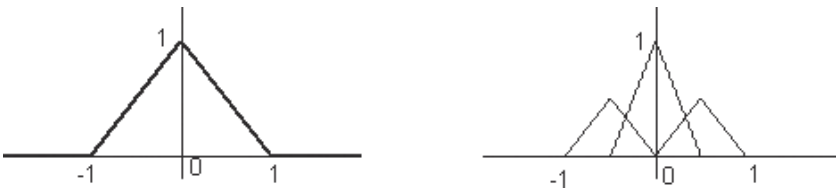


Рис. 2.2.1. Масштабирующее соотношение для B-сплайна степени 1

Пример 4. B-сплайн $\varphi_2(x)$ степени 2:

$$\varphi_2(x)=\begin{cases}\dfrac{1}{2}(x+1)^2, & x\in[-1,0]\\ \dfrac{3}{4}-\left(x-\dfrac{1}{2}\right)^2, & x\in[0,1]\\ \dfrac{1}{2}(x-2)^2, & x\in[1,2]\\ 0, & x\notin[-1,2]\end{cases}. \qquad (5)$$

Масштабирующее соотношение:

$$\varphi_2(x)=\frac{1}{4}\varphi_2(2x+1)+\frac{3}{4}\varphi_2(2x)+\frac{3}{4}\varphi_2(2x-1)+\frac{1}{4}\varphi_2(2x-2).$$

Следующий рис. 2.2.2 иллюстрирует последнее равенство.

Пример 5. Масштабирующая функция $\varphi(x)=(\sin\pi x)/\pi x$. Коэффициенты $\{h_n\}$ разложения $\varphi(x)=\sqrt{2}\sum_n h_n\varphi(2x-n)$ имеют вид $h_0=1/\sqrt{2}$, $h_n=(-1)^{(n-1)/2}\sqrt{2}/(\pi n)$, когда n – нечетное и $h_n=0$ – для остальных n. Подробнее об этой функции см. в § 5 данной главы.

Замечание 1. Если масштабирующая функция $\varphi(x)$ имеет компактный носитель, лежащий на промежутке длины N, то сумма в ра-

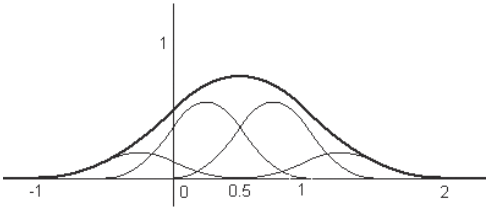


Рис. 2.2.2. Масштабирующее соотношение для B-сплайна степени 2

венстве (1) является конечной, состоящей не более чем из $N + 1$ слагаемого. Это хорошо видно на приведенных выше примерах 1–3.

Пусть $\varphi(x)$ – масштабирующая функция. Образуем следующие функции:

$$\varphi_{j,n}(x) = \sqrt{2^j}\,\varphi(2^j x - n), \quad j, n \in \mathbf{Z}. \tag{6}$$

Предложение 1. *Если* $\varphi(x)$ *– масштабирующая функция с масштабирующим фильтром* $\{h_n, n \in \mathbf{Z}\}$*, то для любых* $j, k \in \mathbf{Z}$ *имеет место следующее разложение*:

$$\varphi_{j-1,k}(x) = \sum_n h_{n-2k}\,\varphi_{j,n}(x) = \sum_n h_n\,\varphi_{j,n+2k}(x). \tag{7}$$

В частности,

$$\varphi_{0,k}(x) = \sqrt{2}\sum_n h_{n-2k}\varphi(2x - n), \quad \varphi_{j-1,0}(x) = \sum_n h_n \varphi_{j,n}(x). \tag{8}$$

Доказательство.

$$\varphi_{j-1,k}(x) = \sqrt{2^{j-1}}\,\varphi(2^{j-1}x - k) = \sqrt{2}\sqrt{2^{j-1}}\sum_m h_m \varphi(2(2^{j-1}x - k) - m) =$$

$$= \sqrt{2^j}\sum_m h_m \varphi(2^j x - 2k - m) = \sqrt{2^j}\sum_m h_m \varphi_{j,m+2k}(x) = \sum_n h_{n-2k}\varphi_{j,n}(x).$$

Сделаем преобразование Фурье масштабирующего соотношения $\varphi(x) = \sqrt{2}\sum_{n\in\mathbf{Z}} h_n \varphi(2x - n)$. Поскольку $\varphi(2x - n) = \varphi(2(x - n/2))$, то $F[\varphi(2x - n)] = e^{-in\omega/2}F[\varphi](\omega/2)/2$. Поэтому

$$\hat{\varphi}(\omega) = \frac{1}{\sqrt{2}}\sum_{n\in\mathbf{Z}} h_n\, e^{-in\frac{\omega}{2}}\,\hat{\varphi}\left(\frac{\omega}{2}\right).$$

Пусть

$$H_0(\omega) = \frac{1}{\sqrt{2}}\sum_{n\in\mathbf{Z}} h_n\, e^{-in\omega}, \tag{9}$$

тогда

$$\hat{\varphi}(\omega) = H_0\left(\frac{\omega}{2}\right)\hat{\varphi}\left(\frac{\omega}{2}\right). \qquad (10)$$

Соотношение (10) также называется *масштабирующим уравнением*. Функцию $H_0(\omega)$ будем называть *частотной функцией* масштабирующей функции $\varphi(x)$.

Замечание 2. Из условия (2) на коэффициенты $\{h_n\}$ следует, что функция $H_0(\omega)$ определена корректно. Функция $H_0(\omega)$ является 2π-периодической, принадлежащей пространству $L^2[0, 2\pi]$. Если вместо (2) предполагать условие $\sum_{n\in\mathbf{Z}}|h_n| < \infty$, то ряд (9) будет сходиться равномерно и поэтому его сумма $H_0(\omega)$ будет непрерывной 2π-периодической функцией. Отметим, что коэффициенты $\{h_n\}$ масштабирующей функции из примера 4 не удовлетворяют условию $\sum_{n\in\mathbf{Z}}|h_n| < \infty$.

Замечание 3. Из масштабирующего уравнения сразу следует, что масштабирующая функция определена с точностью до константы. Легко видеть, что если $\varphi(x)$ – масштабирующая функция, то для целого числа L функция $\varphi(x - L)$ также является масштабирующей с фильтром $b_n := h_{n-L}$. Также легко видеть, что если $\varphi(x)$ – масштабирующая функция, то функция $\varphi(-x)$ также является масштабирующей с фильтром $b_n := h_{-n}$. Кроме того, если $\varphi_1(x)$ и $\varphi_2(x)$ – две масштабирующие функции с частотными функциями $H_{01}(\omega)$ и $H_{02}(\omega)$, то их свертка $\varphi_1(x)*\varphi_2(x)$ является масштабирующей функцией с частотной функцией, равной $H_{01}(\omega)H_{02}(\omega)$. Действительно, перемножая масштабирующие соотношения $\hat{\varphi}_1(\omega) = H_{01}(\omega/2)\,\hat{\varphi}_1(\omega/2)$ и $\hat{\varphi}_2(\omega) = H_{02}(\omega/2)\,\hat{\varphi}_2(\omega/2)$, получаем масштабирующее соотношение для свертки $\varphi_1(x)*\varphi_2(x)$:

$$\hat{\varphi}_1(\omega)\hat{\varphi}_2(\omega) = H_{01}(\omega/2)H_{02}(\omega/2)\,\hat{\varphi}_1(\omega/2)\hat{\varphi}_2(\omega/2).$$

Таким образом, множество $M_2(\mathbf{R})$ масштабирующих функций является подмножеством в $L^2(\mathbf{R})$, замкнутым относительно умножения на числа, целочисленных сдвигов и отражений, а также относительно операции свертки.

Замечание 4. Из масштабирующего уравнения (10) вытекает очевидное, но важное наблюдение. Если функция $\hat{\varphi}(\omega)$ непрерывна и интеграл от функции $\varphi(x)$ отличен от нуля, то, подставляя в (10) значение $\omega = 0$, получаем $\hat{\varphi}(0) = H_0(0)\,\hat{\varphi}(0)$. Следовательно,

$$H_0(0) = \frac{1}{\sqrt{2}} \sum_n h_n = 1. \qquad (11)$$

Замечание 5. В дальнейшем будет показано, что масштабирующая функция $\varphi(x)$ должна иметь ненулевой интеграл. Поскольку она определена с точностью до числового сомножителя, для однозначного выбора требуется нормировка $\varphi(x)$. В этом параграфе будем ее нормировать так, чтобы $\hat{\varphi}(0) = 1$. В следующих параграфах будет предполагаться, что

$$\| \varphi(x) \|_{L^2}^2 = \int_{-\infty}^{\infty} | \varphi(x) |^2 dx = 1.$$

Пример 6. Для функции Хаара фильтр коэффициентов $\{h_n\}$ состоит из двух ненулевых элементов $h_0 = 1/\sqrt{2}$, $h_1 = 1/\sqrt{2}$. Поэтому

$$H_0(\omega) = \frac{1}{\sqrt{2}} \left(\frac{1}{\sqrt{2}} + \frac{1}{\sqrt{2}} e^{-i\omega} \right) = \frac{1}{2}(1 + e^{-i\omega}) = e^{-i\omega/2} \cos \omega / 2 \ .$$

Пример 7. Для функции «шляпы» фильтр коэффициентов $\{h_n\}$ состоит из трех ненулевых элементов: $h_{-1} = 1/2\sqrt{2}$, $h_0 = 1/\sqrt{2}$, $h_1 = 1/2\sqrt{2}$. Поэтому

$$H_0(\omega) = \frac{1}{\sqrt{2}} \frac{1}{2\sqrt{2}} (e^{i\omega} + 2 + e^{-i\omega}) = \frac{1}{2}(1 + \cos \omega) = \cos^2 \omega / 2.$$

2.2.2. Построение масштабирующей функции

Далеко не каждая функция является масштабирующей. Поэтому важно научиться находить такие функции. Будем сначала считать, что коэффициенты $\{h_n\}$ известны и требуется найти функцию $\varphi(x)$ (условия для коэффициентов и методы их нахождения будут рассмотрены далее). Есть несколько способов построения функции $\varphi(x)$. Рассмотрим три из них.

1. Итерационный метод. Рассмотрим масштабирующее уравнение $\varphi(x) = \sqrt{2} \sum_{n \in \mathbf{Z}} h_n \varphi(2x - n)$, где числа $\{h_n\}$ удовлетворяют условию $\sum_{n \in \mathbf{Z}} | h_n |^2 < \infty$. Запишем это уравнение в виде операторного уравнения. Для этого введем два унитарных оператора:

- оператор A_2 изменения масштаба, $A_2(f)(x) = \sqrt{2} f(2x)$;
- оператор T_n сдвига на число n, $T_n(f)(x) = f(x - n)$, $n \in \mathbf{Z}$.

Тогда очевидно, что $\sqrt{2}\varphi(2x - n) = A_2 \circ T_n(\varphi(x))$. Поэтому масштабирующее соотношение принимает вид:

$$\varphi(x) = \sum_{n \in Z} h_n A_2 \circ T_n(\varphi(x)) = M(\varphi(x)),$$

где оператор M: $L^2(\mathbf{R}) \to L^2(\mathbf{R})$ есть линейная комбинация операторов $A_2 \circ T_n$, $M = \sum_{n \in Z} h_n A_2 \circ T_n$. Масштабирующее уравнение, записанное в виде

$$\varphi(x) = M(\varphi(x)),$$

показывает, что масштабирующая функция $\varphi(x)$ является неподвижной точкой оператора M. Поэтому $\varphi(x)$ может быть определена последовательными приближениями

$$\varphi_{n+1}(x) = M(\varphi_n(x)), \quad n \to \infty,$$

где в качестве начального приближения можно взять, например, функцию $\varphi_0(x)$ Хаара. При этом возникает вопрос о сходимости последовательных приближений. Все определяется конкретным выбором коэффициентов h_n. Пока неизвестно, при каких коэффициентах итерации сходятся к достаточно хорошей функции.

Пример 8. Возьмем в качестве коэффициентов следующие:

$$h_0 = \frac{1+\sqrt{3}}{4\sqrt{2}}, \quad h_1 = \frac{3+\sqrt{3}}{4\sqrt{2}}, \quad h_2 = \frac{3-\sqrt{3}}{4\sqrt{2}}, \quad h_3 = \frac{1-\sqrt{3}}{4\sqrt{2}},$$

а в качестве начального приближения – функцию $\varphi_0(x)$ Хаара. На рисунке ниже изображены первые пять приближений. В пределе получим известную $\varphi(x)$-функцию Добеши.

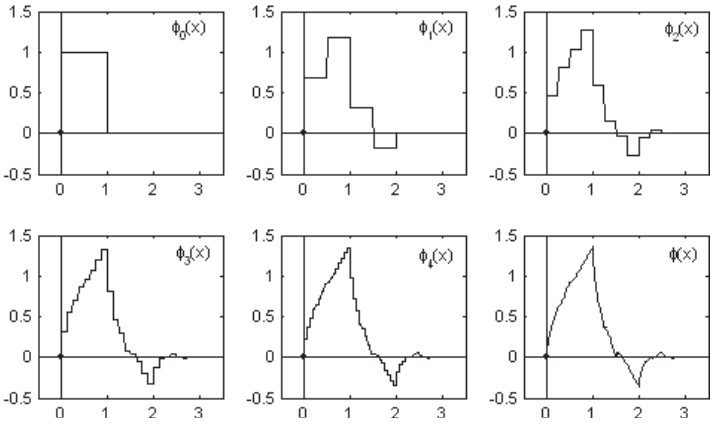


Рис. 2.2.3. Последовательные приближения $\varphi(x)$-функции Добеши

Полученная функция удовлетворяет уравнению

$$\varphi(x)=\frac{1+\sqrt{3}}{4}\varphi(2x)+\frac{3+\sqrt{3}}{4}\varphi(2x-1)+\frac{3-\sqrt{3}}{4}\varphi(2x-2)+\frac{1-\sqrt{3}}{4}\varphi(2x-3).$$

Замечание 6. Укажем простые программы в MATLAB для нахождения итераций $\varphi_n(x)$.

```
function y=Ha01(x); %Начальная функция Хаара
for k=1:length(x);
      if x(k) < 0; y(k)=0;
      elseif x(k) < 1; y(k)=1;
      else y(k)=0;
end end

function y=Phi1(x); %Первая итерация
h0=(1+sqrt(3))/4; h1=(3+sqrt(3))/4; % Коэффициенты
h2=(3-sqrt(3))/4; h3=(1-sqrt(3))/4;
for k=1:length(x);
     y0(k)=Ha01(2*x(k)); y1(k)=Ha01(2*x(k)-1);
     y2(k)=Ha01(2*x(k)-2); y3(k)=Ha01(2*x(k)-3);
     y(k)=h0*y0(k)+h1*y1(k)+h2*y2(k)+h3*y3(k);
end
```

Вторая итерация `y=Phi2(x)` получается простой заменой заголовка `y=Phi1(x)` на `y=Phi2(x)` и заменой функции `Ha01(...)` на `Phi1(...)` в листинге программы функции `y= Phi2(x)`.

Замечание 7. Поскольку итерационный процесс обладает очевидными свойствами самоподобия, предельная масштабирующая функция часто обладает фрактальными свойствами.

2. Рекуррентный метод. Предполагается, что носитель функции $\varphi(x)$ лежит на промежутке $[0, N]$. Тогда масштабирующее уравнение принимает вид конечной суммы

$$\varphi(x)=\sqrt{2}\sum_{n=0}^{N}h_n\varphi(2x-n), \tag{13}$$

поскольку остальные коэффициенты h_n являются нулевыми. Предположим сначала, что известны значения масштабирующей функции $\varphi(x)$ в целых точках, тогда из уравнения (13) можно найти значения масштабирующей функции в полуцелых числах:

$$\varphi(k/2)=\sqrt{2}\sum_{n=0}^{N}h_n\varphi(k-n),\quad k\in\mathbf{Z}.$$

Затем по значениям в полуцелых числах находим значения в четвертях целых чисел и т. д. Таким образом мы можем получить все значения $\varphi(x)$ в диадических точках $x = k/2^j$, где $k \in \mathbf{Z}$, $j = 1, 2, \ldots$.

Поскольку любое действительное число x может быть аппроксимировано диадической точкой вида $k/2^j$ с любой заданной точностью, то непрерывная функция $\varphi(x)$ в любой точке x может быть вычислена с любой заданной степенью точности. Практически мы можем вычислить значения $\varphi(x)$ в диадических точках $x = k/2^j$ с интерполяцией в промежуточных значениях x.

Рассмотрим задачу нахождения значений $\varphi(x)$ в целых числах. Пусть $x = k$, тогда из уравнения (13), используя конечность фильтра $\{h_n\}$ и конечность носителя функции $\varphi(x)$, получаем:

$$\varphi(k) = \sqrt{2}\sum_{n=0}^{N} h_n \varphi(2k-n) = \sqrt{2}\sum_{n\in Z} h_n \varphi(2k-n) = \sqrt{2}\sum_{s=0}^{N} h_{2k-s}\varphi(s). \qquad (14)$$

Поскольку носитель $\varphi(x)$ находится на отрезке $[0, N]$, то вектор значений в целых точках можно считать конечным, $\varphi = (\varphi(0), \varphi(1), \ldots, \varphi(N))$. Последнее уравнение (14) можно тогда записать в матричном виде $H\varphi = \varphi$, где $(N + 1)\times(N + 1)$ матрица H имеет элементы $H_{ks} = \sqrt{2}h_{2k-s}$. Следовательно, вектор φ является собственным вектором, относящимся к собственному значению 1. В матричной форме имеем:

$$\sqrt{2}\begin{pmatrix} h_0 & 0 & 0 & 0 & & & \\ h_2 & h_1 & h_0 & 0 & & \cdots & \\ h_4 & h_3 & h_2 & h_1 & & & \\ h_6 & h_5 & h_4 & h_3 & & & \\ & & & & \cdots & \cdots & \cdots \\ & \cdots & & & \cdots & h_{N-1} & h_{N-2} \\ & & & & \cdots & 0 & h_N \end{pmatrix}\begin{pmatrix} \varphi(0) \\ \varphi(1) \\ \varphi(2) \\ \varphi(3) \\ \vdots \\ \varphi(N-1) \\ \varphi(N) \end{pmatrix} = \begin{pmatrix} \varphi(0) \\ \varphi(1) \\ \varphi(2) \\ \varphi(3) \\ \vdots \\ \varphi(N-1) \\ \varphi(N) \end{pmatrix}.$$

Так значения $\varphi(x)$ в целых числах находятся как собственный вектор оператора H, соответствующий собственному значению 1. Мы нормализуем этот вектор требованием $\sum_k \varphi(k) = 1$. Напомним, что после нахождения значений $\varphi(x)$ в целых числах мы можем по рекуррентной формуле (13) найти значения $\varphi(x)$ в полуцелых, четвертях целых и т. д.

Отметим, что если функция $\varphi(x)$ непрерывна и ее носитель находится на промежутке $[0, N]$, то $\varphi(0) = 0$ и $\varphi(N) = 0$, тогда размер матрицы H будет $(N - 1)\times(N - 1)$.

Пример 9. Возьмем в качестве коэффициентов следующие:

$$h_0 = \frac{1+\sqrt{3}}{4\sqrt{2}}, \quad h_1 = \frac{3+\sqrt{3}}{4\sqrt{2}}, \quad h_2 = \frac{3-\sqrt{3}}{4\sqrt{2}}, \quad h_3 = \frac{1-\sqrt{3}}{4\sqrt{2}}.$$

Тогда матричное уравнение принимает вид:

$$\frac{1}{4}\begin{pmatrix} 1+\sqrt{3} & 0 & 0 & 0 \\ 3-\sqrt{3} & 3+\sqrt{3} & 1+\sqrt{3} & 0 \\ 0 & 1-\sqrt{3} & 3-\sqrt{3} & 3+\sqrt{3} \\ 0 & 0 & 0 & 1-\sqrt{3} \end{pmatrix}\begin{pmatrix} \varphi(0) \\ \varphi(1) \\ \varphi(2) \\ \varphi(3) \end{pmatrix} = \begin{pmatrix} \varphi(0) \\ \varphi(1) \\ \varphi(2) \\ \varphi(3) \end{pmatrix}.$$

Поскольку функция Добеши $\varphi(x)$ непрерывна и имеет носитель на промежутке $[0, 3]$, то $\varphi(0) = 0$ и $\varphi(3) = 0$, поэтому уравнение принимает вид:

$$\frac{1}{4}\begin{pmatrix} 3+\sqrt{3} & 1+\sqrt{3} \\ 1-\sqrt{3} & 3-\sqrt{3} \end{pmatrix}\begin{pmatrix} \varphi(1) \\ \varphi(2) \end{pmatrix} = \begin{pmatrix} \varphi(1) \\ \varphi(2) \end{pmatrix}, \quad \begin{pmatrix} -1+\sqrt{3} & 1+\sqrt{3} \\ 1-\sqrt{3} & -1-\sqrt{3} \end{pmatrix}\begin{pmatrix} \varphi(1) \\ \varphi(2) \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \end{pmatrix}.$$

Последнее легко решается:

$$\varphi(1) = \frac{1+\sqrt{3}}{1-\sqrt{3}}\varphi(2).$$

Нормализуем $\varphi(x)$, полагая $\sum_k \varphi(k) = 1$. Тогда получаем ненулевые значения функции Добеши $\varphi(x)$ в целых точках:

$$\varphi(1) = \frac{1+\sqrt{3}}{2}, \quad \varphi(2) = \frac{1-\sqrt{3}}{2}.$$

3. Бесконечное произведение. Третий способ построения масштабирующей функции использует масштабирующее уравнение для образов Фурье $\hat{\varphi}(\omega) = H_0(\omega/2)\hat{\varphi}(\omega/2)$. Если в последнем соотношении вместо функции $\hat{\varphi}(\omega/2)$ подставить выражение $H(\omega/4)\hat{\varphi}(\omega/4)$, то получим $\hat{\varphi}(\omega) = H_0(\omega/2)\, H_0(\omega/4)\, \hat{\varphi}(\omega/4)$. Повторяя эту процедуру, получаем выражение образа Фурье масштабирующей функции через функцию $H_0(\omega)$, $\hat{\varphi}(\omega) = c\prod_{j=1}^{\infty} H_0(\omega/2^j)$, где константа c равна пределу функции $\hat{\varphi}(\omega)$ в нуле. Полагая, что функция $\hat{\varphi}(\omega)$ непрерывна и $\varphi(x)$ нормирована соотношением $\hat{\varphi}(0) = 1$, имеем:

$$\hat{\varphi}(\omega) = \prod_{j=1}^{\infty} H_0\left(\frac{\omega}{2^j}\right). \tag{15}$$

Полученная формула дает метод для нахождения масштабирующей функции через частотную функцию $H_0(\omega)$.

Теорема 1. *Если для коэффициентов* h_n *в разложении* $H_0(\omega)=\sqrt{2^{-1}}\sum_{n\in\mathbf{Z}} h_n\, e^{-i\,n\omega}$ *выполнено условие* $\sum_{n\in\mathbf{Z}}|h_n|\,|n|<\infty$ *и* $H_0(0)=1$, *тогда бесконечное произведение* (15) *сходится абсолютно и равномерно на компактных подмножествах из* **R**.

Доказательство. Поскольку $H_0(0)=1$, то

$$|H_0(\omega)|=|H_0(\omega)-1+1|\le 1+|H_0(\omega)-1|=1+|H_0(\omega)-H_0(0)|=$$

$$=1+\frac{1}{\sqrt{2}}\left|\sum_{n\in\mathbf{Z}} h_n\,(e^{-i\,n\omega}-1)\right|\le 1+\sqrt{2}\sum_{n\in\mathbf{Z}}|h_n|\,|\sin n\omega/2|=$$

$$=1+\sqrt{2}\sum_{n\in\mathbf{Z}}|h_n|\,|n|\,\frac{|\sin n\omega/2|}{|n\omega/2|}\,|\omega/2|\le 1+C|\omega|\le e^{C|\omega|}.$$

Тогда если $|\omega|\le M$, то получаем равномерную оценку на промежутке $[-M, M]$:

$$H_0\left(\frac{\omega}{2^j}\right)\le e^{CM/2^j}.$$

По признаку Вейерштрасса [АСЧ] бесконечное произведение сходится абсолютно и равномерно на любом конечном промежутке $[-M, M]$:

$$\prod_{j=1}^{\infty}\left|H_0\left(\frac{\omega}{2^j}\right)\right|\le\prod_{j=1}^{\infty}e^{CM/2^j}=\exp\left(\sum_{j=1}^{\infty}CM/2^j\right)=e^{CM}.$$

Таким образом, для сходимости бесконечного произведения (15) от коэффициентов $\{h_n\}$ требуется лишь достаточно быстрое убывание. Это условие заведомо выполнено в случае, когда лишь конечное число коэффициентов $\{h_n\}$ отлично от нуля, то есть когда функция $\varphi(x)$ – с компактным носителем.

Пример 10. Пусть $H_0(\omega)=\cos(\omega)$. Функция $H_0(\omega)$ является 2π-периодической, $H_0(0)=1$ и $\cos(\omega)=\dfrac{e^{-i\omega}+e^{i\omega}}{2}$. Условия теоремы 1 выполнены. Найдем $\hat{\varphi}(\omega)$ из бесконечного произведения, считая, что $\hat{\varphi}(0)=1$.

$$\hat{\varphi}(\omega)=\prod_{j=1}^{\infty}H_0\left(\frac{\omega}{2^j}\right)=\prod_{j=1}^{\infty}\cos\left(\frac{\omega}{2^j}\right).$$

Лемма 1. $\displaystyle\prod_{j=1}^{\infty}\cos\frac{x}{2^j}=\frac{\sin x}{x}.$

Доказательство.

$$\sin x = 2\sin\frac{x}{2}\cos\frac{x}{2}, \qquad \cos\frac{x}{2} = \frac{\sin x}{2\sin\frac{x}{2}}, \qquad \cos\frac{x}{4} = \frac{\sin\frac{x}{2}}{2\sin\frac{x}{4}}, \quad \ldots$$

Тогда легко находится произведение конечного числа сомножителей и предел при $N \to \infty$:

$$\prod_{j=1}^{N}\cos\frac{x}{2^j} = \frac{\sin x}{2^N \sin\frac{x}{2^N}}, \quad \lim_{N\to\infty} 2^N \sin\frac{x}{2^N} = \lim_{N\to\infty}\frac{\sin\frac{x}{2^N}}{\frac{1}{2^N}} = x.$$

$$\prod_{j=1}^{\infty}\cos\frac{x}{2^j} = \lim_{N\to\infty}\prod_{j=1}^{N}\cos\frac{x}{2^j} = \lim_{N\to\infty}\frac{\sin x}{2^N\sin\frac{x}{2^N}} = \frac{\sin x}{x}.$$

Возвращаясь к примеру 10, по лемме 1 получаем:

$$\hat{\varphi}(\omega) = \prod_{j=1}^{\infty}\cos\frac{\omega}{2^j} = \frac{\sin\omega}{\omega},$$

и тогда

$$\varphi(x) = F^{-1}\left[\frac{\sin\omega}{\omega}\right] = \begin{cases} 1/2, & x\in[-1,1) \\ 0, & x\notin[-1,1) \end{cases}.$$

Мы получили масштабирующую функцию $\varphi(x)$ примера 2, масштабирующее соотношение имеет вид: $\varphi(x) = \varphi(2x - 1) + \varphi(2x + 1)$. Отметим, что масштабирующее соотношение в частотной области $\hat{\varphi}(\omega) = H_0(\omega/2)\hat{\varphi}(\omega/2)$ для функции $\hat{\varphi}(\omega) = \dfrac{\sin\omega}{\omega}$ принимает вид следующего очевидного равенства:

$$\frac{\sin\omega}{\omega} = \cos\omega/2\,\frac{\sin\omega/2}{\omega/2}.$$

Теорема 1 о сходимости бесконечного произведения дает достаточно слабые требования на коэффициенты $\{h_n\}$. Однако не следует, что коэффициенты $\{h_n\}$ можно выбирать произвольно с учетом ограничения (11). Дело в том, что теорема 1 не обеспечивает условия $\varphi(x) \in L^2(\mathbf{R})$. Для этого необходимо дополнительное условие на коэффициенты.

Теорема 2. ([Дб], стр. 245.) *Если* 2π*-периодическая функция* $H_0(\omega)$ *обладает свойством*

$$|H_0(\omega)|^2 + |H_0(\omega+\pi)|^2 = 1 \text{ } п.\,в., \tag{16}$$

а произведение (15) *сходится почти всюду, то его предел* $\hat{\varphi}(\omega)$ *принадлежит пространству* $L^2(\mathbf{R})$ *и* $\|\hat{\varphi}(\omega)\|_{L^2} \le 1$.

Замечание 8. Условие (16) эквивалентно следующему условию на коэффициенты:

$$\sum_n h_n \overline{h}_{n-2k} = \delta_{0k}. \qquad (17)$$

Для того чтобы убедиться в этом, достаточно разложить в ряд Фурье функцию $|H_0(\omega)|^2 + |H_0(\omega + \pi)|^2$ с учетом выражения (9).

Пусть V_0 – замкнутое подпространство $L^2(\mathbf{R})$, порожденное функциями $\varphi_{0,n}(x) = \varphi(x - n)$, $n \in \mathbf{Z}$, и V_1 – замкнутое подпространство $L^2(\mathbf{R})$, порожденное масштабированными сдвигами функции $\varphi(x)$, то есть функциями $\varphi(x) = \sqrt{2}\varphi(2x - n)$. Из масштабирующего соотношения (1) следует, что $\varphi(x) \in V_1$. Следовательно, $V_0 \subset V_1$. Совершенно аналогично для любого фиксированного $j \in \mathbf{Z}$ функции $\varphi_{j,n}(x) = \sqrt{2^j}\varphi(2^j x - n)$ порождают замкнутое подпространство V_j, причем $V_j \subset V_{j+1}$. Получаем последовательность замкнутых вложенных друг в друга подпространств

$$... \subset V_{-1} \subset V_0 \subset V_1 \subset V_2 \subset ...$$

пространства $L^2(\mathbf{R})$, обладающих свойством: $f(x) \in V_j \Leftrightarrow f(2x) \in V_{j+1}$. В связи с этим возникают следующие естественные вопросы:

Когда сдвиги $\varphi_{0,n}(x) = \varphi(x - n)$, $n \in \mathbf{Z}$, функции $\varphi(x)$ порождают безусловный базис пространства V_0?

При каких условиях подпространства $... \subset V_{-1} \subset V_0 \subset V_1 \subset V_2 \subset ...$, расширяясь, будут давать все пространство $L^2(\mathbf{R})$? Будет ли нулевым пересечение этих подпространств?

Что представляет собой дополнительное подпространство к V_0 в пространстве V_1?

Их мы рассмотрим в следующем параграфе.

2.3. Ортогональный кратномасштабный анализ

В первом параграфе мы на основе функции Хаара построили серию вложенных друг в друга подпространств $... \subset V_{-1} \subset V_0 \subset V_1 \subset V_2 \subset ...$ пространства $L^2(\mathbf{R})$, причем в каждом подпространстве V_j существовал ортонормированный базис, порожденный сдвигами одной функции. В данном параграфе мы рассмотрим эти вопросы в общей ситуации.

2.3.1. Ортогональное кратномасштабное разложение

Имея в виду обобщение полученных ранее конструкций и для образования новых разложений пространства $L^2(\mathbf{R})$, дадим следующее определение.

Определение 1. *Ортогональным кратномасштабным разложением пространства* $L^2(\mathbf{R})$ *называется последовательность замкнутых вложенных друг в друга подпространств*

$$\ldots \subset V_{-1} \subset V_0 \subset V_1 \subset V_2 \subset \ldots,$$

обладающая свойствами:

1. *Замыкание их объединения совпадает с* $L^2(\mathbf{R})$, $\overline{\bigcup_{j\in\mathbf{Z}} V_j} = L^2(\mathbf{R})$.
2. *Пересечение всех подпространств – нулевое,* $\bigcap_{j\in\mathbf{Z}} V_j = \{0\}$.
3. $f(x) \in V_j \Leftrightarrow f(2x) \in V_j + 1$.
4. *Существует функция* $\varphi(x) \in V_0$, *такая что функции* $\varphi_{0,n}(x) = \varphi(x - n)$, $n \in \mathbf{Z}$, *образуют ортонормированный базис пространства* V_0.

Кратномасштабное разложение называют часто *ортогональным кратномасштабным анализом* (multiresolution analysis), сокращенно – КМА.

Из свойств 3 и 4 сразу следует, что функции $\varphi_{1,n}(x) = \sqrt{2}\varphi(2x - n)$, $n \in \mathbf{Z}$, образуют ортонормированный базис пространства V_1, а для любого $j \in \mathbf{Z}$ ортонормированный базис пространства V_j образуют функции $\varphi_{j,n}(x) = \sqrt{2^j}\varphi(2^j x - n)$, $n \in \mathbf{Z}$.

Поскольку $\varphi(x) \in V_0 \subset V_1$, то функция $\varphi(x)$ раскладывается по базису $\varphi_{1,n}(x) = \sqrt{2}\varphi(2x - n)$ пространства V_1:

$$\varphi(x) = \sqrt{2}\sum_{n\in\mathbf{Z}} h_n \varphi(2x - n),$$

причем коэффициенты разложения $\{h_n, n \in \mathbf{Z}\}$ удовлетворяют условию $\sum_{n\in\mathbf{Z}} |h_n|^2 < \infty$. Поэтому функция $\varphi(x)$ является масштабирующей.

Обратно предположим, что задана масштабирующая функция $\varphi(x)$. Тогда она обычным образом порождает последовательность вложенных подпространств $\ldots \subset V_{-1} \subset V_0 \subset V_1 \subset V_2 \subset \ldots$, при этом выполняется свойство 3. Относительно условий 1 и 2 кратномасштабного разложения отметим, что в книге [Дб], стр. 203 и 205, доказаны следующие факты. Если сдвиги $\{\varphi(x - n), n \in \mathbf{Z}\}$ масштабирующей функции $\varphi(x) \in L^2(\mathbf{R})$ образуют ортонормированный базис про-

странства V_0, то $\bigcap_{j\in\mathbf{Z}} V_j = \{0\}$. Если сдвиги $\varphi(x-n)$ масштабирующей функции $\varphi(x) \in L^2(\mathbf{R})$ образуют ортонормированный базис пространства V_0 и дополнительно $\hat{\varphi}(\omega)$ ограничена для всех ω и непрерывна в окрестности точки $\omega = 0$, причем $\hat{\varphi}(0) \neq 0$, то $\overline{\bigcup_{j\in\mathbf{Z}} V_j} = L^2(\mathbf{R})$. Поэтому в дальнейшем мы будем относительно масштабирующей функции предполагать следующее.

Масштабирующая функция $\varphi(x)$ *имеет ограниченное преобразование Фурье* $\hat{\varphi}(\omega)$, *непрерывное в окрестности* $\omega = 0$ *и* $\hat{\varphi}(0) = \int_{\mathbf{R}} \varphi(x)dx \neq 0$.

Рассмотрим вопрос о том, когда для данной масштабирующей функции $\varphi(x)$ система $\varphi_{0,n}(x) = \varphi(x-n)$, $n \in \mathbf{Z}$ образует ортонормированный базис пространства V_0. Для этого нужно проверить бесконечное число условий: $(\varphi_{0,k}, \varphi_{0,m}) = \delta_{km}$, $\forall\, k, m \in \mathbf{Z}$. Хотелось бы получить более компактное условие ортонормированности. Напомним масштабирующее уравнение в частотной области:

$$\hat{\varphi}(\omega) = H_0\left(\frac{\omega}{2}\right)\hat{\varphi}\left(\frac{\omega}{2}\right), \tag{1}$$

где

$$H_0(\omega) = \frac{1}{\sqrt{2}}\sum_{n\in\mathbf{Z}} h_n e^{-i\,n\omega} \tag{2}$$

– частотная функция для $\varphi(x)$. Функция $H_0(\omega)$ является 2π-периодической функцией, принадлежащей пространству $L^2[0, 2\pi]$. Мы нормируем масштабирующую функцию так, что $\|\varphi(x)\|^2_{L^2} = \int_{-\infty}^{\infty} |\varphi(x)|^2 dx = 1$.

Теорема 1. *Функции* $\varphi_{0,n}(x) = \varphi(x-n)$ *образуют ортонормированный базис подпространства* $V_0 \subset L^2(\mathbf{R})$ *тогда и только тогда, когда*

$$\sum_{n\in Z} |\hat{\varphi}(\omega + 2\pi n)|^2 = 1 \text{ } п.\,в. \tag{3}$$

Доказательство. Заменой переменной любое скалярное произведение $(\varphi_{0,k}, \varphi_{0,m})$ можно привести к виду $(\varphi, \varphi_{0,n})$. Тогда имеем:

$$(\varphi, \varphi_{0,n}) = \int_R \varphi(x)\overline{\varphi(x-n)}\,dx = \frac{}{\pi}\int_R \hat{\varphi}(\omega)\overline{e^{-in\omega}\hat{\varphi}(\omega)}d\omega =$$

$$= \frac{1}{2\pi}\int_R e^{in\omega}|\,\hat{\varphi}(\omega)\,|^2 d\omega = \frac{1}{2\pi}\left(\cdots + \int_{-2\pi}^{0} + \int_{0}^{2\pi} + \int_{2\pi}^{4\pi} + \cdots\right) =$$

$$= \frac{1}{2\pi}\int_0^{2\pi} e^{in\omega}\sum_{k\in Z}\left|\hat{\varphi}(\omega + 2\pi k)\right|^2 d\omega = \delta_{0n}.$$

Поскольку система функций $\{e^{-in\omega}\}$ образует полную ортогональную систему функций на промежутке $[0, 2\pi]$, то интеграл равен δ_{0n} тогда и только тогда, когда сумма под знаком интеграла равна 1 почти всюду, $\sum_{k\in Z}\left|\hat{\varphi}(\omega+2\pi k)\right|^2 = 1$ *п. в.*

Полученное условие (3) также выглядит достаточно сложно. Возникают естественные вопросы о сходимости ряда и о нахождении суммы.

Сначала отметим, что рассматриваемый ряд $\sum_{k\in Z}\left|\hat{\varphi}(\omega+2\pi k)\right|^2$ для функции $\varphi(x) \in L^2(\mathbf{R})$ сходится почти всюду на промежутке $[0, 2\pi]$ по теореме Б. Леви [КФ]. Действительно, члены ряда неотрицательны, интегралы частичных сумм ограничены в совокупности (интегралом функции $|\hat{\varphi}(\omega)|^2$ по всей числовой прямой). Поэтому ряд сходится почти всюду. Его сумма интегрируема, и интеграл от суммы равен сумме интегралов.

Для вычисления суммы $\sum_{k\in Z}\left|\hat{\varphi}(\omega+2\pi k)\right|^2$ любой функции $\varphi(x) \in L^2(\mathbf{R})$ полезно разложить эту сумму в ряд Фурье. Оказывается, что коэффициенты Фурье находятся достаточно просто.

Лемма 1. *Коэффициенты Фурье суммы* $\sum_{k\in Z}\left|\hat{\varphi}(\omega+2\pi k)\right|^2$ *могут быть вычислены по формуле*

$$c_n = \int_{\mathbf{R}} \varphi(x)\overline{\varphi(x-n)}dx. \tag{4}$$

Доказательство.

$$c_n = \frac{1}{2\pi}\int_0^{2\pi}\sum_{k\in Z}\left|\hat{\varphi}(\omega+2\pi k)\right|^2 e^{in\omega}d\omega = \frac{1}{2\pi}\int_{\mathbf{R}}\left|\hat{\varphi}(\omega)\right|^2 e^{in\omega}d\omega =$$

$$= \frac{1}{2\pi}\int_{\mathbf{R}}\hat{\varphi}(\omega)\overline{\hat{\varphi}(\omega)e^{-in\omega}}d\omega = \int_{\mathbf{R}}\varphi(x)\overline{\varphi(x-n)}dx.$$

Тогда

$$\sum_{k\in Z}\left|\hat{\varphi}(\omega+2\pi k)\right|^2 = \sum_{n\in Z}c_n e^{-in\omega}.$$

Замечание 1. Лемма 1 не облегчает проверку ортонормированности $(\varphi_{0,k}, \varphi_{0,m}) = \delta_{km}$, $\forall\, k, m \in \mathbf{Z}$, поскольку также требует вычисления бесконечного числа коэффициентов Фурье. Однако даже если известно, что система $\{\varphi(x - n)\}$ не является ортогональной, в дальнейшем для неортогонального случая нам будет необходимо иметь для функции $\varphi(x) \in L^2(\mathbf{R})$ выражение суммы $\sum_{k\in Z}\left|\hat{\varphi}(\omega+2\pi k)\right|^2$, которое может быть найдено при помощи леммы 1.

Замечание 2. В случае ортогонального кратномасштабного анализа функции $\varphi_{j,n}(x) = \sqrt{2^j}\varphi(2^j x - n)$, $n \in \mathbf{Z}$ образуют ортонормированный базис пространства V_j для любого j. Коэффициенты масштабирующего фильтра $\{h_n\}$ в ортогональном случае обладают свойством

$$\sum_n h_n \overline{h}_{n-2k} = \delta_{0k}, \tag{5}$$

которое сразу следует из равенства $\varphi_{0,k}(x) = \sum_n h_{n-2k}\varphi_{1,n}(x)$ (предложение 1 из параграфа 2.2) и из того, что $(\varphi_{0,0}, \varphi_{0,k}) = \delta_{0,k}$. В частности,

$$\sum_n |h_n|^2 = 1. \tag{6}$$

Рассмотрим вопрос о построении масштабирующей функции $\varphi(x)$, порождающей ортогональный кратномасштабный анализ. Теорема 1 предыдущего параграфа 2.2 о сходимости бесконечного произведения

$$\hat{\varphi}(\omega) = \prod_{j=1}^{\infty} H_0\left(\frac{\omega}{2^j}\right) \tag{7}$$

дает достаточно слабые требования на коэффициенты $\{h_n\}$. Однако она не обеспечивает условия $\varphi(x) \in L^2(\mathbf{R})$. По теореме 2 из 2.2 для этого достаточно потребовать выполнения $|H_0(\omega)|^2 + |H_0(\omega + \pi)|^2 = 1$ п. в. В ортогональном случае последнее условие всегда выполняется.

Теорема 2. *Если сдвиги* $\varphi_n(x) = \varphi(x - n)$ *масштабирующей функции* $\varphi(x)$ *образуют ортонормированный базис подпространства* V_0, *то частотная функция* $H_0(\omega)$ *обладает следующим свойством:*

$$|H_0(\omega)|^2 + |H_0(\omega + \pi)|^2 = 1 \text{ } п.\ в. \tag{8}$$

Доказательство. Подставим в (3) выражение из масштабирующего соотношения (1) и заменим $\omega/2 = \zeta$. Тогда

$$\sum_{k \in Z} \left|H_0(\zeta + \pi k)\hat{\varphi}(\zeta + \pi k)\right|^2 = 1 \text{ } п.\ в.$$

Последнюю сумму разложим на четные и нечетные слагаемые.

$$\sum_{l \in Z} \left|H_0(\zeta + \pi 2l)\hat{\varphi}(\zeta + \pi 2l)\right|^2 + \left|H_0(\zeta + \pi + \pi 2l)\hat{\varphi}(\zeta + \pi + \pi 2l)\right|^2 = 1.$$

Используя 2π-периодичность функции $H_0(\omega)$ и равенство (2), получаем:

$$\sum_{l \in Z} \left|H_0(\zeta)|^2 |\hat{\varphi}(\zeta + \pi 2l)\right|^2 + \left|H_0(\zeta + \pi)|^2 |\hat{\varphi}(\zeta + \pi + \pi 2l)\right|^2 = 1,$$

$|H_0(\varsigma)|^2 + |H_0(\varsigma + \pi)|^2 = 1$ *п. в.*

Вывод. (*Условия ортогональности.*) Из теоремы 2 раздела 2.2.2 следует, что если частотная функция $H_0(\omega)$ обладает свойством

$$|H_0(\omega)|^2 + |H_0(\omega + \pi)|^2 = 1 \text{ п. в.}, \qquad (*)$$

бесконечное произведение (7) сходится почти всюду и функция $\hat{\varphi}(\omega)$ удовлетворяет условию

$$\sum_{n \in Z} |\hat{\varphi}(\omega + 2\pi n)|^2 = 1 \text{ п. в.}, \qquad (**)$$

то функция $\varphi(x)$ порождает ортогональный кратномасштабный анализ.

Замечание 3. Хотя условие (*) является следствием (**), но оно гораздо легче, чем (**). Поэтому для построения функции $\varphi(x)$ сначала решается более легкая задача нахождения частотной функции $H_0(\omega)$, удовлетворяющей условию (*). Затем по формуле (7) находится функция $\hat{\varphi}(\omega)$. Для ортогональности необходимо еще обеспечить выполнение свойства (**). В случае когда фильтр коэффициентов $\{h_n\}$ имеет конечное число ненулевых коэффициентов, существуют простые достаточные условия для выполнения (**), см. параграф 2.8.

Замечание 4. Условие (*) эквивалентно следующему условию на коэффициенты:

$$\sum_n h_n \overline{h}_{n-2k} = \delta_{0k}.$$

Для того чтобы убедиться в этом, достаточно разложить в ряд Фурье функцию $|H_0(\omega)|^2 + |H_0(\omega + \pi)|^2$ с учетом выражения (2) для $H_0(\omega)$.

2.3.2. Вейвлеты

Предположим, что дано орторгональное кратномасштабное разложение, определенное функцией $\varphi(x)$. Наша цель – построить базис всего пространства $L^2(\mathbf{R})$. Из свойств 3 и 4 определения 1 следует, что для любого $j \in \mathbf{Z}$ функции

$$\varphi_{j,n}(x) = \sqrt{2^j}\,\varphi(2^j x - n), n \in \mathbf{Z}$$

образуют ортонормированный базис пространства V_j. Для каждого $j \in \mathbf{Z}$ мы имеем $V_j \subset V_{j+1}$. При переходе от V_j к V_{j+1} базисы в этих пространствах меняются полностью. Было бы лучше, если бы базис пространства V_{j+1} получался из базиса V_j добавлением новых элементов.

Для того чтобы реализовать эту идею, рассмотрим замкнутое подпространство V_0 в $L^2(\mathbf{R})$, порожденное всевозможными целочисленными сдвигами $\varphi(x-n)$ функции $\varphi(x)$ и подпространство V_1, порожденное полуцелыми сдвигами $\varphi_{1,n}(x) = \sqrt{2}\varphi(2x-n)$ функции $\varphi(2x)$.

Поскольку $V_0 \subset V_1$, то в пространстве V_1 имеется ортогональное дополнение W_0 к пространству V_0:

$$V_1 = V_0 \oplus W_0. \tag{9}$$

Оказывается, что существует функция $\psi(x) \in W_0$, такая что ее целочисленные сдвиги $\{\psi(x-m),\ m \in \mathbf{Z}\}$ образуют базис пространства W_0. Тогда базис пространства V_1 может быть получен из базиса $\{\varphi(x-n),\ n \in \mathbf{Z}\}$ пространства V_0 добавлением элементов $\{\psi(x-m)\}$ базиса пространства W_0:

$$\{\varphi(x-n), \psi(x-m),\ n, m \in \mathbf{Z}\}.$$

Далее, поскольку $V_1 \subset V_2$, то в пространстве V_2 имеется ортогональное дополнение W_1 к пространству V_1, и мы имеем следующее ортогональное разложение:

$$V_2 = V_1 \oplus W_1. \tag{10}$$

По построению пространство V_2 является масштабированной версией пространства V_1, другими словами, $f(x) \in V_1 \Leftrightarrow f(2x) \in V_2$. Легко видеть, что при соответствии $f(x) \to f(2x)$ ортогональность функций сохраняется. Поэтому и пространство W_1 является масштабированной версией пространства W_0.

$$\begin{array}{lll} V_1 = & V_0 \oplus W_0 & \\ \downarrow 2x & \downarrow 2x & \downarrow 2x \\ V_2 = & V_1 \oplus W_1 & \end{array}$$

Это означает, что $f(x) \in W_0 \Leftrightarrow f(2x) \in W_1$. Следовательно, полуцелые сдвиги $\psi_{1,n}(x) = \sqrt{2}\psi(2x-n)$ функции $\psi(2x)$ образуют ортонормированный базис пространства W_1. Из (9) и (10) вытекает следующее ортогональное разложение:

$$V_2 = V_1 \oplus W_1 = V_0 \oplus W_0 \oplus W_1.$$

Поэтому базис пространства V_2 образован функциями

$$\{\phi(x-n), \psi(x-m), \sqrt{2}\psi(2x-k),\ n, m, k \in \mathbf{Z}\}.$$

Он получен из базиса пространства V_0 добавлением новых элементов $\psi_{0,m}(x)$ и $\psi_{1,k}(x)$. Ясно, что процесс можно продолжить. Тогда для

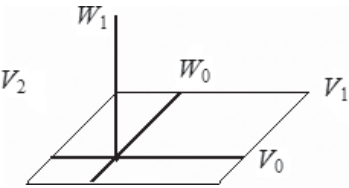


Рис. 2.3.1. Ортогональное разложение пространства V_2

любого $j \in \mathbf{Z}$ получаем $V_j = V_0 \oplus W_0 \oplus W_1 \oplus ... \oplus W_{j-1}$. Учитывая, что пространства V_j, расширяясь, дают все пространство $L^2(\mathbf{R})$, получаем в пределе:

$$L^2(\mathbf{R}) = V_0 \oplus W_0 \oplus W_1 \oplus ... \oplus W_{j-1} \oplus ...$$

Из этого разложения пространства $L^2(\mathbf{R})$ мы получаем, что базис $L^2(\mathbf{R})$ будут образовывать функции

$$\{\varphi(x-n),\ \sqrt{2^j}\psi(2^j x - m), \quad n, m \in \mathbf{Z},\ j = 0, 1, 2, \ldots\}.$$

Очевидна некоторая незавершенность построения базиса. Функции $\varphi(x - n)$ – на особом положении. Однако от них можно освободиться. Действительно, пространство V_0 также может быть разложено в ортогональную сумму:

$$V_0 = V_{-1} \oplus W_{-1}.$$

Поэтому базис $\{\varphi(x - n), n \in \mathbf{Z}\}$ пространства V_0 может быть заменен на такой:

$$\{\sqrt{2^{-1}}\varphi(2^{-1}x - n), \sqrt{2^{-1}}\psi(2^{-1}x - m), \quad n, m \in \mathbf{Z}\}.$$

Далее берем разложение $V_0 = V_{-2} \oplus V_{-2} \oplus W_{-1}$. Продолжая этот процесс до бесконечности при $j \to -\infty$ и учитывая, что пространства V_j уменьшаются до нуля, мы получаем в пределе разложение всего пространства $L^2(\mathbf{R})$:

$$L^2(\mathbf{R}) = ... \oplus W_{-j} \oplus ... \oplus W_{-2} \oplus W_{-1} \oplus W_0 \oplus W_1 \oplus ... \oplus W_j \oplus ...,$$
$$L^2(\mathbf{R}) = \oplus_{j=-\infty}^{\infty} W_j.$$

Тогда базис $L^2(\mathbf{R})$ будут образовывать функции

$$\{\psi_{jn}(x) = \sqrt{2^j}\psi(2^j x - n), \quad j, n \in \mathbf{Z}\}. \tag{11}$$

Определение 2. *Если последовательность подпространств*

$$... \subset V_{-1} \subset V_0 \subset V_1 \subset V_2 \subset ...$$

образует ортогональное кратномасштабное разложение, то для каждого $j \in \mathbf{Z}$ *ортогональное дополнение* W_j *к пространству* V_j *в пространстве* V_{j+1} *называется пространством вейвлетов, а его элементы называются вейвлетами.*

Таким образом, мы построили базис пространства $L^2(\mathbf{R})$ в предположении, что задано ортогональное кратномасштабное разложение пространства $L^2(\mathbf{R})$ с масштабирующей функцией $\varphi(x)$, и в предположении, что в разложении $V_1 = V_0 \oplus W_0$ базис пространства W_0 порождается одной функцией $\psi(x)$.

Теперь покажем, что такая функция $\psi(x)$ действительно существует. Мы уже установили ранее, что функции $\varphi_n(x) = \varphi(x - n)$ образуют ортонормированный базис подпространства $V_0 \subset L^2(\mathbf{R})$ тогда и только тогда, когда

$$\sum_{n \in Z} |\hat{\varphi}(\omega + 2\pi n)|^2 = 1 \text{ п. в.}$$

В этом случае функция $H_0(\omega)$ обладает следующим свойством:

$$|H_0(\omega)|^2 + |H_0(\omega + \pi)|^2 = 1 \text{ п. в.}$$

Из последней формулы следует, что частотная функция $H_0(\omega)$ может служить основой для построения сопряженных квадратурных фильтров разложения и восстановления (см. параграф 1.6 главы 1). Тогда мы можем образовать следующие функции:

- $H(\omega) = \sqrt{2} H_0(\omega)$ – низкочастотный фильтр разложения;
- $G(\omega) = e^{i\omega} \overline{H(\omega + \pi)}$ – высокочастотный фильтр разложения;
- $\tilde{H}(\omega) = \overline{H(\omega)}$ – низкочастотный фильтр восстановления;
- $\tilde{G}(\omega) = e^{-i\omega} H(\omega + \pi)$ – высокочастотный фильтр восстановления.

Из дальнейшего будет видно, что связь вейвлетов с фильтрами имеет глубокую основу. Сначала отметим следующее. Пусть P_j: $L^2(R) \to V_j$ – оператор ортогонального проектирования пространства $L^2(\mathbf{R})$ на подпространство V_j:

$$P_j(f) = \sum_{n \in \mathbf{Z}} (f, \varphi_{j,n}) \varphi_{j,n}.$$

В соответствии с ортогональным разложением $V_{j+1} = V_j \oplus W_j$ имеем:

$$P_{j+1}(f) = P_j(f) + P_j^W(f)$$

– разложение приближения $P_{j+1}(f)$ в пространстве V_{j+1} на более грубую часть $P_{j+1}(f)$ и уточняющее слагаемое $P_j^W(f)$. Ситуация вполне

аналогична разложению сигнала на низкочастотную и высокочастотную составляющие с использованием фильтров $H(\omega)$ и $G(\omega)$.

Функция $H(\omega)$ является передаточной функцией фильтра коэффициентов $\{h_n\}$ масштабирующей функции $\varphi(x)$, $\varphi(x) = \sum_n h_n \varphi_{1,n}(x)$. Естественно предположить, что базис пространства вейвлетов W_0 будут образовывать сдвиги такой функции $\psi(x)$, у которой коэффициенты разложения по базису $\varphi_{1,n}(x)$ образуют фильтр с передаточной функцией $G(\omega) = e^{i\omega}\overline{H(\omega+\pi)}$. Следующая теорема показывает, что это предположение верно.

Теорема 3. *Если последовательность замкнутых вложенных друг в друга подпространств*

$$\ldots \subset V_{-1} \subset V_0 \subset V_1 \subset V_2 \subset \ldots$$

образует ортогональное кратномасштабное разложение пространства $L^2(\mathbf{R})$ *с масштабирующей функцией* $\varphi(x)$ *и частотной функцией* $H_0(\omega)$, *то существует такая функция* $\psi(x)$, *что ее сдвиги* $\psi(x-n)$ *образуют ортонормированный базис пространства* W_0, *дополнительного к* V_0 *при ортогональном разложении* $V_1 = V_0 \oplus W_0$. *В качестве такой функции можно взять*

$$\hat{\psi}(\omega) = e^{i\omega/2}\,\overline{H_0\left(\frac{\omega}{2}+\pi\right)}\,\hat{\varphi}\left(\frac{\omega}{2}\right), \tag{12}$$

или эквивалентно:

$$\psi(x) = \sqrt{2}\sum_n (-1)^{n+1}\,\overline{h_{-n-1}}\,\varphi(2x-n), \tag{13}$$

где $\{h_n\}$ *– масштабирующий фильтр функции* $\varphi(x)$.

Доказательство. Покажем, что сдвиги функции $\psi(x)$, определенной формулой (12), образуют ортонормированный базис пространства W_0.

Часть 1. Эквивалентность формул (12) *и* (13). Поскольку $H_0(\omega) = \sqrt{2^{-1}}\sum_{n\in\mathbf{Z}} h_n e^{-in\omega}$, то

$$\hat{\psi}(\omega) = e^{i\omega/2}\,\overline{H_0\left(\frac{\omega}{2}+\pi\right)}\,\hat{\varphi}\left(\frac{\omega}{2}\right) = e^{i\omega/2}\,\frac{1}{\sqrt{2}}\sum_{n\in\mathbf{Z}}\overline{h_n}\,e^{in(\omega/2+\pi)}\hat{\varphi}(\omega/2) =$$

$$= \frac{1}{\sqrt{2}}\sum_{n\in\mathbf{Z}}\overline{h_n}\,e^{i(n+1)\omega/2}e^{in\pi}\hat{\varphi}(\omega/2) = \frac{1}{\sqrt{2}}\sum_{n\in\mathbf{Z}}(-1)^n\,\overline{h_n}\,e^{i(n+1)\omega/2}\hat{\varphi}(\omega/2) =$$

$$= \frac{1}{\sqrt{2}}\sum_{k\in\mathbf{Z}}(-1)^{k+1}\,\overline{h_{-k-1}}\,e^{-ik\omega/2}\hat{\varphi}(\omega/2).$$

Точно такое же выражение мы получим, если сделаем преобразование Фурье в формуле (13):

$$\hat{\psi}(\omega) = \sqrt{2}\sum_n (-1)^{n+1}\,\overline{h_{-n-1}}\,\frac{1}{2}e^{-i\omega n/2}\hat{\varphi}(\omega/2).$$

Часть 2. Ортогональность системы функций $\psi_{0,n}(x) = \psi(x-n)$. Поскольку $\sum_n |h_n|^2 = 1$, то ряд (13) сходится в пространстве $L^2(\mathbf{R})$, поэтому функции $\psi(x)$ и $\hat{\psi}(\omega)$ интегрируемы с квадратом. Из (13) следует, что функция $\psi(x)$ лежит в пространстве V_1. Рассмотрим ее сдвиги $\psi_{0,n}(x) = \psi(x-n)$, $n \in \mathbf{Z}$. Ортонормированность системы функций $\{\psi_{0,n}(x)\}$ следует из теоремы 1 и свойств ортогональности (*) и (**) функции $\varphi(x)$. Действительно,

$$\sum_n |\hat{\psi}(\omega+2\pi n)|^2 = \sum_{n\in\mathbf{Z}} \left|H_0\left(\frac{\omega}{2}+\pi n+\pi\right)\right|^2 \left|\hat{\varphi}\left(\frac{\omega}{2}+\pi n\right)\right|^2 =$$

[раскладываем на четные и нечетные слагаемые]

[и используем 2π-периодичность функции H, замена $\zeta = \omega/2$]

$$= \sum_k |H_0(\zeta+\pi)|^2 \left|\hat{\varphi}(\zeta+2\pi k)\right|^2 + \sum_k |H_0(\zeta)|^2 \left|\hat{\varphi}(\zeta+2\pi k+\pi)\right|^2 =$$

$$= |H_0(\zeta+\pi)|^2 \sum \left|\hat{\varphi}(\zeta+2\pi k)\right|^2 + |H_0(\zeta)|^2 \sum \left|\hat{\varphi}(\zeta+\pi+2\pi k)\right|^2 =$$

[по свойству (**)]

$$= |H_0(\varsigma+\pi)|^2 + |H_0(\varsigma)|^2 = 1 \text{ } п.\,в.$$

Часть 3. Ортогональность пространств V_0 *и* W_0. Покажем, что каждая функция системы $\psi_{0,n}(x) = \psi(x-n)$, $n \in \mathbf{Z}$ ортогональна пространству V_0. Для этого достаточно проверить ортогональность $\psi_{0,n}(x)$ к образующей функции $\varphi_{0,0}(x) = \varphi(x) \in V_0$,

$$(\varphi_{00}, \psi_{0n}) = \int_{\mathbf{R}} \varphi(x)\,\overline{\psi(x-n)}\,dx = \frac{1}{2\pi}\int_{\mathbf{R}} \hat{\varphi}(\omega)\,\overline{\hat{\psi}(\omega)}\,e^{in\omega}d\omega =$$

[используя (1) и (12)]

$$= \frac{1}{2\pi}\int_{\mathbf{R}} \left|\hat{\varphi}(\omega/2)\right|^2 H_0(\omega/2)H_0(\omega/2+\pi)e^{in\omega}e^{-i\omega/2}d\omega =$$

$$= \frac{1}{\pi}\int_{\mathbf{R}} \left|\hat{\varphi}(\omega)\right|^2 H_0(\omega)H_0(\omega+\pi)\,e^{i(2n-1)\omega}d\omega \;=$$

[воспользуемся 2π-периодичностью H и теоремой 1]

$$= \frac{1}{\pi}\int_0^{2\pi} \sum_n \left|\hat{\varphi}(\omega+2\pi n)\right|^2 H_0(\omega)H_0(\omega+\pi)e^{i(2n-1)\omega}d\omega =$$

$$= \frac{1}{\pi}\int_0^{2\pi} H_0(\omega)H_0(\omega+\pi)e^{i(2n-1)\omega}d\omega =$$

$$= \frac{1}{2\pi}\int_0^{2\pi} \sum_{m\in Z} h_m e^{-im\omega} \sum_{k\in Z} h_k e^{-ik(\omega+\pi)} e^{i(2n-1)\omega} d\omega =$$

$$= \frac{1}{2\pi}\int_0^{2\pi} \sum_{k,m\in Z} (-1)^k h_m h_k e^{-i(m+k-2n+1)\omega} d\omega = \sum_{k\in Z} (-1)^k h_{2n-1-k} h_k = 0\,.$$

Достаточно очевидно, что последняя сумма равна нулю для любого n. Действительно, слагаемые с номерами $k = p$ и $k = 2n - 1 - p$ сокращаются,

$$(-1)^p h_{2n-1-p} h_p + (-1)^{2n-1-p} h_p h_{2n-1-p} = 0.$$

Часть 4. Эквивалентность базисов. Мы получили ортонормированную систему $\{\psi_{0,k}(x) = \psi(x - k)\}$ функций, принадлежащих W_0. В пространстве V_0 также имеется базис $\{\varphi_{0,n}(x) = \varphi(x - n)\}$. Покажем, что ортонормированное семейство функций $\{\varphi_{0,n}(x), \psi_{0,k}(x), n, k \in \mathbf{Z}\}$ образует базис всего пространства V_1. Для этого достаточно показать, что элементы базиса $\{\varphi_{1,n}(x)\}$ пространства V_1 могут быть разложены по базису $\{\varphi_{0,n}(x), \psi_{0,k}(x)\}$.

Пусть $H_0(\omega)$ – частотная функция для масштабирующей функции $\varphi(x)$. Обратимся к следующим функциям:

$$H(\omega) = \sqrt{2}H_0(\omega), \quad G(\omega) = e^{i\omega}\overline{H(\omega+\pi)}, \quad \tilde{H}(\omega) = \overline{H(\omega)}, \quad \tilde{G}(\omega) = e^{-i\omega}H(\omega+\pi).$$

Отметим, что функция $G_0(\omega) = G(\omega)/\sqrt{2}$ участвует в образовании функции $\hat{\psi}(\omega)$ по формуле (12), $\hat{\psi}(\omega) = G_0(\omega/2)\hat{\varphi}(\omega/2)$. Поскольку $H(\omega) = \sum_{k\in\mathbf{Z}} h_k e^{-ik\omega}$, то получаем

$$G(\omega) = \sum_{k\in\mathbf{Z}} g_k e^{-ik\omega} = \sum_{k\in\mathbf{Z}} (-1)^{k+1}\overline{h}_{-k-1} e^{-ik\omega},$$

$$\tilde{H}(\omega) = \sum_{k\in\mathbf{Z}} \tilde{h}_k e^{-ik\omega} = \sum_{k\in\mathbf{Z}} \overline{h_{-k}} e^{-ik\omega},$$

$$\tilde{G}(\omega) = \sum_{k\in\mathbf{Z}} \tilde{g}_k e^{-ik\omega} = \sum_{k\in\mathbf{Z}} (-1)^{k-1} h_{k-1} e^{-ik\omega}.$$

Положим $z = e^{-i\omega}$, тогда $-z = -e^{-i\omega} = e^{-i(\omega+\pi)}$ и $z^{-1} = e^{i\omega} = \bar{z}$. Из условия $|H_0(\omega)|^2 + |H_0(\omega+\pi)|^2 = 1$ вытекает, что $|H(\omega)|^2 + |H(\omega+\pi)|^2 = 2$, и тогда для определенных выше функций $H(z)$, $G(z)$, $\tilde{H}(z)$ и $\tilde{G}(z)$ выполнены следующие равенства (см. также параграф 1.6):

$$\left.\begin{aligned} H(z)\tilde{H}(z) + G(z)\tilde{G}(z) &= 2 \\ H(z)\tilde{H}(-z) + G(z)\tilde{G}(-z) &= 0 \end{aligned}\right\}.$$

Последнюю систему можно переписать в виде:

$$\left.\begin{aligned} H(z)(\tilde{H}(z)+\tilde{H}(-z))+G(z)(\tilde{G}(z)+\tilde{G}(-z))=2\\ H(z)(\tilde{H}(z)-\tilde{H}(-z))+G(z)(\tilde{G}(z)-\tilde{G}(-z))=2\end{aligned}\right\}.$$

В первом равенстве остались удвоенные четные степени $\tilde{H}(z)$ и $\tilde{G}(z)$, во втором – нечетные:

$$\left.\begin{aligned} 2H(z)\sum_k \tilde{h}_{2k}z^{2k}+2G(z)\sum_k \tilde{g}_{2k}z^{2k}=2\\ 2H(z)\sum_k \tilde{h}_{2k-1}z^{2k-1}+2G(z)\sum_k \tilde{g}_{2k-1}z^{2k-1}=2\end{aligned}\right\}.$$

Положим в последних равенствах $z = e^{-i\omega/2}$ и умножим их соответственно на $\hat{\varphi}(\omega/2)$ и $z\hat{\varphi}(\omega/2)$. Тогда

$$\left.\begin{aligned} H(\omega/2)\hat{\varphi}(\omega/2)\sum_k \tilde{h}_{2k}z^{2k}+G(\omega/2)\hat{\varphi}(\omega/2)\sum_k \tilde{g}_{2k}z^{2k}=\hat{\varphi}(\omega/2)\\ H(\omega/2)\hat{\varphi}(\omega/2)\sum_k \tilde{h}_{2k-1}z^{2k}+G(\omega/2)\hat{\varphi}(\omega/2)\sum_k \tilde{g}_{2k-1}z^{2k}=e^{-i\omega/2}\hat{\varphi}(\omega/2)\end{aligned}\right\}.$$

Из масштабирующего соотношения $\hat{\varphi}(\omega) = H_0(\omega/2)\hat{\varphi}(\omega/2)$, определения $\hat{\psi}(\omega) = G_0(\omega/2)\hat{\varphi}(\omega/2)$ функции $\hat{\psi}$ и из соотношений $H(\omega) = \sqrt{2}H_0(\omega)$ и $G(\omega) = \sqrt{2}G_0(\omega)$ получаем:

$$\left.\begin{aligned} \sqrt{2}\sum \tilde{h}_{2k}z^{2k}\hat{\varphi}(\omega)+\sqrt{2}\sum \tilde{g}_{2k}z^{2k}\hat{\psi}(\omega)=\hat{\varphi}(\omega/2)\\ \sqrt{2}\sum \tilde{h}_{2k-1}z^{2k}\hat{\varphi}(\omega)+\sqrt{2}\sum \tilde{g}_{2k-1}z^{2k}\hat{\psi}(\omega)=e^{-i\omega/2}\hat{\varphi}(\omega/2)\end{aligned}\right\}$$

Теперь применяем обратное преобразование Фурье, учитывая, что $z = e^{-i\omega/2}$:

$$\left.\begin{aligned} \sqrt{2}\sum_k \tilde{h}_{2k}\varphi(x-k)+\sqrt{2}\sum_k \tilde{g}_{2k}\psi(x-k)=2\varphi(2x)\\ \sqrt{2}\sum_k \tilde{h}_{2k-1}\varphi(x-k)+\sqrt{2}\sum_k \tilde{g}_{2k-1}\psi(x-k)=2\varphi(2x-1)\end{aligned}\right\}.$$

Последние равенства эквивалентны следующему (для любого n):

$$\sqrt{2}\varphi(2x-n)=\sum_k \tilde{h}_{2k-n}\varphi(x-k)+\sum_k \tilde{g}_{2k-n}\psi(x-k), \qquad (14)$$

которое и дает разложение элементов базиса $\{\varphi_{1,n}(x)\}$ пространства V_1 по базису $\{\varphi_{0,n}(x), \psi_{0,k}(x), n, k \in \mathbf{Z}\}$.

Замечание 4. Коэффициенты $\{h_n\}$ и $\{g_n\}$ фильтров разложения $H(\omega) = \sqrt{2}H_0(\omega)$ и $G(\omega) = e^{-i\omega}\overline{H(\omega+\pi)}$ являются коэффициентами разложения функций $\varphi(x)$ и $\psi(x)$ по базису $\{\varphi_{1,n}(x)\}$ пространства V_1.

Коэффициенты $\tilde{h}_{2k}$ и $\tilde{g}_{2k}$ фильтров восстановления $\tilde{H}(\omega)$ и $\tilde{G}(\omega)$ являются коэффициентами разложения функций $\varphi_{1,n}(x)$ по второму базису $\{\varphi_{0,n}(x), \psi_{0,k}(x), n, k \in \mathbf{Z}\}$ пространства V_1.

Следствие 1. *Для каждого $j \in \mathbf{Z}$ функции*

$$\psi_{j,n}(x) = \sqrt{2^j}\,\psi(2^j x - n), \quad n \in \mathbf{Z} \tag{15}$$

образуют ортонормированный базис пространства W_j.

Замечание 5. Ясно, что порождающая функция $\psi(x)$ пространства W_0 определена неоднозначно. Вместо $\psi(x)$ можно использовать любой ее сдвиг как в области координат $x := x - N$, так и фазовый. Тогда в качестве коэффициентов разложения

$$\psi(x) = \sqrt{2}\sum_n g_n\,\varphi(2x - n)$$

функции $\psi(x)$ по базису пространства V_1 можно взять не только такие, как в теореме 4. Возможны, например, следующие значения:

$$g_n = (-1)^n \bar{h}_{1-n} \quad \text{или} \quad g_n = (-1)^n \bar{h}_{N-n}. \tag{16}$$

Для того чтобы получить первое выражение, достаточно в качестве $G(\omega)$ взять следующую функцию: $G(\omega) = -e^{-i\omega}\,\overline{H(\omega + \pi)}$. Для второго выражения выбираем

$$\hat{\psi}(\omega) = -\,e^{-iN\omega/2}\,\overline{H_0\left(\frac{\omega}{2} + \pi\right)}\,\hat{\varphi}\left(\frac{\omega}{2}\right).$$

2.3.3. О единственности порождающих функций

Рассмотрим вопрос о том, является ли единственной функция $\varphi(x)$, сдвиги которой порождают данное подпространство V_0.

Предположим, что имеются две различные функции $f(x)$ и $g(x)$, сдвиги которых $f_n(x) = f(x - n)$ и $g_n(x) = g(x - n)$, $n \in \mathbf{Z}$, образуют ортонормированные базисы в одном и том же подпространстве $V \subset L^2(\mathbf{R})$. Тогда одну из них можно разложить по ортогональной системе другой функции. Установим (следуя [Дб]) две теоремы, первая из которых утверждает, что преобразование Фурье последовательности коэффициентов разложения является функцией, равной единице по модулю. Вторая теорема утверждает, что если порождающая функция имеет компактный носитель, то она является единственной с точностью до сдвигов и умножения на число, равное единице по модулю.

Теорема 4. *Пусть функции* $f_n(x) = f(x - n)$ *и* $g_n(x) = g(x - n)$, $n \in \mathbf{Z}$, *образуют ортонормированные базисы в одном и том же подпространстве* $V \subset L^2(\mathbf{R})$. *Пусть далее* $g(x) = \sum_n c_n f_n(x)$ *– разложение функции* $g(x)$ *по базису* $\{f_n(x)\}$ *и* $C(\omega) = \sum_n c_n e^{-in\omega}$ *– преобразование Фурье набора коэффициентов разложения. Тогда*

$$|C(\omega)| = 1 \quad \text{и} \quad \hat{g}(\omega) = C(\omega)\hat{f}(\omega).$$

Доказательство. Поскольку $g(x) \in V$ и $\{f_n(x)\}$ – ортонормированный базис, то $g(x) = \sum_n c_n f_n(x) = \sum_n c_n f(x-n)$ и $\sum_n |c_n|^2 = \|g\|^2 = 1$. Применяя преобразование Фурье к равенству $g(x) = \sum_n c_n f(x-n)$, получаем $\hat{g}(\omega) = C(\omega)\hat{f}(\omega)$, где $C(\omega) = \sum_n c_n e^{-in\omega}$ является 2π-периодической. Из ортонормированности систем $\{f_n(x)\}$ и $\{g_n(x)\}$ следует:

$$\sum_{n\in Z} |\hat{f}(\omega + 2\pi n)|^2 = 1, \quad \sum_{n\in Z} |\hat{g}(\omega + 2\pi n)|^2 = 1.$$

Поэтому $|C| = 1$.

Лемма 2. *Если последовательность* $\{c_n, n \in \mathbf{Z}\}$ *имеет конечное число ненулевых членов и если* $|C(\omega)| = 1$, *то* $c_n = c\delta_{nn_0}$ *для некоторого* $n_0 \in \mathbf{Z}$.

Доказательство. Поскольку преобразование Фурье последовательности $\{c_n, n \in \mathbf{Z}\}$ имеет модуль, равный единице, то последовательность $\{c_n\}$ является абсолютно некоррелированной (см.§ 1.4 главы 1). Это означает, что для любого k

$$\sum_n c_n \overline{c_{n+k}} = \delta_{k0}.$$

Пусть p и q – номера крайних ненулевых членов последовательности $\{c_n\}$, то есть $c_n = 0$ при $n < p$ и $c_p \neq 0$, $c_n = 0$ при $n > q$ и $c_q \neq 0$. Тогда $\sum_n c_n \overline{c_{n+q-p}} = \delta_{q-p\,0}$. Нетрудно заметить из определения p и q, что последняя сумма состоит из единственного слагаемого $c_p \overline{c_q}$, неравного нулю по предположению. Поэтому $\delta_{q-p\,0} \neq 0$ и, следовательно, $p = q$. Их общее значение есть искомое число n_0.

Теорема 5. *Если функции* $f(x)$ *и* $g(x)$ *имеют компактные носители и* $f_n(x) = f(x-n)$, $g_n(x) = g(x - n)$, $n \in \mathbf{Z}$, *образуют ортонормированные базисы в одном и том же подпространстве* $V \subset L^2(\mathbf{R})$, *то* $g(x) = c\,f(x - n_0)$, *где* $c \in \mathbf{C}$ *– некоторое число, по модулю равное единице, и* $n_0 \in \mathbf{Z}$.

Доказательство. Поскольку $g(x) = \sum_n c_n f(x-n)$, то из ортонормированности систем $\{f_n(x)\}$ и $\{g_n(x)\}$ следует, что

$$c_n = \int_R g(x)\overline{f(x-n)}\,dx.$$

Носители функций $f(x)$ и $g(x)$ являются компактными, поэтому лишь конечное число интегралов и, следовательно, чисел c_n отлично от нуля. Тогда из леммы следует, что отлично от нуля только одно число $c_{n0} = c$, причем $|c| = 1$. Поэтому $g(x) = c_{n0} f(x - n_0) = cf(x - n_0)$.

Теорема 6 [Дб]. *Предположим, что* φ *и* ψ *– масштабирующая функция и вейвлет в некотором кратномасштабном разложении имеют компактные носители и являются вещественными. Если* ψ *имеет ось симметрии или антисимметрии, то* ψ *является функцией Хаара.*

2.3.4. Неортогональный случай

Мы видели в 2.2.1, что сдвиги $\{c_n, n \in \mathbf{Z}\}$ масштабирующих функций на основе B-сплайнов не образуют ортогональной системы в пространстве V_0. Ослабляя требование ортогональности, естественным компромиссом является соглашение использовать базисы Рисса, которые обладают рядом хороших свойств.

Определение 3. *Базис* $\{u_n, n \in \mathbf{Z}\}$ *в гильбертовом пространстве* H *называется базисом Рисса, если существуют положительные константы* A_1 *и* A_2*, такие что для любой последовательности* $\{c_n, n \in \mathbf{Z}\}$*, для которой* $\sum_n |c_n|^2 < \infty$*, справедливы неравенства*

$$A_1 \sum_n |c_n|^2 \le \left\| \sum_n c_n u_n \right\|^2 \le A_2 \sum_n |c_n|^2. \qquad (17)$$

Напомним, что система элементов $\{u_n, n \in \mathbf{Z}\}$ пространства H называется базисом, если каждый элемент $f \in H$ может быть единственным образом представлен в виде сходящегося ряда $f = \sum_{n\in\mathbf{Z}} c_n u_n$. Базис в пространстве H называется *безусловным*, если он остается базисом при любой перестановке его элементов. В гильбертовом пространстве любой ортогональный базис $\{e_n\}$ является безусловным и любой безусловный базис может быть получен из ортогонального базиса действием линейного непрерывного оператора с непрерывным обратным. Такие безусловные базисы в гильбертовом пространстве и называются базисами Рисса.

Упражнение. Пусть $\varphi(x)$ – характеристическая функция промежутка $[0, 3)$ и V_0 – подпространство, порожденное целочисленными сдвигами $\varphi(x - n)$. Показать, что система $\{\varphi(x - n), n \in \mathbf{Z}\}$ образует линейно независимую систему, не являющуюся базисом Рисса в V_0.

Условием, эквивалентным (17), является следующее: для любого элемента $f \in H$ справедливы неравенства

$$A_1\|f\|^2 \le \sum_n |(f,u_n)|^2 \le A_2\|f\|^2. \qquad (18)$$

Замечание 6. Близким является понятие *каркаса* (*фрейма*). Каркас – это семейство векторов $\{u_n,\ n \in \mathbf{Z}\}$ гильбертова пространства, которое однозначно характеризует любой элемент $f \in H$ по его скалярным произведениям $\{(f, u_n)\}_{n\in\mathbf{Z}}$. Более формально последовательность $\{u_n,\ n \in \mathbf{Z}\}$ в гильбертовом пространстве H называется каркасом (фреймом), если существуют положительные константы A_1 и A_2, такие что для любого элемента $f \in H$ справедливо неравенство (18). Отличие каркаса от базиса Рисса заключается только в том, что для каркаса не требуется линейная независимость векторов $\{u_n\}$. Каркасы являются избыточными наборами векторов. В случае линейной независимости выполняется свойство $A_1 \le 1 \le A_2$. Каркас является избыточным, если $A_1 > 1$. Каркас называется жестким, если $A_1 = A_2$. Каркас является ортонормированным базисом тогда и только тогда, когда $A_1 = A_2 = 1$. Подробнее об этом см. в [Дб], [М] и [Бл].

Определение 4. *Кратномасштабным разложением пространства* $L^2(\mathbf{R})$ *называется последовательность замкнутых вложенных друг в друга подпространств*

$$\ldots \subset V_{-1} \subset V_0 \subset V_1 \subset V_2 \subset \ldots,$$

обладающая свойствами:

1. *Замыкание их объединения совпадает с* $L^2(\mathbf{R})$, $\overline{\bigcup_{j\in\mathbf{Z}} V_j} = L^2(\mathbf{R})$.
2. *Пересечение всех подпространств – нулевое,* $\bigcap_{j\in\mathbf{Z}} V_j = \{0\}$.
3. $f(x) \in V_j \ \Leftrightarrow\ f(2x) \in V_{j+1}$.
4. *Существует функция* $\varphi(x) \in V_0$, *называемая масштабирующей, такая что ее сдвиги* $\varphi_{0,n}(x) = \varphi(x - n)$, $n \in \mathbf{Z}$, *образуют базис Рисса пространства* V_0.

Теорема 7. *Система* $\{\varphi(x - n),\ n \in \mathbf{Z}\}$, *полученная сдвигами некоторой функции* $\varphi(x) \in L^2(\mathbf{R})$, *образует базис Рисса пространства* V_0 *тогда и только тогда, когда существуют положительные константы* A_1 *и* A_2, *такие что*

$$A_1 \le \sum_{k\in Z} \left|\hat{\varphi}(\omega + 2\pi k)\right|^2 \le A_2 \ \text{п. в.} \qquad (19)$$

Доказательство. Пусть $\{c_n,\ n \in \mathbf{Z}\}$ – любая последовательность, для которой $\sum_n |c_n|^2 < \infty$. Для того чтобы показать эквивалентность условий (19) и (17), преобразуем среднее выражение в неравенстве (17) для $u_n = \varphi(x - n)$, используя преобразование Фурье:

$$\left\| \sum_n c_n \varphi(x-n) \right\|^2 = \frac{1}{2\pi} \left\| \sum_n c_n F[\varphi(x-n)] \right\|^2 = \frac{1}{2\pi} \left\| \sum_n c_n e^{-in\omega} \hat{\varphi}(\omega) \right\|^2 =$$

$$= \frac{1}{2\pi} \int_{-\infty}^{\infty} \left| \sum_n c_n e^{-in\omega} \hat{\varphi}(\omega) \right|^2 d\omega = \frac{1}{2\pi} \int_{-\infty}^{\infty} \left| \sum_n c_n e^{-in\omega} \right|^2 |\hat{\varphi}(\omega)|^2 \, d\omega =$$

$$= \frac{1}{2\pi} \left(\cdots + \int_{-2\pi}^{0} + \int_{0}^{2\pi} + \int_{2\pi}^{4\pi} + \cdots \right) =$$

$$= \frac{1}{2\pi} \int_0^{2\pi} \left| \sum_n c_n e^{-in\omega} \right|^2 \sum_k \left| \hat{\varphi}(\omega + 2\pi k) \right|^2 d\omega = \frac{1}{2\pi} \int_0^{2\pi} |C(\omega)|^2 B^2(\omega) d\omega,$$

где $C(\omega) = \sum_n c_n e^{-in\omega}$ и $B^2(\omega) = \sum_{k \in Z} \left| \hat{\varphi}(\omega + 2\pi k) \right|^2$. Тогда, используя равенство Парсеваля для 2π-периодической функции $C(\omega)$, условие (17) можно переписать в виде:

$$A_1 \frac{1}{2\pi} \| C(\omega) \|^2 = A_1 \sum_n |c_n|^2 \le \frac{1}{2\pi} \int_0^{2\pi} |C(\omega)|^2 B^2(\omega) \, d\omega \le$$

$$\le A_2 \sum_n |c_n|^2 = A_2 \frac{1}{2\pi} \| C(\omega) \|^2,$$

$$A_1 \|C(\omega)\|^2 \le \int_0^{2\pi} |C(\omega)|^2 B^2(\omega) d\omega \le A_2 \|C(\omega)\|^2.$$

Последнее двойное неравенство может выполняться для любых $C(\omega)$ только тогда, когда

$A_1 \le B^2(\omega) \le A_2$ *п. в.*

Поскольку подпространства V_j являются масштабированными версиями пространства V_0, то для любого $j \in \mathbf{Z}$ функции $\varphi_{j,n}(x) = \sqrt{2^j} \varphi(2^j x - n)$ образуют базис Рисса в V_j тогда и только тогда, когда выполнено условие (19) теоремы 7.

Приведем еще две теоремы, доказанные в книге [Дб], стр. 203 и 205.

Теорема 8. *Если сдвиги* $\{\varphi(x-n), n \in \mathbf{Z}\}$ *масштабирующей функции* $\varphi(x) \in L^2(\mathbf{R})$ *образуют базис Рисса пространства* V_0 (*то есть выполнено условие* (19)), то $\bigcap_{j \in \mathbf{Z}} V_j = \{0\}$.

Теорема 9. *Пусть масштабирующая функция* $\varphi(x) \in L^2(\mathbf{R})$ *удовлетворяет условию* (19) *и пусть дополнительно* $\hat{\varphi}(\omega)$ *ограничена для всех* ω *и непрерывна в окрестности* $\omega = 0$, *причем* $\hat{\varphi}(0) \ne 0$. *Тогда* $\overline{\bigcup_{j \in \mathbf{Z}} V_j} = L^2(\mathbf{R})$.

Пример 1. Рассмотрим масштабирующую функцию «шляпа». Это B-сплайн $\varphi(x)$ степени 1:

$$\varphi(x) = \begin{cases} 1-|x|, & x \in [-1,1], \\ 0 & \text{при остальных } x \end{cases}.$$

Функция имеет компактный носитель $[-1, 1]$ и удовлетворяет масштабирующему уравнению

$$\varphi(x) = {}^1\!/_2\varphi(2x+1) + \varphi(2x) + {}^1\!/_2\varphi(2x-1).$$

Для того чтобы выяснить, образуют ли сдвиги $\{\varphi(x-n), n \in \mathbf{Z}\}$ базис Рисса пространства V_0, найдем выражение $\sum_{k\in Z}\left|\hat{\varphi}(\omega+2\pi k)\right|^2$ разложением в ряд Фурье. Находим коэффициенты Фурье по лемме 1, $c_n = \int_{\mathbf{R}} \varphi(x)\overline{\varphi(x-n)}dx$.

$$c_0 = \int_{-\infty}^{\infty} \varphi^2(x)\,dx = 2\int_0^1 (1-x)^2\,dx = \frac{2}{3},$$

$$c_{-1} = c_1 = \int_{-\infty}^{\infty} \varphi(x)\varphi(x-1)\,dx = \int_0^1 x(1-x)\,dx = \frac{1}{6}.$$

Остальные коэффициенты нулевые. Поэтому

$$\sum_{k\in Z}\left|\hat{\varphi}(\omega+2\pi k)\right|^2 = \frac{1}{6}e^{i\omega} + \frac{2}{3} + \frac{1}{6}e^{-i\omega} = \frac{1}{3}(\cos\omega + 2).$$

Тогда имеем:

$$\frac{1}{3} \le \sum_{k\in Z}\left|\hat{\varphi}(\omega+2\pi k)\right|^2 \le 1.$$

Поэтому сдвиги $\varphi(x-n)$ образуют базис Рисса пространства V_0. Кроме того, $\varphi(x) \in L^1(\mathbf{R})$ и $\hat{\varphi}(0) = \int_{\mathbf{R}} \varphi(x)dx = 1$. По теоремам 8 и 9 выполнены все свойства кратномасштабного разложения. Пространства V_j состоят из кусочно-линейных функций с узлами в точках $n/2^j$.

Совершенно аналогично показывается, что B-сплайн $\varphi(x)$ любого порядка определяет неортогональное кратномасштабное разложение пространства $L^2(\mathbf{R})$.

Пример 2. Масштабирующая функция, для которой условие (19) не выполнено. Пусть

$$\varphi(x) = \begin{cases} 1/3, & x \in [0,3), \\ 0 & \text{при остальных } x \end{cases}.$$

Легко видеть, что масштабирующее соотношение выполнено:

$\varphi(x) = \varphi(2x) + \varphi(2x - 3)$.

Фильтр состоит из двух ненулевых чисел $h_0 = 1/\sqrt{2}, h_3 = 1/\sqrt{2}$. Преобразование Фурье легко вычисляется:

$$\hat{\varphi}(\omega) = e^{-i3\omega/2}\frac{\sin 3\omega/2}{3\omega/2}, \quad \sum_{n\in Z}|\hat{\varphi}(\omega+2\pi n)|^2 = \frac{1}{3}+\frac{4}{9}\cos\omega+\frac{2}{9}\cos 2\omega.$$

Последняя сумма находится прямым разложением ее в ряд Фурье по формуле (3) из леммы 1. Мы видим, что сумма является тригонометрическим полиномом, который обращается в нуль в точке $\omega = 2\pi/3$ и поэтому не удовлетворяет условию

$$0 < A_1 \le \sum_{k\in Z}\left|\hat{\varphi}(\omega+2\pi k)\right|^2 \le A_2 \text{ п. в.}$$

Хотя приведенная выше функция $\varphi(x)$ напоминает функцию Хаара, но, в отличие от последней, ее сдвиги $\varphi_{0,n}(x) = \varphi(x - n)$, $n \in \mathbf{Z}$, не образуют базис Рисса пространства V_0. Отметим также, что в данном случае функция $H_0(\omega) = (1 + e^{-i3\omega})/2 = e^{-i3\omega/2}\cos 3\omega/2$ обращается в нуль в точках $\pm\pi/3$. В то же время соотношение $|H_0(\omega)|^2 + |H_0(\omega + \pi)|^2 = 1$ выполнено и функция $\hat{\varphi}(\omega)$ очевидным образом восстанавливается в виде бесконечного произведения $\hat{\varphi}(\omega) = \prod_{j=1}^{\infty} H_0(\omega/2^j)$.

Замечание 7. Хорошо известно [Чуи], [Дб], что если функция $\varphi(x)$ является B-сплайном, то система ее целочисленных сдвигов $\varphi_{0,n}(x) = \varphi(x - n)$, $n \in \mathbf{Z}$, образует базис Рисса пространства V_0. В работе [Ми] найдены точные границы Рисса A_1 и A_2 в неравенствах (17), (19) для B-сплайна $\varphi(x)$ степени n:

$$A_1 = \frac{2^{2n+2}}{\pi^{2n+2}}\sum_{k=0}^{\infty}\frac{1}{(1+2k)^{2n+2}} \quad \text{и} \quad A_2 = 1.$$

2.3.5. Достаточные условия ортогональности

Как мы уже отмечали, если функция $H_0(\omega)$ найдена и удовлетворяет требованию $|H_0(\omega)|^2 + |H_0(\omega + \pi)|^2 = 1$, то функция $\varphi(x)$ не обязана порождать ортогональный кратномасштабный анализ. Дело в том, что не всегда выполняется свойство ортонормированности $\sum_{n\in Z}|\hat{\varphi}(\omega+2\pi n)|^2 = 1$. Классическим примером является частотная функция

$$H_0(\omega) = \frac{1+e^{-i3\omega}}{2} = e^{-i3\omega/2}\cos 3\omega/2,$$

определяющая масштабирующую функцию $\varphi(x) = {}^1\!/_3\chi_{[0,3)}(x)$ (пример в конце 2.3.4). Для такой функции свойство $|H_0(\omega)|^2 + |H_0(\omega + \pi)|^2 = 1$ выполнено, однако

$$\sum_{n\in Z} |\hat{\varphi}(\omega + 2\pi n)|^2 = \frac{1}{3} + \frac{4}{9}\cos\omega + \frac{2}{9}\cos 2\omega \neq 1.$$

Найдем, следуя [Дб, стр. 262], дополнительные условия на коэффициенты масштабирующего фильтра $\{h_n\} = \{h_0, h_1, \ldots, h_L\}$, при выполнении которых система функций $\varphi_{0,n}(x) = \varphi(x - n)$ будет ортонормированной в пространстве V_0.

Пусть $H_0(\omega) = \sqrt{2^{-1}}\sum_{n=0}^{L} h_n e^{-in\omega}$, тогда функция $\varphi(x)$ будет иметь носитель на промежутке $[0, L]$. Для ортогональности системы $\varphi_{0,n}(x) = \varphi(x - n)$ необходимо и достаточно, чтобы $\int_{\mathbf{R}} \varphi(x)\overline{\varphi(x-l)}dx = \delta_{0,l}$. Поскольку носитель функции $\varphi(x)$ лежит на промежутке $[0, L]$, то интегралы $\int_{\mathbf{R}} \varphi(x)\overline{\varphi(x-l)}dx$ будут нулевыми при $|l| \geq L$. Поэтому можно образовать следующий вектор возможных ненулевых скалярных произведений

$$\alpha_l = \int_{\mathbf{R}} \varphi(x)\overline{\varphi(x-l)}dx, \quad -L+1 \leq l \leq L-1.$$

Сделаем следующие преобразования:

$$\alpha_l = \int_{\mathbf{R}} \varphi(x)\overline{\varphi(x-l)}dx = 2\int_{\mathbf{R}} \sum_{n,m=0}^{L} h_n\overline{h_m}\varphi(2x-n)\overline{\varphi(2x-2l-m)}dx =$$

$$= \sum_{n,m=0}^{L} h_n\overline{h_m}\int_{\mathbf{R}} \varphi(y)\overline{\varphi(y-2l-m+n)}dy = \sum_{n,m=0}^{L} h_n\overline{h_m}\alpha_{2l+m-n} =$$

$$= \sum_{k=-L+1}^{L-1}\left(\sum_{n=0}^{N} h_n\overline{h}_{k-2l+n}\right)\alpha_k .$$

Мы видим, что вектор α удовлетворяет уравнению

$$\alpha_l = \sum_{k=-L+1}^{L-1}\left(\sum_{n=0}^{N} h_n\overline{h}_{k-2l+n}\right)\alpha_k.$$

Следовательно, если мы определим матрицу A порядка $2L - 1$ по формуле

$$A_{kl} = \sum_{n=0}^{L} h_n\overline{h}_{k-2l+n}, \quad -L+1 \leq k,\ l \leq L-1,$$

где $h_m = 0$, если $m < 0$ или $m > L$, то $A\alpha = \alpha$, то есть α является собственным вектором матрицы A для собственного значения 1. За-

метим, что 1 всегда является собственным значением A, если $|H_0(\omega)|^2 + H_0(\omega + \pi)|^2 = 1$. Действительно, определим вектор β вида $\beta = (0, ..., 0, 1, 0, ..., 0)$ с единицей в середине, тогда $\beta_l = \delta_{l,0}$. Легко видеть, что он является собственным для A:

$$(A\beta)_l = \sum_k A_{kl}\,\delta_{k,0} = A_{0l} = \sum_n h_n \overline{h}_{n-2l} = \delta_{0l} = \beta_l.$$

Напомним, что условие $|H_0(\omega)|^2 + |H_0(\omega + \pi)|^2 = 1$ эквивалентно следующему условию на коэффициенты $\sum_n h_n \overline{h}_{n-2l} = \delta_{0l}$. Если собственное значение 1 является простым, то вектор α должен быть пропорциональным β, $\alpha = c\beta$. Тогда $\int_{\mathbf{R}} \varphi(x)\overline{\varphi(x-l)}dx = c\delta_{l,0}$. После нормировки $\int_{\mathbf{R}} |\varphi(x)|^2 dx = 1$ получаем ортонормированную систему функций $\varphi_{0,n}(x) = \varphi(x - n)$, $n \in \mathbf{Z}$. Мы получили следующий результат.

Теорема 10 (Лоутон [La]). *Предположим, что частотная функция является тригонометрическим полиномом вида* $H_0(\omega) = \sqrt{2^{-1}} \sum_{n=0}^{L} h_n e^{-in\omega}$, *удовлетворяющим условиям* $|H_0(\omega)|^2 + |H_0(\omega + \pi)|^2 = 1$ *и* $H_0(0) = 1$, *а функция* $\varphi(x)$ *определена при помощи соотношения* $\hat{\varphi}(\omega) = \prod_{j=1}^{\infty} H_0(\omega / 2^j)$. *Если собственное значение* 1 *матрицы* $A_{kl} = \sum_{n=0}^{L} h_n \overline{h_{k-2l+n}}$ *порядка* $2L - 1$ *не является кратным, то функции* $\varphi_{0,n}(x) = \varphi(x - n)$ *образуют ортонормированную систему.*

Замечание. Подробнее о необходимых и достаточных условиях ортогональности см. в книге И. Добеши [Дб, стр. 253–268]. Приведем еще одно простое достаточное условие Коэна А.

Теорема 11 (Коэн [Co]). *Предположим, что частотная функция является тригонометрическим полиномом, удовлетворяющим условиям* $|H_0(\omega)|^2 + |H_0(\omega + \pi)|^2 = 1$ *и* $H_0(0) = 1$, *а функция* $\varphi(x)$ *определена формулой* $\hat{\varphi}(\omega) = \prod_{j=1}^{\infty} H_0(\omega / 2^j)$. *Если функция* $H_0(\omega)$ *не имеет нулей на промежутке* $[-\pi/3, \pi/3]$, *то функции* $\varphi_{0,n}(x) = \varphi(x - n)$ *образуют ортонормированную систему.*

2.4. Вейвлет-преобразование

В этом параграфе мы рассмотрим вейвлет-разложение сигнала $f(x) \in L^2(\mathbf{R})$, покажем, что при вычислении коэффициентов вейвлет-разложения можно обойтись без интегрирования (быстрое вейвлет-преобразование), установим формулы восстановления сигнала после его вейвлет-разложения и обсудим пакетное вейвлет-разложение.

2.4.1. Вейвлет-разложение

Предположим, что мы имеем дело с ортонормированными вейвлетами. Тогда у нас есть две функции: масштабирующая функция $\varphi(x)$ и вейвлет $\psi(x)$. Для каждого $j \in \mathbf{Z}$ система функций $\varphi_{j,n}(x) = \sqrt{2^j}\varphi(2^j x - n)$, $n \in \mathbf{Z}$ является ортонормированным базисом пространства $V_j \subset L^2(\mathbf{R})$. Система подпространств V_j, $j \in \mathbf{Z}$, образует бесконечную в обе стороны последовательность вложенных подпространств

$$... \subset V_{-1} \subset V_0 \subset V_1 \subset V_2 \subset ... \subset V_j \subset ...,$$

причем такую, что $\bigcap_{j=-\infty}^{\infty} V_j = \{0\}$, и $\overline{\bigcup_{j=-\infty}^{\infty} V_j} = L^2(\mathbf{R})$. Каждое подпространство V_{j+1} раскладывается в прямую ортогональную сумму $V_{j+1} = V_j \oplus W_j$, где пространство W_j называется пространством вейвлетов и имеет базис, состоящий из функций $\psi_{j,n}(x) = \sqrt{2^j}\psi(2^j x - n)$, $n \in \mathbf{Z}$. Поскольку $L^2(\mathbf{R}) = \oplus_{j=-\infty}^{\infty} W_j$, то базис пространства $L^2(\mathbf{R})$ будут образовывать функции:

$$\{\psi_{jn}(x) = \sqrt{2^j}\psi(2^j x - n), \quad j, n \in \mathbf{Z}\}. \tag{1}$$

При анализе Фурье находится разложение функции $f(x) \in L^2(0, 2\pi)$ по тригонометрическому базису. Затем анализируются коэффициенты Фурье c_n и, наконец, производится восстановление функции $f(x)$ по (может быть) измененным коэффициентам c_n. Коэффициенты разложения функции в ряд Фурье имеют достаточно ясный качественный смысл. Они показывают, насколько $f(x)$ содержит в себе эталонные колебания $\cos(nx)$ и $\sin(nx)$.

Предположим, что у нас имеется функция $f(x) \in L^2(\mathbf{R})$ и мы хотим провести вейвлет-анализ этого сигнала $f(x)$. Если следовать той же схеме, что и при анализе Фурье, то мы должны разложить функцию $f(x)$ по вейвлет-базису (1) пространства $L^2(\mathbf{R})$:

$$f(x) = \sum_{j,n \in \mathbf{Z}} d_{j,n} \psi_{j,n}(x),$$

где

$$d_{j,n} = (f, \psi_{j,n}) = \int_{\mathbf{R}} f(x)\overline{\psi_{j,n}(x)}dx.$$

Затем нужно проанализировать коэффициенты $\{d_{j,n}\}$. Здесь возникают проблемы, поскольку качественный смысл каждого коэффициента $d_{j,n}$ неясен. Поэтому при вейвлет-анализе используется другой подход, состоящий из следующих двух основных этапов.

Этап 1. Начальное приближение. Сначала мы берем приближение функции $f(x)$ в некотором, достаточно большом пространстве V_j. Это можно сделать, спроектировав функцию $f(x) \in L^2(\mathbf{R})$ на пространство V_j:

$$P_j : L^2(\mathbf{R}) \to V_j, \quad f(x) \to P_j(f) \in V_j.$$

Операторы проектирования P_j дают приближения $P_j(f)$ элементов $f(x) \in L^2(\mathbf{R})$, все более точные с возрастанием j. Поскольку V_j имеет ортонормированный базис $\{\varphi_{j,n}(x),\ n \in \mathbf{Z}\}$, то оператор проектирования P_j имеет вид:

$$P_j(f) = \sum_{n \in \mathbf{Z}} (f, \varphi_{j,n}) \varphi_{j,n}(x) = \sum_{n \in \mathbf{Z}} a_{j,n} \varphi_{j,n}(x). \tag{2}$$

Обозначим символом cA_0 набор *коэффициентов разложения* начального приближения $P_j(f)$:

$$cA_0 = \{a_{j,n}\}, \quad a_{j,n} = (f, \varphi_{j,n}) = \int_{\mathbf{R}} f(x) \overline{\varphi_{j,n}(x)}\, dx. \tag{3}$$

Коэффициенты $cA_0 = \{a_{j,n}\}$ дают информацию относительно функции $f(x) \in L^2(\mathbf{R})$, однако качественный смысл каждого коэффициента неясен, поскольку вейвлеты $\varphi_{j,n}(x)$ и $\psi_{j,n}(x)$ не являются эталонными колебаниями. Это затрудняет анализ и обработку массива коэффициентов $\{a_{j,n}\}$. Однако здесь имеются другие возможности. А именно будем рассматривать коэффициенты $cA_0 = \{a_{j,n}\}$ в сравнении с коэффициентами разложения при более низком разрешении $j - 1$. Для того чтобы не потерять информацию при переходе на более низкий уровень, нужно учесть эти потери.

Этап 2. Вейвлет-разложение. Подпространство V_j раскладываем в прямую сумму ортогональных подпространств

$$V_j = V_{j-1} \oplus W_{j-1}.$$

Это позволяет j-е приближение $P_j(f)$ разложить в сумму

$$P_j(f) = P_{j-1}(f) + P^W_{j-1}(f). \tag{4}$$

Первое слагаемое есть более грубое приближение в пространстве V_{j-1}, а второе – отражает детали, которыми $P_j(f)$ отличается от $P_{j-1}(f)$. В базисах пространств V_{j-1} и W_{j-1} имеем:

$$P_j(f) = \sum_{k \in Z} a_{j-1,k} \varphi_{j-1,k}(x) + \sum_{k \in Z} d_{j-1,k} \psi_{j-1,k}(x), \tag{5}$$

где $a_{j-1,k}$ – *коэффициенты аппроксимации* $(j - 1)$-го уровня разрешения:

$$a_{j-1,k} = (f, \varphi_{j-1,k}) = \int_{\mathbf{R}} f(x)\overline{\varphi_{j-1,k}(x)}\,dx$$

и $d_{j-1,k}$ – *детализирующие коэффициенты*:

$$d_{j-1,k} = (f, \psi_{j-1,k}) = \int_{\mathbf{R}} f(x)\overline{\psi_{j-1,k}(x)}\,dx.$$

Обозначим полученные наборы коэффициентов символами cA_1 и cD_1:

$$cA_1 = \{a_{j-1,k}\},\ cD_1 = \{d_{j-1,k}\}. \tag{6}$$

Таким образом, мы получили теперь два набора коэффициентов разложения. По построению коэффициенты cA_1 и cD_1 в совокупности дают ту же информацию, что и первоначальный набор коэффициентов cA_0. Преимущество коэффициентов cA_1 и cD_1 заключается в том, что они несут еще и качественную информацию. А именно коэффициенты cA_1 описывают более грубое (сглаженное) приближение функции $f(x)$ в пространстве V_{j-1}, а коэффициенты cD_1 характеризуют детали, которыми начальное приближение $P_j(f)$ отличается от $P_{j-1}(f)$. Если при разложении в ряд Фурье каждый коэффициент давал некоторую информацию о сигнале, то теперь качественную информацию несут группы коэффициентов.

Второе преимущество такого разложения заключается в том, что массив cA_1, представляющий сглаженную функцию $P_{j-1}(f) \in V_{j-1}$, можно еще раз разложить на более грубую часть $P_{j-2}(f)$ и детализирующую $P^W_{j-2}(f)$ в соответствии с разложением:

$$V_j = V_{j-1} \oplus W_{j-1} = V_{j-2} \oplus W_{j-2} \oplus W_{j-1},$$

$$P_j(f) = P_{j-2}(f) + P^W_{j-2}(f) + P^W_{j-1}(f),$$

$$P_j(f) = \sum_{k\in Z} a_{j-2,k}\varphi_{j-2,k}(x) + \sum_{k\in Z} d_{j-2,k}\psi_{j-2,k}(x) + \sum_{k\in Z} d_{j-1,k}\psi_{j-1,k}(x).$$

Мы получаем:

- $cA_2 = \{a_{j-2,k}\}$ – коэффициенты аппроксимации глубины разложения 2;
- $cD_2 = \{d_{j-2,k}\}$ – детализирующие коэффициенты глубины разложения 2;
- $cD_1 = \{d_{j-1,k}\}$ – детализирующие коэффициенты глубины разложения 1 – они остались с первого шага разложения.

Повторяя процедуру N раз, мы получаем вейвлет-разложение $P_j(f)$ в виде серии коэффициентов

$$P_j(f) = \{cA_N, cD_N, cD_{N-1}, ..., cD_1\}, \tag{7}$$

где $cA_N = \{a_{j-N,k}\}$ – коэффициенты аппроксимации разложения глубины N и $cD_m = \{d_{j-m,k}\}$ – детализирующие коэффициенты разложения глубины $m = 1, 2, ..., N$. Это соответствует следующему разложению пространства V_j:

$$V_j = V_{j-N} \oplus W_{j-N} \oplus W_{j-N+1} \oplus ... \oplus W_{j-2} \oplus W_{j-1}.$$

Графически вейвлет-разложение можно изобразить в виде следующей диаграммы.

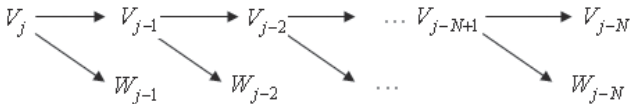


Рис. 2.4.1. Вейвлет-разложение пространства V_j

Или на уровне вейвлет-коэффициентов:

$$P_j(f) = cA_0 \to \{cA_1, cD_1\} \to \{cA_2, cD_2, cD_1\} \to ... \to$$
$$\to \{cA_N, cD_N, cD_{N-1}, ..., cD_1\}.$$

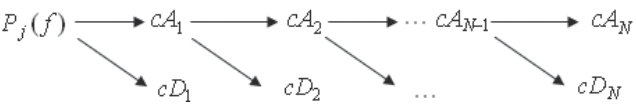


Рис. 2.4.2. Вейвлет-разложение сигнала

Вейвлет-преобразованием функции $f(x)$ называется нахождение коэффициентов вейвлет-разложения:

$\{cA_N, cD_N, cD_{N-1}, ..., cD_1\}$.

Вейвлет-анализ состоит из следующих шагов:

1. Выбираем уровень разрешения j, такой что аппроксимация $P_j(f)$ достаточно точно отражает функцию $f(x)$.
2. Выбираем глубину разложения N и находим коэффициенты разложения:

   $\{cA_N, cD_N, cD_{N-1}, ..., cD_1\}$.
3. Анализируем эти коэффициенты и изменяем их в случае необходимости.
4. Восстанавливаем функцию $f(x)$, используя (возможно, измененные) коэффициенты разложения:

$$f(x) \approx \sum_{k \in Z} a_{j-N,k} \varphi_{j-N,k}(x) + \sum_{k \in Z} d_{j-N,k} \psi_{j-N,k}(x) + \cdots + \sum_{k \in Z} d_{j-1,k} \psi_{j-1,k}(x). \quad (8)$$

2.4.2. Быстрое вейвлет-преобразование

Для вейвлет-преобразования функции $f(x)$ необходимо вычислить серию коэффициентов $\{cA_N, cD_N, cD_{N-1}, ..., cD_1\}$. Каждый коэффициент находится интегрированием:

$$a_{j-N,k} = (f, \varphi_{j-N,k}) = \int_R f(x)\overline{\varphi_{j-N,k}(x)}\,dx, \tag{9}$$

$$d_{j-m,k} = (f, \psi_{j-m,k}) = \int_R f(x)\overline{\psi_{j-m,k}(x)}\,dx,\ m = 1, 2, ..., N. \tag{10}$$

Возникает проблема вычисления большого количества интегралов с необходимой точностью. Следует также учитывать, что при высоком уровне разрешения j носители функций $\varphi_{j,k}(x)$ и $\psi_{j,k}(x)$ становятся малыми порядка $1/2^j$, что затрудняет вычисление интегралов с необходимой точностью.

Быстрое вейвлет-преобразование, предложенное Малла [Ма], позволяет решить эту проблему. Алгоритм Малла дает возможность вычислять коэффициенты вейвлет-разложения без интегрирования, с использованием алгебраических операций на основе свертки.

Предположим, что мы знаем разложение функции $f(x)$ для некоторого уровня разрешения j, то есть знаем коэффициенты $cA_0 = \{a_{j,n}\}$. Проблема заключается в том, чтобы сделать первый шаг разложения

$$P_j(f) = \sum_{k\in Z} a_{j-1,k}\varphi_{j-1,k}(x) + \sum_{k\in Z} d_{j-1,k}\psi_{j-1,k}(x),$$

без интегрирования. Напомним масштабирующее соотношение для $\varphi(x)$ и выражение вейвлета $\psi(x)$:

$$\varphi(x) = \sum_{n\in\mathbf{Z}} h_n\varphi_{1,n}(x), \quad \psi(x) = \sum_n g_n\,\varphi_{1,n}(x),$$

где $\{h_n\}$ – масштабирующий фильтр функции $\varphi(x)$ и $g_n = (-1)^{n+1}\overline{h_{-n-1}}$ – фильтр вейвлета $\psi(x)$. Из этих соотношений сразу вытекают следующие (см. предложение 1 раздела 2.2.1):

$$\varphi_{j-1,k}(x) = \sum_n h_n\varphi_{j,n+2k}(x), \quad \psi_{j-1,k}(x) = \sum_n g_n\varphi_{j,n+2k}(x).$$

Последние равенства выражают базисные функции $\varphi_{j-1,k}(x)$ и $\psi_{j-1,k}(x)$ через базисные функции $\varphi_{j,k}(x)$ пространства V_j. Тогда:

$$a_{j-1,k} = (f, \varphi_{j-1,k}) = \left(f, \sum_n h_n\varphi_{j,n+2k}\right) = \sum_n \overline{h_n}\,a_{j,n+2k},$$

$$d_{j-1,k} = (f, \psi_{j-1,k}) = \left(f, \sum_n g_n \varphi_{j,n+2k} \right) = \sum_n \overline{g_n}\, a_{j,n+2k}.$$

Получили формулы

$$a_{j-1,k} = \sum_n \overline{h_n}\, a_{j,n+2k}, \quad d_{j-1,k} = \sum_n \overline{g_n}\, a_{j,n+2k}, \tag{11}$$

которые выражают коэффициенты первого уровня разложения через начальные коэффициенты при помощи алгебраических операций умножения и суммирования. Обратите внимание на индекс $2k$ в сумме справа, он говорит о проведенной децимации.

Сделаем в данных суммах замену n на $-n$ и учтем, что $\overline{h}_{-n} = h_n^*$ и $\overline{g}_{-n} = g_n^*$ – это коэффициенты сопряженных фильтров $\overline{H}(\omega)$ и $\overline{G}(\omega)$. Тогда коэффициенты разложения $cA_1 = \{a_{j-1,n}\}$ и $cD_1 = \{d_{j-1,n}\}$ получаются в виде свертки

$$a_{j-1,k} = \sum_n h_n^* a_{j,2k-n}, \quad d_{j-1,k} = \sum_n g_n^* a_{j,2k-n} \tag{12}$$

массива $cA_0 = \{a_{j,n}\}$ первоначальных коэффициентов с сопряженными фильтрами $h_n^* = \overline{h}_{-n}$, $g_n^* = \overline{g}_{-n}$ и последующей децимации (выбор коэффициентов с четными номерами $2k$). Схематически это выглядит так:

$$a_{j,k} \;\to\; \tilde{a}_{j-1,s} = \sum_n h_n^* a_{j,s-n} \;\xrightarrow{\downarrow 2}\; a_{j-1,k} = \tilde{a}_{j-1,2k},$$

$$d_{j,k} \;\to\; \tilde{d}_{j-1,s} = \sum_n h_n^* a_{j,s-n} \;\xrightarrow{\downarrow 2}\; d_{j-1,k} = \tilde{d}_{j-1,2k}.$$

Или еще более схематично:

$$cA_0 \;\to\; h^* * cA_0 \;\xrightarrow{\downarrow 2}\; cA_1,$$

$$cA_0 \;\to\; g^* * cA_0 \;\xrightarrow{\downarrow 2}\; cD_1.$$

Эти равенства (11), (12) обеспечивают быстрые алгоритмы вычисления вейвлет-коэффициентов (каскадные алгоритмы, алгоритмы Малла). Термин «быстрые» означает не только то, что в (12) используются более быстрые алгебраические процедуры, но и то, что при каждом преобразовании общее число новых коэффициентов не увеличивается в два раза, а остается прежним за счет децимации.

Ясно, что процедуру быстрого вейвлет-преобразования (12) можно повторить для набора коэффициентов $cA_1 = \{a_{j-1,n}\}$ и таким образом найти все коэффициенты $\{cA_N, cD_N, cD_{N-1}, ..., cD_1\}$ вейвлет-разложения глубины N.

2.4.3. Вопрос о начальных коэффициентах

Остается проблема начальных коэффициентов $cA_0 = \{a_{j,n}\}$. Их прямое вычисление в виде интегралов $a_{j,n} = \int_R s(x)\overline{\varphi_{j,n}(x)}dx$, во-первых, представляет трудоемкую операцию, а во-вторых, может не обеспечить требуемой точности, когда j – достаточно большое (мелкий масштаб). Есть несколько способов решения этой проблемы.

Вариант 1. Функция $f(x)$ практически задается как массив значений $y_n = f(x_n)$ в точках $x = n\Delta x$ достаточно мелкой сетки. Тогда самый простой способ решения проблемы – взять в качестве начальных коэффициентов cA_0 массив значений функции $f(x)$, то есть $a_{j,n} = y_n$. Такой подход используется в MatLab Wavelets Toolbox. Отметим также, что в этом случае нет необходимости использовать формулу $P_j(f) = \sum_{n\in Z} a_{j,n}\varphi_{j,n}(x)$ для восстановления начального приближения $P_j(f)$ по коэффициентам $\{a_{j,n}\}$.

Вариант 2. Если функция $f(x)$ задана как массив значений $y_n = f(x_n)$ в точках $x = x_n$, то можно считать $f(x)$ кусочно-постоянной, $f(x) = y_n$, $x \in [x_n, x_{n+1})$. Изменяя масштаб, можно считать, что $x_n = n$ и $f(n) = y_n$, $x \in [n, n+1)$. Вместо функций $\varphi_{j,k}(x)$ теперь следует взять $\varphi_{0,k}(x)$. Тогда если масштабирующая функция $\varphi(x)$ имеет носитель на промежутке $[0, N]$, то (с точностью до масштабного коэффициента) имеем:

$$a_{j,k} = (f, \varphi_{0,k}) = \int_R f(x)\overline{\varphi(x-k)}dx = \int_0^N f(x+k)\overline{\varphi(x)}dx =$$

$$= \int_0^1 f(x+k)\overline{\varphi(x)}dx + \int_1^2 f(x+k)\overline{\varphi(x)}dx + \cdots + \int_{N-1}^N f(x+k)\overline{\varphi(x)}dx =$$

$$= y_k\int_0^1 \overline{\varphi(x)}dx + y_{k+1}\int_1^2 \overline{\varphi(x)}dx + \cdots + y_{k+N-1}\int_{N-1}^N \overline{\varphi(x)}dx =$$

$$= y_k c_0 + y_{k+1}c_1 + \cdots + y_{k+N-1}c_{N-1}.$$

Поэтому начальные коэффициенты могут быть найдены усреднением по формуле

$$a_{j,k} = \sum_{s=0}^{N-1} y_{k+s}c_s, \qquad (13)$$

где числа c_s можно считать известными.

Вариант 3. Пусть снова (после перемасштабирования) функция $f(x)$ задана своими значениями $y_n = f(n)$. Тогда коэффициен-

ты разложения $a_k = (f, \varphi_{0,k})$ можно вычислить при помощи свертки следующим образом. Мы считаем, что $f = \sum_k (f, \varphi_{0,k})\varphi_{0,k}$. Поэтому $f(n) = \sum_k (f, \varphi_{0,k})\varphi(n-k)$. Тогда

$$\sum_n f(n)e^{-in\omega} = \sum_k (f, \varphi_{0,k})e^{-ik\omega}\sum_m \varphi(m)e^{-im\omega}, \ \ m = n-k,$$

то есть коэффициенты $a_k = (f, \varphi_{0,k})$ являются коэффициентами Фурье функции

$$\sum_n f(n)e^{-in\omega}\left(\sum_m \varphi(m)e^{-im\omega}\right)^{-1} = \sum_k (f, \varphi_{0,k})e^{-ik\omega}.$$

Разложим в ряд Фурье второй сомножитель

$$c_s = \frac{1}{2\pi}\int_0^{2\pi} e^{is\omega}\left(\sum_m \varphi(m)e^{-im\omega}\right)^{-1} d\omega.$$

Тогда

$$\sum_k (f, \varphi_{0,k})e^{-ik\omega} = \sum_n f(n)e^{-in\omega}\sum_s c_s e^{-is\omega} = \sum_k\sum_n f(n)c_{k-n}e^{-ik\omega}.$$

Получаем формулу для вычисления начальных коэффициентов разложения

$$a_k = (f, \varphi_{0,k}) = \sum_n f(n)c_{k-n}.$$

Вариант 4. В разделе 2.9 будет показано, что для койфлетов $\varphi(x)$ и $\psi(x)$ коэффициенты $a_{j,k}$ начального приближения можно получить по формуле $a_{jk} = \sqrt{2^{-j}}f(2^{-j}k)$.

2.4.4. Восстановление

Рассмотрим вопрос о восстановлении начального приближения $P_j(f)$ по набору коэффициентов $\{cA_N, cD_N, cD_{N-1}, ..., cD_1\}$ вейвлет-разложения. Достаточно описать восстановление cA_0 по коэффициентам $\{cA_1, cD_1\}$. Коэффициенты $cA_0 = \{a_{j,n}\}$ есть коэффициенты разложения

$$P_j(f) = \sum_{n\in\mathbf{Z}} a_{j,n}\varphi_{j,n}(x) \qquad (14)$$

по базису $\varphi_{j,n}(x) = \sqrt{2^j}\varphi(2^j x - n)$ пространства V_j. Коэффициенты $cA_1 = \{a_{j-1,n}\}$ и $cD_1 = \{d_{j-1,n}\}$ есть коэффициенты разложения

$$P_j(f) = \sum_k a_{j-1,k}\varphi_{j-1,k}(x) + \sum_k d_{j-1,k}\psi_{j-1,k}(x) \qquad (15)$$

по базису $\varphi_{j-1,k}(x)$ и $\psi_{j-1,k}(x)$ пространства V_j, соответствующего разложению $V_j = V_{j-1} \oplus W_{j-1}$. Поэтому для нахождения коэффициентов $cA_0 = \{a_{j,n}\}$ достаточно от разложения (15) перейти к разложению вида (14). Это легко сделать с использованием следующих формул (см. предложение 1 в разделе 2.2.1):

$$\varphi_{j-1,k}(x) = \sum_n h_{n-2k}\,\varphi_{j,n}(x), \quad \psi_{j-1,k}(x) = \sum_n g_{n-2k}\,\varphi_{j,n}(x).$$

А именно:

$$P_j(f) = \sum_{k\in Z} a_{j-1,k}\varphi_{j-1,k}(x) + \sum_{k\in Z} d_{j-1,k}\psi_{j-1,k}(x) =$$

$$= \sum_k a_{j-1,k} \sum_n h_{n-2k}\,\varphi_{j,n}(x) + \sum_k d_{j-1,k} \sum_n g_{n-2k}\,\varphi_{j,n}(x) =$$

$$= \sum_n \left(\sum_k (a_{j-1,k} h_{n-2k} + d_{j-1,k} g_{n-2k}) \right) \varphi_{j,n}(x).$$

Поэтому

$$a_{j,n} = \sum_k (h_{n-2k}\, a_{j-1,k} + g_{n-2k}\, d_{j-1,k}). \tag{16}$$

Формула (16) дает нам требуемое *восстановление* коэффициентов $cA_0 = \{a_{j,n}\}$ по коэффициентам $\{cA_1, cD_1\}$.

Последнюю формулу можно записать в виде свертки, сделав обратную децимацию массивов $cA_1 = \{a_{j-1,n}\}$ и $cD_1 = \{d_{j-1,n}\}$:

$$\tilde{a}_{j-1,m} = \begin{cases} a_{j-1,k}, \text{ если } m = 2k, \\ 0, \text{ если } m = 2k+1 \end{cases}, \quad \tilde{d}_{j-1,m} = \begin{cases} d_{j-1,k}, \text{ если } m = 2k, \\ 0, \text{ если } m = 2k+1 \end{cases}.$$

Тогда формула (16) принимает вид свертки:

$$a_{j,n} = \sum_{m\in Z} (h_{n-m}\tilde{a}_{j-1,m} + g_{n-m}\tilde{d}_{j-1,m}). \tag{17}$$

Схематически это выглядит так:

$$\left.\begin{array}{lllll} a_{j-1,k} & \xrightarrow{\uparrow 2} & \tilde{a}_{j-1,m} & \to & \sum_m h_{n-m}\,\tilde{a}_{j-1,m} \\ d_{j-1,k} & \xrightarrow{\uparrow 2} & \tilde{d}_{j-1,m} & \to & \sum_m g_{n-m}\,\tilde{d}_{j-1,m} \end{array}\right\} \xrightarrow{+} \quad a_{j,k}.$$

Или еще более схематично:

$$\left.\begin{array}{lll} cA_1 & \xrightarrow{\uparrow 2} & c\tilde{A}_1 \to c\tilde{A}_1 * h \\ cD_1 & \xrightarrow{\uparrow 2} & c\tilde{D}_1 \to c\tilde{D}_1 * g \end{array}\right\} \xrightarrow{+} \quad cA_0.$$

Вывод. Вейвлет-разложение производится сопряженными фильтрами $h_n^* = \overline{h}_{-n}$ и $g_n^* = \overline{g}_{-n}$ вейвлетов $\varphi(x)$ и $\psi(x)$ с последующей децимацией, а восстановление производится фильтрами $\{h_n\}$ и $\{g_n\}$ вейвлетов $\varphi(x)$ и $\psi(x)$ с предварительной обратной децимацией.

Напомним, что в случае ортогонального кратномасштабного разложения высокочастотный фильтр $\{g_n\}$ вейвлета $\psi(x)$ выражается через низкочастотный фильтр $\{h_n\}$ масштабирующей функции $\varphi(x)$ по формуле $g_n = (-1)^n \overline{h}_{1-n}$.

2.4.5. Вейвлет-пакеты

Для того чтобы понять идею пакетного вейвлет-разложения, рассмотрим, например, вейвлет-разложение пространства V_2:

$$V_2 = V_1 \oplus W_1 = V_0 \oplus W_0 \oplus W_1.$$

Два первых пространства, V_0 и W_0, имеют одинаковый «размер» – они порождены целочисленными сдвигами функций $\varphi(x)$, $\psi(x)$, то есть функциями $\varphi(x-n)$, $\psi(x-k)$, а третье пространство – «в два раза больше», оно порождено полуцелыми сдвигами функции $\psi(2x)$, то есть функциями $\psi(2x-k)$. Желая исправить эту несправедливость, рассмотрим вопрос: можно ли разложить и пространство W_1 так, чтобы все пространства в разложении V_2 имели бы одинаковый «размер»?

Для того чтобы выяснить, как можно разложить пространство вейвлетов W_1, рассмотрим более подробно процедуру вейвлет-разложения

$$V_1 = V_0 \oplus W_0.$$

Пространство V_1 имеет ортонормированный базис, состоящий из функций $\sqrt{2}\varphi(2x-n)$. Пространство V_0 имеет ортонормированный базис, состоящий из функций $\varphi(x-m)$, и пространство W_0 имеет ортонормированный базис, состоящий из функций $\psi(x-k)$. Поэтому пространство $V_1 = V_0 \oplus W_0$ имеет еще один ортонормированный базис, состоящий из функций $\{\varphi(x-m), \psi(x-k), m, k \in \mathbf{Z}\}$.

Из основных соотношений

$$\varphi(x) = \sqrt{2}\sum_{n\in\mathbf{Z}} h_n \varphi(2x-n), \quad \psi(x) = \sqrt{2}\sum_{n\in\mathbf{Z}} g_n \varphi(2x-n) \tag{18}$$

легко получаются формулы перехода от одного базиса к другому:

$$\varphi(x-m) = \sqrt{2}\sum_{n\in\mathbf{Z}} h_n \varphi(2x-2m-n) = \sqrt{2}\sum_{p\in\mathbf{Z}} h_{p-2m} \varphi(2x-p),$$

$$\psi(x-k) = \sqrt{2}\sum_{n\in\mathbf{Z}} g_n \varphi(2x-2k-n) = \sqrt{2}\sum_{p\in\mathbf{Z}} g_{q-2k} \varphi(2x-p).$$

Матрица перехода A имеет блочный вид с элементами:

$$A = (A_{pm}(\varphi), A_{pk}(\psi)) = (h_{p-2m}, g_{p-2k}),\ p, m, k \in \mathbf{Z}.$$

Очевидно, что такое разложение указанной матрицей A можно применить к любому пространству E, имеющему ортонормированный базис из сдвигов $\{\sqrt{2}\, f(2x - n),\ n \in \mathbf{Z}$, вообще говоря, любой (не обязательно масштабирующей) функции $f(x) \in L^2(\mathbf{R})$. Формулы

$$f_1(x-m) = \sum_{p\in\mathbf{Z}} h_{p-2m} f(2x-p), \quad f_2(x-k) = \sum_{p\in\mathbf{Z}} g_{p-2k} f(2x-p) \tag{19}$$

определяют новый ортонормированный базис пространства E, состоящий из элементов $\{f_1(x - m), f_2(x - k),\ m, k \in \mathbf{Z}\}$. Первая группа элементов $f_1(x - m)$ порождает некоторое пространство E_1, а вторая группа элементов $f_2(x - m)$ порождает пространство E_2. Мы получаем ортогональное разложение $E = E_1 \oplus E_2$ на два подпространства, в каждом из которых имеется ортонормированный базис, состоящий из целочисленных сдвигов функций.

Таким образом, процедуру разложения (19) можно применить к пространству W_1, имеющему базис $\{\sqrt{2}\psi(2x - n),\ n \in \mathbf{Z}\}$. Тогда у нас появляются два новых пространства

$$W_1 = W_{10} \oplus W_{11},$$

каждое из которых порождено целочисленными сдвигами, соответственно, функций:

$$w_2(x) = \sqrt{2}\sum_{n\in\mathbf{Z}} h_n \psi(2x-n), \quad w_3(x) = \sqrt{2}\sum_{n\in\mathbf{Z}} g_n \psi(2x-n). \tag{20}$$

Получили разложение

$$V_2 = V_0 \oplus W_0 \oplus W_{10} \oplus W_{11}$$

на ортогональные пространства «одного размера».

Рассмотрим теперь пространство V_3. Применяя вейвлет-разложение, получаем

$$V_3 = V_0 \oplus W_0 \oplus W_1 \oplus W_2,$$

где пространство W_2 является масштабированной версией пространства W_1, то есть $f(x) \in W_1 \Leftrightarrow f(2x) \in W_2$. Выше мы установили, что пространство $W_1 = W_{10} \oplus W_{11}$ имеет ортогональный базис, порожденный сдвигами $w_2(x - m)$, $w_3(x - k)$ функций $w_2(x)$, $w_3(x)$ соответственно. Поскольку W_2 есть масштабированная версия пространства $W_1 = W_{10} \oplus W_{11}$, то имеет место ортогональное разложение

$W_2 = W_{20} \oplus W_{21}$,

где W_{20} и W_{21} – масштабированные версии пространств W_{10} и W_{11} соответственно. В пространствах W_{20} и W_{21} ортонормированные базисы образованы полуцелыми сдвигами $w_2(2x - m)$ и $w_3(2x - k)$ функций $w_2(2x)$ и $w_3(2x)$ соответственно.

Каждое из полученных пространств W_{20} и W_{21} разложим, в соответствии с формулами (19), на два ортогональных подпространства

$W_{20} = W_{200} \oplus W_{201}$ и $W_{21} = W_{210} \oplus W_{211}$.

Каждое из этих новых подпространств порождается, соответственно, целочисленными сдвигами функций

$$w_4(x) = \sqrt{2}\sum_{n\in\mathbf{Z}} h_n w_2(2x-n), \quad w_5(x) = \sqrt{2}\sum_{n\in\mathbf{Z}} g_n w_2(2x-n)$$

и

$$w_6(x) = \sqrt{2}\sum_{n\in\mathbf{Z}} h_n w_3(2x-n), \quad w_7(x) = \sqrt{2}\sum_{n\in\mathbf{Z}} g_n w_3(2x-n).$$

В итоге получаем разложение

$V_3 = V_0 \oplus W_0 \oplus W_{10} \oplus W_{11} \oplus W_{200} \oplus W_{201} \oplus W_{210} \oplus W_{211}$

на подпространства, каждое из которых имеет ортогональный базис, порожденный целочисленными сдвигами, каждое – своей функции. Соответствующий базис пространства V_3 называется вейвлет-пакетным.

Продолжая эту процедуру, мы получаем разложение пространства $L^2(\mathbf{R})$ в прямую сумму ортогональных замкнутых подпространств

$L^2(\mathbf{R}) = \Omega_0 + \Omega_1 + \Omega_2 + ...$,

где $\Omega_0 = V_0$ и $\Omega_1 = W_0$ и каждое следующее пространство Ω_n порождено целочисленными сдвигами одной функции $w_n(x)$. При этом функции $w_n(x)$ определяются рекурсивно. Полагаем $w_0(x) = \varphi(x)$ и $w_1(x) = \psi(x)$ и определим *вейвлет-пакетные* функции для любого натурального n следующим образом:

$$w_{2n}(x) = \sqrt{2}\sum_{k\in\mathbf{Z}} h_k w_n(2x-k), \quad w_{2n+1}(x) = \sqrt{2}\sum_{n\in\mathbf{Z}} g_n w_n(2x-n). \qquad (21)$$

В качестве примера рассмотрим вейвлет-пакеты, порожденные вейвлетами Хаара. Фильтры вейвлетов Хаара имеют коэффициенты $h_0 = h_1 = 1/\sqrt{2}$ и $g_0 = -g_1 = 1/\sqrt{2}$. Поэтому

$w_{2n}(x) = w_n(2x) + w_n(2x - 1)$

и

$$w_{2n+1}(x) = w_n(2x) - w_n(2x - 1),$$

где $w_0(x) = \varphi(x)$ – характеристическая функция единичного интервала и $w_1(x) = \psi(x)$ – вейвлет Хаара. Полученные функции – это так называемые функции Уолша (Walsh). Они все ортогональны друг к другу и даже к их собственным сдвигам. Приведем несколько графиков функций вейвлет-пакета Хаара (рис. 2.4.3, 2.4.4).

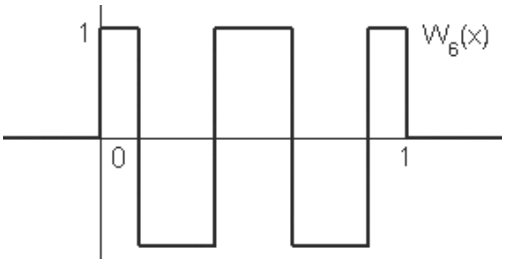


Рис. 2.4.3. График функции $w_6(x)$

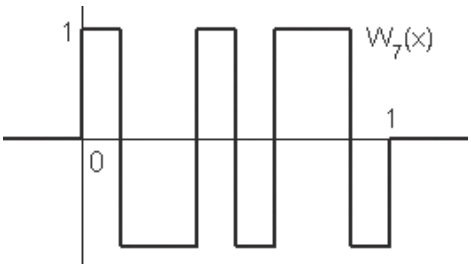


Рис. 2.4.4. График функции $w_7(x)$

Разложение функции $f(x)$ по базису $w_n(x - k)$ называется *вейвлет-пакетным разложением*. Использование пары низкочастотных и высокочастотных фильтров для разложения пространства соответствует разделению сигнала на низкочастотную и высокочастотную компоненты. При вейвлет-преобразовании мы оставляем высокочастотную часть в покое и продолжаем раскладывать низкочастотную часть. В пакетном вейвлет-преобразовании мы раскладываем также и высокочастотную часть. Таким образом, пакетное вейвлет-преобразование способствует лучшей частотной локализации сигналов.

Вейвлет-пакетное преобразование данных не создает никаких дополнительных трудностей. Оно делается точно так же, как и вейвлет-преобразование, – сверткой с фильтрами h_n и g_n или их сопряженными. Схематически вейвлет-пакетное разложение сигнала можно изобразить в виде следующего дерева, рис. 2.4.5.

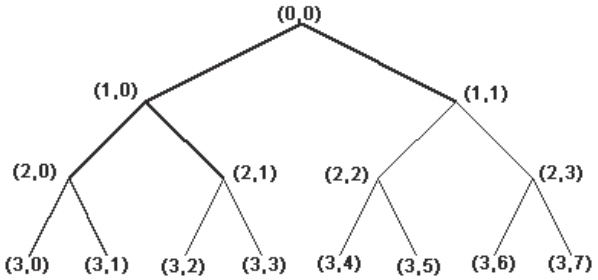


Рис. 2.4.5. Схема пакетного разложения

На вершине этого дерева – исходный сигнал, а ниже – его пакетные вейвлет-коэффициенты. Ветви влево указывают на аппроксимирующие коэффициенты, а правые ветви идут к детализирующим коэффициентам предыдущего узла. Обычное вейвлет-разложение – это левая часть пакетного дерева. Вейвлет-коэффициентов достаточно для восстановления сигнала, и мы можем раскладывать высокочастотные коэффициенты деталей или отказаться от этого. Таким образом, у нас появляется гораздо больше возможностей выбора базиса для разложения – от «минимального» вейвлет-разложения до полного пакетного разложения на всех уровнях. Так что есть много вариантов, чтобы разложить сигнал. Из числа всех представлений мы должны выбрать то, которое представляет сигнал наиболее эффективно. Под «эффективным» мы подразумеваем такое, когда сигнал представлен небольшим количеством коэффициентов разложения, то есть базис для разложения должен быть таким, чтобы было мало больших коэффициентов разложения и много коэффициентов, близких к нулю. Тогда сигнал представлен меньшим числом коэффициентов. Обычно в качестве критерия для выбора самого эффективного или лучшего базиса для данного сигнала используется критерий минимальности энтропии. Энтропия характеризует усредненность, «размазанность» сигнала. Энтропия должна обладать свойством аддитивности по отношению к сигналам. Существует несколько вариантов определения энтропии.

Рассмотрим в качестве примера энтропию Шеннона. Пусть s – сигнал и s_i – его значения. Можно считать, что сигнал нормирован $\sum_i s_i^2 = 1$. Энтропия Шеннона определяется формулой

$$E1(s) = -\sum_i s_i^2 \log_2(s_i^2)$$

при следующем соглашении: 0*log(0) = 0. Использование суммы обеспечивает свойство аддитивности по отношению к объединению массивов, задающих сигналы.

Лучший базис – это тот базис, который дает наименьшую энтропию. Интуитивно энтропия, определенная выше, дает критерий того, сколько эффективных компонент необходимо, чтобы представить сигнал в определенном базисе. Например, если в некотором базисе при разложении получаем все нулевые коэффициенты, кроме одного (то есть сигнал совпадает с одной из базисных функций), тогда энтропия достигает своего минимального значения – нуля. С другой стороны, если в некотором базисе коэффициенты разложения все одинаковы, скажем $s_i = 1/\sqrt{N}$, где N является длиной сигнала, энтропия в этом случае принимает максимальное значение, равное $\log_2 N$. Любое другое разложение находится между этими двумя крайними случаями. Вообще, чем меньше энтропия, тем меньше существенных коэффициентов нужно для представления сигнала.

Критерий энтропии используется для выбора оптимального базиса следующим образом. Если при разложении коэффициентов некоторого узла сумма энтропий, полученных при разложении компонент, меньше, чем энтропия коэффициентов в исходном узле, то разложение применяется, в противном случае коэффициенты остаются без изменения.

2.5. Примеры кратномасштабного анализа и вейвлетов

В этом параграфе мы приведем два примера кратномасштабного анализа и вейвлетов. Они будут получены из масштабирующего уравнения в частотной области.

2.5.1. Вейвлеты Шеннона–Котельникова

В системе вейвлетов Хаара масштабирующая функция $\varphi(x)$ является ступенчатой, и ее образ Фурье $\hat{\varphi}(\omega)$ с точностью до фазового множителя совпадает с $\sin(\omega/2)/(\omega/2)$. Теперь возьмем за основу ступенчатую функцию в частотной области. Положим

$$\hat{\varphi}(\omega) = \begin{cases} 1, & \omega \in [-\pi, \pi] \\ 0, & \text{при других } \omega \end{cases},$$

тогда

$$\varphi(x) = \frac{1}{2\pi} \int_{-\infty}^{\infty} \hat{\varphi}(\omega) e^{i\omega x}\, d\omega = \frac{\sin \pi x}{\pi x}.$$

Пусть V_0 – пространство функций из $L^2(\mathbf{R})$, порожденное системой сдвигов $\{\varphi(x - n), n \in \mathbf{Z}\}$. Данная система является ортонормированной, поскольку из определения функции $\hat{\varphi}(\omega)$ сразу следует, что $\sum_{n \in Z} |\hat{\varphi}(\omega + 2\pi n)|^2 = 1$ *п. в.*

Выясним, из каких функций состоит V_0. По определению, любой элемент пространства V_0 может быть представлен в виде сходящегося в $L^2(\mathbf{R})$ ряда вида

$$f(x) = \sum_{n \in Z} c_n \frac{\sin \pi(x - n)}{\pi(x - n)}, \quad \sum_{n \in Z} |c_n|^2 < \infty.$$

Для того чтобы понять, что собой представляет такая функция, возьмем ее преобразование Фурье:

$$\hat{f}(\omega) = \sum_{n \in Z} c_n e^{-in\omega} \chi_{[-\pi, \pi]}(\omega)$$

– это функция из $L^2(\mathbf{R})$ с носителем на промежутке $[-\pi, \pi]$. Таким образом, пространство V_0 состоит из функций с ограниченной шириной полосы, равной $[-\pi, \pi]$. Напомним также, что по теореме Котельникова–Шеннона функция $f(x)$ ограниченной шириной полосы, равной $[-\pi, \pi]$, может быть записана так:

$$f(x) = \sum_{n \in Z} f(n) \frac{\sin \pi(x - n)}{\pi(x - n)}.$$

Образуем последовательность пространств V_j, каждое из которых является масштабированной версией V_0. Например, пространство V_1 состоит из всех интегрируемых с квадратом функций вида $f(2x)$, где $f(x) \in V_0$. Поскольку $F[f(2x)] = \hat{f}(\omega / 2) / 2$, то преобразования Фурье функций из V_1 имеют носитель на промежутке $[-2\pi, 2\pi]$. Поэтому $V_0 \subset V_1$. Следовательно, функция $\varphi(x)$ является масштабирующей. Пространство V_j состоит из функций, преобразования Фурье которых имеют носитель на промежутке $[-2^j\pi, 2^j\pi]$.

Мы получили последовательность подпространств

$$\ldots \subset V_{-1} \subset V_0 \subset V_1 \subset V_2 \subset \ldots$$

в пространстве $L^2(\mathbf{R})$. Каждое подпространство V_j состоит из функций с ограниченной шириной полосы $[-2^j\pi, 2^j\pi]$. Поэтому замыкание их объединения совпадает с $L^2(\mathbf{R})$, а пересечение всех подпространств – нулевое. Все условия ортогонального кратномасштабного разложения выполнены.

Найдем коэффициенты $\{h_n\}$ масштабирующего фильтра функции $\varphi(x)$. Для этого мы найдем частотную функцию

$$H_0(\omega) = \frac{1}{\sqrt{2}} \sum_{n \in \mathbf{Z}} h_n e^{-i n \omega}$$

и вычислим ее коэффициенты. Для нахождения функции $H_0(\omega)$ рассмотрим масштабирующее соотношение в частотной области, $\hat{\varphi}(\omega) = H_0\left(\frac{\omega}{2}\right)\hat{\varphi}\left(\frac{\omega}{2}\right)$. Поскольку $\hat{\varphi}(\omega/2) = 1$ на промежутке $[-2\pi, 2\pi]$ и нуль в противном случае, то $H_0(\omega/2) = \hat{\varphi}(\omega)$ на $[-2\pi, 2\pi]$, или $H_0(\omega) = \hat{\varphi}(2\omega)$ на промежутке $[-\pi, \pi]$. Поэтому на основном промежутке $[-\pi, \pi]$:

$$H_0(\omega) = \begin{cases} 1 & \omega \in [-\pi/2, \pi/2], \\ 0 & \omega \in [-\pi, -\pi/2) \cup (\pi/2, \pi] \end{cases}. \qquad (1)$$

Вне промежутка $[-\pi, \pi]$ функция $H_0(\omega)$ продолжается периодически. Поскольку $\sqrt{2}H_0(\omega) = \sum_{n \in \mathbf{Z}} h_n e^{-i n \omega}$, то коэффициенты h_n являются коэффициентами Фурье функции $\sqrt{2}H_0(\omega)$:

$$h_n = \sqrt{2}\frac{1}{2\pi}\int_{-\pi}^{\pi} H_0(\omega) e^{i\omega n}\, d\omega = \sqrt{2}\frac{1}{2\pi}\int_{-\pi/2}^{\pi/2} e^{i\omega n}\, d\omega =$$

$$= \begin{cases} \dfrac{1}{\sqrt{2}} & n = 0, \\ (-1)^{(n-1)/2}\dfrac{\sqrt{2}}{\pi n} & n \text{ нечетное} \\ 0 & \text{для остальных } n \end{cases}.$$

Таким образом, функция $\varphi(x) = \sin(\pi x)/(\pi x)$ является масштабирующей с коэффициентами $\{h_n\}$. Отметим, что коэффициенты фильтра $\{h_n\}$ убывают достаточно медленно, как n^{-1}.

Найдем вейвлет $\psi(x) \in W_0$. Согласно результатам параграфа 2.3, функция $\psi(x) \in W_0$ определяется из формулы

$$\hat{\psi}(\omega) = e^{i\omega/2}\,\overline{H_0\left(\frac{\omega}{2}+\pi\right)}\,\hat{\varphi}\left(\frac{\omega}{2}\right) = \begin{cases} e^{i\omega/2} & \pi \le |\,\omega\,| \le 2\pi \\ 0 & \text{при остальных } \omega \end{cases}.$$

Тогда мы имеем:

$$\psi(x) = \frac{1}{2\pi}\int_{-\infty}^{\infty}\hat{\psi}(\omega)e^{i\omega x}d\omega = \frac{1}{2\pi}\int_{-2\pi}^{-\pi}e^{i\omega/2}e^{i\omega x}d\omega + \frac{1}{2\pi}\int_{\pi}^{2\pi}e^{i\omega/2}e^{i\omega x}d\omega =$$

$$= \frac{1}{2\pi}\int_{-2\pi}^{-\pi}e^{i(x+1/2)\omega}d\omega + \frac{1}{2\pi}\int_{\pi}^{2\pi}e^{i(x+1/2)\omega}d\omega =$$

$$= \frac{\sin 2\pi(x+1/2) - \sin\pi(x+1/2)}{\pi(x+1/2)}.$$

Таким образом,

$$\psi(x) = \frac{\sin 2\pi(x+1/2) - \sin\pi(x+1/2)}{\pi(x+1/2)}. \qquad (2)$$

Данную функцию назовем *вейвлетом Шеннона–Котельникова*. Отметим, что вейвлет-базис, порожденный функцией $\psi(x)$, называется также базисом Литвуда–Пэли. Ниже приведены графики функций $\varphi(x)$ и $\psi(x)$.

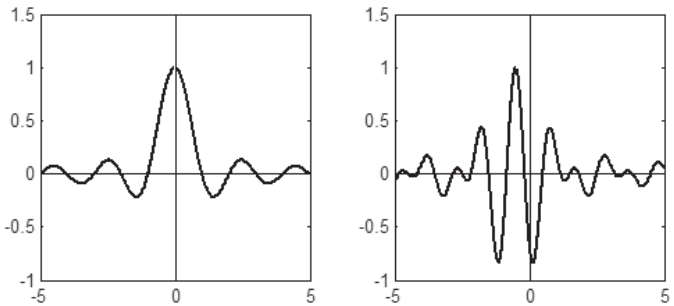


Рис. 2.5.1. Вейвлеты Шеннона–Котельникова $\varphi(x)$ и $\psi(x)$

Замечание 1. По построению вейвлеты Шеннона–Котельникова соответствуют идеальным фильтрам $H_0(\omega)$ и $G_0(\omega) = 1 - H_0(\omega)$. Вследствие разрывности функции $\hat{\varphi}(\omega)$ масштабирующая функция $\varphi(x)$ и вейвлет $\psi(x)$ убывают очень медленно, как n^{-1}. Поэтому вейвлеты Шеннона–Котельникова слабо локализованы в пространстве. Медленное убывание коэффициентов фильтра делает вейвлет Шеннона–Котельникова не очень удобным в практических применениях.

Вейвлет Хаара есть противоположный крайний случай, когда $\varphi(x)$ является разрывной по времени.

2.5.2. Вейвлеты Мейера

Вейвлеты Мейера являются сглаженным вариантом вейвлетов Шеннона–Котельникова. Разрывы масштабирующей функции $\hat{\varphi}(\omega)$ вейвлетов Шеннона–Котельникова в точках $-\pi$ и π можно удалить сглаживанием многими способами. Однако нужно, чтобы сглаженная функция $\hat{\varphi}(\omega)$ обладала свойством $\sum_{n\in Z}|\hat{\varphi}(\omega+2\pi n)|^2=1$, которое обеспечивает ортогональность системы функций $\varphi(x-n)$. Отметим, что при сглаживании может быть улучшено медленное убывание вейвлета Шеннона–Котельникова.

Масштабирующая функция Мейера $\varphi(x)$ определяется из равенства

$$\hat{\varphi}(\omega)=\begin{cases}1, & \omega\in[-2\pi/3,\,2\pi/3],\\ \cos\left(\dfrac{\pi}{2}\nu\left(\dfrac{3}{2\pi}|\omega|-1\right)\right), & 2\pi/3\le|\omega|\le 4\pi/3,\\ 0, & \text{для остальных } \omega,\end{cases} \tag{4}$$

где $\nu(x)$ есть вспомогательная функция, удовлетворяющая двум условиям:

$$\nu(x)=\begin{cases}0, & x\le 0\\ 1 & x\ge 1\end{cases} \quad \text{и} \quad \nu(x)+\nu(1-x)=1. \tag{5}$$

Например, можно считать, что $\nu(x) = x$ на промежутке [0, 1]. Это линейная интерполяция между 0 и 1. Распространенным выбором функции $\nu(x)$ является следующая полиномиальная интерполяция $\nu(x) = x^4(35 - 84x + 70x^2 - 20x^3)$ между 0 и 1 на промежутке [0, 1]. Ниже приведен график $\hat{\varphi}(\omega)$ с данной вспомогательной функцией $\nu(x)$.



Рис. 2.5.2. График $\hat{\varphi}(\omega)$

Таким образом, сглаживание производится только на промежутках $[-4\pi/3, -2\pi/3]$ и $[2\pi/3, 4\pi/3]$. За счет выбора функции $\nu(x)$ обеспечивается необходимая гладкость в точках склейки. Функция cos удобна для обеспечения ортогональности системы сдвигов $\varphi(x - n)$.

Пусть V_0 – подпространство в $L^2(\mathbf{R})$, порожденное сдвигами функции $\varphi(x)$. Покажем, что функции $\varphi(x - n)$ образуют ортонормированный базис пространства V_0. Для этого достаточно проверить равенство

$$\sum_{n\in Z}|\hat{\varphi}(\omega+2\pi n)|^2=1.$$

По построению $\hat{\varphi}(\omega)$ мы видим, что достаточно показать равенство

$$\cos^2\left(\frac{\pi}{2}\nu\left(\frac{3}{2\pi}\omega-1\right)\right)+\cos^2\left(\frac{\pi}{2}\nu\left(\frac{3}{2\pi}|\omega-2\pi|-1\right)\right)=1$$

только на промежутке $[2\pi/3, 2\pi]$. Это равенство верно, так как второй член в левой части уравнения может быть написан в виде

$$\cos^2\left(\frac{\pi}{2}\nu\left(\frac{3}{2\pi}|\omega-2\pi|-1\right)\right)=\cos^2\left(\frac{\pi}{2}\nu\left(-\frac{3}{2\pi}\omega+2\right)\right)=$$

[воспользуемся равенством $\nu(1 - x) = 1 - \nu(x)$]

$$=\cos^2\left(\frac{\pi}{2}-\frac{\pi}{2}\nu\left(\frac{3}{2\pi}\omega-1\right)\right)=\sin^2\left(\frac{\pi}{2}\nu\left(\frac{3}{2\pi}\omega-1\right)\right).$$

Построим кратномасштабное разложение пространства $L^2(\mathbf{R})$. Пусть V_0 – подпространство в $L^2(\mathbf{R})$, порожденное сдвигами функции $\varphi(x)$. Функция $f(x) \in V_0$ представляется в виде

$$f(x)=\sum_{n\in Z}c_n\varphi(x-n),$$

где $\sum_{n\in Z}|c_n|^2<\infty$. Последнее разложение позволяет записать преобразование Фурье функции $f(x) \in V_0$ в виде

$$\hat{f}(\omega)=\sum_{n\in Z}c_n e^{-in\omega}\hat{\varphi}(\omega)=F_0(\omega)\hat{\varphi}(\omega), \qquad (6)$$

где $F_0(\omega)=\sum_{n\in Z}c_n e^{-in\omega}$ – 2π-периодическая функция, принадлежащая пространству $L^2[-\pi, \pi]$. За счет умножения на $\hat{\varphi}(\omega)$ данное выражение (6) обращается в нуль вне промежутка $[-4\pi/3, 4\pi/3]$. Таким образом, каждая функция $f(x) \in V_0$ является функцией с ограниченной шириной полосы $\Omega = 4\pi/3$. Однако определенное нами пространство V_0 несколько отличается от аналогичного пространства V_0 в разложе-

нии Шеннона. Действительно, преобразование Фурье $\hat{f}(\omega)$ функции $f(x) \in V_0$ – это не произвольная функция из $L^2(\mathbf{R})$, равная нулю вне промежутка $[-4\pi/3, 4\pi/3]$. Из (6) следует, что функция $\hat{f}(\omega)$ получается из 2π-периодической функции $F_0(\omega)$, принадлежащей пространству $L^2[-\pi, \pi]$, умножением на функцию $\hat{\varphi}(\omega)$ вида (4), см. рис. 2.5.2:

$$\hat{f}(\omega) = F_0(\omega)\hat{\varphi}(\omega).$$

Возьмем последовательность пространств V_j, каждое из которых является масштабированной версией V_0. Пространство V_1 состоит из всех интегрируемых с квадратом функций вида $g(x) = f(2x)$, где $f(x) \in V_0$. Охарактеризуем элементы из V_1 через их преобразования Фурье. Поскольку $\hat{g}(\omega) = F[f(2x)] = \hat{f}(\omega/2)/2$, то преобразования Фурье функций $g(x)$ из V_1 имеют носитель на промежутке $[-8\pi/3, 8\pi/3]$ и имеют вид функций, «сглаженных» вблизи концов промежутка умножением на $\hat{\varphi}(\omega/2)$:

$$\hat{g}(\omega) = 1/2\, F_0(\omega/2)\hat{\varphi}(\omega/2) = F_1(\omega)\hat{\varphi}(\omega/2),$$

где $F_1(\omega) = 1/2F_0(\omega/2)$ – 4π-периодическая функция, принадлежащая пространству $L^2[-2\pi, 2\pi]$. Поэтому $V_0 \subset V_1$. Следовательно, функция $\varphi(x)$ является масштабирующей.

Пространство V_j состоит из функций $g(x)$, преобразования Фурье которых имеют вид

$$\hat{g}(\omega) = F_j(\omega)\hat{\varphi}(\omega/2^j),$$

где $F_j(\omega)$ – это $2^j\pi$-периодическая функция, принадлежащая пространству $L^2[-2^j\pi, 2^j\pi]$. Мы получили последовательность подпространств

$$... \subset V_{-1} \subset V_0 \subset V_1 \subset V_2 \subset ...$$

в пространстве $L^2(\mathbf{R})$, образующую кратномасштабное разложение $L^2(\mathbf{R})$.

Найдем коэффициенты масштабирующего фильтра $\{h_n\}$ функции $\varphi(x)$. Для этого рассмотрим масштабирующее соотношение в частотной области

$$\hat{\varphi}(\omega) = H_0\left(\frac{\omega}{2}\right)\hat{\varphi}\left(\frac{\omega}{2}\right)$$

и найдем частотную функцию $H_0(\omega)$. Поскольку $\hat{\varphi}(\omega/2) = 1$ на промежутке $[-4\pi/3, 4\pi/3]$, а функция $\hat{\varphi}(\omega)$ обращается в нуль вне промежутка $[-4\pi/3, 4\pi/3]$, то $H_0(\omega/2) = \hat{\varphi}(\omega)$ на $[-4\pi/3, 4\pi/3]$, или $H_0(\omega) = \hat{\varphi}(2\omega)$ на промежутке $[-2\pi/3, 2\pi/3]$,

$$H_0(\omega)=\begin{cases}\hat{\varphi}(2\omega) & \omega\in[-2\pi/3,\,2\pi/3],\\ 0 & \omega\in[-\pi,-2\pi/3)\cup(2\pi/3,\,\pi]\end{cases}. \tag{7}$$

Вне промежутка $[-\pi, \pi]$ функция $H_0(\omega)$ продолжается периодически. Тогда легко видеть, что

$$H_0(\omega)=\sum_{n\in\mathbf{Z}}\hat{\varphi}(2(\omega+2\pi n)). \tag{8}$$

Коэффициенты $\{h_n\}$ масштабирующего соотношения находятся разложением в ряд Фурье функции $H_0(\omega)$:

$$h_n=\sqrt{2}\frac{1}{2\pi}\int_{-\pi}^{\pi}H_0(\omega)e^{in\omega}d\omega=\sqrt{2}\frac{1}{2\pi}\int_{-\pi}^{\pi}\hat{\varphi}(2\omega)\cos n\omega\,d\omega=$$

$$=\sqrt{2}\frac{1}{\pi}\left(\int_0^{\pi/3}\cos n\omega\,d\omega+\int_{\pi/3}^{2\pi/3}\cos\left(\frac{\pi}{2}\nu\left(\frac{3}{\pi}\omega-1\right)\right)\cos n\omega\,d\omega\right).$$

Найдем вейвлет $\psi(x)$. По формуле (12) из раздела 2.3.2 с учетом формулы (8) этого параграфа вейвлет $\psi(x)\in W_0$ определяется так:

$$\hat{\psi}(\omega)=e^{i\omega/2}\overline{H_0\left(\frac{\omega}{2}+\pi\right)}\,\hat{\varphi}\left(\frac{\omega}{2}\right)=e^{i\omega/2}\sum_n\hat{\varphi}(\omega+2\pi+4\pi n)\,\hat{\varphi}\left(\frac{\omega}{2}\right).$$

Поскольку носитель $\hat{\varphi}(\omega/2)$ есть $[-8\pi/3, 8\pi/3]$, то в сумме можно ограничиться только двумя слагаемыми

$$\hat{\psi}(\omega)=\ e^{i\omega/2}(\hat{\varphi}(\omega-2\pi)+\hat{\varphi}(\omega+2\pi))\,\hat{\varphi}\left(\frac{\omega}{2}\right). \tag{9}$$

На верхнем рис. 2.5.3 приведены графики $\hat{\varphi}(\omega/2)$ (сплошная линия) и выражения в скобках (пунктирная линия) на интервале $[-4\pi, 4\pi]$. Произведение этих двух графиков дает график функции $|\hat{\psi}(\omega)|$ (нижний рисунок).

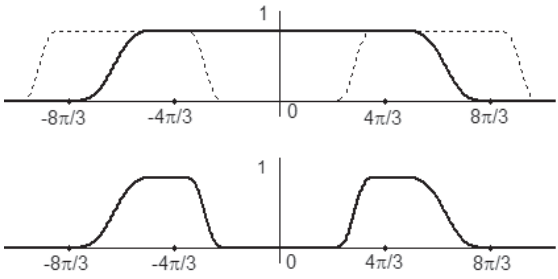


Рис. 2.5.3. Вейвлеты Мейера в частотной области

Из равенства (9) сразу следует, что $\hat{\psi}(\omega)$ задается следующим образом:

$$\hat{\psi}(\omega)=\begin{cases} e^{i\omega/2}\sin\left(\frac{\pi}{2}\nu\left(\frac{3}{2\pi}|\omega|-1\right)\right), & \frac{2\pi}{3}\le|\omega|\le\frac{4\pi}{3} \\ e^{i\omega/2}\cos\left(\frac{\pi}{2}\nu\left(\frac{3}{4\pi}|\omega|-1\right)\right), & \frac{4\pi}{3}\le|\omega|\le\frac{8\pi}{3} \\ 0, & |\omega|\notin\left[\frac{2\pi}{3},\frac{8\pi}{3}\right] \end{cases} \qquad (10)$$

Масштабирующая функция и вейвлет в пространственной области получаются при помощи обратного преобразования Фурье:

$$\varphi(x)=\frac{1}{2\pi}\int_{-\infty}^{\infty}\hat{\varphi}(\omega)e^{i\omega x}d\omega=\frac{1}{\pi}\int_{0}^{4\pi/3}\hat{\varphi}(\omega)\cos\omega x\,d\omega,$$

$$\psi(x)=\frac{1}{2\pi}\int_{-\infty}^{\infty}e^{i\omega/2}\,\hat{\varphi}(\omega/2)(\hat{\varphi}(\omega-2\pi)+\hat{\varphi}(\omega+2\pi))e^{i\omega x}d\omega=$$

$$=\frac{1}{\pi}\int_{0}^{\infty}\hat{\varphi}(\omega/2)\,\hat{\varphi}(\omega-2\pi)\cos\omega(x+1/2)d\omega=$$

$$=\frac{1}{\pi}\int_{2\pi/3}^{8\pi/3}\hat{\varphi}(\omega/2)\,\hat{\varphi}(\omega-2\pi)\cos\omega(x+1/2)d\omega.$$

Данную функцию называют *вейвлетом Мейера*. Вообще, мы не можем получить аналитической формы для масштабирующей функции или вейвлета и ограничиваемся их приближениями.

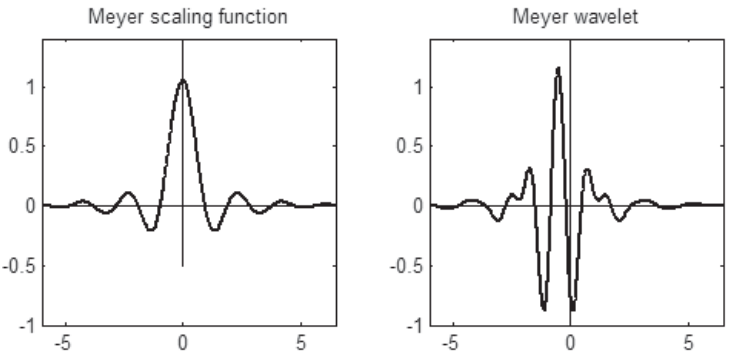


Рис. 2.5.4. Вейвлеты Мейера $\varphi(x)$ и $\psi(x)$

Замечание 2. Если функция $\hat{\varphi}(\omega)$ имеет n непрерывных производных, то функции $\varphi(x)$ и $\psi(x)$ называются вейвлетами Мейера порядка n. Если мы потребуем n непрерывных производных функции $\hat{\varphi}(\omega)$ в точках $2\pi/3$ и $4\pi/3$, тогда можно показать, что функция $\nu(x)$ (если она полиномиальная) должна иметь вид $\nu(x) = x^{n+1}(c_0 + c_1x + ... + c_nx^n)$. Коэффициенты c_n находятся единственным образом из условия $\nu(x) + \nu(1 - x) = 1$. Примеры функций $\nu(x)$:

$\nu(x) = x$;
$\nu(x) = x^2(3 - 2x)$;
$\nu(x) = x^3(10 - 15x + 6x^2)$;
$\nu(x) = x^4(35 - 84x + 70x^2 - 20x^3)$,
$\nu(x) = x^5(126 - 420x + 540x^2 - 315x^3 + 70x^4)$.

2.6. Вейвлеты Батла–Лемарье. B-сплайны

При построении вейвлетов Хаара мы использовали в качестве масштабирующей функции характеристическую функцию интервала [0, 1). Соответствующие базисные вейвлеты получились разрывными. Поэтому разложение гладких функций по базису вейвлетов Хаара не очень эффективно. Хотелось бы найти другие, более гладкие, простые масштабирующие функции и, соответственно, вейвлеты. Однако далеко не каждая функция удовлетворяет масштабирующему уравнению. Нахождение таких функций представляет достаточно сложную задачу. Замечательно, что существует целое семейство функций, *B*-сплайнов, которые являются масштабирующими. Каждая такая функция порождает семейство вложенных подпространств $... \subset V_{-1} \subset V_0 \subset V_1 \subset V_2 \subset ...$ обычным образом. При этом будут выполнены все свойства кратномасштабного разложения, кроме ортонормированности семейства сдвигов $\varphi_{0,n}(x) = \varphi(x - n)$, $n \in \mathbf{Z}$. Однако существует прием, который позволяет провести ортогонализацию. Он был предложен Франклином [Fr] еще в 1928 г. для *B*-сплайнов степени 1. В этом параграфе будет рассмотрена процедура ортогонализации и будут построены ортогональные вейвлеты на основе *B*-сплайнов. Рассмотрим сначала все эти конструкции на примере *B*-сплайна степени 1.

2.6.1. Вейвлеты на основе B-сплайна степени 1

Рассмотрим функцию

$$\varphi(x) = \begin{cases} 1 - |x|, & x \in [-1,1], \\ 0 & \text{при остальных } x \end{cases}. \qquad (1)$$

Она имеет компактный носитель $[-1, 1]$ и удовлетворяет масштабирующему уравнению

$$\varphi(x) = \frac{1}{2}\varphi(2x+1) + \varphi(2x) + \frac{1}{2}\varphi(2x-1). \qquad (2)$$

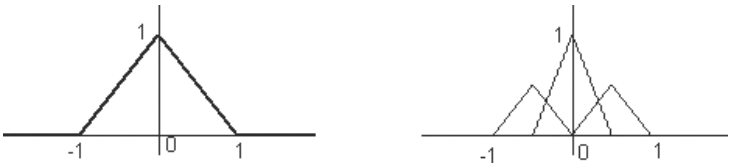


Рис. 2.6.1. Масштабирующее соотношение для B-сплайна степени 1

Фильтр состоит из трех ненулевых чисел $h_{-1} = 1/2\sqrt{2}$, $h_0 = 1/\sqrt{2}$, $h_1 = 1/2\sqrt{2}$. Соответствующая частотная функция $H_0(\omega)$ имеет вид:

$$H_0(\omega) = \frac{1}{\sqrt{2}}\frac{1}{2\sqrt{2}}(e^{i\omega} + 2 + e^{-i\omega}) = \frac{e^{i\omega}}{4}(1 + 2e^{-i\omega} + e^{-i2\omega}) = \frac{e^{i\omega}}{4}(1 + e^{-i\omega})^2 .$$

Ее можно представить также следующим образом:

$$H_0(\omega) = \frac{1}{2}(1 + \cos\omega) = \cos^2 \omega/2. \qquad (3)$$

Пусть V_0 – подпространство в $L^2(\mathbf{R})$, порожденное функциями $\varphi_{0,n}(x) = \varphi(x - n)$, $n \in \mathbf{Z}$. Эти функции образуют базис пространства V_0, но не ортогональный. Действительно, функции $\varphi(x)$ и $\varphi(x - 1)$ имеют пересекающиеся по промежутку $[0, 1]$ носители и легко видеть, что

$$(\varphi(x), \varphi(x-1)) = \int_{\mathbf{R}} \varphi(x)\varphi(x-1)dx = \int_0^1 x(1-x)dx = 1/6 > 0.$$

Образуем пространства V_j как масштабированные версии V_0 обычным образом. Например, пространство V_1 состоит из всех функций вида $f(2x)$, где $f(x) \in V_0$. Из равенства (2) следует, что $V_0 \subset V_1$, и поэтому для любого $j \in \mathbf{Z}$ имеет место включение $V_j \subset V_{j+1}$. Мы получили последовательность замкнутых вложенных друг в друга подпространств

$$... \subset V_{-1} \subset V_0 \subset V_1 \subset V_2 \subset ...,$$

при этом выполнены все свойства кратномасштабного анализа, кроме ортогональности базиса $\{\varphi(x - n)\}$ (см. пример 1 из раздела 2.3.4).

Для ортогонализации базиса можно применить обычную процедуру Грамма–Шмидта, однако новый базис, который при этом получается, не будет образован сдвигами одной функции. Применим другой способ, который использует то, что наш базис является базисом Рисса.

Напомним, что базис $\{u_n\}_{n\in \mathbf{Z}}$ в гильбертовом пространстве H называется *базисом Рисса*, если существуют положительные константы A_1 и A_2, такие что для любого элемента $f \in H$ справедливы неравенства

$$A_1 \| f \|^2 \le \sum_n |(f, u_n)|^2 \le A_2 \| f \|^2. \tag{4}$$

Ранее было показано, что система $\{\varphi(x - n),\ n \in \mathbf{Z}\}$, полученная сдвигами некоторой функции $\varphi(x) \in L^2(\mathbf{R})$, образует базис Рисса пространства V_0 тогда и только тогда, когда существуют положительные константы A_1 и A_2, такие что

$$A_1 \le \sum_{k\in Z} \left|\hat{\varphi}(\omega + 2\pi k)\right|^2 \le A_2 \text{ п. в.} \tag{5}$$

Из базиса Рисса вида $\{\varphi(x - n),\ n \in \mathbf{Z}\}$ можно получить ортонормированный базис способом, который был предложен Франклином в 1928 г. [Fr], см. также [Бл].

Ортогонализационный прием. Пусть $\{\varphi_{0,n}(x) = \varphi(x - n),\ n \in \mathbf{Z}\}$ – система функций, образующая базис Рисса пространства $V_0 \subset L^2(\mathbf{R})$. Выполнено условие (5), а для ортонормированности требуется выполнение равенства

$$\sum_{k\in Z} \left|\hat{\varphi}(\omega + 2\pi k)\right|^2 = 1 \text{ п. в.} \tag{6}$$

Если это равенство не выполнено, то мы определим другую функцию

$$\hat{\varphi}^*(\omega) = \frac{\hat{\varphi}(\omega)}{\sqrt{\sum_{k\in Z} \left|\hat{\varphi}(\omega + 2\pi k)\right|^2}}. \tag{7}$$

Для последней функции условие (6), очевидно, имеет место.

Положим

$$A(\omega) = \frac{1}{\sqrt{\sum_{k\in Z} \left|\hat{\varphi}(\omega + 2\pi k)\right|^2}} \quad \text{и} \quad B(\omega) = \sqrt{\sum_{k\in Z} \left|\hat{\varphi}(\omega + 2\pi k)\right|^2}. \tag{8}$$

Функции $A(\omega)$ и $B(\omega)$ являются четными, 2π-периодическими, принадлежащими пространству $L^2[0, 2\pi]$.

Рассмотрим функцию $\varphi^*(x) = F^{-1}[\hat{\varphi}^*(\omega)]$. Для нее выполнено необходимое и достаточное условие ортонормированности. Поэтому ее сдвиги $\{\varphi^*(x-n)\}_{n\in\mathbf{Z}}$ порождают ортонормированную систему. Однако возникают два вопроса: является ли функция $\varphi^*(x)$ масштабирующей и принадлежит ли она пространству V_0? Сначала заметим, что из равенства $\hat{\varphi}^*(\omega) = A(\omega)\hat{\varphi}(\omega)$, используя разложение функции $A(\omega)$ в ряд Фурье, получаем:

$$\hat{\varphi}^*(\omega) = A(\omega)\hat{\varphi}(\omega) = \sum_n a_n e^{-in\omega}\hat{\varphi}(\omega) = F[\sum_n a_n \varphi(x-n)].$$

Поэтому $\varphi^*(x) = \sum_n a_n \varphi(x-n)$. Это означает, что функция $\varphi^*(x)$ раскладывается по исходному базису пространства V_0. Отметим, что функция $\varphi(x) \in V_0$ также имеет разложение $\varphi(x) = \sum_n b_n \varphi^*(x-n)$ по базису $\{\varphi^*(x-x)\}$, поскольку $\hat{\varphi}(\omega) = B(\omega)\hat{\varphi}^*(\omega)$. Следовательно, система функций $\{\varphi^*(x-n)\}_{n\in\mathbf{Z}}$ является базисом исходного пространства V_0. Масштабируемость функции $\varphi^*(x)$ вытекает теперь из включения $V_0 \subset V_1$ и того, что базис V_1 состоит из функций $\sqrt{2}\varphi^*(2x-n)$.

Обратимся опять к нашему примеру B-сплайна $\varphi(x)$ степени 1. Преобразование Фурье масштабирующей функции $\varphi(x) \in V_0$ хорошо известно:

$$\hat{\varphi}(\omega) = \frac{4\sin^2 \omega/2}{\omega^2}. \tag{9}$$

Сумма $\sum_k \left|\hat{\varphi}(\omega + 2\pi k)\right|^2$ вычислена в примере 1 раздела 2.3.4:

$$\sum_{k\in Z} \left|\hat{\varphi}(\omega + 2\pi k)\right|^2 = \frac{1}{3}(2 + \cos\omega) = \frac{1}{3}(1 + 2\cos^2 \omega/2). \tag{10}$$

Мы видим, что она ограничена и сверху, и снизу положительными числами. Поэтому функции $\{\varphi(x-n)\}$ образуют базис Рисса и можно применить ортогонализационный прием. Тогда выражение для $\hat{\varphi}^*(\omega)$ получается по формуле (7):

$$\hat{\varphi}^*(\omega) = \frac{\sqrt{3}}{(1 + 2\cos^2 \omega/2)^{1/2}} \frac{4\sin^2 \omega/2}{\omega^2}. \tag{11}$$

Для нахождения масштабирующей функции $\varphi^*(x)$ нужно применить обратное преобразование Фурье к $\hat{\varphi}^*(\omega)$. Поскольку $\hat{\varphi}^*(\omega) = A(\omega)\hat{\varphi}(\omega)$, то более простой способ нахождения $\phi^*(x)$ заключается в разложении функции $A(\omega)$ в ряд Фурье:

$$A(\omega) = \frac{\sqrt{3}}{(1+2\cos^2\omega/2)^{1/2}} = \sum_n a_n e^{-in\omega}.$$

Тогда $\hat{\varphi}^*(\omega) = \sum_n a_n e^{-in\omega}\,\hat{\varphi}(\omega)$, и поэтому масштабирующая функция задается рядом: $\varphi^*(x) = \sum_n a_n \varphi(x-n)$. Следовательно, она является линейным сплайном с узлами в целых точках.

Поскольку $A(\omega)$ является тригонометрическим полиномом в отрицательной степени $-1/2$, то ее разложение в ряд Фурье будет иметь бесконечное число членов. Поэтому ортогонализованная масштабирующая функция $\varphi^*(x)$ будет иметь бесконечный носитель, в отличие от исходной функции $\varphi(x)$. Однако функция $A(\omega)$ является гладкой, и поэтому ее коэффициенты Фурье быстро убывают. Поэтому мы можем усечь ряд Фурье функции $A(\omega)$ при некотором n в зависимости от требуемой точности. В частности, в нашем примере линейного B-сплайна имеем: $a_0 = 1.2917$; $a_1 = a_{-1} = -0.1747$; $a_2 = a_{-2} = 0.0352$; $a_3 = a_{-3} = -0.0079$; $a_4 = a_{-4} = 0.0018$; $a_5 = a_{-5} = -0.0004$.

Для сравнения покажем на рис. 2.6.2 линейный B-сплайн $\varphi(x)$ вместе с ортогонализованной масштабирующей функцией $\varphi^*(x)$.

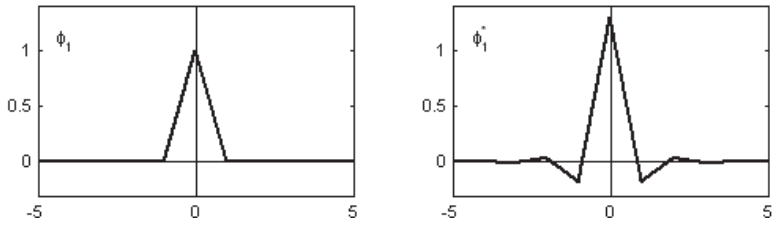


Рис. 2.6.2. Линейный B-сплайн и его ортогонализация

Мы видим, что ортогонализованная масштабирующая функция имеет форму линейного сплайна и достаточно быстро убывает. Значения функции $\varphi^*(x)$ в целых точках от 0 до 5 равны соответственно: $a_0 = 1.2917$; $a_1 = -0.1747$; $a_2 = 0.0352$; $a_3 = -0.0079$; $a_4 = 0.0018$; $a_5 = -0.0004$.

Вейвлеты. Найдем теперь соответствующий вейвлет $\psi^*(x)$. Поскольку мы построили ортогональный кратномасштабный анализ, то можно воспользоваться формулой, установленной в теореме 3 раздела 2.3.2:

$$\hat{\psi}^*(\omega) = e^{i\omega/2}\,\overline{H_0^*\left(\frac{\omega}{2}+\pi\right)}\,\hat{\varphi}^*\left(\frac{\omega}{2}\right). \qquad (12)$$

Нам требуется функция $H_0^*(\omega)$. Она может быть найдена из масштабирующего уравнения $\hat{\varphi}^*(\omega) = H_0^*(\omega/2)\,\hat{\varphi}^*(\omega/2)$. Имеем:

$$H_0^*(\omega) = \frac{\hat{\varphi}^*(2\omega)}{\hat{\varphi}^*(\omega)} = \frac{4\sqrt{3}\sin^2\omega}{4\omega^2(1+2\cos^2\omega)^{1/2}} \frac{\omega^2(1+2\cos^2\omega/2)^{1/2}}{4\sqrt{3}\sin^2\omega/2}.$$

$$H_0^*(\omega) = \cos^2\left(\frac{\omega}{2}\right)\left(\frac{1+2\cos^2\omega/2}{1+2\cos^2\omega}\right)^{1/2}. \qquad (13)$$

Подставляя в (12) выражения из (13) и (11) и учитывая выражение для $\hat{\varphi}^*(\omega/2)$, получаем:

$$\hat{\psi}^*(\omega) =$$

$$= \sqrt{3}\,e^{i\omega/2}\sin^2(\omega/4)\left(\frac{1+2\sin^2\omega/4}{(1+2\cos^2\omega/2)(1+2\cos^2\omega/4}\right)^{1/2}\hat{\varphi}(\omega/2). \quad (14)$$

Для нахождения вейвлета $\psi^*(x)$ нужно применить обратное преобразование Фурье к $\hat{\psi}^*(x)$. Другой способ нахождения вейвлета $\psi^*(x)$ использует представление (14) функции $\psi^*(x)$ в виде произведения $\hat{\psi}^*(\omega) = C(\omega/2)\hat{\varphi}(\omega/2)$ и разложение функции

$$C(\omega) = \sqrt{3}\,e^{i\omega}\sin^2(\omega/2)\left(\frac{1+2\sin^2\omega/2}{(1+2\cos^2\omega)(1+2\cos^2\omega/2)}\right)^{1/2}$$

в ряд Фурье $C(\omega) = \sum_n c_n e^{-in\omega}$. Тогда функция $\psi^*(x)$ представляется рядом $\psi^*(x) = 2\sum_n c_n \varphi(2x-n)$. Следовательно, она является линейным сплайном с узлами в полуцелых точках.

Построенные функции $\varphi^*(x)$ и $\psi^*(x)$ называются вейвлетами *Батла–Лемарье*.

2.6.2. B-сплайны

Напомним, что *сплайном порядка n минимального дефекта* называется функция $f(x)$, которая является полиномом степени, не более чем n на каждом интервале $[k, k+1)$ и непрерывно дифференцируема $n-1$ раз во всех точках, включая узлы. Для простоты мы будем называть их просто сплайнами. Рассмотрим пространство S_n полиномиальных сплайнов $f(x)$ порядка n с узлами в целых числах.

Например, когда $n = 0$, пространство S_0 состоит из кусочно-постоянных функций с возможными разрывами в целых числах. Функция Хаара, определенная ранее, и ее сдвиги образуют базис для S_0, поскольку любая кусочно-постоянная функция из S_0 может быть запи-

сана как комбинация функций базиса. Поэтому функцию Хаара $\varphi_0(x)$ называют *базисным сплайном* порядка нуль, или *B*-сплайном, где *B* обозначает, что это базисная функция.

Для $n = 1$ пространство S_1 сплайнов порядка 1 состоит из непрерывных кусочно-линейных функций с возможным разрывами первой производной в целых числах. Тогда производная любого сплайна порядка 1 есть сплайн порядка нуль. Поэтому функции из S_1 могут быть получены интегрированием сплайнов нулевого порядка. Если мы хотим получить базис в S_1 из сплайнов с компактным носителем, мы должны интегрировать вейвлеты Хаара. Однако есть способ получить базисные функции в S_1 интегрированием базисных функций из S_0 следующим образом:

$$\int_0^1 \varphi_0(x-t)\,dt = \int_{\max(0,x-1)}^{\min(1,x)} dt = \begin{cases} x & x \in [0,1] \\ 2-x & x \in [1,2] \\ 0 & \text{при остальных } x \end{cases}.$$

Первый рисунок ниже поясняет данную формулу. Второй рисунок есть график полученной функции.

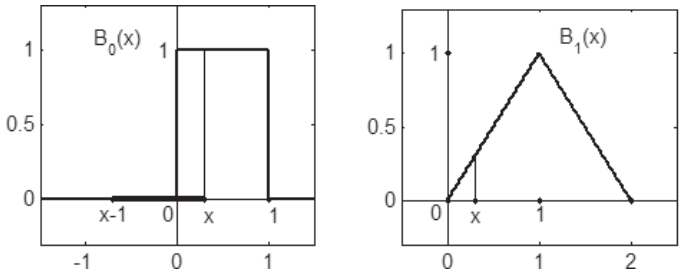


Рис. 2.6.3. B-сплайны

Получили *B*-сплайн $\varphi_1(x)$ порядка 1 с носителем на промежутке [0, 2]. Его можно также записать в виде свертки $\varphi_0(x)$ с собой:

$$\varphi_1(x) = \int_0^1 \varphi_0(x-t)\,dt = \int_{-\infty}^{\infty} \varphi_0(t)\varphi_0(x-t)\,dt = \varphi_0 * \varphi_0.$$

B-сплайны высших порядков определяются индуктивно. Если определен *B*-сплайн $\varphi_{n-1}(x)$, то *B*-сплайн порядка *n* определяется сверткой

$$\varphi_n = \varphi_{n-1} * \varphi_0. \qquad (15)$$

Установим несколько свойств B-сплайнов $\varphi_n(x)$.

Теорема 1. *Для любого* $n \geq 0$ *B-сплайны* $\varphi_n = \varphi_{n-1} * \varphi_0$ *обладают следующими свойствами*:

B-сплайн $\varphi_n(x)$ *порядка* n *имеет носитель на промежутке* $[0, n+1]$, *и график* $\varphi_n(x)$ *симметричен относительно точки* $(n+1)/2$.

$$\sum_k \varphi_n(x-k) = 1, \forall x.$$

$$\int_0^{n+1} \varphi_n(x)dx = 1.$$

$$\varphi'_{n+1}(x) = \varphi_n(x) - \varphi_n(x-1).$$

$$\varphi_n(x) = \frac{x}{n}\varphi_{n-1}(x) + \frac{n+1-x}{n}\varphi_{n-1}(x-1).$$

Доказательство предлагается читателю в качестве упражнения (например, для доказательства свойства 2 см. формулу (16) ниже и формулу (22) раздела 1.3.7).

Поскольку

$$\hat{\varphi}_0(\omega) = e^{-i\omega/2}\frac{\sin(\omega/2)}{\omega/2},$$

то из $\varphi_n = \varphi_{n-1} * \varphi_0$ по свойству преобразования Фурье свертки для любого n получаем $\hat{\varphi}_n(\omega) = \hat{\varphi}_0(\omega)^{n+1}$,

$$\hat{\varphi}_n(\omega) = \hat{\varphi}_0(\omega)^{n+1} = e^{-i(n+1)\omega/2}\left(\frac{\sin(\omega/2)}{\omega/2}\right)^{n+1}. \qquad (16)$$

Теорема 2. *Для любого* $n \geq 0$ *B-сплайн* $\varphi_n(x)$ *является масштабирующей функцией с частотной функцией, равной* $H_{0,n}(\omega) = H_0(\omega)^{n+1}$, *где* $H_0(\omega) = (1 + e^{-i\omega}) / 2$ *– частотная функция для сплайна Хаара* $\varphi_0(x)$ *степени* 0. *Фильтр коэффициентов* $\{h_n\}$ *состоит из чисел*

$$h_k = \frac{1}{2^n\sqrt{2}}C_{n+1}^k, \quad k = 0, 1, \ldots, n+1, \qquad (17)$$

где $C_{n+1}^k = \dfrac{(n+1)!}{k!(n+1-k)!}$ *– биномиальные коэффициенты.*

Доказательство. Для B-сплайна $\varphi_0(x)$ степени 0 имеем следующее масштабирующее соотношение в частотной области:

$$\hat{\varphi}_0(\omega) = H_0(\omega/2)\hat{\varphi}_0(\omega/2).$$

Возводя в степень $n + 1$ это соотношение и учитывая $\hat{\varphi}_n(\omega) = \hat{\varphi}_0(\omega)^{n+1}$, получаем:

$$\hat{\varphi}_n(\omega) = \hat{\varphi}_0(\omega)^{n+1} = H_0(\omega/2)^{n+1}\hat{\varphi}_0(\omega/2)^{n+1} = H_0(\omega/2)^{n+1}\hat{\varphi}_n(\omega/2),$$

$$\hat{\varphi}_n(\omega) = H_0(\omega/2)^{n+1}\hat{\varphi}_n(\omega/2).$$

Мы получили, что B-сплайн $\varphi_n(x)$ удовлетворяет масштабирующему соотношению с частотной функцией $H_{0,n}(\omega) = H_0(\omega)^{n+1}$, где $H_0(\omega) = (1 + e^{-i\omega}) / 2$ – частотная функция для сплайна Хаара $\varphi_0(x)$ степени 0. Тогда

$$H_{0,n}(\omega) = H_0(\omega)^{n+1} = \left(\frac{1}{2}(1+e^{-i\omega})\right)^{n+1} = \frac{1}{2^{n+1}}\sum_{k=0}^{n+1} C_{n+1}^k e^{-ik\omega}, \tag{18}$$

откуда и следует выражение для коэффициентов масштабирующего фильтра.

В дальнейшем будет удобно сдвинуть точку симметрии B-сплайна $\varphi_n(x)$ ближе к нулю. А именно будем считать, что для $\varphi_n(x)$ носитель находится на промежутке $[-(n + 1)/2, (n + 1)/2]$ в случае нечетного n, а в случае четного n – на промежутке $[-n/2, n/2 + 1]$. Тогда

$$\hat{\varphi}_n(\omega) = e^{-iK\omega/2}\left(\frac{\sin(\omega/2)}{\omega/2}\right)^{n+1}, \tag{19}$$

где $K = 0$ в случае нечетного n и $K = 1$ – в случае четного n. В этом случае коэффициенты $\{h_n\}$ будут теми же самыми (17), но будут занумерованы от $-(n + 1)/2$ до $(n + 1)/2$ – в случае нечетного n и от $-n/2$ до $(n + 2)/2$ – в случае четного n.

Из (19) легко получается еще одно выражение для частотной функции $H_{0,n}(\omega)$ B-сплайна $\varphi_n(x)$:

$$H_{0n}(\omega) = \frac{\hat{\varphi}_n(2\omega)}{\hat{\varphi}_n(\omega)} = e^{-iK\omega/2}(\cos\omega/2)^{n+1}. \tag{20}$$

Замечание 1. Хорошо известно [Чуи], [Дб], что если функция $\varphi(x)$ является B-сплайном, то система ее целочисленных сдвигов $\varphi_{0,n}(x) = \varphi(x - n)$, $n \in \mathbf{Z}$, образует базис Рисса пространства V_0. В работе [Ми] найдены точные границы Рисса A_1 и A_2 в неравенствах (17), (19) раздела 2.3.4 для B-сплайна $\varphi_n(x)$ степени n:

$$A_1 = \frac{2^{2n+2}}{\pi^{2n+2}}\sum_{k=0}^{\infty}\frac{1}{(1+2k)^{2n+2}} \quad \text{и} \quad A_2 = 1.$$

2.6.3. Сплайновые вейвлеты

Кратномасштабный анализ, порожденный B-сплайном $\varphi_n(x)$, строится точно так же, как в рассмотренном выше случае B-сплайна $\varphi_1(x)$

порядка 1. В соответствии с формулой (8) определяются функции $A_n(\omega)$ и $B_n(\omega)$:

$$A_n(\omega)=\frac{1}{\sqrt{\sum\limits_{k\in Z}\left|\hat{\varphi}_n(\omega+2\pi k)\right|^2}} \quad \text{и} \quad B_n(\omega)=\sqrt{\sum\limits_{k\in Z}\left|\hat{\varphi}_n(\omega+2\pi k)\right|^2}.$$

Для B-сплайна $\varphi_n(x)$ любого порядка функция $B_n^2(\omega)$ является положительным тригонометрическим многочленом [Чуи], поэтому можно применить ортогонализацию (7) по той же схеме, что и для B-сплайна первого порядка.

Покажем сначала, как можно вычислить функцию $B_n^2(\omega)$ для любого n. Из формулы (19) получаем:

$$B_n^2(\omega)=\sum_{k\in Z}\left|\hat{\varphi}_n(\omega+2\pi k)\right|^2=\sum_{k\in Z}\left(\frac{2\sin\omega/2}{\omega+2\pi k}\right)^{2(n+1)}=$$

$$=(2\sin\omega/2)^{2(n+1)}\sum_{k\in Z}\frac{1}{(\omega+2\pi k)^{2(n+1)}}=(2\sin\omega/2)^{2(n+1)}\,\Sigma_{2(n+1)}(\omega).$$

В последнем равенстве мы ввели обозначение

$$\Sigma_m(\omega)=\sum_{k\in Z}\frac{1}{(\omega+2\pi k)^m}. \tag{21}$$

Кажется, что такой ряд трудно просуммировать. Однако непосредственно видно, что

$$\Sigma_m(\omega)=\frac{(-1)^{m-2}}{(m-1)!}\frac{d^{m-2}}{d\omega^{m-2}}\Sigma_2(\omega). \tag{22}$$

Выражение для $\Sigma_2(\omega)$ легко находится, поскольку оно соответствует B-сплайну Хаара $\varphi_0(x)$. Как известно, B-сплайн Хаара $\varphi_0(x)$ порождает ортогональный анализ. Поэтому для него

$$B_0^2(\omega)=\sum_{k\in Z}\left|\hat{\varphi}_0(\omega+2\pi k)\right|^2=1.$$

С другой стороны, $B_0^2(\omega)=(2\sin\omega/2)^2\,\Sigma_2(\omega)=1$. Поэтому

$$\Sigma_2(\omega)=\sum_{k\in Z}\frac{1}{(\omega+2\pi k)^2}=\frac{1}{4\sin^2\omega/2}. \tag{23}$$

Таким образом, можно считать функцию $\Sigma_{2n+2}(\omega)$, а с ней и $B_n^2(\omega)$ вычисленной по формулам (22) и (23). Это, в свою очередь, позволяет найти масштабирующую функцию $\hat{\varphi}_n^*(\omega)$ ортогонального кратномасштабного анализа по формуле

$$\hat{\varphi}_n^*(\omega) = \frac{\hat{\varphi}_n(\omega)}{B_n(\omega)} = A_n(\omega)\hat{\varphi}_n(\omega). \tag{24}$$

В пространственной области система сдвигов $\{\varphi_n^*(x-k)\}$ образует ортонормированный базис пространства $V_0 \subset L^2(\mathbf{R})$.

Поскольку $A_n(\omega)$ является тригонометрическим полиномом, возведенным в отрицательную степень 1/2, то его разложение в ряд Фурье будет иметь бесконечное число членов. Это, в свою очередь, показывает, что, в отличие от неортогональной масштабирующей функции $\varphi_n(x)$, ортогонализованная масштабирующая функция $\varphi_n^*(x)$ не будет иметь конечного носителя. Это означает, что в принципе мы должны вычислить бесконечно много коэффициентов, чтобы найти $\varphi_n^*(x)$. Однако практически коэффициенты убывают экспоненциально быстро [Дб] и мы можем усечь ряд при некотором k в зависимости от требуемой точности.

Вейвлеты. Поскольку известна масштабирующая функция $\varphi_n^*(x)$, порождающая ортогональный кратномасштабный анализ, то вейвлет $\psi_n^*(x)$ находится обычным образом (теорема 3 раздела 2.3.2) по формуле

$$\hat{\psi}_n^*(\omega) = e^{i\omega/2}\,\overline{H_{0n}^*\left(\frac{\omega}{2}+\pi\right)}\,\hat{\varphi}_n^*\left(\frac{\omega}{2}\right), \tag{25}$$

в которой все элементы уже можно считать известными. Действительно, масштабирующая функция задана соотношением (24), а частотная функция $H_{0,n}^*(\omega)$ может быть найдена из масштабирующего уравнения $\hat{\varphi}_n^*(\omega) = H_{0,n}^*(\omega/2)\,\hat{\varphi}_n^*(\omega/2)$. Такие вейвлеты (25) называются вейвлетами *Батла–Лемарье* [Ba], [Le], [Ma].

Пример. *B*-сплайн $\varphi_2(x)$ степени 2.

$$\varphi_2(x) = \begin{cases} \frac{1}{2}(x+1)^2, & x \in [-1,0] \\ \frac{3}{4} - \left(x - \frac{1}{2}\right)^2, & x \in [0,1] \\ \frac{1}{2}(x-2)^2, & x \in [1,2] \\ 0, & x \notin [-1,2] \end{cases}, \quad \hat{\varphi}(\omega) = e^{-i\omega/2}\left(\frac{\sin\omega/2}{\omega/2}\right)^3.$$

Масштабирующее соотношение, функция $B_2^2(\omega)$ и вейвлет Батла–Лемарье для *B*-сплайна $\varphi_2(x)$ степени 2 находятся достаточно просто:

$$\varphi_2(x) = \frac{1}{4}\varphi_2(2x+1) + \frac{3}{4}\varphi_2(2x) + \frac{3}{4}\varphi_2(2x-1) + \frac{1}{4}\varphi_2(2x-2),$$

$$B_2^2(\omega)=\sum_{k\in Z}\left|\hat{\varphi}_2(\omega+2\pi k)\right|^2=\frac{8}{15}+\frac{13}{30}\cos\omega+\frac{1}{30}\cos^2\omega,$$

$$\hat{\varphi}^*(\omega)=e^{-i\omega/2}\frac{8\sin^3(\omega/2)}{\omega^3}\left(\frac{8}{15}+\frac{13}{30}\cos\omega+\frac{1}{30}\cos^2\omega\right)^{-1/2}.$$

Для нахождения масштабирующей функции $\varphi^*(x)$ нужно применить обратное преобразование Фурье к $\hat{\varphi}^*(\omega)$. Поскольку $\hat{\varphi}^*(\omega)=A_2(\omega)\hat{\varphi}(\omega)$, то более простой способ нахождения $\varphi^*(x)$ заключается в разложении функции $A_2(\omega)$ в ряд Фурье $A_2(\omega)=\sum_n c_n e^{-in\omega}$. Тогда $\hat{\varphi}^*(\omega)=\sum_n c_n e^{-in\omega}\hat{\varphi}(\omega)$, и поэтому масштабирующая функция задается рядом: $\varphi^*(x)=\sum_n c_n\varphi(x-n)$. В частности, в нашем примере B-сплайна степени 2 имеем: $c_0=1.5855$; $c_1=c_{-1}=-0.3833$; $c_2=c_{-2}=0.1224$; $c_3=c_{-3}=-0.0438$; $c_4=c_{-4}=0.0165$; $c_5=c_{-5}=-0.0064$.

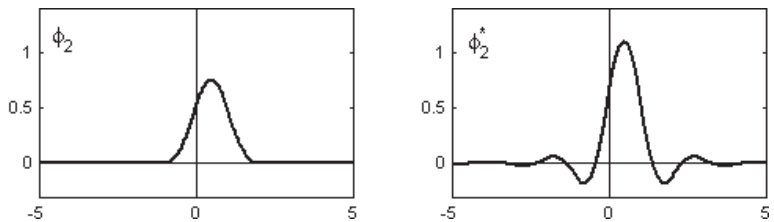


Рис. 2.6.4. B-сплайн степени 2 и его ортогонализация

Вейвлет $\psi^*(x)$ находится обычным образом по масштабирующей функции.

Замечание 1. Вейвлеты Батла–Лемарье являются сплайнами, то есть кусочно-полиномиальными функциями степени n, имеющими непрерывные производные порядка $n-1$ в узлах. Хотя они имеют некомпактный носитель, они убывают достаточно быстро. Экспоненциальное убывание на бесконечности вейвлетов Батла–Лемарье и соответствующих масштабирующих функций показано, например, в книге И. Добеши [Дб], стр. 214–216.

Замечание 2. Дополнительно о суммировании ряда (21). Отметим, что в случае четного $m=2n+2$ сумму ряда $\Sigma_m(\omega)$ можно выразить через многочлен $G_n(y)$ переменной $y=1/\sin^2(\omega/2)$:

$$\sum_{k\in Z}\frac{1}{(\omega+2\pi k)^{2n+2}}=\frac{1}{(2n+1)!}\frac{d^{2n}}{d\omega^{2n}}\left(\frac{1}{4\sin^2\omega/2}\right)=\frac{1}{2^{2n+2}(2n+1)!}G_{n+1}(y).$$

Полиномы $G_n(y)$ найдены З. Ж. Жемаловым [Со], стр. 753, при построении оптимальных квадратурных формул. Коэффициенты многочлена $G_n(y) = \sum_{k=1}^{n} g_k^{(n)} y^k$ могут быть найдены по рекуррентным формулам:

$$g_1^{(n+1)} = -4g_1^{(n)},$$
$$g_k^{(n+1)} = (2k-2)(2k-1)g_{k-1}^{(n)} - 4k^2 g_k^{(n)}, \quad 1 < k \le n,$$
$$g_{n+1}^{(n+1)} = 2n(2n+1)g_n^{(n)}.$$

Укажем несколько первых полиномов:

$$G_1(y) = y, \quad G_2(y) = 6y^2 - 4y, \quad G_3(y) = 120y^3 - 120y^2 + 16y,$$
$$G_4(y) = 5040y^4 - 6720y^3 - 2016y^2 - 64y.$$

В работе [Ми] Мищенко Е. В. также получены выражения для суммы ряда $\Sigma_m(\omega)$ в случае четного $m = 2n + 2$. С точностью до числового сомножителя они представляются в виде $\dfrac{S_n(\sin^2(\omega/2))}{\sin^{2n+2}(\omega/2)}$ или в виде $\dfrac{C\ (\cos^2(\omega/2))}{(1-\cos^2(\omega/2))^{n+1}}$, где $S_n(y)$ и $C_n(y)$ – полиномы степени n, коэффициенты которых находятся по рекуррентным формулам.

2.7. Регулярность и нулевые моменты

В параграфе 2.4 мы привели формулы быстрых алгоритмов вейвлет-преобразования. В них используются фильтры коэффициентов вейвлетов. Естественно, что если вейвлеты имеют некомпактный носитель, то эти фильтры состоят из бесконечного числа коэффициентов. Тогда формулы представляют собой ряды, а не конечные суммы. Удобнее иметь конечные суммы, соответствующие вейвлетам с компактным носителем. Однако пока в нашем распоряжении имеется только один вейвлет Хаара, который обладает этим свойством. Он, к сожалению, имеет разрывы. Поэтому разложение гладких функций по базису вейвлетов Хаара не очень эффективно. Хотелось бы найти другие вейвлеты, которые имеют компактный носитель и были бы достаточно регулярными. Есть еще одно свойство, которое также хотелось бы обеспечить, – это достаточное количество нулевых моментов. В этом параграфе мы обсудим значение этого свойства, а в следующем – рассмотрим построение вейвлетов с компактным носителем и нулевыми моментами.

Определение 1. *Функция* $\psi(x)$ *имеет* N *нулевых моментов, если для любого* $k = 0, ..., N - 1$ *имеют место равенства*

$$\int_{-\infty}^{\infty} x^k \psi(x)\,dx = 0. \qquad (1)$$

Если функция $\psi(x)$ имеет N нулевых моментов, то она ортогональна любому полиному степени не более чем $N-1$. Кроме того, любая функция вида $\psi_{j,k}(x) = \sqrt{2^j}\,\psi(2^j x - k)$ также имеет N нулевых моментов.

Нулевые моменты имеют большое значение в вейвлет-разложении. Дело в том, что если вейвлет $\psi(x)$ имеет N нулевых моментов, то коэффициенты d_{jk} разложения гладкой функции по базису вейвлетов достаточно быстро убывают с ростом j. Это легко показать. Пусть $d_{jk} = (f, \psi_{jk})$. Предположим, что $f(x) \in C^{N-1}(\mathbf{R})$, то есть функция $f(x)$ имеет $N-1$ непрерывных производных на $\mathbf{R}$. Разложим ее по формуле Тейлора в окрестности точки x_0, принадлежащей носителю функции $\psi_{jk}(x)$:

$$f(x) = f(x_0) + f'(x_0)(x - x_0) + \cdots + \frac{f^{(N-1)}(x_0)}{(N-1)!}(x - x_0)^{N-1} + \beta(x)(x - x_0)^{N-1},$$

где $\beta(x) \to 0$ при $x \to x_0$. Поскольку $\psi(x)$ имеет N нулевых моментов, то при вычислении коэффициентов $d_{jk} = (f, \psi_{jk})$ скалярное произведение $\psi(x)$ с полиномом Тейлора будет равно нулю и остается только остаточный член:

$$d_{jk} = (f, \psi_{jk}) = (\beta(x)(x - x_0)^{N-1}, \psi_{jk}(x)).$$

Если $\psi(x)$ имеет компактный носитель, то с ростом j носитель функции $\psi_{j,k}(x) = \sqrt{2^j}\,\psi(2^j x - k)$ становится все меньше (уменьшается в 2^j раз, по сравнению с первоначальным). В формуле Тейлора точка x_0 принадлежит носителю функции $\psi_{jk}(x)$. Учитывая, что остаточный член является бесконечно малой более высокого порядка, чем $(x - x_0)^{N-1}$, получаем, что $d_{jk} = (f, \psi_{jk}) = (\beta(x)(x - x_0)^{N-1}, \psi_{jk}(x))$ будет тем меньше, чем меньше носитель функции $\psi_{jk}(x)$. Это и означает, что коэффициенты d_{jk} убывают с ростом j. Таким образом, вейвлет-коэффициенты для больших j будут преимущественно нулевыми там, где функция гладкая.

Приведем, следуя [Дб], основные теоремы, которые связывают свойство функции иметь нулевые моменты с ее регулярностью и характером убывания на бесконечности.

Теорема 1. *Пусть $f(x), \tilde{f}(x)$ – две функции, не равные тождественно константе, такие что $f(x) \in C^{N-1}(\mathbf{R})$ и производные $f^{(l)}(x)$ ограничены при $l \le N-1$, а для функции $\tilde{f}(x)$ выполнена оценка*

$$|\tilde{f}(x)| \le \frac{C}{(1+|x|)^{\alpha}}, \text{ где } \alpha > N. \tag{2}$$

Предположим, что

$$(f_{i,k}, \tilde{f}_{j,m}) = \delta_{ij}\delta_{km} \quad \forall\, i,j,k,m \in \mathbf{Z}, \tag{3}$$

где $f_{i,k}(x) = \sqrt{2^i}\, f(2^i x - k)$ *и* $\tilde{f}_{j,m}(x) = \sqrt{2^j}\, \tilde{f}(2^j x - m)$. *Тогда*

$$\int_{-\infty}^{\infty} x^n \tilde{f}(x)\, dx = 0, \quad n = 0, 1, \ldots, N-1.$$

Доказательство. Будем доказывать индукцией по n. При $n = 0$ утверждение верно. Оно следует из соотношения $(f_{i,0}, \tilde{f}_{0,0}) = 0$ при $i \to -\infty$ (тогда $f_{i,0} = f(2^i x)$ «распрямляется» при $i \to -\infty$, превращаясь в константу $f(0)$, которую можно считать отличной от нуля). Предположим, что утверждение теоремы верно для $n < l$, и докажем его для $n = l$. Рациональные числа вида $x_{j,k} = k/2^j, j, k \in \mathbf{Z}$ образуют всюду плотное множество. Поскольку производные $f^{(l)}(x)$ непрерывны ($l \le N-1$), то существуют числа J, K, такие что $f^{(l)}(x_{JK}) \ne 0$. В противном случае $f^{(l)}(x) \equiv 0$, тогда либо $f(x) = \text{const}$, что невозможно по предположению, либо $f(x)$ – полином степени $l - 1$, следовательно, неограниченная функция, что также невозможно по условию теоремы.

Разложим $f(x)$ по формуле Тейлора в форме Пеано в окрестности такой точки x_{JK}:

$$f(x) = f(x_{JK}) + f'(x_{JK})(x - x_{JK}) + \cdots + \frac{1}{l!} f^{(l)}(x_{JK})(x - x_{JK})^l + \beta(x)(x - x_{JK})^l,$$

где $\beta(x) \to 0$ при $x \to x_{JK}$. Поэтому для любого $\varepsilon > 0$ существует $\delta > 0$, такое что

$$|\beta(x)(x - x_{JK})^l| \le \varepsilon |x - x_{JK}|^l$$

при $|x - x_{JK}| < \delta$. Тогда

$$|f(x) - T_l(x_{JK}, x - x_{JK})| \le \varepsilon |x - x_{JK}|^l, \tag{4}$$

где $T_l(x_{JK}, x - x_{JK})$ – полином Тейлора.

Пусть теперь $j > J$. Тогда $2^j x_{JK} = 2^j K / 2^J = p$ – целое число. Поэтому можно применить условие (3) теоремы для случая $(f_{0,0}, \tilde{f}_{j,p}) = 0$. Выпишем подробнее это равенство, используя для $f_{00}(x) = f(x)$ разложение Тейлора:

$$0 = \int_{-\infty}^{\infty} f(x)\, \overline{\tilde{f}(2^j x - p)}\, dx = \int_{-\infty}^{\infty} T_l(x_{JK}, x - x_{JK})\, \overline{\tilde{f}(2^j x - 2^j x_{JK})}\, dx +$$

$$+ \int_{-\infty}^{\infty} \left(f(x) - T_l(x_{JK}, x - x_{JK})\right) \overline{\tilde{f}(2^j x - 2^j x_{JK})}\, dx.$$

По индуктивному предположению от первого слагаемого останется только интеграл последнего члена степени l полинома Тейлора. Преобразуем его, сделав замену $y = 2^j x - 2^j x_{JK} = 2^j (x - x^{JK})$,

$$\int_{-\infty}^{\infty} T_l(x_{JK}, x-x_{JK})\,\overline{\tilde{f}(2^j x-2^j x_{JK})}\,dx = \frac{f^{(l)}(x_{JK})}{l!}\int_{-\infty}^{\infty}(x-x_{JK})^l\,\overline{\tilde{f}(2^j x-2^j x_{JK})}\,dx =$$

$$= \frac{1}{l!} f^{(l)}(x_{JK})\frac{1}{2^{j(l+1)}}\int_{-\infty}^{\infty} y^l\,\overline{\tilde{f}(y)}\,dx\,.$$

Используя это выражение в предыдущей формуле, получаем

$$\int_{-\infty}^{\infty} x^l\,\overline{\tilde{f}(x)}\,dx = -C_0\,2^{j(l+1)}\left(\int_{-\infty}^{\infty}\left(f(x)-T_l(x_{JK},x-x_{JK})\right)\overline{\tilde{f}(2^j x-2^j x_{JK})}\,dx\right), \quad (5)$$

где $C_0 = l!(f^{(l)}(x_{JK}))^{-1}$. Оценим интеграл I справа. Поскольку

$$f(x) - T_l(x_{JK}, x - x_{JK}) = \beta(x)(x - x_{JK})^l,$$

то для малых $|x - x_{JK}| < \delta$ можно использовать оценку (4). При остальных, $|x - x_{JK}| > \delta$, будем использовать ограниченность функции $\beta(x)$, которая вытекает из предположения ограниченности функции $f(x)$ и ее производных до порядка l. Действительно, $\beta(x)$ можно записать в виде

$$\beta(x) = \frac{f(x)-f(x_{JK})}{(x-x_{JK})^l} - \frac{f'(x_{JK})}{(x-x_{JK})^{l-1}} - \cdots - \frac{1}{l!} f^{(l)}(x_{JK}).$$

Последнее выражение ограничено в области $|x - x_{JK}| > \delta$. Поэтому при $|x - x_{JK}| > \delta$ имеет место следующая оценка:

$$|f(x) - T_l(x_{JK}, x - x_{JK})| \le C'|x - x_{JK}|^l \le C'(1 + |x - x_{JK}|^l).$$

При оценке интеграла I в правой части (5) разобьем область интегрирования на две области: $|x - x_{JK}| < \delta$ и $|x - x_{JK}| > \delta$. В первой из них будем использовать неравенство (4), а во второй – только что полученную оценку. Сделаем замены $y = x - x_{JK}$ и $t = 2^j y$. Тогда, используя условие (2) теоремы, получаем:

$$|I| \le \varepsilon \int_{|y|<\delta} |y|^l\,|\tilde{f}(2^j y|\,dy + C' \int_{|y|>\delta}(1 + |y|^l)|\tilde{f}(2^j y|\,dy \le$$

$$\le 2\varepsilon C \frac{1}{2^{j(l+1)}}\int_0^{2^j\delta} t^l (1+t)^{-\alpha}\,dt + 2C'C\int_{\delta}^{+\infty}(1+y)^l(1+2^j y)^{-\alpha}\,dy \le$$

$$\le C_1\varepsilon\frac{1}{2^{j(l+1)}} + C_2\frac{1}{2^{j\alpha}}\frac{(1+\delta)^\alpha}{\delta^\alpha},$$

в первом интеграле верхняя граница интегрирования заменена на $+\infty$, а во втором интеграле сначала использована оценка

$$(1+2^j y)^{-1} \le \frac{1+\delta}{1+2^j\delta}(1+y)^{-1} \le \frac{1+\delta}{2^j\delta}(1+y)^{-1},$$

которая вытекает из очевидного неравенства $(1 + y)(1 + 2^j\delta) \le (1 + 2^j y)(1 + \delta)$ при $y \ge \delta$. Затем нижняя граница интегрирования заменена на 0. Из полученных оценок получаем

$$\left|\int_{-\infty}^{\infty} x^l \tilde{f}(x)\,dx\right| \le C_0\left(C_1\varepsilon + C_2\frac{1}{2^{j(\alpha-l-1)}}\frac{(1+\delta)^\alpha}{\delta^\alpha}\right).$$

Здесь $\varepsilon > 0$ можно взять произвольно малым, а при выбранном $\delta > 0$ можно подобрать такое j, что второе слагаемое тоже будет сколь угодно малым. Следовательно, $\int_{-\infty}^{\infty} x^l \tilde{f}(x)\,dx = 0$.

Полагая $f(x) = \tilde{f}(x) = \psi(x)$ в тереме 1, получаем следующее утверждение.

Теорема 2. *Пусть функция $\psi(x)$ класса C^{N-1} имеет ограниченные производные $\psi^{(l)}(x)$ при $l \le N - 1$ и*

$$|\psi(x)| \le \frac{C}{(1+|x|)^{N+\varepsilon}}.$$

Если функции $\psi_{j,k}(x) = \sqrt{2^j}\psi(2^j x - k)$ образуют ортонормированное семейство в $L^2(\mathbf{R})$, тогда

$$\int_{-\infty}^{\infty} x^n \psi(x)\,dx = 0, \quad n = 0, 1, \ldots, N-1.$$

Следствие 1. *Пусть функция $\psi(x)$ класса C^{N-1} имеет компактный носитель. Если функции $\psi_{j,k}(x) = \sqrt{2^j}\psi(2^j x - k)$ образуют ортонормированное семейство в $L^2(\mathbf{R})$, тогда*

$$\int_{-\infty}^{\infty} x^n \psi(x)\,dx = 0, \quad n = 0, 1, \ldots, N-1.$$

Таким образом, ортогональный вейвлет $\psi(x)$ класса C^{N-1} с компактным носителем имеет N нулевых моментов. Отметим, что компактность носителя ортогонального вейвлета $\psi(x)$ препятствует тому, чтобы он был класса C^∞.

Следствие 2. *Предположим, что функции $\psi_{j,k}(x)$ ортонормированы. Тогда функция $\psi(x)$ не может одновременно иметь экспоненциальное убывание, быть класса C^∞ и иметь ограниченными все производные.*

Доказательство. Если $\psi(x)$ имеет экспоненциальное убывание, то $\hat{\psi}(z)$ аналитическая в некоторой полосе $|\mathrm{Im}(z)| < \lambda$. Если функция $\psi(x)$ класса C^∞ и имеет ограниченные производные, то по теореме 2 $\int_{-\infty}^{\infty} x^n \psi(x)dx = 0$ для всех $n \in \mathbf{N}$. Из свойств преобразования Фурье получаем, что производные $\hat{\psi}^{(n)}(0) = 0$ для всех $n \in \mathbf{N}$. Тогда из аналитичности функции $\hat{\psi}(z)$ следует, что $\hat{\psi}(z) = 0$. Поэтому $\psi(x) = 0$, что противоречит ортонормированности системы $\psi_{j,k}(x)$.

Теорема 3. *Предположим, что функция $\psi(x)$ порождает ортонормированный базис пространства вейвлетов W_0 в ортогональном кратномасштабном разложении пространства $L^2(\mathbf{R})$, порожденном функцией $\varphi(x)$. Если $\psi(x)$ класса C^{N-1}, ее производные $\psi^{(l)}(x)$ ограничены при $l \le N-1$ и*

$$|\varphi(x)|, |\psi(x)| \le \frac{C}{(1+|x|)^{N+\varepsilon}},$$

тогда частотная функция $H_0(\omega)$ масштабирующей функции $\varphi(x)$ раскладывается на множители:

$$H_0(\omega) = (1 + e^{-i\omega})^N L(\omega), \qquad (6)$$

где $L(\omega)$ – 2π-периодическая функция класса C^{N-1}.

Доказательство. Из оценок поведения функций $\varphi(x)$ и $\psi(x)$ следует, что их преобразования Фурье $\hat{\varphi}(\omega)$ и $\hat{\psi}(\omega)$ являются функциями класса C^{N-1}. Из теоремы 2 следует, что функция $\psi(x)$ имеет $N-1$ нулевых моментов. Тогда, как и в доказательстве следствия 2, мы получаем $\hat{\psi}^{(n)}(0) = 0$ при $n \le N-1$. Из непрерывности $\hat{\varphi}(\omega)$ следует [Дб, стр. 207, 244], что $\hat{\varphi}(0) = \int_R \varphi(x)dx \ne 0$. С другой стороны,

$$\hat{\psi}(\omega) = e^{i\omega/2}\,\overline{H_0(\omega/2+\pi)}\,\hat{\varphi}(\omega/2).$$

Из этого равенства и из дифференцируемости функций $\hat{\varphi}(\omega)$ и $\hat{\psi}(\omega)$ с учетом того, что $\hat{\varphi}(0) \ne 0$ и $\hat{\psi}^{(n)}(0) = 0$ при $n = 0, 1, \ldots N-1$, следует, что функция $H_0(\omega)$ дифференцируема $N-1$ раз в точке $\omega = \pi$ и имеет нулевые производные:

$$\frac{d^n}{d\omega^n} H_0 \Big|_{\omega=\pi} = 0 \quad \text{для} \quad n \le N-1.$$

Это означает, что функция $H_0(\omega)$ имеет в точке $\omega = \pi$ нуль порядка N. Полагая $z = e^{-i\omega}$, получаем, что в z-представлении степенной ряд $H_0(\omega)$ имеет в точке $z = -1$ нуль порядка N. Поэтому $H_0(z) = (1+z)^N \tilde{L}(z)$, и тогда

$H_0(\omega) = (1 + e^{-i\omega})^N L(\omega)$.

Поскольку $H_0(\omega)$ класса C^{N-1}, то и $L(\omega)$ – тоже класса C^{N-1}.

Теорема 4. *Предположим, что функции $\varphi(x)$ и $\psi(x)$ порождают ортонормированный базис вейвлетов, связанный с кратномасштабным анализом. Если функция $\psi(x)$ имеет N нулевых моментов и*

$$|\varphi(x)|, |\psi(x)| \le \frac{C}{(1+|x|)^{N+\varepsilon}}, \tag{7}$$

тогда коэффициенты $\{h_n\}$ фильтра вейвлета $\varphi(x)$ обладают свойствами:

$$\sum_n h_n = \sqrt{2};$$

$$\sum_n (-1)^n n^k h_n = 0, \quad k = 0, 1, \ldots, N-1.$$

Доказательство. Из оценок (7) поведения функций $\varphi(x)$ и $\psi(x)$ следует, что их преобразования Фурье $\hat{\varphi}(\omega)$ и $\hat{\psi}(\omega)$ являются функциями класса C^{N-1}. Поскольку функция $\psi(x)$ имеет N нулевых моментов, то $\hat{\psi}^{(k)}(0) = 0$ при $k \le N - 1$. Из непрерывности $\hat{\varphi}(\omega)$ следует, что $\hat{\varphi}(0) \ne 0$. Тогда из масштабирующего уравнения $\hat{\varphi}(\omega) = H_0(\omega/2)\,\hat{\varphi}(\omega/2)$ следует, что $H_0(0) = 1$. Из равенства $\hat{\psi}(\omega) = e^{i\omega/2} \times \overline{H_0(\omega/2 + \pi)}\,\hat{\varphi}(\omega/2)$ и из дифференцируемости функций $\hat{\varphi}(\omega)$ и $\hat{\psi}(\omega)$ с учетом того, что $\hat{\varphi}(0) \ne 0$, следует, что функция $H_0(\omega)$ дифференцируема $N - 1$ раз в точке $\omega = \pi$ и все производные нулевые, $H_0^{(k)}(\pi) = 0$ для $k \le N - 1$. Поскольку $H_0(\omega) = \frac{1}{\sqrt{2}} \sum_n h_n e^{-in\omega}$, то

- из $H_0(0) = 1$ получаем $\sum_n h_n = \sqrt{2}$;
- из $H_0^{(k)}(\pi) = 0$ получаем $\sum_n (-1)^n n^k h_n = 0, \quad k = 0, 1, \ldots, N-1$.

2.8. Построение вейвлетов Добеши с компактным носителем

В этом параграфе мы приведем основные конструкции, И. Добеши (см. [Дб], гл. 6), построения вещественных вейвлетов $\varphi(x)$ и $\psi(x)$ с компактным носителем и с N нулевыми моментами, которые определяют ортогональное кратномасштабное разложение. Мы имеем следующее условие ортогональности:

$$\sum_{n\in Z} |\hat{\varphi}(\omega + 2\pi n)|^2 = 1. \qquad (1)$$

И, как следствие, равенство

$$|H_0(\omega)|^2 + |H_0(\omega + \pi)|^2 = 1 \qquad (2)$$

для частотной функции $H_0(\omega) = \sqrt{2^{-1}} \sum_n h_n e^{-i n\omega}$. Сначала мы построим частотную функцию $H_0(\omega)$, удовлетворяющую (2). Тогда масштабирующая функция $\varphi(x)$ находится по формуле

$$\hat{\varphi}(\omega) = \prod_{j=1}^{\infty} H_0\left(\frac{\omega}{2^j}\right)$$

с последующей нормировкой $\varphi(x)$ так, чтобы ее L^2-норма была равна 1. Затем мы установим дополнительные условия на коэффициенты $\{h_n\}$ фильтра вейвлета $\varphi(x)$, достаточные для того, чтобы выполнялось (1).

Из предположения компактности носителя следует, что фильтр коэффициентов $\{h_n\}$ разложения $\varphi(x) = \sum_n h_n \varphi_{1,n}(x)$ состоит из конечного числа ненулевых членов. Если длина носителя $\varphi(x)$ равна N, то имеется не более $N+1$ ненулевых коэффициентов h_n. Из предположения вещественности $\varphi(x)$ следует, что коэффициенты $\{h_n\}$ являются вещественными. Поэтому частотная функция $H_0(\omega) = \sqrt{2^{-1}} \sum_n h_n e^{-i n\omega}$ является тригонометрическим многочленом. Поэтому в условии (2) можно опустить «*п. в.*».

Преобразование Фурье $\hat{\varphi}(\omega)$ является ограничением на $\mathbf{R}$ целой аналитической функции экспоненциального типа. В частности, $\hat{\varphi}(\omega)$ является гладкой класса C^∞. Из непрерывности $\hat{\varphi}(\omega)$ следует [Дб, стр. 207, 244], что $\hat{\varphi}(0) \neq 0$. Тогда из масштабирующего уравнения $\hat{\varphi}(\omega) = H_0(\omega/2)\hat{\varphi}(\omega/2)$ вытекает: $H_0(0) = 1$, а из равенства (2) следует, что $H_0(\pi) = 0$.

Если требовать N нулевых моментов функции $\psi(x)$, то в соответствии с результатами предыдущего параграфа функция $H_0(\omega)$ имеет специальный вид

$$H_0(\omega) = \left(\frac{1 + e^{-i\omega}}{2}\right)^N T(\omega), \qquad (3)$$

где $T(\omega)$ – тригонометрический полином. Кроме того, коэффициенты $\{h_n\}$ фильтра вейвлета $\varphi(x)$ обладают свойствами:

$$\sum_n (-1)^n n^k h_n = 0, \quad k = 0, 1, \ldots, N-1.$$

2.8.1. Построение функции $H_0(\omega)$

Нам нужно найти функцию $H_0(\omega)$ в виде (3), удовлетворяющую (2). Будем искать сначала функцию $M_0(\omega) = |H_0(\omega)|^2$, удовлетворяющую соотношению

$$M_0(\omega) + M_0(\omega + \pi) = 1. \tag{4}$$

Имеем:

$$M_0(\omega) = |H_0(\omega)|^2 = \left(\frac{|1+\cos\omega - i\sin\omega|^2}{4}\right)^N |T(\omega)|^2 =$$

$$= \left(\frac{1+\cos\omega}{2}\right)^N |T(\omega)|^2 = (\cos^2\omega/2)^N L(\omega),$$

где $L(\omega) = |T(\omega)|^2$ – также тригонометрический полином по $\cos\omega$. Действительно, тригонометрический полином $T(\omega)$ по степеням $e^{-i\omega}$ имеет вещественные коэффициенты, поэтому $T(-\omega) = \overline{T(\omega)}$, и тогда $L(\omega) = |T(\omega)|^2 = T(\omega)T(-\omega)$ – четная 2π-периодическая функция, следовательно, полином по $\cos\omega$. Поскольку $1 + \cos\omega = 2\sin^2(\omega/2)$, то его можно записать как полином $P(\sin^2(\omega/2))$. Тогда

$$M_0(\omega) = (\cos^2\omega/2)^N P(\sin^2(\omega/2)) = (1 - \sin^2\omega/2)^N P(\sin^2(\omega/2)).$$

При этом очевидно:

$$M_0(\omega + \pi) = (\sin^2\omega/2)^N P(1 - \sin^2(\omega/2)).$$

Итак, мы ищем функцию $M_0(\omega)$ вида

$$M_0(\omega) = (1 - \sin^2\omega/2)^N P(\sin^2(\omega/2)), \tag{5}$$

удовлетворяющую условию (4). Обозначим $y = \sin^2\omega/2$. Тогда из двух последних соотношений и из (4) получаем

$$(1 - y)^N P(y) + y^N P(1 - y) = 1. \tag{6}$$

Это равенство выполняется для любого $y \in [0, 1]$, следовательно, и для любого $y \in \mathbf{R}$. Для нахождения $P(y)$ из соотношения (6) воспользуемся следующим частным случаем теоремы Безу (см. раздел 1.6 главы 1).

Лемма 1 (Безу). *Если $P_1(y)$ и $P_2(y)$ – полиномы степеней $n \geq 1$ и $m \geq 1$ без общих нулей, то существуют единственные полиномы $Q_1(y)$ и $Q_2(y)$ степени $m - 1$, $n - 1$, такие что*

$$P_1(y)Q_1(y) + P_2(y)Q_2(y) = 1.$$

Используем лемму для многочленов $P_1(y) = (1 - y)^N$, $P_2(y) = y^N$. Тогда существуют единственные полиномы $Q_1(y)$ и $Q_2(y)$ степени меньше N, такие что

$$(1 - y)^N Q_1(y) + y^N Q_2(y) = 1.$$

Сделаем замену $y \to 1 - y$. Тогда $y^N Q_1(1 - y) + (1 - y)^N Q_2(1 - y) = 1$. Из единственности многочленов $Q_1(y)$ и $Q_2(y)$ получаем $Q_2(y) = Q_1(1 - y)$. Тогда

$$(1 - y)^N Q_1(y) + y^N Q_1(1 - y) = 1. \quad (7)$$

Поэтому искомое выражение для $P(y)$ есть $P(y) = Q_1(y)$. Найдем $Q_1(y)$ в явном виде из (7):

$$Q_1(y) = \frac{1}{(1-y)^N}(1 - y^N Q_1(1-y)).$$

Раскладываем первый сомножитель в ряд Тейлора:

$$\frac{1}{(1-y)^N} = \sum_{k=0}^{\infty} \frac{(N+k-1)(N+k-2)\cdots(N+1)N}{k!} y^k = \sum_{k=0}^{\infty} C_{N+k-1}^k y^k,$$

где $C_n^k = n!/(k!(n - k)!)$ – биномиальные коэффициенты. Тогда

$$Q_1(y) = \frac{1}{(1-y)^N}(1 - y^N Q_1(1-y)) = \sum_{k=0}^{N-1} C_{N+k-1}^k y^k + O(y^N) = \sum_{k=0}^{N-1} C_{N+k-1}^k y^k,$$

поскольку степень $Q_1(y)$ не превосходит $N - 1$. Итак, искомое решение $P(y)$ уравнения (6) степени $N - 1$ имеет вид

$$P_{N-1}(y) = \sum_{k=0}^{N-1} C_{N+k-1}^k y^k = 1 + Ny + \frac{N(N+1)}{2!} y^2 + \frac{N(N+1)(N+2)}{3!} y^3 + \cdots. \quad (8)$$

Замечание 1. Мы получили единственное решение $P_{N-1}(y)$ минимальной степени $N - 1$. Если $P(y)$ – решение более высокой степени, то разность $P(y) - P_{N-1}(y)$ – это решение однородного уравнения

$$(1 - y)^N (P(y) - P_{N-1}(y)) + y^N (P(1 - y) - P_{N-1}(1 - y)) = 0. \quad (9)$$

Поскольку сомножитель $(1 - y)^N$ не делится на y^N, то $P(y) - P_{N-1}(y)$ делится на y^N, следовательно, $P(y) - P_{N-1}(y) = y^N R(y)$. Подставляя это выражение в уравнение (9), получаем единственное условие на многочлен $R(y)$:

$$R(y) + R(1 - y) = 0.$$

Последнее условие означает антисимметричность относительно $1/2$, что, в свою очередь, означает, что R является многочленом, со-

держащим нечетные степени $^1/_2 - y$. Тогда решением уравнения (6) является

$$P(y) = \sum_{k=0}^{N-1} C_{N+k-1}^{k} y^k + y^N R(1-2y), \tag{10}$$

где $R(z)$ – многочлен, состоящий из нечетных степеней.

Вывод. Функция

$$H_0(\omega) = \left(\frac{1+e^{-i\omega}}{2} \right)^N T(\omega)$$

удовлетворяет соотношению $|H_0(\omega)|^2 + |H_0(\omega + \pi)|^2 = 1$ тогда и только тогда, когда функция $L(\omega) = |T(\omega)|^2$ является тригонометрическим полиномом вида $L(\omega) = P(\sin^2(\omega/2))$, где $P(y)$ – многочлен вида (8) или (10), в котором $R(z)$ подобран так, что $P(y) \geq 0$ на отрезке $[0, 1]$.

По существу мы пока нашли $M_0(\omega) = |H_0(\omega)|^2$. Для нахождения $H_0(\omega)$ нужно «извлечь квадратный корень» из уравнения $M_0(\omega) = |H_0(\omega)|^2$. Это позволяет сделать следующая

Лемма Рисса. *Пусть $A(\omega)$ – неотрицательный тригонометрический полином вида*

$$A(\omega) = \sum_{k=0}^{N} a_k \cos k\omega, \quad a_k \in \mathbf{R}.$$

Тогда существует тригонометрический полином $B(\omega)$ вида

$$B(\omega) = \sum_{k=0}^{N} b_k e^{ik\omega}, \quad b_k \in \mathbf{R},$$

такой что $|B(\omega)|^2 = A(\omega)$.

Доказательство. Функцию $A(\omega)$ можно представить в виде многочлена от $\cos\omega$, $A(\omega) = p_A(\cos\omega)$ степени N. Поскольку $|B(\omega)|^2 = A(\omega)$, то нам нужно извлечь квадратный корень из $A(\omega)$ для нахождения $B(\omega)$. Если $z = e^{-i\omega}$, то $\cos\omega = (e^{-i\omega} + e^{i\omega})/2 = (z + z^{-1})/2$. Поэтому

$$A(\omega) = p_A(\cos\omega) = p_A\left(\frac{z+z^{-1}}{2} \right) = P_A(z).$$

Последний многочлен комплексного переменного z имеет вещественные коэффициенты и $p(z) = p(z^{-1})$, поэтому его комплексные корни возникают четверками $\{z_j, \bar{z}_j, z_j^{-1}, \bar{z}_j^{-1}\}$, а вещественные – парами $\{r_j, r_j^{-1}\}$. Так что мы можем записать

$$P_A(z) = c z^{-N} \prod_j (z-z_j)(z-\bar{z}_j)(z-z_j^{-1})(z-\bar{z}_j^{-1}) \prod_k (z-z_k)^2 (z-\bar{z}_k)^2 \prod_l (z-r_l)(z-r_l^{-1}),$$

где особо выделен случай, когда четверка вырождается в кратную пару степени 2 при $z_k = e^{ia_k} = \bar{z}_k^{-1}$. На единичном круге $z = e^{-i\omega}$ мы имеем:

$$|(e^{i\omega} - z_0)(e^{i\omega} - \bar{z}_0^{-1})| = |z_0|^{-1}|e^{i\omega} - z_0|^2.$$

Следовательно,

$$A(\omega) = |A(\omega)| = |P_A(e^{-i\omega})| =$$

$$= |c| \left| \prod_j \frac{1}{|z_j|}(e^{-i\omega} - z_j)(e^{-i\omega} - \bar{z}_j) \right|^2 \left| \prod_k (e^{-i\omega} - e^{ia_k})(e^{-i\omega} - e^{-ia_k}) \right|^2 \prod_l \frac{1}{|r_l|} |e^{-i\omega} - r_l|^2 = |B(\omega)|^2.$$

Теперь в качестве $B(\omega)$ берем

$$B(\omega) = |c|^{1/2} \prod_j \frac{(e^{-i\omega} - z_j)(e^{-i\omega} - \bar{z}_j)}{|z_j|} \prod_k (e^{-i\omega} - e^{ia_k})(e^{-i\omega} - e^{-ia_k}) \prod_l \frac{(e^{-i\omega} - r_l)}{|r_l|^{1/2}}. \quad (11)$$

Легко видеть, что полученный многочлен от переменной $z = e^{-i\omega}$ имеет вещественные коэффициенты.

Замечание 2. Многочлен $B(\omega)$ находится по многочлену $A(\omega)$ неоднозначно, например можно умножить $B(\omega)$ на $e^{ik\omega}$, где k – любое целое. Другие возможности вытекают из неоднозначности выбора первого корня z_j из четверки корней $\{z_j, \bar{z}_j, z_j^{-1}, \bar{z}_j^{-1}\}$.

Замечание 3. Указанная в доказательстве процедура нахождения $B(\omega)$ из равенства $|B(\omega)|^2 = A(\omega)$ называется *спектральной факторизацией*. «Извлечение квадратного корня» из $A(\omega)$ заключается в выборе половины корней $A(\omega)$ и образовании $\prod_j (z - z_j)$, где j пробегает все индексы выбранных корней.

Четверка комплексных корней $\{z_j, \bar{z}_j, z_j^{-1}, \bar{z}_j^{-1}\}$ и пара вещественных $\{r_j, r_j^{-1}\}$, возникающие при спектральной факторизации, инвариантны при инверсиях относительно единичной окружности (см. рис. 2.8.1).

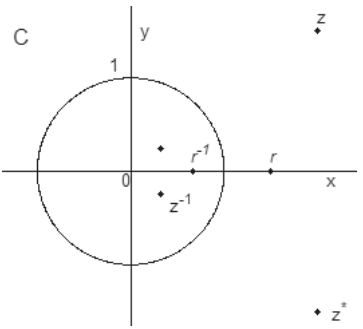


Рис. 2.8.1. Симметрия корней

Для каждой четверки комплексных корней $\{z_j, \bar{z}_j, z_j^{-1}, \bar{z}_j^{-1}\}$ мы выбираем пару $\{z_j, \bar{z}_j\}$ или $\{z_j^{-1}, \bar{z}_j^{-1}\}$, такую что оба корня лежат либо внутри, либо снаружи единичного круга на комплексной плоскости. Хотя существуют шесть способов выбора двух корней из четырех, легко видеть, что другие комбинации двух корней не будут давать разложения на

множители с действительными коэффициентами. Из каждой пары $\{r_j, r_j^{-1}\}$ действительных корней мы выбираем одну любую внутри или снаружи единичного круга. Среди корней на единичном круге мы выбираем один корень из пары вырожденных корней.

В нашем случае ортогонального вейвлета $\varphi(x)$ с компактным носителем частотная функция $H_0(\omega)$ является тригонометрическим полиномом, удовлетворяющим соотношению $|H_0(\omega)|^2 + |H_0(\omega + \pi)|^2 = 1$. Если требовать N нулевых моментов функции $\psi(x)$, то функция $H_0(\omega)$ имеет специальный вид

$$H_0(\omega) = \left(\frac{1+e^{-i\omega}}{2}\right)^N T(\omega),$$

где $T(\omega)$ – тригонометрический полином. Полагаем $M_0(\omega) = |H_0(\omega)|^2$, тогда $M_0(\omega) = (1 - \sin^2\omega/2)^N P(\sin^2(\omega/2))$, где многочлен $P(y)$ (минимальной) степени $N - 1$ определяется формулой (8). Функция $H_0(\omega)$ находится спектральной факторизацией многочлена $M_0(\omega)$ учетом того, что $\sin^2\omega/2 = (1 - \cos\omega)/2 = (1 - (z + z^{-1})/2)/2$, где $z = e^{-i\omega}$. Тогда $H_0(\omega)$ задается в виде:

$$H_0(\omega) = \left(\frac{1+z}{2}\right)^N \prod_{j=1}^{N-1}(z - z_j),$$

где j пробегает индексы всех выбранных корней.

Пример. Вейвлет Добеши db2. Рассмотрим наиболее простой нетривиальный случай, когда $N = 2$, а тригонометрический полином T имеет степень 1. В этом случае мы потребовали двух нулевых моментов для функции $\psi(x)$. Поскольку тригонометрический многочлен $H_0(\omega)$ имеет степень 3, то фильтр коэффициентов $\{h_n\}$ будет иметь 4 ненулевых элемента, а носитель будет иметь длину 3. Учитывая возможность сдвига носителя, потребуем, чтобы функция $\varphi(x)$ имела бы носитель на промежутке $[0, 3]$. Частотная функция $H_0(\omega)$ является тригонометрическим полиномом степени 3. Из равенства нулю двух моментов имеем:

$$H_0(\omega) = \left(\frac{1+e^{-i\omega}}{2}\right)^2 T(\omega),$$

где $T(\omega)$ – тригонометрический полином степени 1. Рассмотрим функцию $M_0(\omega) = |H_0(\omega)|^2$. По формуле (8) многочлен P имеет вид $P(y) = 1 + 2y$. Тогда

$$M_0(\omega) = (1 - \sin^2\omega/2)^2 P(\sin^2(\omega/2)) = (\cos^2\omega/2)^2(1 + 2\sin^2(\omega/2)).$$

$$M_0(\omega)=\left(\frac{1+(z+z^{-1})/2}{2}\right)^2\left(1+2\frac{1-(z+z^{-1})/2}{2}\right)=$$

$$=\frac{1}{32}(2+z+z^{-1})^2(4-z-z^{-1})=\frac{-1}{32z^3}(z^2+2z+1)^2(z^2-4z+1)=$$

$$=\frac{-1}{32z^3}(z+1)^4(z-(2+\sqrt{3}))(z-(2-\sqrt{3})).$$

По формуле (11) имеем для вещественного корня $r_1 = 2 + \sqrt{3}$ вне единичного круга

$$H_0(\omega)=\frac{1}{\sqrt{32}}\frac{1}{\sqrt{2+\sqrt{3}}}(z+1)^2(z-(2+\sqrt{3})).$$

Поскольку

$$\frac{1}{\sqrt{2+\sqrt{3}}}=\frac{\sqrt{2}}{1+\sqrt{3}} \quad \text{и} \quad \sqrt{2+\sqrt{3}}=\frac{1+\sqrt{3}}{\sqrt{2}},$$

то

$$H_0(\omega)=\frac{1}{4\sqrt{2}}(z+1)^2\left(\frac{\sqrt{2}}{1+\sqrt{3}}z-\frac{1+\sqrt{3}}{\sqrt{2}}\right)=$$

$$=\frac{1}{4\sqrt{2}}(z+1)^2\left(\frac{(1-\sqrt{3})\sqrt{2}\sqrt{2}}{\sqrt{2}(1-3)}z-\frac{1+\sqrt{3}}{\sqrt{2}}\right)=-\frac{1}{8}(z+1)^2(1+\sqrt{3}+(1-\sqrt{3})z).$$

Учитывая, что знак «минус» можно убрать, и полагая $z = e^{-i\omega}$, окончательно получаем:

$$H_0(\omega)=\frac{1}{8}(1+e^{-i\omega})^2(1+\sqrt{3}+(1-\sqrt{3})e^{-i\omega}).$$

Раскрывая скобки и учитывая $\sqrt{2}$ в выражении функции $H_0(\omega)$, находим следующие коэффициенты:

$$h_0=\frac{1+\sqrt{3}}{4\sqrt{2}}, \quad h_1=\frac{3+\sqrt{3}}{4\sqrt{2}}, \quad h_2=\frac{3-\sqrt{3}}{4\sqrt{2}}, \quad h_3=\frac{1-\sqrt{3}}{4\sqrt{2}}.$$

Вейвлет-функция $\varphi(x)$ строится через бесконечное произведение. Она называется вейвлетом Добеши порядка 2 и обозначается символом «db2». На рис. 2.8.2 приведены графики вейвлетов Добеши db2.

Упражнение. Найти фильтр коэффициентов $\{h_n\}$ вейвлета Добеши db2, используя только свойства коэффициентов:

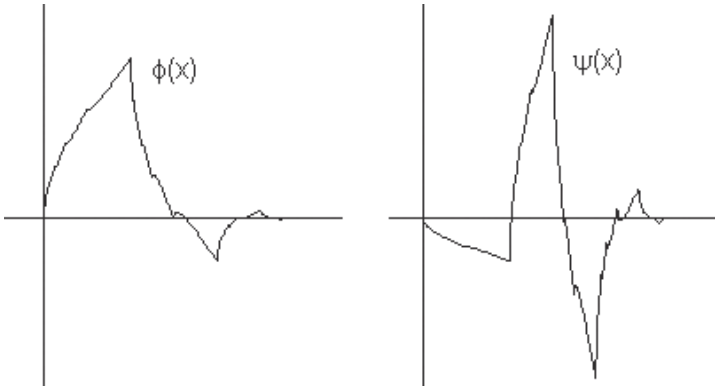


Рис. 2.8.2. Вейвлеты Добеши $\varphi(x)$ и $\psi(x)$

- ортонормированность $\sum_k h_k h_{k-2m} = \delta_{0,m}$;
- нормировка $H_0(0) = 1, \sum_n h_n = \sqrt{2}$;
- нулевые моменты $\sum_n (-1)^n n^k h_n = 0, \quad k = 0, 1.$

2.8.2. Симлеты

При нахождении частотной функции $H_0(\omega)$ ортогонального вейвлета $\varphi(x)$ с компактным носителем используется процедура спектральной факторизации, «извлечения квадратного корня» из $M_0(\omega) = |H_0(\omega)|^2$. Как уже отмечалось, $H_0(\omega)$ находится по многочлену $M_0(\omega)$ неоднозначно, например можно умножить $H_0(\omega)$ на e^{ib}, где b – любое число. Имеется также произвол в выборе половины корней многочлена $M_0(\omega)$. Эти различные выборы корней приводят к различным фазам функции $H_0(\omega)$ и, в свою очередь, к различным коэффициентам фильтра и вейвлетам, все из которых удовлетворяют условию ортогональности и условию нулевых моментов. В случае вейвлетов Добеши используется систематическое удержание нулей внутри единичной окружности.

Для практических целей (например, в обработке изображений) хотелось бы иметь фильтры с некоторыми свойствами симметрии. Легко видеть, что если действительный фильтр $\{h_n\} = \{h_0, h_1, \ldots, h_N\}$ является симметричным относительно центрального коэффициента, то есть $h_k = h_{N-k}$, то

$$H_0(\omega) = \frac{1}{\sqrt{2}} \sum_{n=0}^{N} h_n e^{-in\omega} = e^{-iN\omega/2} \sqrt{2} \left(h_0 \cos\frac{N}{2}\omega + h_1 \cos\left(\frac{N}{2} - 1\right)\omega + \cdots \right).$$

Если действительный фильтр $\{h_n\} = \{h_0, h_1, ..., h_N\}$ является антисимметричным относительно центрального коэффициента, то есть $h_k = -h_{N-k}$, то

$$H_0(\omega) = \frac{1}{\sqrt{2}} \sum_{n=0}^{N} h_n e^{-i n\omega} = e^{-iN\omega/2} \sqrt{2} i \left(h_0 \sin \frac{N}{2}\omega + h_1 \sin\left(\frac{N}{2} - 1\right)\omega + \cdots \right).$$

В обоих случаях частотная функция $H_0(\omega) = e^{i\phi\omega}|H_0(\omega)|$ имеет фазу, которая является линейной по ω, $\phi(\omega) = a\omega + b$. Мы называем такие фильтры фильтрами *с линейной фазой*.

Как известно, кроме системы Хаара, никакая система функций $\psi(x)$ и $\varphi(x)$ не может одновременно иметь компактный носитель и быть симметричной. Однако можно попробовать приблизиться, насколько возможно, к симметрии. Для симметричных вейвлетов фаза частотной функции $H_0(\omega)$ нулевая. Поэтому можно потребовать, чтобы фаза $H_0(\omega)$ была минимальной среди всех $H_0(\omega)$ с тем же самым значением $|H_0(\omega)|$. Это требование определяет некоторый выбор тригонометрического полинома $T(\omega)$. Такие вейвлеты, полученные из вейвлетов Добеши, называются *симлетами*.

Идея построения симлетов состоит в том, чтобы, выбирая нули в формуле (11), получить наименее асимметричный вейвлет. Для этого нужно научиться вычислять фазу функции $H_0(\omega)$ в зависимости от выбора корней.

Если игнорировать линейную фазу, то фаза может быть вычислена следующим образом. Поскольку $H_0(\omega)$ является произведением сомножителей вида $(z - z_j)(z - \bar{z}_j)$ или $(z - r_j)$, полная фаза есть сумма фазовых вкладов каждого сомножителя. Для коэффициента вида $(z - z_j)(z - \bar{z}_j)$, где $z = e^{-i\omega}$ и $z_j = R_j e^{ia_j}$, мы имеем

$$(z - z_j)(z - \bar{z}_j) = e^{i\omega}(1 - e^{-i\omega} R_j e^{ia_j})(e^{i\omega} - R_j e^{-ia_j}) = e^{i\omega}(e^{i\omega} - 2R_j \cos a_j + R_j^2 e^{-i\omega}),$$

и соответствующая фаза, игнорируя линейный вклад, имеет вид

$$\phi_j(\omega) = \arctan\left(\frac{(1 - R_j^2)\sin\omega}{(1 + R_j^2)\cos\omega - 2R_j \cos a_j} \right).$$

Аналогично для коэффициента вида $(z - r_j)$, где r_j является действительным, мы имеем

$$(z - r_j) = e^{i\omega} - r_j = e^{i\omega/2}(e^{i\omega/2} - r_j e^{-i\omega/2}),$$

и соответствующая фаза, снова игнорируя линейный вклад:

$$\phi_j(\omega) = \arctan\left(\frac{1 + r_j}{1 - r_j} \tan\frac{\omega}{2} \right).$$

Отклонение фазы $H_0(\omega)$ от линейной фазы определяется суммой разовых фазовых отклонений.

Для нахождения наименее асимметричного фильтра необходимо найти все корни. Выбирая пары корней либо внутри единичного круга, либо вне круга, мы строим $H_0(\omega)$ с минимальной фазой. Корни, лежащие на единичном круге, имеют линейную фазу.

Замечание 5. Для вейвлета Добеши малого порядка недостаточно свободы для различных вариантов выбора корней, тогда симлеты совпадают с вейвлетами Добеши. Различия наблюдаются с порядка $N = 4$. См. рис. 2.8.3 и 2.8.4.



Рис. 2.8.3. Графики вейвлетов $\varphi(x)$ и $\psi(x)$ Добеши db4

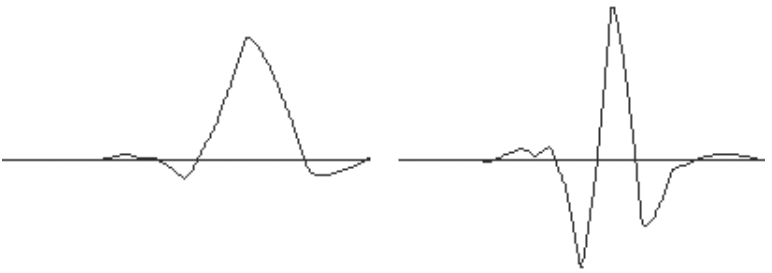


Рис. 2.8.4. Графики $\varphi(x)$ и $\psi(x)$ симлетов sym4

2.9. Койфлеты

В разделе 2.7 мы видели, что нулевые моменты функции $\psi(x)$ приводят к убыванию вейвлет-коэффициентов $d_{jk} = (f, \psi_{jk})$ с ростом j там, где функция гладкая. Однако это не касается коэффициентов $a_{jk} = (f, \varphi_{jk})$. Если потребовать выполнения следующих нулевых моментов:

$$\int_{-\infty}^{\infty} x^k \varphi(x)\,dx = 0\,, \quad k = 1, \ldots, N-1, \tag{1}$$

тогда можно применить те же рассуждения (см. раздел 2.7), связанные с разложением по формуле Тейлора. А именно, раскладывая $f(x)$ по формуле Тейлора

$$f(x) = f(x_0) + f'(x_0)(x - x_0) + \cdots + \frac{f^{(N-1)}(x_0)}{(N-1)!}(x - x_0)^{N-1} + \beta(x)(x - x_0)^{N-1}$$

в окрестности точки x_0, принадлежащей носителю функции $\varphi_{jk}(x)$, вследствие (1) получаем

$$a_{jk} = (f, \varphi_{jk}) = (f(x_0), \varphi_{jk}) + (\beta(x)(x - x_0)^{N-1}, \varphi_{jk}(x)) \approx$$

$$\approx \int_R f(x_0)\sqrt{2^j}\,\varphi(2^j x - k)\,dx = f(x_0)\sqrt{2^j}\int_R \varphi(2^j x - k)\,dx = \frac{1}{\sqrt{2^j}} f(x_0)\,.$$

В последнем равенстве мы считаем, что $\int_{-\infty}^{\infty} \varphi(x)\,dx = 1$. Предположим, что носитель функции $\varphi(x)$ сосредоточен вблизи точки $x_0 = 0$. Тогда носитель функции $\varphi_{jk}(x) = \sqrt{2^j}\,\varphi(2^j x - k)$ сосредоточен вблизи точки $x_{j,k} = k/2^j$ и будет в 2^j раз у́же, чем носитель $\varphi(x)$. Поэтому для нахождения коэффициента $a_{jk} = (f, \varphi_{jk})$ следует брать разложение по формуле Тейлора в окрестности точки $x_0 = x_{j,k} = k/2^j$. Тогда при больших J для гладкой функции $f(x)$ мы имеем приближенное равенство

$$a_{Jk} = (f, \varphi_{Jk}) \approx \frac{1}{\sqrt{2^J}} f(2^{-J} k). \tag{2}$$

Последнее дает эффективное правило для перехода от выборки, представляющей $f(x)$, к коэффициентам cA_0 мелкого масштаба. Поэтому важно иметь не только нулевые моменты функции $\psi(x)$, но и функции $\varphi(x)$. Поскольку вопрос о таких функциях поставлен впервые Р. Койфманом, то подобные вейвлеты называются *койфлетами*.

Нашей целью является нахождение таких функций $\psi(x)$ и $\varphi(x)$ с компактными носителями, что

$$\int_{-\infty}^{\infty} x^k \psi(x)\,dx = 0\,, \quad k = 0, \ldots, N-1, \tag{3}$$

$$\int_{-\infty}^{\infty} \varphi(x)\,dx = 1, \quad \int_{-\infty}^{\infty} x^k \varphi(x)\,dx = 0\,, \quad k = 1, \ldots, N-1. \tag{4}$$

Число N называется порядком койфлета. Как мы видели ранее,

свойства (3) и (4) эквивалентны таким:

$$\hat{\psi}^{(k)}(0) = 0, \quad k = 0, 1, \ldots, N-1. \tag{5}$$

$$\hat{\varphi}(0) = 1, \quad \hat{\varphi}^{(k)}(0) = 0, \quad k = 1, \ldots, N-1. \tag{6}$$

Их условий (5) и теоремы 3 раздела 2.7 следует, что для койфлетов с компактным носителем функция $H_0(\omega)$ имеет вид

$$H_0(\omega) = \left(\frac{1+e^{-i\omega}}{2}\right)^N T(\omega), \tag{7}$$

где $T(\omega)$ является тригонометрическим полиномом.

Из масштабирующего уравнения $\hat{\varphi}(\omega) = H_0(\omega/2)\hat{\varphi}(\omega/2)$ и из (6) следует, что

$$H_0(0) = 1, H_0^{(k)}(0) = 0, k = 1, ..., N-1. \tag{8}$$

Это означает, что функция $H_0(\omega) - 1$ имеет в точке $\omega = 0$ нуль порядка N. Полагая $z = e^{-i\omega}$, получаем, что в z-представлении функция $H_0(\omega)$ имеет в точке $z = 1$ нуль порядка N. Поэтому $H_0(z) - 1 = (1-z)^{N-1}S(z)$ и

$$H_0(\omega) = 1 + (1 - e^{-i\omega})^N S(\omega), \tag{9}$$

где $S(\omega)$ есть тригонометрический полином. Кроме того, должно еще выполняться равенство

$$|H_0(\omega)|^2 + |H_0(\omega + \pi)|^2 = 1. \tag{10}$$

Построение койфлетов использует, как и в случае вейвлетов Добеши, теорему Безу. Рассмотрим для простоты случай четного $N = 2K$. Поскольку

$$\left(\frac{1+e^{-i\omega}}{2}\right)^{2K} = e^{-i\omega K}(\cos^2\omega/2)^K, \quad (1-e^{-i\omega})^{2K} = e^{-i\omega K}(2i\sin\omega/2)^{2K},$$

то из (7) и (9) получаем

$$(\cos^2\omega/2)^K P_1(\omega) = 1 + (\sin\omega/2)^K P_2(\omega), \tag{11}$$

где $P_1(\omega)$ и $P_2(\omega)$ – тригонометрические полиномы. Они могут быть найдены по теореме Безу, как в разделе 2.8.1. В частности, $P_1(\omega)$ имеет вид

$$P_1(\omega) = \sum_{k=0}^{K-1} C_{K-1+k}^k (\sin^2\omega/2)^k + (\sin^2\omega/2)^K R(\omega), \tag{12}$$

где $R(\omega)$ – произвольный тригонометрический полином. Этот произвол будет использоваться для удовлетворения нужного тож-

дества (10). Полагая $R(\omega) = \sum_{n=0}^{2K-1} R_n e^{-in\omega}$ и подставляя $H_0(\omega) = ((1 + e^{-i\omega})/2)^{2K} P_1(\omega)$ в соотношение (10), получаем систему квадратных уравнений на неопределенные коэффициенты R_n. Подробнее об этом см. в книге [Дб].

Замечание. Вейвлеты с частотной функцией вида

$$H_0(\omega) = \left(\frac{1+e^{-i\omega}}{2}\right)^{2K} P_1(\omega),$$

где $P_1(\omega)$ имеет вид (12), называются *койфлетами порядка K*. Отметим, что койфлеты несимметричны, но значительно симметричнее вейвлетов Добеши и кроме нулевых $N = 2K$ моментов (3) и (4) имеют длину носителя $6K - 1$.

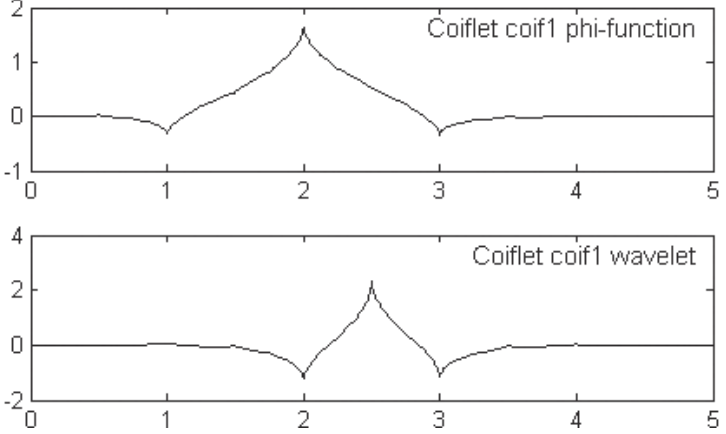


Рис. 2.9.1. Графики койфлетов coif1

2.10. Биортогональные вейвлеты

Свойство ортогональности вейвлетов $\varphi_{j,n}$ и $\psi_{j,n}$ является не единственным хорошим свойством. Во многих вопросах желательна симметричность. Однако не существует базиса симметричных вейвлетов ортогональных и с компактным носителем, кроме вейвлета Хаара. Следовательно, если мы хотим получить гладкие симметричные вейвлеты с компактным носителем, то нужно, наверное, пожертвовать ортогональностью. Но тогда возникает проблема с восстановлением сигнала $f(x)$ по его коэффициентам. Рассмотрим подробнее этот вопрос.

2.10.1. Мотивировка и определение

Пусть $\{\psi_{j,n}(x)\}$ – вейвлет-базис пространства $L^2(\mathbf{R})$, который не является ортогональным. Для функции $f(x) \in L^2(\mathbf{R})$ найдем коэффициенты

$$d_{j,n} = (f, \psi_{j,n}) = \int_R f(x)\overline{\psi_{j,n}(x)}dx, \quad j,n \in \mathbf{Z}. \tag{1}$$

Если мы хотим восстановить функцию $f(x)$ по вычисленным коэффициентам, то не можем надеяться на равенство

$$f(x) = \sum_{i,k\in\mathbf{Z}} d_{i,k}\psi_{i,k}, \tag{2}$$

поскольку функции $\{\psi_{j,n}(x)\}$ не ортонормированы и даже не ортогональны. Действительно, если предположить формулу (2), то получаем: $(f, \psi_{j,n}) = \sum_{i,k\in\mathbf{Z}} d_{i,k}(\psi_{i,k}, \psi_{j,n}) \neq d_{j,n}$. Можно применить ортогонализацию базиса, описанную ранее, но тогда вейвлеты $\psi_{j,n}(x)$ потеряют ряд хороших свойств. Например, после ортогонализации B-сплайнов получаются вейвлеты Батла–Лемарье с некомпактным носителем.

Проблема восстановления функции $f(x)$ по ее коэффициентам $d_{j,n}$ может быть решена с использованием дуального базиса в $L^2(\mathbf{R})$ следующим образом. Рассмотрим сначала случай конечномерного евклидова пространства V. Возьмем базис (не ортогональный) $e_1, e_2, ..., e_n$ в пространстве V. Для любого вектора $f \in V$ пусть

$c_i = (f, e_i), \quad i = 1, 2, ..., n.$

Мы хотим восстановить f по его коэффициентам c_i. Поскольку базис не ортонормированный, то c_i – это не коэффициенты разложения по базису $\{e_i\}$, поэтому $f \neq \sum_{i=1}^n c_i e_i$. Попробуем найти другой базис $\tilde{e}_1$, ..., $\tilde{e}_n$, в котором

$$f = \sum_{j=1}^n c_j \tilde{e}_j. \tag{3}$$

Если такой базис $\tilde{e}_j$ существует, то мы должны иметь следующее (берем скалярное произведение с вектором e_i обеих частей последнего равенства (3)):

$$c_i = (f, e_i) = (\sum_{j=1}^n c_j \tilde{e}_j, e_i) = \sum_{j=1}^n c_j (e_i, \tilde{e}_j)$$

для любого $i = 1, 2, ..., n$. Это имеет место только тогда, когда $(e_i, \tilde{e}_j) = \delta_{ij}$. Из линейной алгебры известно, что базис $\tilde{e}_j$, удовлетворяющий

соотношениям $(e_i, \tilde{e}_j) = \delta_{ij}$, существует. Для его построения мы сначала определяем дуальный базис линейных функционалов на V, то есть таких функционалов $\tilde{e}^j$: $V \to \mathbf{R}$, что

$$\tilde{e}^j(e_i) = \delta_{ij}, \quad \forall\, i, j = 1, 2, \ldots, n.$$

Далее, поскольку наше пространство V является гильбертовым, то по теореме Рисса каждый функционал $\tilde{e}^j$: $V \to \mathbf{R}$ задается как скалярное произведение с некоторым вектором $\tilde{e}_j$. Полученные векторы $\tilde{e}_j$ и составляют искомый дуальный базис пространства V. Развивая эти идеи, мы приходим к так называемым биортогональным вейвлетам.

Определение 1. *Биортогональные вейвлеты – это две пары функций* $\varphi(x)$, $\psi(x)$ *и* $\tilde{\varphi}(x)$, $\tilde{\psi}(x)$, *такие что:*

1) *функции* $\varphi(x)$ *и* $\tilde{\varphi}(x)$ *являются масштабирующими и порождают кратномасштабные разложения* $L^2(\mathbf{R})$

$$\ldots \subset V_{-1} \subset V_0 \subset V_1 \subset \ldots \subset V_j \subset \ldots \quad \text{и} \quad \ldots \subset \tilde{V}_0 \subset \tilde{V}_1 \subset \ldots \subset \tilde{V}_j \subset \ldots;$$

2) *функции* $\psi_{j,n}(x) = \sqrt{2^j}\,\psi(2^j x - n)$ *порождают базис Рисса пространства* W_j, *дополнительного к* V_j *в пространстве* V_{j+1}, *а функции* $\tilde{\psi}_{j,n}(x) = \sqrt{2}\tilde{\psi}(2^j x - n)$ *образуют базис Рисса пространства* $\tilde{W}_j$, *дополнительного к* $\tilde{V}_j$ *в пространстве* $\tilde{V}_{j+1}$;

3) *функции* $\{\tilde{\varphi}_{i,n}(x)\}$ *и* $\{\tilde{\psi}_{j,n}(x)\}$ *образуют дуальные базисы для систем функций* $\{\varphi_{j,n}(x)\}$ *и* $\{\psi_{j,n}(x)\}$. *Другими словами, для любых целых* j, k, l, n *выполняются равенства*

$$\begin{aligned} &(\varphi_{j,n}, \tilde{\varphi}_{j,k}) = \delta_{nk}, && (\varphi_{j,n}, \tilde{\psi}_{jk}) = 0, \\ &(\psi_{j,n}, \tilde{\varphi}_{j,k}) = 0, && (\psi_{j,n}, \tilde{\psi}_{l,k}) = \delta_{jl}\delta_{nk}. \end{aligned} \tag{4}$$

Отметим, что в определении не требуется, чтобы $\tilde{\varphi}(x) \in V_0$ и $\tilde{\psi}(x) \in W_0$. Эти функции будут порождать, возможно, другую последовательность подпространств $\ldots \subset \tilde{V}_0 \subset \tilde{V}_1 \subset \ldots \subset \tilde{V}_j \subset \ldots$ и разложений $\tilde{V}_{j+1} = \tilde{V}_j \oplus \tilde{W}_j$. Условия (4) означают, что

$$V_j \perp \tilde{W}_j, \quad \tilde{V}_j \perp W_j, \quad W_j \perp \tilde{W}_l \quad \text{при } j \neq l.$$

Пусть

$$\varphi(x) = \sqrt{2}\sum\nolimits_n h_n \varphi(2x - n) \quad \text{и} \quad \psi(x) = \sqrt{2}\sum\nolimits_n g_n \varphi(2x - n)$$

– масштабирующие соотношения для функций $\varphi(x)$ и $\psi(x)$. При использовании биортогональных вейвлетов разложение сигнала $\{a_n\}$ производится при помощи фильтров $\{h_n\}$ и $\{g_n\}$ функций $\varphi(x)$ и $\psi(x)$ по формулам

$$a_{1,k} = \sum_n \overline{h}_n a_{j,n+2k}, \quad d_k = \sum_n \overline{g}_n a_{j,n+2k}.$$

Восстановление сигнала производится при помощи фильтров дуальных функций $\tilde{\varphi}(x)$ и $\tilde{\psi}(x)$. Если

$$\tilde{\varphi}(x) = \sqrt{2}\sum_n \tilde{h}_n \tilde{\varphi}(2x-n) \quad \text{и} \quad \tilde{\psi}(x) = \sqrt{2}\sum_n \tilde{g}_n \tilde{\varphi}(2x-n)$$

– масштабирующие соотношения для функций $\tilde{\varphi}(x)$ и $\tilde{\psi}(x)$, то восстановление сигнала производится по формуле

$$a_n = \sum_k (\tilde{h}_{n-2k} a_{1,k} + \tilde{g}_{n-2k} d_{1,k}).$$

В случае ортогонального кратномасштабного анализа функция $\psi(x)$ может быть найдена через масштабирующую функцию $\varphi(x)$, а дуальные функции совпадают с исходными: $\tilde{\varphi}(x) = \varphi(x)$ и $\tilde{\psi}(x) = \psi(x)$.

В общем случае вопрос о построении биортогональных вейвлетов $\varphi(x)$, $\psi(x)$ и $\tilde{\varphi}(x)$, $\tilde{\psi}(x)$ является достаточно сложным, см. [Чуи] стр. 221–276 и [Дб] стр. 348–378. В основе лежит знакомый уже метод нахождения фильтров восстановления $\tilde{H}_0(\omega)$и $\tilde{G}_0(\omega)$.

Хорошо известно [Чуи], [Дб], что если функция $\varphi(x)$ является B-сплайном, то система ее целочисленных сдвигов $\varphi_{0,n}(x) = \varphi(x-n)$, $n \in \mathbf{Z}$, образует базис Рисса пространства V_0. Тогда из теорем 8 и 9 раздела 2.3.4 следует, что B-сплайн $\varphi(x)$ порождает неортогональное кратномасштабное разложение $... \subset V_0 \subset V_1 \subset ... \subset V_j \subset ...$ пространства $L^2(\mathbf{R})$. Поэтому в качестве первой масштабирующей функции $\varphi(x)$ можно взять B-сплайн.

Рассмотрим основные идеи нахождения вейвлет-функций $\varphi(x)$, $\psi(x)$ и $\tilde{\varphi}(x)$, $\tilde{\psi}(x)$ с компактным носителем.

2.10.2. Условия на функции $\varphi(x)$ и $\psi(x)$

Хотя в качестве функции $\varphi(x)$ можно взять B-сплайн, будем в дальнейшем считать, что $\varphi(x)$ – произвольная масштабирующая функция, обладающая свойствами:

- функция $\varphi(x)$ вещественная, имеет компактный носитель и удовлетворяет масштабирующему соотношению $\varphi(x) = \sqrt{2}\sum_n h_n \varphi(2x-n)$;
- интеграл от $\varphi(x)$ положителен. Нормируем функцию $\varphi(x)$ так, чтобы $\int_R \varphi(x)\,dx = 1$;

- функция $\varphi(x)$ порождает (неортогональное) кратномасштабное разложение

$$\ldots \subset V_{-1} \subset V_0 \subset V_1 \subset \ldots \subset V_j \subset \ldots .$$

В частности, система функций $\{\varphi(x-n); n \in \mathbf{Z}\}$ образует базис Рисса пространства V_0. Напомним, что из вещественности $\varphi(x)$ и компактности носителя следует, что в масштабирующем соотношении коэффициенты $\{h_n\}$ вещественные и только конечное число коэффициентов $\{h_n\}$ отлично от нуля. Тогда частотная функция $H_0(\omega) = \sqrt{2^{-1}} \sum h_n e^{-i n\omega}$ является тригонометрическим полиномом. Из непрерывности $\hat{\varphi}(\omega)$, положительности интеграла от $\varphi(x)$ и из масштабирующего соотношения $\hat{\varphi}(\omega) = H_0(\omega/2)\hat{\varphi}(\omega/2)$ следует, что

$$H_0(0) = \frac{1}{\sqrt{2}} \sum_n h_n = 1. \tag{5}$$

Символом $H_0(z) = \sqrt{2^{-1}} \sum h_n z^n$ будем обозначать соответствующий полином комплексной переменной z. Его значения на единичной окружности $z = e^{-i\omega}$ дают $H_0(\omega)$.

Легко видеть, что B-сплайн $\varphi(x)$ удовлетворяет всем перечисленным выше условиям. Рассмотрим вопрос о функции $\psi(x)$, которая порождала бы дополнительные подпространства W_j. Поскольку требуется, чтобы $W_0 \subset V_1$, то функцию $\psi(x)$ следует искать в виде $\psi(x) = \sqrt{2} \sum_n g_n \varphi(2x-n)$. Будем предполагать, что искомая функция $\psi(x)$ также имеет компактный носитель, что означает конечное число ненулевых коэффициентов g_n. Пусть

$$G_0(\omega) = \frac{1}{\sqrt{2}} \sum_n g_n e^{-i n\omega} \tag{6}$$

– соответствующая (и пока неизвестная) частотная функция. Она также является тригонометрическим полиномом.

Пусть V_0 – подпространство в V_1, порожденное сдвигами $\{\varphi(x-n)\}$, $n \in \mathbf{Z}$, и W_0 – подпространство в V_1, порожденное сдвигами $\{\psi(x-n)\}$, $n \in \mathbf{Z}$. Необходимо, чтобы данные подпространства V_0 и W_0 были дополнительными в V_1, то есть чтобы

$$V_0 \oplus W_0 = V_1 \quad \text{и} \quad V_0 \cap W_0 = \{0\}. \tag{7}$$

Вопрос о возможности этого разложения решается следующей теоремой.

Теорема 1 [Чуи]. *Необходимое и достаточное условие разложимости в прямую сумму* (7) *состоит в том, что* (*непрерывная*) *функция*

$$\Delta_{HG} = H_0(z)G_0(-z) - H_0(-z)G_0(z) \tag{8}$$

нигде не обращается в нуль на единичной окружности $z = e^{-i\omega}$. *Если* $\Delta_{HG}(z) \neq 0$ *на единичной окружности, то семейство функций* $\{\psi(x-n);\ n \in \mathbf{Z}\}$ *является базисом Рисса пространства* W_0 *и имеет место разложение*

$$\sqrt{2}\varphi(2x-n) = \sum_k h_{2k-n}\varphi(x-k) + \sum_k g_{2k-n}\psi(x-k).$$

Доказательство может быть получено на основе третьей части доказательства теоремы 3 раздела 2.3. Полное доказательство можно найти в книге [Чуи], стр. 225.

Поскольку все пространства V_j являются масштабированными версиями пространства V_0, то из разложения $V_1 = V_0 \oplus W_0$ следуют разложения $V_{j+1} = V_j \oplus W_j$ при любом j. При этом базис Рисса пространства W_j образуют функции $\psi_{j,n}(x) = \sqrt{2^j}\psi(2^j x - n),\ n \in \mathbf{Z}$. Однако все семейство функций $\{\psi_{j,n}(x),\ j, n \in \mathbf{Z}\}$ не обязательно является базисом Рисса всего пространства $L^2(\mathbf{R})$. Известно [Чуи] следующее необходимое условие: если функция $\psi(x)$ порождает базис Рисса всего пространства $L^2(\mathbf{R})$ и $\hat{\psi}(\omega)$ непрерывна, то интеграл от функции $\psi(x)$ должен быть равен нулю:

$$\hat{\psi}(0) = \int_{-\infty}^{\infty} \psi(x)\,dx = 0. \tag{9}$$

Заметим, что в ортогональном случае функция $\psi(x)$ может быть найдена по формуле $\psi(x) = \sqrt{2}\sum_n (-1)^{n+1}\overline{h_{-1-n}}\varphi(2x-n)$. В нашем случае мы пока не можем найти $\psi(x)$, однако можно построить функцию $\tilde{\varphi}(x)$.

2.10.3. Построение функции $\tilde{\varphi}(x)$

Теперь рассмотрим свойства, которыми должны обладать масштабирующая функция $\tilde{\varphi}(x)$, порождающая дуальный базис. Выпишем масштабирующее соотношение с неизвестными коэффициентами $\{\tilde{h}_n\}$:

$$\tilde{\varphi}(x) = \sqrt{2}\sum_n \tilde{h}_n\tilde{\varphi}(2x-n), \qquad \tilde{H}_0(\omega) = \frac{1}{\sqrt{2}}\sum_n \tilde{h}_n e^{-in\omega}. \tag{10}$$

Функция $\tilde{\varphi}(x)$ будет найдена, если мы найдем коэффициенты $\{\tilde{h}_n\}$ или эквивалентно – ее частотную функцию $\tilde{H}_0(\omega)$.

Определение 2. *Масштабирующие функции* $\varphi(x)$ *и* $\tilde{\varphi}(x)$ *называются двойственными, если*

$$(\varphi(x-n),\tilde{\varphi}(x-k))=\int_{-\infty}^{\infty}\varphi(x-n)\overline{\tilde{\varphi}(x-k)}\,dx=\delta_{nk},\ n,k\in\mathbf{Z}. \tag{11}$$

Лемма 1. *Масштабирующие функции* $\varphi(x)$ *и* $\tilde{\varphi}(x)$ *являются двойственными тогда и только тогда, когда*

$$\sum_k \hat{\varphi}(\omega+2\pi k)\overline{\hat{\tilde{\varphi}}(\omega+2\pi k)}=1 \ \text{ п. в.} \tag{12}$$

Доказательство. Поскольку интеграл в (11) инвариантен относительно сдвигов, достаточно рассмотреть только $(\varphi(x-n),\tilde{\varphi}(x))$:

$$(\varphi(x-n),\tilde{\varphi}(x))=\frac{1}{2\pi}\int_{-\infty}^{\infty}e^{-in\omega}\hat{\varphi}(\omega)\overline{\hat{\tilde{\varphi}}(\omega)}\,d\omega=$$

$$=\frac{1}{2\pi}\int_0^{2\pi}e^{-in\omega}\left(\sum_k\hat{\varphi}(\omega+2\pi k)\overline{\hat{\tilde{\varphi}}(\omega+2\pi k)}\right)d\omega=\delta_{n0}.$$

Последнее равенство верно лишь в том случае, когда выполнено (12).

Выпишем масштабирующее соотношение для образа Фурье $\hat{\tilde{\varphi}}(x)$:

$$\hat{\tilde{\varphi}}(\omega)=\tilde{H}_0(\omega/2)\,\hat{\tilde{\varphi}}(\omega/2).$$

Лемма 2. *Если масштабирующие функции* $\varphi(x)$ *и* $\tilde{\varphi}(x)$ *являются двойственными, то*

$$\overline{H_0(z)}\tilde{H}_0(z)+\overline{H_0(-z)}\tilde{H}_0(-z)=1 \ \textit{п. в.} \tag{13}$$

на единичной окружности $z=e^{-i\omega}$.

Доказательство. В следующих вычислениях мы разбиваем интеграл на сумму по k отдельно по четным $k=2m$ и по нечетным $k=2m+1$ и используем 2π-периодичность функций $H_0(\omega)$ и $\tilde{H}_0(\omega)$. Пусть также $z=e^{-i\omega/2}$ и $\varsigma=\omega/2$. Тогда

$$(\tilde{\varphi}(x),\varphi(x-n))=\frac{1}{2\pi}\int_{-\infty}^{\infty}e^{in\omega}\overline{\hat{\varphi}(\omega)}\hat{\tilde{\varphi}}(\omega)\,d\omega=$$

$$=\frac{1}{2\pi}\int_{-\infty}^{\infty}e^{in\omega}\overline{H_0}(\omega/2)\,\overline{\tilde{\varphi}}(\omega/2)\tilde{H}_0(\omega/2)\,\hat{\tilde{\varphi}}(\omega/2)\,d\omega=$$

$$=\frac{1}{2\pi}\sum_k\int_0^{2\pi}e^{in\omega}\left(\overline{H_0}\left(\frac{\omega+2\pi k}{2}\right)\tilde{H}_0\left(\frac{\omega+2\pi k}{2}\right)\overline{\hat{\tilde{\varphi}}}\left(\frac{\omega+2\pi k}{2}\right)\hat{\tilde{\varphi}}\left(\frac{\omega+2\pi k}{2}\right)\right)d\omega=$$

$$=\frac{1}{2\pi}\int_0^{2\pi}e^{in\omega}\left(\overline{H_0(z)}\tilde{H}_0(z)\sum_m\overline{\hat{\varphi}(\varsigma+2\pi m)}\,\hat{\tilde{\varphi}}(\varsigma+2\pi m)\right)d\omega+$$

$$+\frac{1}{2\pi}\int_0^{2\pi} e^{in\omega}\left(\overline{H_0}(-z)\tilde{H}_0(-z)\sum_m \overline{\hat{\varphi}(\varsigma+\pi+2\pi m)}\,\hat{\tilde{\varphi}}(\varsigma+\pi+2\pi m)\right)d\omega =$$

$$=\frac{1}{2\pi}\int_0^{2\pi} e^{in\omega}(\overline{H_0(z)}\tilde{H}_0(z)+\overline{H_0(-z)}\tilde{H}_0(-z))\,d\omega=\delta_{n0}.$$

Последнее равенство имеет место только в том случае, когда выполнено (13).

В нашем случае многочлен $H_0(z)$ имеет вещественные коэффициенты. Поэтому на единичной окружности $z = e^{-i\omega}$ будет выполняться равенство $\overline{H_0(z)} = H_0(z^{-1})$.

Замечание 1. Лемма 2 дает достаточно простые условия на функции $H_0(z)$ и $\tilde{H}_0(z)$. Учитывая, что масштабирующие функции $\varphi(x)$ и $\tilde{\varphi}(x)$ определяются по своим частотным функциям $H_0(z)$ и $\tilde{H}_0(z)$, лемма 2 в большой степени решает вопрос о нахождении функций $\varphi(x)$ и $\tilde{\varphi}(x)$.

Замечание 2. Условие (13) равносильно следующему условию:

$$\sum_{n\in Z}\overline{h}_n\tilde{h}_{n+2k}=\delta_{0k} \tag{14}$$

на коэффициенты $\{h_n\}$ и $\{\tilde{h}_n\}$ фильтров $H_0(z)$ и $\tilde{H}_0(z)$. Это легко видеть из разложения правой части равенства $\overline{H_0}(\omega)\tilde{H}_0(\omega)+\overline{H_0}(\omega+\pi)\times$ $\times\tilde{H}_0(\omega+\pi)=1$ в ряд Фурье:

$$\overline{H_0}(\omega)\tilde{H}_0(\omega)+\overline{H_0}(\omega+\pi)\tilde{H}_0(\omega+\pi)=$$

$$=\frac{1}{\sqrt{2}}\sum_{n\in\mathbf{Z}}\overline{h}_n e^{in\omega}\frac{1}{\sqrt{2}}\sum_{m\in\mathbf{Z}}\tilde{h}_m e^{-im\omega}+\frac{1}{\sqrt{2}}\sum_{n\in\mathbf{Z}}\overline{h}_n e^{in(\omega+\pi)}\frac{1}{\sqrt{2}}\sum_{m\in\mathbf{Z}}\tilde{h}_m e^{-im(\omega+\pi)}=$$

$$=\frac{1}{2}\sum_{n,m\in\mathbf{Z}}\overline{h}_n\,\tilde{h}_m e^{-i(m-n)\omega}+\frac{1}{2}\sum_{n,m\in\mathbf{Z}}\overline{h}_n\,\tilde{h}_m\,e^{-i(m-n)\pi}e^{-i(m-n)\omega}=$$

$$=\frac{1}{2}\sum_{n,p\in\mathbf{Z}}\overline{h}_n\,\tilde{h}_{n+p}(1+(-1)^p)e^{-ip\omega}=\sum_{k\in\mathbf{Z}}\sum_{n\in\mathbf{Z}}\overline{h}_n\,\tilde{h}_{n+2k}e^{-i2k\omega}.$$

Поэтому равенство (13) выполняется только тогда, когда $\sum_n \overline{h}_n\,\tilde{h}_{n+2k} = \delta_{0k}$.

В нашем случае вещественных коэффициентов $\{h_n\}$ равенство (14) принимает вид $\sum_n h_n\tilde{h}_{n+2k}=\delta_{0k}$.

В случае, когда масштабирующие функции $\varphi(x)$ и $\tilde{\varphi}(x)$ имеют компактные носители, их частотные функции $H_0(z)$ и $\tilde{H}_0(z)$ являются многочленами. Тогда, считая заданной вещественную функ-

цию $\varphi(x)$ и ее фильтр $H_0(z)$, можно найти функцию $\tilde{H}_0(z)$ из уравнения $\overline{H_0(z)}\tilde{H}_0(z)+\overline{H_0(-z)}\tilde{H}_0(-z)=1$ по теореме Безу. Учитывая, что $\overline{H_0(z)}=H_0(z^{-1})$, уравнение принимает вид

$$H_0(z^{-1})\tilde{H}_0(z)+H_0(-z^{-1})\tilde{H}_0(-z)=1.$$

Полином $\tilde{H}_0(z)$ находится единственным образом и с вещественными коэффициентами $\tilde{h}_n$. Если мы нашли решение $\tilde{H}_0(z)$, удовлетворяющее условию

$$\tilde{H}_0(1)=\frac{1}{\sqrt{2}}\sum_{n\in Z}\tilde{h}_n=1, \tag{15}$$

то функция $\tilde{\varphi}(x)$ определяется из масштабирующего уравнения

$$\hat{\tilde{\varphi}}(\omega)=\tilde{H}_0(\omega/2)\,\hat{\tilde{\varphi}}(\omega/2) \tag{16}$$

в виде бесконечного произведения

$$\hat{\tilde{\varphi}}(\omega)=\prod_{j=1}^{\infty}\tilde{H}_0\left(\frac{\omega}{2^j}\right).$$

2.10.4. Построение функций $\psi(x)$ и $\tilde{\psi}(x)$

Полагаем, что масштабирующие функции $\varphi(x)$ и $\tilde{\varphi}(x)$ уже найдены по своим частотным функциям $H_0(z)$ и $\tilde{H}_0(\omega)$, которые удовлетворяют соотношению (13). Найдем функции $\psi(x)$ и $\tilde{\psi}(x)$.

Предположим, что функции $\psi(x)$ и $\tilde{\psi}(x)$ найдены. Тогда пара функций $\varphi(x)$ и $\psi(x)$ определяет вейвлет-разложение сигнала $\{f_n\}$ по формулам

$$a_k=\sum_n h_n^* f_{2k-n},\qquad d_k=\sum_n g_n^* f_{2k-n},$$

где $h_n^*=\overline{\tilde{h}}_{-n}$ и $g_n^*=\overline{\tilde{g}}_{-n}$ – сопряженные фильтры к фильтрам коэффициентов $\{h_n\}$ и $\{g_n\}$ функций $\varphi(x)$ и $\psi(x)$. Поэтому их передаточные функции будут сопряженными к исходным: $\overline{H_0(\omega)}$ и $\overline{G_0(\omega)}$.

Функции $\tilde{\varphi}(x)$, $\tilde{\psi}(x)$ используются для восстановления сигнала, и их основное свойство заключается в том, что они обеспечивают точное восстановление своими фильтрами $\tilde{h}_n$ и $\tilde{g}_n$ по формуле

$$f_n=\sum\nolimits_k a_k\tilde{h}_{n-2k}+d_k\tilde{g}_{n-2k}.$$

В соответствии с результатами § 6 главы 1 точное восстановление сигнала возможно в том случае, когда фильтры $\overline{H_0}(z)$, $\overline{G_0}(z)$ и $\tilde{H}_0(z)$,

$\tilde{G}_0(z)$, участвующие в разложении и восстановлении, удовлетворяют следующей системе (с учетом $\sqrt{2}$ в частотных функциях):

$$\left.\begin{aligned}\overline{H_0}(z)\tilde{H}_0(z)+\overline{G_0}(z)\tilde{G}_0(z)=1\\ \overline{H_0}(-z)\tilde{H}_0(z)+\overline{G_0}(-z)\tilde{G}_0(z)=0\end{aligned}\right\}. \tag{17}$$

Как уже отмечалось, можно считать известными функции $H_0(z)$ и $\tilde{H}_0(z)$, остальные подлежат определению. Если мы найдем функции $G_0(z)$ и $\tilde{G}_0(\omega)$ удовлетворяющими (17), то функции $\psi(x)$ и $\tilde{\psi}(x)$ определятся равенствами

$$\hat{\psi}(\omega)=G_0(\omega/2)\,\varphi(\omega/2) \quad \text{и} \quad \hat{\tilde{\psi}}(\omega)=\tilde{G}_0(\omega/2)\,\tilde{\varphi}(\omega/2).$$

По предположению все наши функции $\varphi(x)$, $\psi(x)$ и $\tilde{\varphi}(x)$, $\tilde{\psi}(x)$ имеют компактные носители. Тогда все функции $H_0(z)$, $G_0(z)$ и $\tilde{H}_0(z)$, $\tilde{G}_0(z)$ являются тригонометрическими полиномами. Второе соотношение в (17) можно решить, полагая

$$\tilde{H}_0(z)=-z\overline{G_0}(-z), \quad \tilde{G}_0(z)=z\overline{H_0}(-z).$$

Тогда $\overline{G_0}(z)=z^{-1}\tilde{H}_0(-z)$. Подставляя эти выражения в первое уравнение системы (17), получаем равенство

$$\overline{H_0(z)}\tilde{H}_0(z)+\overline{H_0(-z)}\tilde{H}_0(-z)=1,$$

которое, как предполагается, решено в предыдущем разделе при нахождении функции $\tilde{\varphi}(x)$.

Если $z=e^{-i\omega}$, то $-z=-e^{-i\omega}=-e^{-i(\omega+\pi)}$ и $z^{-1}=\bar{z}$. Тогда получаем:

$$G_0(\omega)=e^{-i\omega}\overline{\tilde{H}_0(\omega+\pi)}, \quad \tilde{G}_0(\omega)=e^{-i\omega}\overline{H_0(\omega+\pi)}. \tag{18}$$

Из последних выражений следует, что если

$$G_0(\omega)=\frac{1}{\sqrt{2}}\sum_k g_k e^{-i\omega k} \quad \text{и} \quad \tilde{G}_0(\omega)=\frac{1}{\sqrt{2}}\sum_k \tilde{g}_k e^{-i\omega k},$$

то коэффициенты $\{g_k\}$ и $\{\tilde{g}_k\}$ определяются равенствами

$$g_k=(-1)^{k+1}\tilde{h}_{1-k}, \quad \tilde{g}_k=(-1)^{k+1}h_{1-k}. \tag{19}$$

Поскольку $\hat{\psi}(\omega)=G_0(\omega/2)\,\varphi(\omega/2)$ и $\hat{\tilde{\psi}}(\omega)=\tilde{G}_0(\omega/2)\,\tilde{\varphi}(\omega/2)$, то функции $\psi(x)$ и $\tilde{\psi}(x)$ находятся в виде следующих линейных комбинаций:

$$\psi(x)=\sqrt{2}\sum_k g_k\varphi(2x-k), \quad \tilde{\psi}(x)=\sqrt{2}\sum_k \tilde{g}_k\tilde{\varphi}(2x-k). \tag{20}$$

Для того чтобы функции $\psi(x)$ и $\tilde{\psi}(x)$ образовывали базисы Рисса всего пространства $L^2(\mathbf{R})$, должны выполняться условия $\hat{\psi}(0) = 0 = \hat{\tilde{\psi}}(0)$. Последнее свойство выражается в частотных функциях в виде равенства $G_0(0) = 0 = \tilde{G}_0(0)$. Это означает, с учетом (19), следующие равенства для коэффициентов $\{h_k\}$ и $\{\tilde{h}_k\}$:

$$\sum_{k\in Z}(-1)^{k+1}\tilde{h}_{1-k} = 0, \quad \sum_{k\in Z}(-1)^{k+1}h_{1-k} = 0.$$

Упражнение. Проверить, что условие $\overline{H_0(z)}\tilde{H}_0(z) + \overline{H_0(-z)}\tilde{H}_0(-z) = 1$ обеспечивает выполнение условий

$$\Delta_{HG} = H_0(z)G_0(-z) - H_0(-z)G_0(z) \neq 0,$$
$$\Delta_{\tilde{H}\tilde{G}} = \tilde{H}_0(z)\tilde{G}_0(-z) - \tilde{H}_0(-z)\tilde{G}_0(z) \neq 0,$$

необходимых для того, чтобы функции $\psi(x)$ и $\tilde{\psi}(x)$, определенные по частотным функциям $G_0(z)$ и $G\tilde{}_0(z)$, образовывали бы (по теореме 1) базисы Рисса пространств W_j и $\tilde{W}_j$ и чтобы имели место разложения $V_{j+1} = V_j \oplus W_j$ и $\tilde{V}_{j+1} = \tilde{V}_j \oplus \tilde{W}_j$.

2.10.5. Условия на коэффициенты

Суммируем все условия для коэффициентов $\{h_n\}$ и $\{\tilde{h}_n\}$, необходимые для построения биортогональных вейвлетов.

1. Условие $\overline{H_0}(z)\tilde{H}_0(z) + \overline{H_0}(-z)\tilde{H}_0(-z) = 1$ (при $z = e^{-i\omega}$) двойственности функций $\varphi(x)$ и $\tilde{\varphi}(x)$, или в терминах коэффициентов фильтров:
$$\sum_{n\in Z} h_n\tilde{h}_{n+2k} = \delta_{0k}. \tag{21}$$
2. Нормировка коэффициентов $\{h_n\}$ и $\{\tilde{h}_n\}$.
$$\sum_{n\in Z} h_n = \sqrt{2}, \quad \sum_{n\in Z}\tilde{h}_n = \sqrt{2}. \tag{22}$$
3. Необходимое условие образования функциями $\psi(x)$ и $\tilde{\psi}(x)$ базисов Рисса всего пространства $L^2(\mathbf{R})$:
$$\sum_{k\in Z}(-1)^{k+1}\tilde{h}_{1-k} = 0, \quad \sum_{k\in Z}(-1)^{k+1}h_{1-k} = 0. \tag{23}$$

2.10.6. Симметричные биортогональные вейвлеты

Пусть масштабирующая функция $\varphi(x)$ имеет компактный носитель и симметрична. Тогда коэффициенты $\{h_n\}$ будут симметричными от-

носительно серединного значения. Если число коэффициентов N, то носитель функции $\varphi(x)$ имеет длину $N-1$.

Будем искать остальные функции $\psi(x)$ и $\tilde{\varphi}(x)$, $\tilde{\psi}(x)$, которые также предполагаем симметричными и с компактным носителем. Предположим сначала, что ненулевых коэффициентов $\{h_n\}$ – нечетное число и k – середина ненулевого массива $\{h_n\}$ (четный случай рассматривается совершенно аналогично). Тогда в силу симметричности $H_0(-\omega) = e^{i2k\omega}H_0(\omega)$. Поэтому $H_0(\omega)$ имеет вид

$$H_0(\omega) = e^{-ik\omega}p_0(\cos\omega), \tag{24}$$

где $p_0(x)$ – полином. Будем искать $\tilde{H}_0(\omega)$ в аналогичном виде:

$$\tilde{H}_0(\omega) = e^{-ik\omega}\tilde{p}_0(\cos\omega), \tag{25}$$

где $\tilde{p}_0(x)$ – искомый полином. Соотношение (13), которое мы должны решить, теперь выглядит следующим образом:

$$\overline{p_0(x)}\tilde{p}_0(x) + \overline{p_0(-x)}\tilde{p}_0(-x) = 1, \tag{26}$$

или, с учетом $z = e^{-i\omega}$ и $-z = -e^{-i\omega} = -e^{-i(\omega+\pi)}$,

$$\overline{H_0(\omega)}\tilde{H}_0(\omega) + \overline{H_0(\omega+\pi)}\tilde{H}_0(\omega+\pi) = 1. \tag{27}$$

Если многочлены $p_0(x)$ и $p_0(-x)$ не имеют общих корней, то по теореме Безу всегда существует точное решение $\tilde{p}_0(x)$. Для нахождения $\tilde{p}_0(x)$ достаточно решить линейную систему уравнений.

Желая сделать вейвлеты $\varphi(x)$, $\psi(x)$ и $\tilde{\varphi}(x)$, $\tilde{\psi}(x)$ более регулярными, потребуем, чтобы функции $H_0(\omega)$ и $\tilde{H}_0(\omega)$ имели множитель вида $(1 + e^{-i\omega})^l$ для некоторого $l > 0$. Из вида (24), (25) функций $H_0(\omega)$ и $\tilde{H}_0(\omega)$ и того, что многочлены $p_0(x)$, $\tilde{p}_0(x)$ зависят только от $\cos\omega$, следует, что степень l – четная. Действительно,

$$(1 + e^{-i\omega})^2 = 4e^{-i\omega}\cos^2\omega/2 = 2e^{-i\omega}(1 + \cos\omega),$$

а в нечетном случае будут и синусы.

Выбирая точку симметрии за начало координат, можно считать, что в случае нечетного числа коэффициентов

$$H_0(\omega) = (\cos\omega/2)^{2l}q_0(\cos\omega), \quad \tilde{H}_0(\omega) = (\cos\omega/2)^{2s}\tilde{q}_0(\cos\omega). \tag{28}$$

В случае четного числа коэффициентов будем искать наши функции в виде

$$H_0(\omega) = e^{-i\omega/2}(\cos\omega/2)^{2l+1}q_0(\cos\omega), \quad \tilde{H}_0(\omega) = e^{-i\omega/2}(\cos\omega/2)^{2s+1}\tilde{q}_0(\cos\omega).$$

В любом случае, подстановка в (27) дает

$$(\cos\omega/2)^{2L}\overline{q_0(\cos\omega)}\,\tilde{q}_0(\cos\omega) + (\sin\omega/2)^{2L}\overline{q_0(-\cos\omega)}\tilde{q}_0(-\cos\omega) = 1, \tag{29}$$

где $L = l + s$ в первом случае и $L = l + s + 1$ – во втором случае. Обозначим

$$P(\sin^2\omega/2) = \overline{q_0(\cos\omega)}\,\tilde{q}_0(\cos\omega), \quad x = \sin^2\omega/2,$$

тогда, учитывая, что

$$P(1-x) = P(1-\sin^2\omega/2) = P(\cos^2\omega/2) = P(\sin^2(\omega+\pi)/2) =$$

$$= \overline{q_0(\cos(\omega+\pi))}\,q_0(\cos(\omega+\pi)) = \overline{q_0(-\cos\omega)}\,q_0(-\cos\omega),$$

уравнение (29) принимает знакомый нам вид:

$$(1-x)^L P(x) + x^L P(1-x) = 1. \tag{30}$$

Все решения найдены в разделе 2.8, они имеют вид

$$P(x) = \sum_{m=0}^{L-1} C_{L+m-1}^m x^m + x^L R(1-2x), \tag{31}$$

где R – полином с нечетными степенями.

2.10.7. Сплайны

Рассмотрим построение биортогональных вейвлетов, когда в качестве $\varphi(x)$ берется B-сплайн порядка n. Как известно, частотная функция для сплайнов имеет вид $H_{0n}(\omega) = e^{-iK\omega/2}(\cos\omega/2)^{n+1}$. Она имеет тип (28), рассмотренный выше, где $q_0 \equiv 1$. Имея возможность выбирать точку симметрии сплайна, будем считать, что

$$H_0(\omega) = \begin{cases} (\cos\omega/2)^N, & \text{если} \quad N = 2l \\ e^{-i\omega/2}(\cos\omega/2)^N, & \text{если} \quad N = 2l+1 \end{cases}. \tag{32}$$

Тогда, полагая $R = 0$, получаем явное выражение для $\tilde{H}_0(\omega)$. В первом случае нечетного числа коэффициентов $\tilde{N} = 2s$, согласно (28) и $q_0 \equiv 1$, имеем

$$\tilde{H}_0(\omega) = (\cos\omega/2)^{2s} \sum_{m=0}^{L-1} C_{L+m-1}^m (\sin^2\omega/2)^m, \quad L = l + s. \tag{33}$$

Во втором случае четного числа коэффициентов $\tilde{N} = 2s + 1$, согласно (28'):

$$\tilde{H}_0(\omega) = e^{-i\omega/2}(\cos\omega/2)^{2s+1} \sum_{m=0}^{L-1} C_{L+m-1}^m (\sin^2\omega/2)^m, \quad L = l + s + 1. \tag{34}$$

В результате мы получаем биортогональные вейвлеты, где функции $\varphi(x)$, $\psi(x)$ являются сплайнами. Функция $\tilde{\varphi}(x)$ определяется через $\hat{\tilde{\varphi}}(\omega)$ в виде бесконечного произведения (16). Функции $\psi(x)$ и

$\tilde{\psi}(x)$ находятся как линейные комбинации (20) масштабированных сдвигов функций $\varphi(x)$ и $\tilde{\varphi}(x)$ с коэффициентами (19).

Замечание 3. В случае биортогональных вейвлетов на основе сплайнов масштабирующая функция $\varphi(x)$, определенная по формуле (32), обозначается символом $\varphi_N(x)$. Другие вейвлет-функции, для определения которых требуется еще и $\tilde{N} = 2s$, принято обозначать символами $\tilde{\varphi}_{N\tilde{N}}(x), \psi_{N\tilde{N}}(x)$ и $\tilde{\psi}_{N\tilde{N}}(x)$.

Пример 1. Найдем биортогональные вейвлеты $\varphi(x)$, $\tilde{\varphi}(x)$, $\psi(x)$ и $\tilde{\psi}(x)$ в случае, когда функция $\varphi(x)$ есть сплайн степени 1. В этом случае $N = 2$. Пусть $\tilde{N} = 2s = 2$.

1. Функция $\varphi_2(x)$. Сначала по формуле (32) находим частотную функцию H_0:

$H_0 = \cos^2\omega/2$.

Поскольку $(1 + e^{-i\omega})^2 = 4e^{-i\omega}\cos^2\omega/2$, то после замены $z = e^{-i\omega}$ получаем

$$H_0(z) = \frac{(1+z)^2}{4z} = \frac{1}{4}(z^{-1} + 2 + z).$$

Ненулевые коэффициенты фильтра:

$$h_{-1} = \frac{1}{2\sqrt{2}}, \quad h_0 = \frac{1}{\sqrt{2}}, \quad h_1 = \frac{1}{2\sqrt{2}}.$$

Масштабирующая функция $\varphi_2(x)$ есть B-сплайн степени 1:

$$\varphi_2(x) = \begin{cases} 1 - |x|, & x \in [-1,1], \\ 0 & \text{при остальных } x \end{cases}$$

График $\varphi_2(x)$ приведен ниже.

2. Функция $\tilde{\varphi}_{22}(x)$. Сначала по формуле (33) находим частотную функцию $\tilde{H}_0(\omega)$. В данном случае $L = 1 + 1 = 2$, $m = 0, 1$, $C_1^0 = 1$, $C_2^1 = 2$. Поэтому

$$\tilde{H}_0(\omega) = \cos^2\omega/2(1 + 2\sin^2\omega/2) = \cos^2\omega/2(3 - 2\cos^2\omega/2) =$$

$$= \frac{(1+z)^2}{4z}\left(3 - 2\frac{(1+z)^2}{4z}\right) = \frac{1}{8}(-z^{-2} + 2z^{-1} + 6 + 2z - z^2).$$

$$\tilde{H}_0(z) = \frac{1}{8}(-z^{-2} + 2z^{-1} + 6 + 2z - z^2).$$

Ненулевые коэффициенты фильтра:

$$\tilde{h}_{-2} = -\frac{1}{4\sqrt{2}}, \quad \tilde{h}_{-1} = \frac{1}{2\sqrt{2}}, \quad \tilde{h}_0 = \frac{3}{2\sqrt{2}}, \quad \tilde{h}_1 = \frac{1}{2\sqrt{2}}, \quad \tilde{h}_2 = -\frac{1}{4\sqrt{2}}.$$

Соответствующая масштабирующая функция $\tilde{\varphi}_{22}(x)$ не имеет конечного аналитического выражения. Сначала мы находим ее преобразование Фурье в виде бесконечного произведения

$$\hat{\tilde{\varphi}}(\omega) = \prod_{j=1}^{\infty} \tilde{H}_0\left(\frac{\omega}{2^j}\right),$$

а затем применяем обратное преобразование Фурье.

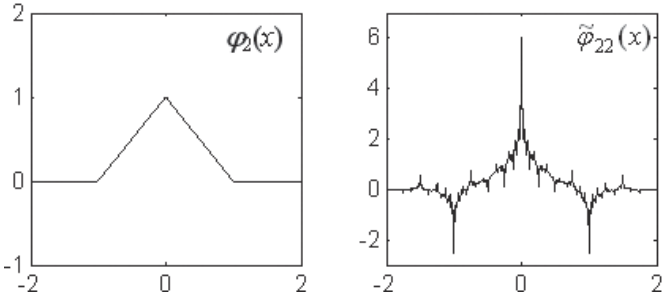


Рис. 2.10.1. Графики функций $\varphi_2(x)$ и $\tilde{\varphi}_{22}(x)$

3. Функция $\psi_{22}(x)$. Она определяется формулой

$$\psi_{22}(x) = \sqrt{2}\sum_n g_n\, \varphi_2(2x-n), \text{ где } g_k = (-1)^{k+1}\,\tilde{h}_{1-k}.$$

Находим ненулевые коэффициенты g_k:

$$g_{-1} = \tilde{h}_2 = -\frac{1}{4\sqrt{2}}, \quad g_0 = -\tilde{h}_1 = -\frac{1}{2\sqrt{2}}, \quad g_1 = \tilde{h}_0 = \frac{3}{2\sqrt{2}},$$

$$g_2 = -\tilde{h}_{-1} = -\frac{1}{2\sqrt{2}}, \quad g_3 = \tilde{h}_{-2} = -\frac{1}{4\sqrt{2}}.$$

$$\psi_{22}(x) = -\frac{1}{4}\varphi_2(2x+1) - \frac{1}{2}\varphi_2(2x) + \frac{3}{2}\varphi_2(2x-1) - \frac{1}{2}\varphi_2(2x-2) - \frac{1}{4}\varphi_2(2x-3).$$

Из данного выражения следует, что вейвлет $\psi_{22}(x)$ является сплайном степени 1 с носителем $[-1, -2]$. График функции $\psi_{22}(x)$ приведен ниже.

4. Функция $\tilde{\psi}_{22}(x)$. Она определяется формулой

$$\tilde{\psi}_{22}(x) = \sqrt{2}\sum_n \tilde{g}_n \tilde{\varphi}_{22}(2x-n), \text{ где } \tilde{g}_k = (-1)^{k+1}\, h_{1-k}.$$

Находим ненулевые коэффициенты $\tilde{g}_k$:

$$\tilde{g}_0 = -h_1 = -\frac{1}{2\sqrt{2}}, \quad \tilde{g}_1 = h_0 = \frac{1}{\sqrt{2}}, \quad \tilde{g}_2 = -h_{-1} = -\frac{1}{2\sqrt{2}}.$$

$$\tilde{\psi}_{22}(x) = -\frac{1}{2}\tilde{\varphi}_{22}(2x) + \tilde{\varphi}_{22}(2x-1) - \frac{1}{2}\tilde{\varphi}_{22}(2x-2).$$

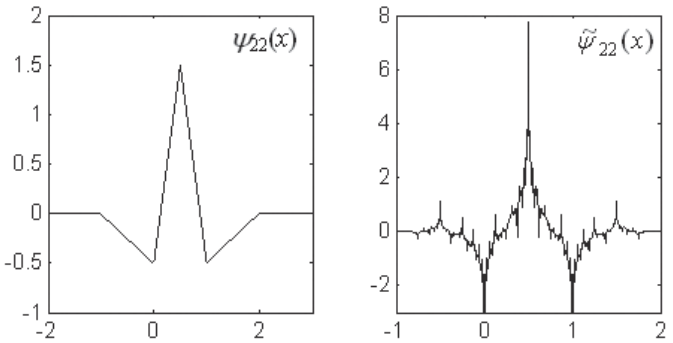


Рис. 2.10.2. Графики функций $\psi_{22}(x)$ и $\tilde{\psi}_{22}(x)$

Замечание 4. Функции $\varphi(x)$ и $\psi(x)$ обычно обладают большей регулярностью, чем $\tilde{\varphi}(x)$ и $\tilde{\psi}(x)$. Поэтому принято (например, в MATLAB) разложение сигнала проводить с использованием функций $\tilde{\varphi}(x)$ и $\tilde{\psi}(x)$, а восстановление – при помощи $\varphi(x)$ и $\psi(x)$.

2.11. Двумерные вейвлеты

Для некоторых задач необходимо анализировать сигналы, которые представляют собой функции двух переменных. Тогда кратномасштабный анализ должен строиться в пространстве $L^2(\mathbf{R}^2)$ функций двух переменных. Существует несколько способов обобщить изложенные ранее конструкции на функции нескольких переменных (мы будем говорить только о функциях двух переменных). Самый простой и широко распространенный путь – это тензорное произведение одномерных вейвлетов. Он основан на том факте, что пространство $L^2(\mathbf{R}^2)$ является тензорным произведением

$$L^2(\mathbf{R}^2) = L^2(\mathbf{R}) \otimes L^2(\mathbf{R})$$

пространств $L^2(\mathbf{R})$ функций одной переменной. Это означает, что линейные комбинации функций вида $f(x)g(y)$ образуют плотное множество в пространстве $L^2(\mathbf{R}^2)$ функций двух переменных. Рассмотрим

кратко идеи данного подхода. Напомним сначала основные элементы одномерного кратномасштабного анализа.

Пусть функции $\varphi(x)$ и $\psi(x)$ порождают ортогональный кратномасштабный анализ пространства $L^2(\mathbf{R})$:

$$... \subset V_{-1} \subset V_0 \subset V_1 \subset V_1 \subset ...$$

Сдвиги $\varphi_{0,n}(x) = \varphi(x - n), n \in \mathbf{Z}$ функции $\varphi(x)$ образуют ортонормированный базис пространства V_0. Остальные пространства V_j являются масштабированными версиями V_0, $f(x) \in V_0 \Leftrightarrow f(2^j x) \in V_j$. В частности, пространство V_j имеет в качестве ортонормированного базиса систему функций $\varphi_{j,k}(x) = \sqrt{2^j}\varphi(2^j x - k)$.

Пространство вейвлетов W_0 определяется как ортогональное дополнение к V_0 в подпространстве V_1. Тогда

$$V_1 = V_0 \oplus W_0$$

– прямая сумма ортогональных подпространств. В пространстве W_0 ортонормированный базис образуют сдвиги $\psi_{0,n}(x) = \psi(x - n)$, $n \in \mathbf{Z}$, функции $\psi(x)$. В случае любого $j \in \mathbf{Z}$ имеем $V_j \subset V_{j+1}$ и

$$V_{j+1} = V_j \oplus W_j.$$

Пространства вейвлетов W_j являются масштабированными версиями W_0, в частности W_j имеет ортонормированный базис из функций $\psi_{j,k}(x) = \sqrt{2^j}\psi(2^j x - k)$.

В двумерном случае определим пространство $V_0^2 \in L^2(\mathbf{R}^2)$ как тензорное произведение пространств V_0 функций одной переменной:

$$V_0^2 = V_0 \oplus V_0.$$

Другими словами, пространство V_0^2 есть замкнутое подпространство в $L^2(\mathbf{R}^2)$, порожденное сдвигами

$$\varphi_{0,k,n}(x, y) = \varphi(x - k)\varphi(y - n), k, n \in \mathbf{Z}.$$

Эти функции образуют ортонормированный базис пространства V_0^2.

Пространства $V_j^2 = V_j \otimes V_j$ являются масштабированными версиями пространства V_0^2:

$$f(x, y) \in V_0^2 \Leftrightarrow f(2^j x, 2^j y) \in V_j^2.$$

Тогда мы получаем последовательность замкнутых вложенных друг в друга подпространств

$$... \subset V_{-1}^2 \subset V_0^2 \subset V_1^2 \subset V_2^2 \subset ...,$$

обладающую аналогичными свойствами.

Поскольку $V_1 = V_0 \oplus W_0$, то для тензорного произведения имеем следующее ортогональное разложение:

$$V_1^2 = V_1 \otimes V_1 = (V_0 + W_0) \otimes (V_0 + W_0) =$$
$$= (V_0 \otimes V_0) + (V_0 \otimes W_0) + (W_0 \otimes V_0) + (W_0 \otimes W_0).$$

Здесь $V_0 \otimes V_0 = V_0^2$. Поэтому остальные три пространства в прямой ортогональной сумме образуют пространство W_0^2 двумерных вейвлетов. При этом:

- пространство $V_0 \otimes W_0$ порождено целочисленными сдвигами функции $\psi^H(x, y) = \varphi(x)\psi(y)$;
- пространство $W_0 \otimes V_0$ порождено целочисленными сдвигами функции $\psi^V(x, y) = \psi(x)\varphi(y)$;
- пространство $W_0 \otimes W_0$ порождено целочисленными сдвигами функции $\psi^D(x, y) = \psi(x)\psi(y)$.

Будем обозначать эти пространства соответственно символами

$$W_0^H = V_0 \otimes W_0, \quad W_0^V = W_0 \otimes V_0 \quad \text{и} \quad W_0^D = W_0 \otimes W_0.$$

Тогда получаем следующее ортогональное разложение:

$$V_1^2 = V_0^2 \oplus W_0^H \oplus W_0^V \oplus W_0^D.$$

Соответствующее разложение получается для любого $j \in \mathbf{Z}$.

$$V_{j+1}^2 = V_{j+1}^2 \oplus W_j^H \oplus W_j^V \oplus W_j^D.$$

Выпишем системы функций, которые образуют ортонормированные базисы в указанных выше пространствах V_j^2, W_j^H, W_j^V, W_j^D:

$$\varphi_{j,k,n}(x, y) = 2^j\varphi(2^j x - k)\varphi(2^j y - n),$$
$$\psi^H_{j,k,n}(x, y) = 2^j\varphi(2^j x - k)\psi(2^j y - n),$$
$$\psi^V_{j,k,n}(x, y) = 2^j\psi(2^j x - k)\varphi(2^j y - n),$$
$$\psi^D_{j,k,n}(x, y) = 2^j\psi(2^j x - k)\psi(2^j y - n).$$

Соответственно, вейвлет-коэффициенты сигнала $f(x, y)$ будут четырех типов. Коэффициенты

$$a_{j,k,n} = (f, \varphi_{j,k,n}) = 2^j \int_{\mathbf{R}^2} f(x,y)\overline{\varphi(2^j x - k)\varphi(2^j y - n)}\, dxdy$$

называются аппроксимирующими. Обозначим $cA_1 = \{a_{j,k,n}\}$ набор этих коэффициентов. Остальные коэффициенты – детализирующие трех типов. Они обозначаются символами cH_1, cV_1 и cD_1 и называются горизонтальными, вертикальными и диагональными детализирующими коэффициентами соответственно. Смысл таких названий объясняется следующим образом. Детализирующие вейвлет-коэффициенты $cH_1 = \{d^H_{j,k,n}\}$ функции $f(x, y)$ находятся по формуле

$$d^H_{j,k,n} = (f, \psi^H_{j,k,n}) = 2^j \int_{\mathbf{R}^2} f(x,y)\overline{\varphi(2^j x - k)\psi(2^j y - n)}\, dxdy.$$

По переменной x происходит интегрирование с масштабирующей функцией $\varphi(x)$, а это, как известно, вызывает сглаживание сигнала $f(x, y)$ по переменной x. Интегрирование по переменной y производится с участием вейвлет-функции $\psi(y)$, и в коэффициентах отражаются изменения функции по переменной y. В результате получается двумерный массив cH_1, в котором больше отражены изменения по вертикальной координате y. Поэтому более отчетливо будут видны горизонтальные полосы (см. рис. 2.11.2). Поэтому такие коэффициенты cH_1 называются горизонтальными. Аналогичный смысл и название имеют и другие вейвлет-коэффициенты: cV_1 – вертикальные, cD_1 – диагональные.

$$d^V_{j,k,n} = (f, \psi^V_{j,k,n}) = 2^j \int_{\mathbf{R}^2} f(x,y)\overline{\psi(2^j x - k)\varphi(2^j y - n)}\, dxdy,$$

$$d^D_{j,k,n} = (f, \psi^D_{j,k,n}) = 2^j \int_{\mathbf{R}^2} f(x,y)\overline{\psi(2^j x - k)\psi(2^j y - n)}\, dxdy.$$

Схема разложения сигнала C_0 может быть изображена в виде:

$$C_0 \rightarrow (cA_1, cH_1, cV_1, cD_1) \rightarrow (cA_2, cH_2, cV_2, cD_2, cH_1, cV_1, cD_1) \rightarrow \ldots$$

Поскольку массив начальных коэффициентов двумерный, то более естественно схему разложения сигнала C_0 изобразить в (принятом в MATLAB) виде.

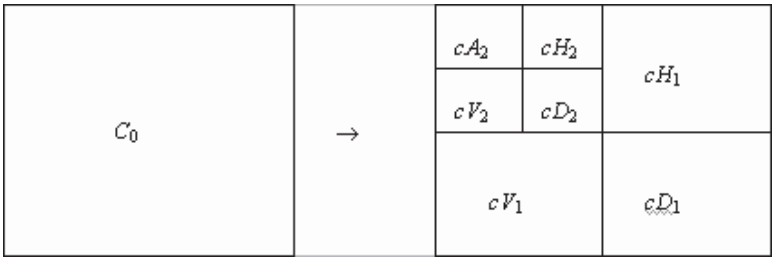


Рис. 2.11.1. Схема разложения двумерного сигнала

Заметим, что эта схема отражает изменение размеров массивов коэффициентов разложения. А именно если сигнал C_0 имел размерность $n \times n$, то коэффициенты первого уровня разложения cA_1, cH_1, cD_1, cV_1 будут иметь размерность $n/2 \times n/2$.

Приведем для иллюстрации рис. 2.11.2 вейвлет-разложения двумерного тестового сигнала «Geometry» системы MATLAB. Слева вверху – исходный сигнал, справа внизу изображены компоненты разложения, справа вверху – более крупное изображение выбранной компоненты разложения cA_2 – коэффициентов аппроксимации второго уровня разложения, слева внизу – восстановленный сигнал (он совпадает с исходным, поскольку обработка коэффициентов не производилась).

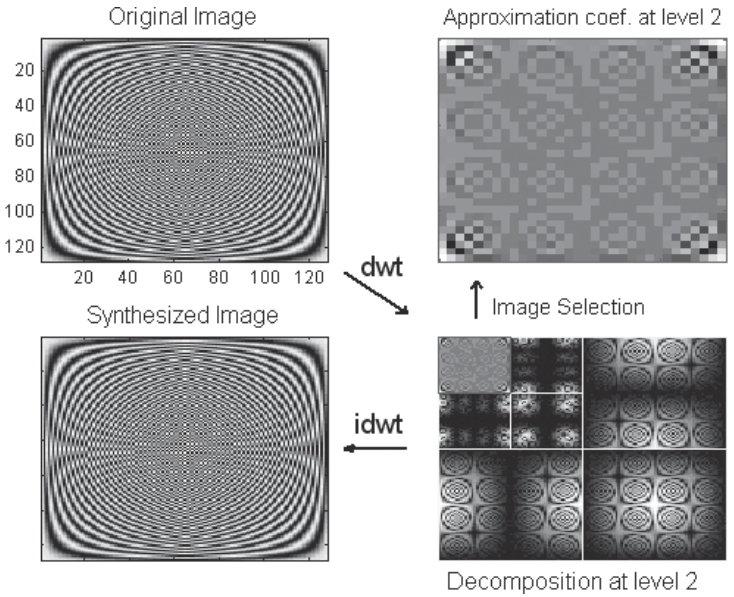


Рис. 2.11.2. Разложение двумерного тестового сигнала в MATLAB

Двумерное вейвлет-разложение можно применять не только к аппроксимирующим коэффициентам cA_j, но и ко всем остальным. Тогда мы приходим к пакетным вейвлетам. Их схему разложения принято изображать в виде дерева, аналогичного дереву пакетных одномерных разложений.

В заключение найдем масштабирующие соотношения для двумерных вейвлетов. Пусть $\varphi(x) = \sqrt{2}\sum_n h_n \varphi(2x-n)$ и $\psi(x) = \sqrt{2}\sum_n g_n \varphi(2x-n)$. Тогда

$$\varphi(x,y)=\varphi(x)\varphi(x)=\sqrt{2}\sum_{k\in\mathbf{Z}}h_k\varphi(2x-k)\sqrt{2}\sum_{n\in\mathbf{Z}}h_n\varphi(2x-n)=$$

$$=2\sum_{k,n\in\mathbf{Z}}h_kh_n\varphi(2x-k)\varphi(2x-n)=2\sum_{k,n\in\mathbf{Z}}h_kh_n\varphi(2x-k,2x-n).$$

Получаем масштабирующее соотношение вида:

$$\varphi(x,y)=2\sum_{k,n\in\mathbf{Z}}h_kh_n\varphi(2x-k,2x-n).$$

Для вейвлетов $\psi^H(x,\ y)=\varphi(x)\psi(y)$, $\psi^V(x,\ y)=\psi(x)\varphi(y)$ и $\psi^D(x,y)=\psi(x)\psi(y)$ имеем следующие формулы:

$$\psi^H(x,y)=2\sum_{k,n\in\mathbf{Z}}h_kg_n\varphi(2x-k,2x-n),$$

$$\psi^V(x,y)=2\sum_{k,n\in\mathbf{Z}}g_kh_n\varphi(2x-k,2x-n),$$

$$\psi^D(x,y)=2\sum_{k,n\in\mathbf{Z}}g_kg_n\varphi(2x-k,2x-n).$$

2.11.1. Вейвлет-преобразование

Получим формулы быстрого вейвлет-преобразования. Предположим, что мы знаем приближение $P_j(f)$ функции $f(x,y)$ в пространстве V_j^2:

$$P_j(f)=\sum_{k,n\in\mathbf{Z}}a_{k,n}\varphi_{j,k,n}(x,y),$$

где $a_{k,n}=(f,\varphi_{j,k,n})$. Тогда в соответствии с разложением

$$V_j^2=V_{j-1}^2\oplus W_{j-1}^H\oplus W_{j-1}^V\oplus W_{j-1}^D$$

имеем:

$$P_j(f)=\sum_{k,n\in\mathbf{Z}}a_{1,k,n}\varphi_{j-1,k,n}(x,y)+\sum_{k,n\in\mathbf{Z}}d_{1,k,n}^H\psi_{j,k,n}^H(x,y)+$$

$$+\sum_{k,n\in\mathbf{Z}}d_{1,k,n}^V\psi_{j,k,n}^V(x,y)+\sum_{k,n\in\mathbf{Z}}d_{1,k,n}^D\psi_{j,k,n}^D(x,y),$$

где $a_{1,k,n}=(f,\varphi_{j-1,k,n})$ и $d_{1,k,n}^H=(f,\psi_{j-1,k,n}^H)$, $d_{1,k,n}^V=(f,\psi_{j-1,k,n}^V)$, $d_{1,k,n}^D=(f,\psi_{j-1,k,n}^D)$. Выразим базисные функции $\varphi_{j-1,k,n}(x,\ y)$, $\psi_{j-1,k,n}^H(x,\ y)$, $\psi_{j-1,k,n}^V(x,\ y)$ и $\psi_{j-1,k,n}^D(x,\ y)$ в этих формулах через базисные функции $\varphi_{j,k,n}(x,\ y)$ пространства V_j^2 по формулам:

$$\varphi_{j-1,k,n}(x,y)=\sum_{p,q\in\mathbf{Z}}h_ph_q\,\varphi_{j,\,p+2k,q+2n}(x,y),$$

$$\psi_{j-1,k,n}^H(x,y)=\sum_{p,q\in\mathbf{Z}}h_pg_q\,\varphi_{j,\,p+2k,q+2n}(x,y),$$

$$\psi^V_{j-1,k,n}(x,y) = \sum_{p,q\in\mathbf{Z}} g_p h_q \varphi_{j,p+2k,q+2n}(x,y),$$

$$\psi^D_{j-1,k,n}(x,y) = \sum_{p,q\in\mathbf{Z}} g_p g_q \varphi_{j,p+2k,q+2n}(x,y).$$

Тогда получаем формулы вейвлет-разложения массива $A = \{a_{k,n}\}$:

$$a_{1,k,n} = (f,\varphi_{j-1,k,n}) = \left(f, \sum_{p,q\in\mathbf{Z}} h_p h_q \varphi_{j,p+2k,q+2n}\right) = \sum_{p,q\in\mathbf{Z}} \overline{h}_p \overline{h}_q a_{p+2k,q+2n},$$

$$d^H_{1,k,n} = (f,\psi^H_{j-1,k,n}) = \left(f, \sum_{p,q\in\mathbf{Z}} h_p g_q \varphi_{j,p+2k,q+2n}\right) = \sum_{p,q\in\mathbf{Z}} \overline{h}_p \overline{g}_q a_{p+2k,q+2n},$$

$$d^V_{1,k,n} = (f,\psi^V_{j-1,k,n}) = \left(f, \sum_{p,q\in\mathbf{Z}} g_p h_q \varphi_{j,p+2k,q+2n}\right) = \sum_{p,q\in\mathbf{Z}} \overline{g}_p \overline{h}_q a_{p+2k,q+2n},$$

$$d^D_{1,k,n} = (f,\psi^D_{j-1,k,n}) = \left(f, \sum_{p,q\in\mathbf{Z}} g_p g_q \varphi_{j,p+2k,q+2n}\right) = \sum_{p,q\in\mathbf{Z}} \overline{g}_p \overline{g}_q a_{p+2k,q+2n}.$$

Легко видеть, что восстановление массива $A = \{a_{k,n}\}$ по коэффициентам вейвлет-разложения $cA_1 = \{a_{1,k,n}\}$ и $cH_1 = \{d^H_{1,k,n}\}$, $cV_1 = \{d^V_{1,k,n}\}$, $cD_1 = \{d^D_{1,k,n}\}$ производится следующим образом:

$$a_{k,n} = \sum_{p,q\in\mathbf{Z}} h_{k-2p} h_{n-2q} a_{1,p,q} + \sum_{p,q\in\mathbf{Z}} h_{k-2p} g_{n-2q} d^H_{1,p,q} +$$
$$+ \sum_{p,q\in\mathbf{Z}} g_{k-2p} h_{n-2q} d^V_{1,p,q} + \sum_{p,q\in\mathbf{Z}} g_{k-2p} g_{n-2q} d^D_{1,p,q}.$$

2.12. Непрерывное вейвлет-преобразование

В этом параграфе мы определим непрерывное вейвлет-преобразование, установим его некоторые свойства и формулу обращения. Кроме того, мы рассмотрим вопросы многомерных обобщений непрерывного вейвлет-преобразования.

2.12.1. Непрерывное вейвлет-преобразование в одномерном случае

Непрерывное вейвлет-преобразование можно получить, если в выражении вейвлета $\psi_{j,k}(x) = 2^{j/2}\psi(2^j x - k)$ разрешить числам j и k принимать непрерывные значения, а суммы заменить на интегралы. Тогда мы получаем семейство функций $\psi_{a,b}(x) = \sqrt{a}\psi(az - b)$, зависящее от

двух непрерывных параметров a и b. Далее будет показано, что правильнее использовать следующее двухпараметрическое семейство функций:

$$\psi_{ab}(x) = \frac{1}{\sqrt{|a|}}\psi\left(\frac{x-b}{a}\right), \quad a,b \in \mathbf{R}, \quad a \neq 0. \tag{1}$$

Параметр b определяет сдвиг по оси Ox, параметр a – это *масштабный коэффициент*.

Непрерывное (интегральное) *вейвлет-преобразование* функции $f(x)$ из $L^2(\mathbf{R})$ определяется формулой

$$W_\psi[f](a,b) = (f, \psi_{a,b}) = |a|^{-1/2} \int_{-\infty}^{\infty} f(x)\overline{\psi\left(\frac{x-b}{a}\right)}dx. \tag{2}$$

Очевидно, что вейвлет-коэффициенты дискретного разложения функции $f(x)$ в ряд по вейвлетам $\psi_{j,k}(x)$ можно определить через интегральное вейвлет-преобразование:

$$c_{jk} = W_\psi[f]\left(\frac{1}{2^j}, \frac{k}{2^j}\right).$$

При использовании вейвлетов для анализа сигналов непрерывное вейвлет-преобразование иногда более удобно за счет избыточности, связанной с непрерывным изменением масштабного коэффициента a и параметра сдвига b.

Отметим, что семейство функций (1) и преобразование (2) можно определить для любой функции $\psi(x)$ из $L^2(\mathbf{R})$. Однако для того, чтобы из функции $W_\psi[f](a,b)$ двух переменных a и b можно было восстановить функцию $f(x)$, необходимо сделать некоторые предположения относительно $\psi(x)$. Во-первых, удобно считать функцию $\psi(x)$ нормированной:

$$\int_{-\infty}^{\infty} |\psi(x)|^2 dx = 1, \tag{3}$$

а во-вторых, для обращения вейвлет-преобразования (2) необходима конечность следующего интеграла:

$$C_\psi = \int_{-\infty}^{\infty} \frac{|\hat{\psi}(\omega)|^2}{|\omega|} d\omega < \infty. \tag{4}$$

Функцию $\psi(x)$ из $L^2(\mathbf{R})$, удовлетворяющую условиям (3) и (4), будем называть *вейвлетом* для непрерывного вейвлет-преобразования. Отметим, что здесь нет речи о масштабирующих функциях и

сами условия (3) и (4) достаточно слабые. Из (4) следует, что если $\psi(x) \in L^1(\mathbf{R})$, то

$$\hat{\psi}(0) = \int_{-\infty}^{\infty} \psi(x)dx = 0. \tag{5}$$

В противном случае интеграл (4) является расходящимся. Таким образом, условие (5) является необходимым. При дополнительном условии $x\psi(x) \in L^1(\mathbf{R})$ оно является и достаточным. Действительно, если $x\psi(x) \in L^1(\mathbf{R})$, то преобразование Фурье $\hat{\psi}(\omega)$ является непрерывно дифференцируемой функцией. Тогда, учитывая (5), получаем в окрестности нуля $\hat{\psi}(\omega) = A\omega + o(\omega)$. Поэтому особенность в интеграле (4) исчезает и интеграл будет сходящимся.

Пусть $\psi(x) \in L^2(\mathbf{R})$ – некоторый вейвлет для непрерывного вейвлет-преобразования.

Теорема 1. (Формула обращения.) *Если* $f(x) \in L^2(\mathbf{R})$, *то*

$$f(x) = C_\psi^{-1} \iint_{\mathbf{R}^2} W_\psi[f](a,b)\psi_{ab}(x)\frac{da\,db}{a^2}. \tag{6}$$

Доказательство. При доказательстве неоднократно будет использоваться равенство Планшереля

$$\int_{-\infty}^{\infty} f(x)\overline{g(x)}dx = \frac{1}{2\pi}\int_{-\infty}^{\infty} \hat{f}(\omega)\overline{\hat{g}(\omega)}d\omega.$$

Равенство (6) эквивалентно следующему интегральному равенству:

$$C_\psi(f,g) = \int_{\mathbf{R}} \left(\iint_{\mathbf{R}^2} W_\psi[f](a,b)\psi_{ab}(x)\frac{dadb}{a^2} \right) \overline{g(x)}\,dx$$

для любой функции $g(x) \in L^2(\mathbf{R})$. Это и будем доказывать. Сделаем следующие преобразования:

$$\iint_{\mathbf{R}^2} W_\psi[f](a,b)\overline{W_\psi[g](a,b)}\frac{dadb}{a^2} =$$

$$= \iint_{\mathbf{R}^2} W_\psi[f](a,b)\left(\int_{\mathbf{R}} \overline{g(x)}\psi_{ab}(x)\,dx \right)\frac{dadb}{a^2} =$$

$$= \int_{\mathbf{R}} \left(\iint_{\mathbf{R}^2} W_\psi[f](a,b)\psi_{ab}(x)\frac{dadb}{a^2} \right)\overline{g(x)}\,dx.$$

Поэтому для доказательства (6) достаточно показать, что

$$\iint_{\mathbf{R}^2} W_\psi[f](a,b)\overline{W_\psi[g](a,b)}\frac{dadb}{a^2} = C_\psi(f,g).$$

Для этого выразим вейвлет-преобразование, используя равенство Планшереля:

$$W_\psi[f](a,b) = |a|^{-1/2}\int_{-\infty}^{\infty} f(x)\overline{\psi\left(\frac{x-b}{a}\right)}dx = \frac{1}{2\pi}\int_{\mathbf{R}} |a|^{1/2}\hat{f}(\omega)e^{ib\omega}\overline{\hat{\psi}(a\omega)}d\omega.$$

Тогда, многократно используя формулу Планшереля, получаем:

$$\iint_{\mathbf{R}^2} W_\psi[f](a,b)\overline{W_\psi[g](a,b)}\frac{dadb}{a^2} = \iint_{\mathbf{R}^2} W_\psi[f](a,-b)\overline{W_\psi[g](a,-b)}\frac{dadb}{a^2} =$$

$$= \frac{1}{4\pi^2}\iint_{\mathbf{R}^2}\left(\int_{\mathbf{R}} |a|^{1/2}\hat{f}(\omega)e^{-ib\omega}\overline{\hat{\psi}(a\omega)}d\omega\right)\left(\int_{\mathbf{R}} |a|^{1/2}\overline{\hat{g}(\omega)}e^{ib\omega}\hat{\psi}(a\omega)d\omega\right)\frac{dadb}{a^2} =$$

$$= \frac{1}{4\pi^2}\iint_{\mathbf{R}^2}\left(\int_{\mathbf{R}} \hat{f}(\omega)\overline{\hat{\psi}(a\omega)}e^{-ib\omega}d\omega\right)\left(\int_{\mathbf{R}} \overline{\hat{g}(\omega)}\hat{\psi}(a\omega)e^{ib\omega}d\omega\right)\frac{dadb}{|a|} =$$

$$= \frac{1}{4\pi^2}\iint_{\mathbf{R}^2} F_{\omega\to b}[\hat{f}(\omega)\overline{\hat{\psi}(a\omega)}]\overline{F_{\omega\to b}[\hat{g}(\omega)\overline{\hat{\psi}(a\omega)}]}\frac{dadb}{|a|} =$$

$$= \frac{1}{4\pi^2}\int_{\mathbf{R}}\frac{da}{|a|}\int_{\mathbf{R}} F_{\omega\to b}[\hat{f}(\omega)\overline{\hat{\psi}(a\omega)}]\overline{F_{\omega\to b}[\hat{g}(\omega)\overline{\hat{\psi}(a\omega)}]}\,db =$$

$$= \frac{1}{2\pi}\int_{\mathbf{R}}\frac{da}{|a|}\int_{\mathbf{R}} \hat{f}(\omega)\overline{\hat{g}(\omega)}\,|\hat{\psi}(a\omega)|^2\,d\omega = \frac{1}{2\pi}\int_{\mathbf{R}} \hat{f}(\omega)\overline{\hat{g}(\omega)}\left(\int_{\mathbf{R}} |\hat{\psi}(a\omega)|^2\frac{da}{|a|}\right)d\omega =$$

[во внутреннем интеграле делаем замену переменной $a\omega = \zeta$, тогда интеграл в скобках равен C_ψ]

$$= \frac{1}{2\pi}\int_{\mathbf{R}} \hat{f}(\omega)\overline{\hat{g}(\omega)}(C_\psi)d\omega = C_\psi\int_{\mathbf{R}} f(x)\overline{g(x)}d\omega = C_\psi(f,g).$$

Замечание. Для обратимости непрерывного вейвлет-преобразования необходимо, чтобы $W[f(x)] \neq 0$, если $f(x) \neq 0$. Из формулы (2) следует, что это возможно только в том случае, когда система $\{\psi_{ab}(x),\ a \in \mathbf{R},\ a \neq 0,\ b \in \mathbf{R}\}$ является полной (не существует элемента $f(x) \in L^2(\mathbf{R})$, ортогонального всем $\psi_{ab}(x)$). Интересно, что это свойство системы $\{\psi_{ab}(x)\}$ обеспечивается одним интегральным условием (4).

Следствие. *Имеет место следующая формула Планшереля*:

$$\int_{\mathbf{R}} f(x)\overline{g(x)}dx = C_{\psi}^{-1}\iint_{\mathbf{R}^2} W_{\psi}[f](a,b)\overline{W_{\psi}[g](a,b)}\frac{dadb}{a^2}. \qquad (7)$$

Отметим два свойства непрерывного вейвлет-преобразования:

Сдвиг. $W[f(x-b_0)] = W[f](a, b-b_0)$.

Растяжение. $W\left[f\left(\frac{x}{a_0}\right)\right] = \frac{1}{a_0}W[f(x)]\left(\frac{a}{a_0},\frac{b}{a_0}\right)$.

2.12.2. Многомерные обобщения непрерывного вейвлет-преобразования

Рассмотрим непрерывное вейвлет-преобразование в случае пространства $\mathbf{R}^n$, то есть когда функции зависят от $n > 1$ переменных. Сначала напомним определение и основные свойства преобразования Фурье функции многих переменных.

Преобразование Фурье в $\mathbf{R}^n$. Пусть $f(x) \in L^1(\mathbf{R}^n) \cap L^2(\mathbf{R}^n)$. *Преобразование Фурье* функции $f(x)$ определяется формулой

$$\hat{f}(\omega) = \int_{\mathbf{R}^n} f(x)e^{-i(x,\omega)}d^n x, \qquad (8)$$

где $\omega = (\omega_1, \omega_2, ..., \omega_n)$, $x = (x_1, x_2, ..., x_n)$, $(x, \omega) = x_1\omega_1 + x_2\omega_2 + ... + x_n\omega_n$ и $d^n x = dx_1 dx_2 ... dx_n$.

Преобразование Фурье в $\mathbf{R}^n$ обладает аналогичными свойствами, что и в одномерном случае. Отметим некоторые из них.

Формула обращения:

$$f(x) = \frac{1}{(2\pi)^n}\int_{\mathbf{R}^n} \hat{f}(\omega)e^{i(x,\omega)}d^n\omega. \qquad (9)$$

Формула Планшереля:

$$\int_{\mathbf{R}^n} f(x)\overline{g(x)}d^n x = \frac{1}{(2\pi)^n}\int_{\mathbf{R}^n} \hat{f}(\omega)\overline{\hat{g}(\omega)}d^n\omega. \qquad (10)$$

Сдвиг пространственной переменной:

$$F[f(x-b)](\omega) = e^{-i(b,\omega)}\hat{f}(\omega). \qquad (11)$$

Действие линейного оператора на x. *Пусть $y = Ax$ – линейный невырожденный оператор в пространстве $\mathbf{R}^n$. Тогда*

$$F[f(Ax)](\omega) = \frac{1}{|\det(A)|}\hat{f}((A^t)^{-1}\omega), \qquad (12)$$

где $(A^t)^{-1}$ – обратная транспонированная матрица.

Доказательство.

$$F[f(Ax)](\omega) = \int_{\mathbf{R}^n} f(Ax)e^{-i(x,\omega)}d^n x = \left[\begin{array}{c} y = Ax, \\ x = A^{-1}y,\ I = \det(A^{-1}) \end{array}\right] =$$

$$= \int_{\mathbf{R}^n} f(y)e^{-i(A^{-1}y,\omega)} \,|\det(A^{-1})|\, d^n y = \frac{1}{|\det(A)|}\int_{\mathbf{R}^n} f(y)e^{-i(y,(A^{-1})^t\omega)}d^n y =$$

$$= \frac{1}{|\det(A)|}\hat{f}((A^t)^{-1}\omega).$$

В частности, преобразование Фурье функции

$$f\left(\frac{x-b}{a}\right) = f\left(\frac{x_1-b_1}{a_1},\ldots,\frac{x_n-b_n}{a_n}\right)$$

имеет вид:

$$F\left[f\left(\frac{x-b}{a}\right)\right](\omega) = e^{-i(\omega,b)}|a_1\cdots a_n|\hat{f}(a_1\omega_1,\ldots,a_n\omega_n). \qquad (13)$$

Прежде чем переходить к многомерным обобщениям, проанализируем одномерное вейвлет-преобразование с теоретико-групповой точки зрения. Введем два обозначения. Символом $\mathbf{R}^*$ будем обозначать группу по умножению ненулевых действительных чисел, $\mathbf{R}^* = \mathbf{R}\backslash\{0\}$, а символом $\mathbf{R}^+$ – группу по умножению положительных действительных чисел.

Аффинная группа

Преобразование вида $y = ax + b$, $a \neq 0$ – это действие *аффинной группы* $Aff(\mathbf{R})$ на пространстве $\mathbf{R}$. Аффинная группа $Aff(\mathbf{R})$ параметризуется двумя переменными $(a, b) \in \mathbf{R}^*\times\mathbf{R}$, здесь $\mathbf{R}$ – группа по сложению. Однако $Aff(\mathbf{R})$ не является прямым произведением групп $\mathbf{R}^*$ и $\mathbf{R}$. Действительно, если $z = a_1y + b_1$ и $y = a_2x + b_2$, то композиции этих преобразований есть $z = a_1(a_2x + b_2) + b_1 = a_1a_2x + a_1b_2 + b_1$. Поэтому закон перемножения элементов (a_1, b_1) и (a_2, b_2) следующий:

$$(a_1, b_1)\cdot(a_2, b_2) = (a_1a_2, a_1b_2 + b_1).$$

Такая структура называется *полупрямым произведением* групп $\mathbf{R}^*$ и $\mathbf{R}$.

Элементы (a, b) группы $Aff(\mathbf{R})$ удобно представлять как матрицы вида:

$$A = \begin{pmatrix} a & b \\ 0 & 1 \end{pmatrix}.$$

При этом если $z = a_1y + b_1$ и $y = a_2x + b_2$, то композиции этих преобразований $z = a_1(a_2x + b_2) + b_1 = a_1a_2x + a_1b_2 + b_1$ соответствует произведение матриц

$$A_1A_2 = \begin{pmatrix} a_1 & b_1 \\ 0 & 1 \end{pmatrix}\begin{pmatrix} a_2 & b_2 \\ 0 & 1 \end{pmatrix} = \begin{pmatrix} a_1a_2 & a_1b_2 + b_1 \\ 0 & 1 \end{pmatrix}.$$

Обратное преобразование для $y = ax + b$ имеет вид $x = (y - b)/a$, и ему соответствует обратная матрица

$$A^{-1} = \begin{pmatrix} 1/a & -b/a \\ 0 & 1 \end{pmatrix}.$$

Аффинная группа $Aff(\mathbf{R})$ параметризуется двумя переменными $(a, b) \in \mathbf{R}^*\times\mathbf{R}$. Если мы предполагаем интегрировать по этому пространству параметров, то в качестве элемента объема естественно взять $d\mu_0(A) = dadb$. Однако такой выбор будет не самым удачным. Дело в том, что данный элемент объема μ_0 не связан с групповой структурой на $Aff(\mathbf{R})$. В частности, $\mu_0 = dadb$ не будет инвариантным относительно левых сдвигов на группе $Aff(\mathbf{R})$. Выберем другой элемент объема:

$$d\mu(A) = \frac{da\,db}{a^2},$$

зависящий от точки $A \in Aff(\mathbf{R})$. Этот элемент объема уже будет левоинвариантным. Действительно, если

$$L_{A_0} : Aff(\mathbf{R}) \to Aff(\mathbf{R}),\ L_{A_0}(A) = A_0A$$

есть левый сдвиг на группе элементом A_0, то для выбранного элемента объема выполняется свойство:

$$L^*_{A_0} d\mu(A_0A) = d\mu(A).$$

Покажем это. Напомним [ДНФ], что в общем случае преобразование $y = F(x)$ действует на дифференциальных формах $df(x)$ как замена переменной, по формуле $F^*(df)(x) = d(f(F(x)))$. Пусть

$$C = A_0A = \begin{pmatrix} a_0 & b_{01} \\ 0 & 1 \end{pmatrix}\begin{pmatrix} a & b \\ 0 & 1 \end{pmatrix} = \begin{pmatrix} a_0a & a_0b + b_0 \\ 0 & 1 \end{pmatrix} = \begin{pmatrix} c & e \\ 0 & 1 \end{pmatrix},\quad d\mu(C) = \frac{dc\,de}{c^2}.$$

Тогда

$$L^*_{A_0} d\mu(A_0A) = L^*_{A_0}\frac{dc\,de}{c^2} = \frac{d(a_0a)\,d(a_0b + b_0)}{(a_0a)^2} = \frac{a_0da\,a_0db}{(a_0a)^2} = \frac{da\,db}{a^2} = d\mu(A).$$

Каждый элемент A аффинной группы $Aff(\mathbf{R})$ действует на функциях следующим образом: T_A: $f(x) \to f(ax + b)$, то есть $T_A(f) = f \circ A$. Такое действие, хотя и является вполне естественным, обладает двумя недостатками. Во-первых, композиции матриц не соответствует композиция операторов в том же порядке:

$$T_{AC}(f) = f \circ AC = f \circ A \circ C = T_C(f \circ A) = T_C T_A(f),\ T_{AC} = T_C T_A,$$

а во-вторых, оно не унитарно, то есть не сохраняет L^2-норму функции $f(x)$. Первый недостаток можно устранить, определив действие группы $Aff(\mathbf{R})$ на функциях через обратное преобразование: T_A: $f(x) \to \to f((x - b)/a)$, то есть $T_A(f) = f \circ A^{-1}$. В этом случае $T_{AC} = T_A T_C$. Унитарность достигается простым умножением на $|a|^{-1/2}$. Тогда оператор в пространстве $L^2(\mathbf{R})$, определенный формулой

$$U_A(f(x)) = \frac{1}{\sqrt{|a|}} f\left(\frac{x-b}{a}\right), \tag{14}$$

является унитарным – он сохраняет скалярное произведение. Действительно,

$$(U_A(f), U_A(g)) = \int_{-\infty}^{\infty} U_A(f(x))\overline{U_A(g(x))}\,dx = \int_{-\infty}^{\infty} \frac{1}{\sqrt{|a|}} f\left(\frac{x-b}{a}\right) \frac{1}{\sqrt{|a|}} \overline{g\left(\frac{x-b}{a}\right)}\,dx =$$

$$= \frac{1}{|a|}\int_{-\infty}^{\infty} f\left(\frac{x-b}{a}\right)\overline{g\left(\frac{x-b}{a}\right)}\,dx = \frac{1}{|a|}\int_{-\infty}^{\infty} f(y)\overline{g(y)}\,|a|\,dy = (f,g).$$

Таким образом, каждому аффинному преобразованию $A \in Aff(\mathbf{R})$ соответствует унитарный оператор в пространстве $L^2(\mathbf{R})$, определенный формулой (14). При этом легко видеть, что соответствие $A \to U_A$ обладает свойством

$$U_{AB} = U_A U_B. \tag{15}$$

Отметим еще два свойства:

- Единичной матрице E соответствует тождественное преобразование $L^2(\mathbf{R}), U_E = Id_{L^2(\mathbf{R})}$.
- Обратной матрице соответствует обратное преобразование $L^2(\mathbf{R}), U_{A^{-1}} = (U_A)^{-1}$.

Следовательно, соответствие $A \to U_A$ является *унитарным представлением* аффинной группы $Aff(\mathbf{R})$ в пространстве $L^2(\mathbf{R})$.

Для обобщения непрерывного вейвлет-преобразования на многомерный случай, то есть на пространство $\mathbf{R}^n$, $n > 1$, естественно вместо группы $Aff(\mathbf{R})$ взять группу $Aff(\mathbf{R}^n)$ всех аффинных преобразований $\mathbf{R}^n$. Она состоит из матриц вида

$$A_{n+1} = \begin{pmatrix} A & b \\ 0 & 1 \end{pmatrix},$$

где A – невырожденная матрица порядка n и $b \in \mathbf{R}^n$. Обратное преобразование задается матрицей:

$$A_{n+1}^{-1} = \begin{pmatrix} A^{-1} & -A^{-1}b \\ 0 & 1 \end{pmatrix}.$$

Унитарное представление аффинной группы $Aff(\mathbf{R}^n)$ в пространстве $L^2(\mathbf{R}^n)$ определяется формулой

$$U_{A_{n+1}}(f(x)) = \frac{1}{\sqrt{|\det A|}} f(A^{-1}(x-b)).$$

Непрерывное (интегральное) *вейвлет-преобразование* функции $f(x)$ из $L^2(\mathbf{R}^n)$ определяется формулой

$$W_\psi[f](A,b) = \frac{1}{\sqrt{|\det A|}} \int_{\mathbf{R}^n} f(x)\overline{\psi(A^{-1}(x-b))}\, d^n x.$$

В результате получается функция $W_\psi[f](A, b)$ на аффинной группе $Aff(\mathbf{R}^n)$.

Вейвлет-преобразование со сферически симметричным вейвлетом

Полная аффинная группа $Aff(\mathbf{R}^n)$ неудобна для вейвлет-преобразования вследствие большой размерности, равной $n^2 + n$. Рассмотрим некоторые подгруппы меньшей размерности, которые во многих случаях достаточны для непрерывного вейвлет-анализа. Наиболее простая группа, которая содержит растяжения и сдвиги $\mathbf{R}^n$, – это группа $A(\mathbf{R}^n)$, состоящая из матриц вида

$$A = \begin{pmatrix} a & 0 & \cdots & 0 & b_1 \\ 0 & a & \cdots & 0 & b_2 \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ 0 & 0 & \cdots & a & b_n \\ 0 & 0 & \cdots & 0 & 1 \end{pmatrix}, \quad \text{где } a \in \mathbf{R}^*, b \in \mathbf{R}^n. \tag{16}$$

В этом случае левоинвариантный элемент объема и унитарное представление задаются формулами:

$$d\mu(A) = \frac{da\, db_1 db_2 \ldots db_n}{a^{n+1}} = \frac{da\, d^n b}{a^{n+1}},$$

$$U_A(f(x)) = U_A(f(x_1,\ldots,x_n)) = \frac{1}{\sqrt{|a|^n}} f\left(\frac{x_1-b_1}{a},\cdots,\frac{x_n-b_n}{a}\right) = \frac{1}{\sqrt{|a|^n}} f\left(\frac{x-b}{a}\right).$$

Непрерывное вейвлет-преобразование определяем по аналогии с одномерным случаем. Для функции $\psi(x)$ вводим семейство функций, $\psi_{a,b}(x)$, зависящих от параметров (a, b) группы $A(\mathbf{R}^n)$:

$$\psi_{a,b}(x) = \frac{1}{\sqrt{|a|^n}} \psi\left(\frac{x-b}{a}\right), \quad a \in \mathbf{R}^*, b \in \mathbf{R}^n. \tag{17}$$

Формула *непрерывного* вейвлет-преобразования имеет вид:

$$W_\psi[f](a,b) = (f, \psi_{a,b}) = |a|^{-n/2} \int_{\mathbf{R}^n} f(x) \overline{\psi\left(\frac{x-b}{a}\right)} d^n x, \tag{18}$$

где $d^n x = dx_1 dx_2 \ldots dx_n$. В результате получилась функция на группе $A(\mathbf{R}^n)$, $(a, b) \in A(\mathbf{R}^n)$.

Имеет место формула обратного непрерывного вейвлет-преобразования

$$f(x) = C_\psi^{-1} \int_{\mathbf{R}^* \times \mathbf{R}^n} W_\psi[f](a,b) \psi_{ab}(x) \frac{da\, d^n b}{a^{n+1}} \tag{19}$$

при условии сферической симметричности функции $\psi(x)$, то есть $\psi(x) = \psi(\|x\|)$, и конечности следующего интеграла:

$$C_\psi = \int_{\mathbf{R}} \frac{|\hat{\psi}(\zeta)|^2}{|\zeta|} d\zeta < \infty. \tag{20}$$

Доказательство формулы обращения проводится точно так же, как и в одномерном случае (теорема 1), с использованием свойств (11), (12) преобразования Фурье и формулы Планшереля. Требование сферической симметричности появляется в необходимости сделать интеграл

$$\int_{\mathbf{R}} |\hat{\psi}(a\omega)|^2 \frac{da}{|a|} \tag{21}$$

заменой переменных $\zeta = a\|\omega\|, k = 1, \ldots, n$, не зависящим от $\omega = (\omega_1, \ldots, \omega_n)$. Это возможно, поскольку из свойства (12) следует, что преобразование Фурье сферически симметричной функции будет сферически симметричной функцией и поэтому $\hat{\psi}(a\omega)$ зависит на самом деле от $\|a\omega\|$, $\hat{\psi}(a\omega) = \hat{\psi}(\|a\omega\|)$.

Вейвлет-преобразование с различными сжатиями по осям

Рассмотрим чуть более сложную группу $A^n(\mathbf{R}^n)$, которая содержит различные растяжения по каждой координате $\mathbf{R}^n$ и сдвиги на вектор $b \in \mathbf{R}^n$. Она состоит из матриц вида

$$A = \begin{pmatrix} a_1 & 0 & \cdots & 0 & b_1 \\ 0 & a_2 & \cdots & 0 & b_2 \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ 0 & 0 & \cdots & a_n & b_n \\ 0 & 0 & \cdots & 0 & 1 \end{pmatrix}, \quad \text{где } a_k \in \mathbf{R}^*, b \in \mathbf{R}^n. \tag{22}$$

Очевидно, что эта группа $A^n(\mathbf{R}^n)$ является прямым произведением n экземпляров аффинной группы, $A^n(\mathbf{R}^n) = Aff(\mathbf{R}^n) \times ... \times Aff(\mathbf{R}^n)$. В этом случае левоинвариантный элемент объема и унитарное представление задаются формулами:

$$d\mu(A) = \frac{da_1 da_2 \ldots da_n\, db_1 db_2 \ldots db_n}{a_1^2 a_2^2 \ldots a_n^2} = \frac{d^n a\, d^n b}{a^2},$$

$$U_A(f(x)) = \frac{1}{\sqrt{|a_1 \ldots a_n|}} f\left(\frac{x_1 - b_1}{a_1}, \cdots, \frac{x_n - b_n}{a_n}\right) = \frac{1}{\sqrt{|a|}} f\left(\frac{x-b}{a}\right).$$

Действие этой группы – это действие группы $Aff(\mathbf{R}^n)$ отдельно по каждой координате. Поэтому непрерывное вейвлет-преобразование определяем по аналогии с одномерным случаем. Для функции $\psi(x)$ вводим семейство функций $\psi_{a,b}(x)$, зависящих от $2n$ параметров группы $A^n(\mathbf{R}^n)$:

$$\psi_{a,b}(x) = \frac{1}{\sqrt{|a_1 \ldots a_n|}} \psi\left(\frac{x-b}{a}\right), \quad a_k \in \mathbf{R}^*, b \in \mathbf{R}^n. \tag{23}$$

Формула *непрерывного* вейвлет-преобразования:

$$W_\psi[f](a,b) = (f, \psi_{a,b}) = |a_1 \ldots a_n|^{-/2} \int_{\mathbf{R}^n} f(x) \overline{\psi\left(\frac{x-b}{a}\right)} d^n x, \tag{24}$$

где $d^n x = dx_1 dx_2 ... dx_n$.

Имеет место следующая формула обратного непрерывного вейвлет-преобразования:

$$f(x) = C_\psi^{-1} \int_{A^n(R^n)} W_\psi[f](a,b) \psi_{ab}(x) \frac{d^n a\, d^n b}{a_1^2 \ldots a_n^2} \tag{25}$$

при условии конечности следующего интеграла:

$$C_\psi = \int_{R^n} \frac{|\hat{\psi}(\zeta)|^2}{|\zeta_1 \cdots \zeta_n|} d^n\zeta < \infty, \tag{26}$$

где $\zeta = (\zeta_1, ..., \zeta_n)$. Доказательство формулы обращения проводится точно так же, как и в одномерном случае (теорема 1), с использованием свойств (11), (12) преобразования Фурье и формулы Планшереля. Условие обращения (26) появляется при доказательстве (см. теорему 1), когда нужно сделать замену $\zeta_k = a_k\omega_k$, $k = 1, ..., n$, для освобождения от ω в интеграле

$$\int_R |\hat{\psi}(a_1\omega_1, \ldots a_n\omega_n)|^2 \frac{d^n a}{|a_1 \cdots a_n|}.$$

Вейвлет-преобразование с группой подобий

Рассмотрим группу подобий $SIM(\mathbf{R}^n)$ пространства $\mathbf{R}^n$, она состоит из переносов, вращений и растяжений. Элемент группы $SIM(\mathbf{R}^n)$ можно представить в виде произведения матриц:

$$A = \begin{pmatrix} aE_n & b \\ 0 & 1 \end{pmatrix} \begin{pmatrix} R & 0 \\ 0 & 1 \end{pmatrix}, \tag{27}$$

где $a \in \mathbf{R}^*$, E_n – единичная матрица порядка n, $b \in \mathbf{R}^n$, R – ортогональная матрица порядка n. Размерность группы $SIM(\mathbf{R}^n)$ достаточно большая, она равна $(n^2 + n + 2)/2$. Поэтому для простоты рассмотрим случай $n = 2$. Тогда получится четырехпараметрическая группа, состоящая из матриц

$$A = \begin{pmatrix} a & 0 & b_1 \\ 0 & a & b_2 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} \cos\theta & -\sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{pmatrix}. \tag{28}$$

Обозначим $R_\theta = \begin{pmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{pmatrix}$ матрицу поворота на угол θ. Поскольку $R_\pi = -1$, то в комбинации aR_θ нет необходимости рассматривать отрицательные значения параметра a. Поэтому в дальнейшем будем считать a положительным, $a > 0$. Левоинвариантный элемент объема и унитарное представление задаются формулами:

$$d\mu(A) = \frac{da\, db_1 db_2\, d\theta}{a^2} = \frac{da\, d^2b\, d\theta}{a^2},$$

$$U_A(f(x)) = U_A(f(x_1,x_2)) = \frac{1}{a} f\left(R_\theta^{-1}\left(\frac{x_1-b_1}{a}, \frac{x_2-b_2}{a} \right) \right) = \frac{1}{a} f\left(R_\theta^{-1}\left(\frac{x-b}{a} \right) \right).$$

Для функции $\psi(x)$ вводим семейство функций $\psi_{a,b,\theta}(x)$, зависящих от параметров группы $SIM(\mathbf{R}^n)$:

$$\psi_{a,b,\theta}(x) = \frac{1}{a} \psi\left(R_\theta^{-1}\left(\frac{x-b}{a} \right) \right), \tag{29}$$

где $a \in \mathbf{R}^+$, $b \in \mathbf{R}^2$, $\theta \in [0, 2\pi)$ и R_θ – матрица поворота на угол θ. Найдем преобразование Фурье данного семейства функций, учитывая, что $(R_\theta)^t = (R_\theta)^{-1}$:

$$\hat{\psi}_{a,b,\theta}(\omega) = F\left[\frac{1}{a} \psi\left(R_\theta^{-1}\left(\frac{x-b}{a} \right) \right) \right] = \frac{1}{a} e^{-i(\omega,b)} a^2 \hat{\psi}(R_\theta^{-1}(a\omega)) = \\ = a e^{-i(\omega,b)} \hat{\psi}(a R_\theta^{-1}(\omega)). \tag{30}$$

Непрерывное вейвлет-преобразование определяем по формуле

$$W_\psi[f](a,b,\theta) = (f, \psi_{a,b,\theta}) = a^{-1/2} \int_{\mathbf{R}^2} f(x) \overline{\psi\left(R_\theta^{-1}\left(\frac{x-b}{a} \right) \right)} d^2x, \tag{31}$$

где $d^2x = dx_1 dx_2$.

Имеет место следующая формула *обратного* непрерывного вейвлет-преобразования:

$$f(x) = C_\psi^{-1} \int_{SIM(R^2)} W_\psi[f](a,b,\theta) \psi_{a,b,\theta}(x) \frac{da\, d^2b\, d\theta}{a^2} \tag{32}$$

при условии конечности следующего интеграла:

$$C_\psi = \int_0^{+\infty} \frac{dr}{r} \int_0^{2\pi} |\hat{\psi}(r\cos\vartheta, r\sin\vartheta)|^2 d\vartheta < \infty. \tag{33}$$

Доказательство формулы обращения проводится точно так же, как и в одномерном случае (теорема 1), с использованием свойств (11), (12), (30) преобразования Фурье и формулы Планшереля. Условие обращения появляется при доказательстве (см. теорему 1) после замены переменных в интеграле

$$\int_0^{+\infty} \frac{da}{a} \int_0^{2\pi} |\hat{\psi}(a R_\theta^{-1}\omega)|^2 d\theta = \\ = \int_0^{+\infty} \frac{da}{a} \int_0^{2\pi} |\hat{\psi}(a\cos\theta\omega_1 + a\sin\theta\omega_2, -a\sin\theta\omega_1 + a\cos\theta\omega_2)|^2 d\theta$$

с целью освобождения от $\omega = (\omega_1, \omega_2)$. Требуемая замена:

$$\begin{cases} a\cos\theta\omega_1 + a\sin\theta\omega_2 & = r\cos\vartheta \\ -a\sin\theta\omega_1 + a\cos\theta\omega_2 & = r\sin\vartheta \end{cases}.$$

Другими словами, $ae^{i\theta}(\omega_1 - i\omega_2) = re^{i\vartheta}$. Тогда, если $\omega_1 - i\omega_2 = \|\omega\|e^{i\alpha}$, то $ae^{i\theta}\|\omega\|e^{i\alpha} = re^{i\vartheta}$. Из последнего равенства получаем: $r = a\|\omega\|$ и $\vartheta = \theta + \alpha$. Поэтому $dr = \|\omega\|\, da$ и $d\vartheta = d\theta$. Теперь делаем замену переменных и получаем интеграл (33).

Примеры двумерных вейвлетов

Приведем несколько известных примеров *двумерных* вейвлетов. Более подробно эта тема изложена в книге [За], там же описываются приложения двумерных вейвлетов к анализу изображений.

Двумерная «мексиканская шляпа». Это лапласиан функции Гаусса:

$$\psi(x) = -\Delta e^{-(x,x)/2} = (2 - (x, x))e^{-(x,x)/2}, \quad x = (x_1, x_2).$$

Можно рассмотреть также лапласианы более высокого порядка:

$$\psi^{(n)}(x) = (-\Delta)^n e^{-(x,x)/2}.$$

При увеличении n эти вейвлеты имеют все больше нулевых моментов и становятся все более чувствительными к сингулярным деталям сигнала. Сферически симметричный вейвлет «мексиканская шляпа» эффективен для анализа точечных особенностей и неэффективен для выделения направленных элементов сигнала.

Отметим так называемые градиентные вейвлеты. Они получаются как частные производные функции Гаусса:

$$\psi(x) = \partial_1 e^{-(x,x)/2}, \quad \psi(x) = \partial_1\partial_2 e^{-(x,x)/2}.$$

Разностные вейвлеты. Интересный класс вейвлетов получается как разность двух положительных функций, в частности как разность некоторой функции $\varphi(x)$ и ее сжатой версии. Если функция $\varphi(x)$ есть гладкая неотрицательная функция и $\varphi(x) \in L^1(\mathbf{R}^2) \cap L^2(\mathbf{R}^2)$ и все ее моменты первого порядка равны нулю, то функция $\psi(x)$, задаваемая соотношением

$$\psi(x) = \alpha^{-2}\varphi(x/\alpha) - \varphi(x),\ 0 < \alpha < 1,$$

является вейвлетом, удовлетворяющим условию допустимости (33). Типичным примером служит так называемый DOG-вейвлет (Difference of Gaussians), когда в качестве $\varphi(x)$ берется функция Гаусса:

$$\psi(x) = \alpha^{-2}e^{-(x,x)/2\alpha^2} - e^{-(x,x)/2}, \quad 0 < \alpha < 1.$$

Двумерный вейвлет Морле. Вейвлет, который не обладает сферической симметричностью, можно назвать *направленным*. Направленность вейвлета удобно задавать матрицей A в скалярном произведении, когда вместо (x, x) берется (Ax, Ax). Вейвлет Морле является простейшим направленным вейвлетом и может рассматриваться как прототип всех других направленных вейвлетов:

$$\psi(x) = \exp(i(\xi_0, x))\exp(-(Ax, Ax)/2),$$

где параметр ξ_0 является волновым вектором и A – матрица, определяющая анизотропию вейвлета. Например, в качестве такой матрицы можно взять $A = \begin{pmatrix} \varepsilon & 0 \\ 0 & 1 \end{pmatrix}$. Хотя вейвлет Морле не удовлетворяет условию обратимости вейвлет-преобразования, но при больших значениях $\|\xi_0\|$ значение $\hat{\psi}(0)$ близко к нулю, что позволяет его практическое использование. Модуль вейвлета Морле является функцией Гаусса, вытянутой в направлении оси x (если $\varepsilon < 1$), и его фаза постоянна в направлении, ортогональном ξ_0. Таким образом, вейвлет Морле обнаруживает резкие изменения сигнала в направлении, перпендикулярном ξ_0.

Замечание. Более подробно о двумерных направленных ориентированных вейвлетах и примерах их использования см. в книге [За] или в работах [An], [AJV], [AMV]. Об n-мерных обобщениях непрерывного вейвлет-преобразования см. в работе [FO].

Вейвлеты на однородном пространстве

Обобщения вейвлет-преобразования на случай более сложных пространств следуют одной общей схеме. В качестве пространства рассматривается многообразие M с фиксированным элементом объема dx. На многообразии M транзитивно действует группа Ли G (таким образом, M – однородное пространство). Кроме транзитивности, необходимо еще предполагать, что для любой точки $x \in M$ существуют преобразования из группы G, которые являются растяжениями в окрестности точки x. Определено представление группы Ли G в пространстве функций $L^2(M, dx)$ формулой $(T_g f)(x) = f(g^{-1}x)$, $g \in G$. Пусть $\psi(x)$ – некоторая функция с нулевым средним на M. Рассмотрим функцию $\psi(g, x)$ двух переменных $x \in M$ и $g \in G$, определенную формулой $\psi(g, x) = \lambda(g, x)\psi(g^{-1}x)$, где $\lambda(g, x)$ – некоторая функция, обеспечивающая унитарность представления $(U_g\psi)(x) = \lambda(g, x)\psi(g^{-1}x)$ группы Ли G. Тогда непрерывное вейвлет-преобразование определяется формулой

$$W_{\psi}[f](g) = \int_{M} f(x)\overline{\psi(g,x)}\,dx,$$

где dx – фиксированный элемент объема на M. В результате получается функция на группе Ли G. В случае унитарности и неприводимости представления $\psi(g, x) = \lambda(g, x)\psi(g^{-1}x)$ можно ожидать, что обратное вейвлет-преобразование будет иметь вид:

$$f(x) = C\int_{G} W_{\psi}[f](g)\psi(g,x)\,dg,$$

где dg – инвариантный элемент объема на группе Ли G.

В такой общей постановке задача обобщения непрерывного вейвлет-преобразования представляется слишком неопределенной. Однако если многообразие M достаточно простое, например окружность, сфера или тор, то все конструкции могут быть уточнены и развиты в достаточной мере, с тем чтобы использовать непрерывное вейвлет-преобразование на M для решения задач медицины, геофизики и астрономии. Подробнее о непрерывном вейвлет-преобразовании на сфере и на других многообразиях см. в [AV1], [AV2], [Bog], [BVAJM] и [См1]. В работе [AV2] развита теория непрерывного вейвлет-преобразования на сфере размерности n и указаны подходы к определению вейвлет-преобразования на симметрических пространствах. В диссертации И. Богдановой [Bog] изучены вейвлеты на сфере, гиперболоиде и приведены примеры вейвлет-анализа неевклидовых изображений. Более современное состояние исследований в этом направлении можно найти в красочной презентации [McE], в работах [SMAN], [RA], [SLBD] и в диссертации [L], а также в приведенной в этих работах литературе.

2.13. Вейвлеты с коэффициентом масштабирования *N*

Использование степеней двойки при построении теории вейвлетов удобно во многих отношениях, хотя и не является обязательным. Можно вместо 2 применять любое целое число и даже рациональное, большее единицы [Дб, стр. 413 и 418]. При этом имеется одна особенность – если параметр сжатия есть целое число, большее двух, то масштабирующей функции $\varphi(x)$ соответствует несколько вейвлетов $\psi(x)$. Рассмотрим для примера случай параметра сжатия $N = 3$. Пусть $\varphi(x)$ – масштабирующая функция,

$$\varphi(x) = \sqrt{3}\sum_n h_n \varphi(3x - n),$$

целочисленные сдвиги $\{\varphi(x - n)\}$ которой порождают ортонормированный базис пространства V_0. Пространство V_1 порождается функциями вида $\{\varphi(3x - n)\}$, то есть сдвигами функции $\varphi(3x)$ на числа, кратные 1/3. Эквивалентно V_1 порождается целочисленными сдвигами трех функций $\varphi(3x)$, $\varphi(3x - 1)$ и $\varphi(3x - 2)$. Поэтому пространство V_1 «втрое больше», чем V_0. Необходимы два пространства «того же размера», что и V_0, чтобы дополнить V_0 до пространства V_1. Следовательно, ортогональное дополнение к V_0 будет состоять из двух пространств W_0^1 и W_0^2, базисы которых порождены двумя вейвлетами $\psi^1(x)$ и $\psi^2(x)$. Соответственно, получаются два соотношения

$$\psi^1(x) = \sqrt{3}\sum_n g_n^1 \varphi(3x - n), \qquad \psi^2(x) = \sqrt{3}\sum_n g_n^2 \varphi(3x - n).$$

Вейвлеты с параметром масштабирования N, отличным от 2, естественно использовать при анализе временных рядов, возникающих в экономических задачах. Дело в том, что они имеют периодичность по времени. Обычными примерами таких рядов являются массивы данных об объемах, или ценах продаж товарного или фондового рынка. Например, при анализе 15-минутных данных о цене акции естественно использовать вейвлеты с параметром масштабирования $N = 4$ (поскольку в часе таких периодов 4). Далее, при анализе часовых данных о цене ценной бумаги фондового рынка естественно использовать вейвлеты с параметром масштабирования $N = 8$ (поскольку день работы рынка содержит 8 таких отсчетов). При анализе дневных данных естественно использовать вейвлеты с параметром масштабирования $N = 5$ (поскольку неделя работы рынка содержит 5 рабочих дней). При анализе дневных показателей продаж товара естественно использовать вейвлеты с параметром масштабирования $N = 7$ (недельный цикл), с отдельной выборкой данных за выходные дни.

Рассмотрим основные положения N-кратномасштабного анализа и приведем примеры масштабирующих функций и вейвлетов с параметром масштабирования $N > 1$, следуя работам [BJ], [J03], [По1], [По2], [PS1] и [PS2].

2.13.1. Масштабирующие функции

Пусть $N > 1$ – целое число.

Определение 1. *Функция* $\varphi(x) \in L^2(\mathbf{R})$ *называется N-масштабирующей, если она может быть представлена в виде*

$$\varphi(x) = \sqrt{N} \sum_{n \in \mathbf{Z}} h_n \varphi(Nx - n), \tag{1}$$

где числа h_n, $n \in \mathbf{Z}$, *удовлетворяют условию* $\Sigma_{n \in \mathbf{Z}} |h_n|^2 < \infty$. *Равенство* (1) *называется масштабирующим уравнением. Набор* $\{h_n\}$ *коэффициентов разложения называется масштабирующим фильтром.*

Замечание 1. Если масштабирующая функция $\varphi(x)$ имеет компактный носитель, лежащий на промежутке длины L, то сумма в равенстве (1) является конечной, состоящей не более чем из $L(N-1)+1$ слагаемого.

Пусть $\varphi(x)$ – масштабирующая функция. Образуем следующие функции:

$$\varphi_{j,n}(x) = \sqrt{N^j}\, \varphi(N^j x - n), \quad j, n \in \mathbf{Z}. \tag{2}$$

Сомножитель $\sqrt{N^j}$ включен с той целью, чтобы L^2-норма каждой такой функции совпадала бы с нормой $\varphi(x)$.

Теорема 1. *Для любых* $j, k \in \mathbf{Z}$ *имеет место следующее разложение:*

$$\varphi_{j-1,k}(x) = \sum_n h_{n-Nk} \varphi_{j,n}(x) = \sum_n h_n \varphi_{j,n+Nk}(x). \tag{3}$$

В частности,

$$\varphi_{0,k}(x) = \sqrt{N} \sum_n h_{n-Nk} \varphi(Nx - n), \qquad \varphi_{j-1,0}(x) = \sum_n h_n \varphi_{j,n}(x). \tag{4}$$

Доказательство точно такое же, что и доказательство предложения 1 из раздела 2.2.1.

Сделаем преобразование Фурье масштабирующего соотношения (1). Поскольку $\varphi(Nx - n) = \varphi(N(x - n/N))$, то $F[\varphi(Nx - n)] = e^{-in\omega/N} F[\varphi](\omega/N)/N$. Поэтому

$$\hat{\varphi}(\omega) = \frac{1}{\sqrt{N}} \sum_{n \in \mathbf{Z}} h_n e^{-in\frac{\omega}{N}} \hat{\varphi}\left(\frac{\omega}{N}\right).$$

Пусть

$$H_0(\omega) = \frac{1}{\sqrt{N}} \sum_{n \in \mathbf{Z}} h_n e^{-in\omega}, \tag{5}$$

тогда

$$\hat{\varphi}(\omega) = H_0\left(\frac{\omega}{N}\right) \hat{\varphi}\left(\frac{\omega}{N}\right). \tag{6}$$

Соотношение (6) также называется масштабирующим уравнением. Функцию $H_0(\omega)$ будем называть частотной функцией масштаби-

рующей функции $\varphi(x)$. Из (6) следует, что если функция $\hat{\varphi}(\omega)$ непрерывна и интеграл от функции $\varphi(x)$ положителен, то

$$H_0(0) = \frac{1}{\sqrt{N}} \sum_n h_n = 1. \tag{7}$$

При известных коэффициентах $\{h_n\}$ масштабирующая функция $\varphi(x)$ может быть построена так же, как и в разделе 2.2.2: итерационным методом, рекуррентным методом и через бесконечное произведение. Для бесконечного произведения совершенно аналогично доказывается следующая теорема.

Теорема 2. *Если* $\Sigma_{n\in\mathbf{Z}}|h_n||n| < \infty$ *и* $H_0(0) = 1$, *тогда бесконечное произведение*

$$\hat{\varphi}(\omega) = \prod_{j=1}^{\infty} H_0\left(\frac{\omega}{N^j}\right) \tag{9}$$

сходится абсолютно и равномерно на компактных подмножествах из $\mathbf{R}$.

В этой теореме считается что функция $\varphi(x)$ нормирована так, что $\hat{\varphi}(0) = \int_{-\infty}^{\infty} \varphi(x)\,dx = 1$.

Замечание 2. Теорема 2 дает достаточно слабые требования на коэффициенты $\{h_n\}$. Однако не следует, что эти коэффициенты $\{h_n\}$ можно выбирать произвольно с учетом ограничений теоремы. Дело в том, что бесконечное произведение, даже если оно сходится, может не принадлежать $L^2(\mathbf{R})$. Например, произведение

$$\hat{\varphi}(\omega) = e^{-i\omega/2} \prod_{n=1}^{\infty} \cos\left(\frac{\omega}{3^n}\right)$$

сходится поточечно на $\mathbf{R}$ и равномерно на компактных подмножествах числовой прямой. Однако функция $\hat{\varphi}(\omega)$ не принадлежит $L^2(\mathbf{R})$, поскольку соответствующая ей функция $\varphi(x)$ является сингулярной мерой, сосредоточенной в точках канторова множества на отрезке [0, 1]. Такая функция $\varphi(x)$ не является даже локально интегрируемой. Подробнее об этом см. раздел 2.14.3.

2.13.2. N-кратномасштабное разложение

Дадим здесь определение ортогонального *N-кратномасштабного* уточняющего разложения пространства $L^2(\mathbf{R})$ и напомним некоторые свойства масштабирующей функции, необходимые для ортогональности.

Определение 2. *Ортогональным N-кратномасштабным разложением пространства* $L^2(\mathbf{R})$ *называется последовательность замкнутых вложенных друг в друга подпространств*

$$\ldots \subset V_{-1} \subset V_0 \subset V_1 \subset V_2 \subset \ldots,$$

обладающая свойствами:

1. *Замыкание их объединения совпадает с* $L^2(\mathbf{R})$, $\overline{\bigcup_{j\in\mathbf{Z}} V_j} = L^2(\mathbf{R})$.
2. *Пересечение всех подпространств – нулевое,* $\bigcap_{j\in\mathbf{Z}} V_j = \{0\}$.
3. $f \in V_j \Leftrightarrow f(Nx) \in V_{j+1}$.
4. *Существует такая функция* $\varphi(x) \in V_0$, *что функции* $\varphi_{0n}(x) = \varphi(x - n)$, $n \in \mathbf{Z}$, *образуют ортонормированный базис пространства* V_0.

Отметим несколько следствий, которые сразу вытекают из определения. Поскольку пространства V_j являются масштабированными версиями пространства V_0, то из свойства 4 сразу вытекает, что функции $\varphi_{j,n}(x) = \sqrt{N^j}\varphi(N^j x - n)$, $n \in \mathbf{Z}$, образуют ортонормированный базис пространства V_j для любого j. Из того, что $\varphi(x) \in V_0 \subset V_1$, следует, что функция $\varphi(x)$ раскладывается по базису пространства V_1:

$$\varphi(x) = \sqrt{N}\sum_n h_n \varphi(Nx - n),$$

то есть является N-масштабирующей функцией. В кратномасштабном разложении функция $\varphi(x)$ нормируется так, чтобы $\|\varphi\|_{L^2} = 1$.

Замечание 3. В случае ортогонального N-кратномасштабного разложения коэффициенты масштабирующего фильтра $\{h_n\}$ обладают свойством

$$\sum_n h_n \overline{h_{n-Nk}} = \delta_{0k}, \tag{10}$$

которое сразу следует из равенства $\varphi_{0,k}(x) = \sum_n h_{n-Nk}\varphi_{1,n}(x)$ (формула (4)) и из того, что $(\varphi_{0,0}, \varphi_{0,k}) = \delta_{0k}$. В частности,

$$\sum_n |h_n|^2 = 1. \tag{11}$$

Как обычно, функции $\varphi_{0,n}(x) = \varphi(x - n)$ образуют ортонормированный базис подпространства $V_0 \subset L^2(\mathbf{R})$ тогда и только тогда, когда

$$\sum_{n\in Z} |\hat{\varphi}(\omega + 2\pi n)|^2 = 1 \text{ } п.\,в. \tag{12}$$

Более простое условие ортогональности дает

Теорема 3. *Если сдвиги* $\varphi_n(x) = \varphi(x - n)$ *масштабирующей функции* $\varphi(x)$ *образуют ортонормированный базис подпространства* V_0*, то частотная функция* $H_0(\omega)$ *обладает следующим свойством*:

$$\sum_{m=0}^{N-1} |H_0(\omega + 2\pi m / N)|^2 = 1 \ \textit{п. в.} \tag{13}$$

Теорема доказывается точно так же, как и аналогичная теорема 2 из раздела 2.3.1.

Замечание 4. Условия ортогональности (10) и (13) эквивалентны. Действительно, разложим левую часть равенства (13) в ряд Фурье:

$$|H_0(\omega + 2\pi s / N)|^2 = \frac{1}{\sqrt{N}} \sum_{n \in \mathbf{Z}} h_n e^{-i n \omega} e^{-i n 2\pi s / N} \frac{1}{\sqrt{N}} \sum_{m \in \mathbf{Z}} \overline{h_m} e^{i m \omega} e^{i m 2\pi s / N} =$$

$$= \frac{1}{N} \sum_{n,m \in \mathbf{Z}} (e^{-i 2\pi s / N})^{n-m} h_n \overline{h_m} e^{-i(n-m)\omega} = \frac{1}{N} \sum_{k \in \mathbf{Z}} (e^{-i 2\pi s / N})^k \sum_n h_n \overline{h_{n-k}} e^{-i k \omega},$$

$$H_0(\omega)|^2 + |H_0(\omega + \pi / N)|^2 + \cdots + |H_0(\omega + \pi(N-1)/N)|^2 =$$

$$= \frac{1}{N} \sum_{k \in \mathbf{Z}} (1 + (e^{-i 2\pi / N})^k + (e^{-i \pi 4 / N})^k + \cdots + (e^{-i 2\pi (N-1)/N})^k) \sum_n h_n \overline{h_{n-k}} e^{-i k \omega} =$$

$$= \sum_{p \in \mathbf{Z}} \sum_{n \in \mathbf{Z}} h_n \overline{h_{n-Np}} e^{-i N p \omega}.$$

Поэтому равенство (13) выполняется только тогда, когда $\sum_n h_n \overline{h_{n-Np}} = \delta_{0p}$. Мы использовали свойство $1 + \rho + \rho^2 + \ldots + \rho^{N-1} = 0$, которое верно для любого корня $\rho \neq 1$ из единицы степени N. Если $k = Np + s$, то $\rho = e^{-i2\pi s/N}$.

2.13.3. Вейвлеты с коэффициентом масштабирования *N*

Пусть $N > 1$ – некоторое натуральное число. Предположим, что задано ортогональное N-кратномасштабное разложение пространства $L^2(\mathbf{R})$

$$\ldots \subset V_{-1} \subset V_0 \subset V_1 \subset V_2 \subset \ldots,$$

определенное масштабирующей функцией $\varphi(x) \in V_0$.

Определение 3. *Функции* $\psi^1(x), \ldots, \psi^{N-1}(x) \in L^2(\mathbf{R})$ *называются материнскими вейвлетами для* N*-масштабирующей функции* $\varphi(x)$*, если их целочисленные сдвиги образуют ортонормированные базисы подпространств* $W_0^1, W_0^2, \ldots, W_0^{N-1}$ *в* $L^2(\mathbf{R})$*, которые вместе с* V_0 *образуют ортогональное разложение пространства* V_1:

$$V_1 = V_0 \oplus W_0^1 \oplus \ldots \oplus W_0^{N-1}. \tag{14}$$

Для любого j получаем следующее ортогональное разложение:

$$V_{j+1} = V_j \oplus W_j^1 \oplus \ldots \oplus W_j^{N-1}. \tag{15}$$

Функции

$$\psi_{j,n}^l(x) = \sqrt{N^j}\,\psi^l(N^j x - n), \quad j,n \in \mathbf{Z}, \tag{16}$$

образуют ортонормированные базисы пространств вейвлетов W_j^l, $l = 1, 2, \ldots, N-1$. Поскольку $\psi^l(x) \in W_0^1 \subset V_1$, то каждый вейвлет $\psi^l(x)$ раскладывается по базису функций пространства V_1:

$$\psi^l(x) = \sqrt{N}\sum_n g_n^l \varphi(Nx - n), \quad l = 1, 2, \ldots, N-1. \tag{17}$$

Числа $\{g_n^l\}$ называются фильтрами вейвлетов $\psi^l(x)$. Определим соответствующие вейвлетам $\psi^l(x)$ частотные функции:

$$H_l(\omega) = \frac{1}{\sqrt{N}}\sum_n g_n^l\, e^{-in\omega}, \quad l = 1, 2, \ldots, N-1. \tag{18}$$

В частотной области имеем:

$$\psi^l(\omega) = H_l\left(\frac{\omega}{N}\right)\hat{\varphi}\left(\frac{\omega}{N}\right), \quad l = 1, 2, \ldots N-1. \tag{19}$$

Для частотных функций $H_0(\omega)$, $H_1(\omega)$, ..., $H_{N-1}(\omega)$ ортогональных вейвлетов $\varphi(x)$, $\psi^1(x)$, ..., $\psi^{N-1}(x)$ следующая матрица является унитарной [Дб], [BJ]:

$$\begin{pmatrix} H_0(\omega) & H_0\left(\omega+\frac{2\pi}{N}\right) & \cdots & H_0\left(\omega+\frac{2\pi(N-1)}{N}\right) \\ H_1(\omega) & H_1\left(\omega+\frac{2\pi}{N}\right) & \cdots & H_1\left(\omega+\frac{2\pi(N-1)}{N}\right) \\ \cdots & \cdots & \cdots & \cdots \\ H_{N-1}(\omega) & H_{N-1}\left(\omega+\frac{2\pi}{N}\right) & \cdots & H_{N-1}\left(\omega+\frac{2\pi(N-1)}{N}\right) \end{pmatrix} \tag{20}$$

Из унитарности данной матрицы сразу следует, что коэффициенты вейвлетов $\{h_n\}$ и $\{h_n^l\}$ обладают свойствами:

$$\sum_{n\in\mathbf{Z}} |g_n^l|^2 = 1, \quad \sum_{n\in\mathbf{Z}} \overline{g_n^s} g_{n-Nk}^l = 0, \quad \forall\, s \neq l, \quad s, l = 0, 1, \ldots, N-1,$$

где мы считаем, что $g_n^0 = h_n$.

2.13.4. Вейвлет-преобразование

Получим формулы быстрого вейвлет-преобразования для ортогонального N-кратномасштабного разложения

$$... \subset V_{-1} \subset V_0 \subset V_1 \subset V_2 \subset ...,$$

определенного N-масштабирующей функцией $\varphi(x)$. Пусть $\psi^1(x)$, ..., $\psi^{N-1}(x)$ – вейвлеты, определяющие разложение $V_1 = V_0 \oplus W_0^1 \oplus ... \oplus W_0^{N-1}$.

Предположим, что мы знаем приближение $P_j(f)$ функции $f(x)$ в пространстве V_j:

$$P_j(f) = \sum_{k \in Z} a_{j,k} \varphi_{j,k}(x),$$

где $a_{j,k} = (f, \varphi_{j,k})$. Тогда в соответствии с разложением

$$V_j = V_{j-1} \oplus W_{j-1}^1 \oplus ... \oplus W_{j-1}^{N-1}$$

имеем:

$$P_j(f) = P_{j-1}(f) + Q_{j-1}^1(f) + ... + Q_{j-1}^{N-1}(f)$$

и

$$P_j(f) = \sum_{k \in Z} a_{j-1,k} \varphi_{j-1,k}(x) + \sum_{l=1}^{N-1} \sum_{k \in Z} d_{j-1,k}^l \psi_{j-1,k}^l(x), \qquad (21)$$

где $a_{j-1,k} = (f, \varphi_{j-1,k})$ и $d_{j-1,k}^l = (f, \psi_{j-1,k}^l)$, $l = 1, 2, ..., N-1$.

При быстром вейвлет-преобразовании этот первый шаг разложения делается без интегрирования следующим образом. Выразим базисные функции $\varphi_{j-1,k}(x)$ и $\psi_{j-1,k}^l(x)$, по которым должно быть проведено разложение, через базисные функции $\varphi_{j,k}(x)$ пространства V_j:

$$\varphi_{j-1,k}(x) = \sum_n h_n \varphi_{j,n+Nk}(x),$$

$$\psi_{j-1,k}^l(x) = \sum_n g_n^l \varphi_{j,n+Nk}(x), \quad l = 1, 2, \ldots, N-1,$$

где $\{h_n\}_{n \in \mathbf{Z}}$ и $\{g_n^l\}_{n \in \mathbf{Z}}$ – фильтры вейвлетов $\varphi(x)$ и $\psi^l(x)$. Тогда получаем:

$$a_{j-1,k} = (f, \varphi_{j-1,k}) = \left(f, \sum_n h_n \varphi_{j,n+Nk} \right) = \sum_n \overline{h_n} a_{j,n+Nk},$$

$$d_{j-1,k}^l = (f, \psi_{j-1,k}^l) = \left(f, \sum_n g_n^l \varphi_{j,n+Nk} \right) = \sum_n \overline{g_n^l} a_{j,n+Nk}, \; l = 1, 2, \ldots, N-1.$$

Получили формулы *вейвлет-разложения* для ортогональных вейвлетов с коэффициентом масштабирования N:

$$a_{j-1,k} = \sum_n \overline{h}_n a_{j,n+Nk}, \quad d^l_{j-1,k} = \sum_n \overline{g^l_n} a_{j,n+Nk}, \; l = 1, 2, \ldots, N-1. \tag{22}$$

Обратите внимание на индекс Nk в сумме справа, он говорит о проведенной N-адической децимации, то есть выборке элементов с номерами Nk после операции свертки.

Таким образом, для сигнала, заданного массивом $A = \{a_n\}$, вейвлет-разложение производится по формулам:

$$a_{1,k} = \sum_n \overline{h}_n a_{n+Nk}, \quad d^l_{1,k} = \sum_n \overline{g^l_n} a_{n+Nk}, \; l = 1, 2, \ldots, N-1, \tag{23}$$

где $\{h_n\}$ и $\{g^l_n\}$ – фильтры вейвлетов $\varphi(x)$ и $\psi^i(x)$. Запишем последние формулы в виде свертки. Для этого введем следующие коэффициенты: $h^*_n = \overline{h_{-n}}$, $g^{*l}_n = \overline{g^l_{-n}}$. Тогда

$$a_{1,k} = \sum_n h^*_n a_{Nk-n}, \quad d^l_{1,k} = \sum_n g^{*l}_n a_{Nk-n}, \; l = 1, 2, \ldots, N-1. \tag{24}$$

Из данных формул следует, что вейвлет-разложение производится сопряженными фильтрами $\{h^*_n\}$, $\{g^{*l}_n\}$ с последующей N-адической децимацией (выбором элементов только с номерами Nk). В результате разложения мы получаем два массива коэффициентов $cA_1 = \{a_{1,k}\}$ и $cD_1 = \{d^1_k, ..., d^{N-1}_k\}$. Процедуру разложения можно повторить, применив ее к набору коэффициентов аппроксимации cA_1.

$$cA_0 \to \{cA_1, cD_1\} \to \{cA_2, cD_2, cD_1\} \to ... \to \{cA_M, cD_M, cD_{M-1}, ..., cD_1\}.$$

Легко видеть (см. раздел 2.6.4), что *восстановление* массива $A = \{a_n\}$ по коэффициентам вейвлет-разложения $cA_1 = \{a_{1,k}\}$ и $cD_1 = \{d^1_k, ..., d^{N-1}_k\}$ производится следующим образом:

$$a_n = \sum_k h_{n-Nk} a_{1,k} + \sum_{l=1}^{N-1} \sum_{k \in Z} g^l_{n-Nk} d^l_{1,k}. \tag{25}$$

Последнюю формулу можно также записать в виде свертки, сделав обратную децимацию массивов $cA_1 = \{a_{1,k}\}$ и $cD_1 = \{d^1_k, ..., d^{N-1}_k\}$:

$$\tilde{a}_{1,m} = \begin{cases} a_{1,k}, & \text{если } m = Nk, \\ 0, & \text{если } m \text{ не кратно } N \end{cases}, \quad \tilde{d}^l_{1,m} = \begin{cases} d^l_{1,k}, & \text{если } m = Nk, \\ 0, & \text{если } m \text{ не кратно } N. \end{cases}$$

Тогда формула (25) принимает вид:

$$a_n = \sum_{m \in Z} h_{n-m} \tilde{a}_{1,m} + \sum_{l=1}^{N-1} \sum_{m \in Z} g^l_{n-m} \tilde{d}^l_{1,m}. \tag{26}$$

2.13.5. Разложение и восстановление в неортогональном случае

Формулы вейвлет-разложения и восстановления (23)–(26) установлены только для ортогонального случая. Они неприменимы, если разложение $V_j = V_{j-1} \oplus W_{j-1}^1 \oplus ... \oplus W_{j-1}^{N-1}$ не является ортогональным и функции

$$\varphi_{j,n}(x) = \sqrt{N^j}\,\varphi(N^j x - n),$$
$$\psi_{j,n}^l(x) = \sqrt{N^j}\,\psi^l(N^j x - n), \quad n \in \mathbf{Z},\ l = 1, 2, \ldots, N-1$$

не образуют ортонормированных систем. Предположим, что разложение сигнала $\{a_k\}$ производится некоторыми (неортогональными) фильтрами $\{h_k\}$ и $\{g_k^l\}$, $l = 1, 2, ..., N-1$ по формулам

$$a_{1k} = \sum_n h_n a_{Nk-n}, \quad d_{1,k}^l = \sum_n g_n^l a_{Nk-n},\ l = 1, 2, \ldots, N-1, \tag{27}$$

и найдем другие фильтры $\{\tilde{h}_n\}$, $\{\tilde{g}_n^l\}$, $l = 1, 2, ..., N-1$, которые обеспечивают точное восстановление сигнала по формулам типа (26). Обратите внимание, что формулы (27) записаны в виде свертки с последующей децимацией, они отличаются от формул (23). Отметим также, что $\{h_k\}$ и $\{g_n^l\}$ – это некоторые фильтры, пока не связанные с вейвлетами.

Задачу удобно решить на уровне формальных степенных рядов. Пусть $X(z) = \Sigma_n a_n z_n$ – ряд, соответствующий сигналу $\{a_k\}$. Определим передаточные функции заданных фильтров, $H_0(z) = \sum_n h_n z^n$, $H_l(z) = \sum_n g_n^l z^n$, $l = 1, 2, ..., N-1$. Тогда действие фильтров $\{h_k\}$ и $\{g_n^l\}$ на сигнал $\{a_k\}$ соответствует умножению соответствующих функций:

$$X_0(z) = H_0(z)X(z), \quad X_l(z) = H_l(z)X(z),\ l = 1, 2, ..., N-1. \tag{28}$$

Таким образом, сигнал $X(z)$ разбивается на следующие компоненты:

$$X(z) \to \{X_0(z), X_1(z), ..., X_{N-1}(z)\}. \tag{29}$$

При вейвлет-разложении (27) необходимо еще провести N-адическую *децимацию*, то есть выборку элементов с номерами Nk. На уровне формальных степенных рядов (29) для этого достаточно удалить все слагаемые со степенями, отличными от z^{kN}. Например, для ряда $X(z)$:

$$X(z) \to A(z^N) = A(w).$$

Коэффициенты полученного ряда $A(w)$ дают требуемую децимацию. Удаление степеней проведем следующим образом. Пусть $\rho = e^{-i2\pi/N}$, тогда легко видеть, что

$$\frac{1}{N}(X(z)+X(\rho z)+X(\rho^2 z)+\cdots+X(\rho^{N-1}z))=\sum_k a_{Nk}z^{Nk}=A(z^N) \qquad (30)$$

есть многочлен, содержащий только степени z^{kN}. Действительно, выпишем ряды слагаемых в левой части:

$$X(z)=\sum_k(a_{Nk}z^{Nk}+a_{Nk+1}z^{Nk+1}+\cdots+a_{Nk+N-1}z^{Nk+N-1}),$$

$$X(\rho z)=\sum_k\rho^{Nk}(a_{Nk}z^{Nk}+a_{Nk+1}\rho z^{Nk+1}+\cdots+a_{Nk+N-1}\rho^{N-1}z^{Nk+N-1}),$$

$$X(\rho^2 z)=\sum_k\rho^{Nk}(a_{Nk}z^{Nk}+a_{Nk+1}\rho^2 z^{Nk+1}+\cdots+a_{Nk+N-1}\rho^{2(N-1)}z^{Nk+N-1}),$$

…

$$X(\rho^{N-1}z)=\sum_k\rho^{Nk}(a_{Nk}z^{Nk}+a_{Nk+1}\rho^{N-1}z^{Nk+1}+\cdots+a_{Nk+N-1}\rho^{(N-1)(N-1)}z^{Nk+N-1}).$$

Просуммируем, учитывая, что $\rho^{Nk}=1$:

$$\begin{aligned}\frac{1}{N}\sum_{s=0}^{N-1}X(\rho^s z)&=\sum_k a_{Nk}z^{Nk}+\\&+\frac{1}{N}\sum_k a_{Nk+1}z^{Nk+1}(1+\rho+\rho^2+\cdots+\rho^{N-1})+\\&+\frac{1}{N}\sum_k a_{Nk+2}z^{Nk+2}(1+\rho^2+\rho^4+\cdots+\rho^{2(N-1)})+\\&\quad\cdots\\&+\frac{1}{N}\sum_k a_{Nk+N-1}z^{Nk+N-1}(1+\rho^{N-1}+\rho^{2(N-1)}+\cdots+\rho^{(N-1)(N-1)})=\\&=\sum_k a_{Nk}z^{Nk}=\sum_k a_{Nk}(z^N)^k=A(z^N)=A(w).\end{aligned}$$

Здесь мы использовали свойство $1 + \rho + \rho^2 + ... + \rho^{N-1} = 0$, которое верно для любого корня $\rho \neq 1$ из единицы степени N.

Обратимся снова к разложению (29) и проведем децимацию в каждой компоненте $X_l(z)$ разложения. Пусть $A_l(w)$ – многочлены, которые получаются из компоненты $X_l(z)$ после децимации и замены $w = z^N$.

Тогда на уровне степенных рядов вейвлет-разложение (27) производится следующим образом:

$$X(z) \to \{A_0(w), A_1(w), ..., A_{N-1}(w)\},$$

где компоненты разложения $A_l(w)$ определяются из формул:

$$A_0(z^N) = \frac{1}{N}\sum_{s=0}^{N-1} H_0(\rho^s z)X(\rho^s z),$$

$$A_l(z^N) = \frac{1}{N}\sum_{s=0}^{N-1} H_l(\rho^s z)X(\rho^s z), \quad l = 1, 2, \ldots, N-1.$$

Восстановление сигнала производим сверткой с другими фильтрами $G_l(z) = \sum_n \tilde{g}_n^l z^n$, $l = 0, 1, \ldots, N-1$, по формуле:

$$a_n = \sum_m \tilde{g}_{n-m}^0 \tilde{a}_{1,m} + \sum_{l=1}^{N-1}\sum_{m\in Z} \tilde{g}_{n-m}^l \tilde{d}_{1,m}^l,$$

где $\tilde{a}_{1,m}$ и $\tilde{d}_{1,m}^l$ – массивы коэффициентов, в которых проведена обратная децимация.

На уровне степенных рядов обратная децимация эквивалентна замене степенного ряда $A(w)$ по степеням переменной w на ряд $A(z^N)$ по степеням переменной z, $A(w) \to A(z^N)$. Поэтому восстановление на уровне степенных рядов делается по формуле:

$$\{A_0(w), A_1(w), \ldots, A_{N-1}(w)\} \to \{A_0(z^N), A_1(z^N), \ldots, A_{N-1}(z^N)\} \to$$

$$\to \sum_{l=0}^{N-1} G_l(z)A_l(z^N) = \frac{1}{N}\sum_{l=0}^{N-1} G_l(z)\sum_{s=0}^{N-1} H_l(\rho^s z)X(\rho^s z) =$$

$$= \frac{1}{N}\sum_{s=0}^{N-1}\left(\sum_{l=0}^{N-1} G_l(z)H_l(\rho^s z)\right)X(\rho^s z) = X(z).$$

Для точного восстановления достаточно, чтобы выполнялось равенство

$$\sum_{l=0}^{N-1} G_l(z)H_l(z) = N \quad (\text{при } s = 0),$$

$$\sum_{l=0}^{N-1} G_l(z)H_l(\rho^s z) = 0 \quad \text{для } s = 1, 2, \ldots, N-1.$$

Данные условия точного восстановления удобно выразить через матрицы *фильтров разложения* и *восстановления*. Произведение следующих матриц

$$\begin{pmatrix} G_0(z) & G_1(z) & \cdots & G_{N-1}(z) \\ G_0(\rho z) & G_1(\rho z) & \cdots & G_{N-1}(\rho z) \\ \cdots & \cdots & \cdots & \cdots \\ G_0(\rho^{N-1} z) & G_1(\rho^{N-1} z) & \cdots & G_{N-1}(\rho^{N-1} z) \end{pmatrix} \begin{pmatrix} H_0(z) & H_0(\rho z) & \cdots & H_0(\rho^{N-1} z) \\ H_1(z) & H_1(\rho z) & \cdots & H_1(\rho^{N-1} z) \\ \cdots & \cdots & \cdots & \cdots \\ H_{N-1}(z) & H_{N-1}(\rho z) & \cdots & H_{N-1}(\rho^{N-1} z) \end{pmatrix}$$

есть единичная матрица, умноженная на N. Мы получили следующий факт.

Теорема 4. *Если матрица фильтров разложения*

$$H(z)=\frac{1}{\sqrt{N}}\begin{pmatrix} H_0(z) & H_0(\rho z) & \cdots & H_0(\rho^{N-1}z) \\ H_1(z) & H_1(\rho z) & \cdots & H_1(\rho^{N-1}z) \\ \cdots & \cdots & \cdots & \cdots \\ H_{N-1}(z) & H_{N-1}(\rho z) & \cdots & H_{N-1}(\rho^{N-1}z) \end{pmatrix} \tag{31}$$

невырождена при $z = e^{-i\omega}$, то возможно точное восстановление сигнала фильтрами $G_l(z)$, $l = 0, 1, 2, ..., N - 1$, матрица которых

$$G(z)=\frac{1}{\sqrt{N}}\begin{pmatrix} G_0(z) & G_0(\rho z) & \cdots & G_0(\rho^{N-1}z) \\ G_1(z) & G_1(\rho z) & \cdots & G_1(\rho^{N-1}z) \\ \cdots & \cdots & \cdots & \cdots \\ G_{N-1}(z) & G_{N-1}(\rho z) & \cdots & G_{N-1}(\rho^{N-1}z) \end{pmatrix} \tag{32}$$

является транспонированной к обратной матрице $H(z)$ исходных фильтров.

Замечание 4. В ортогональном случае $H(z)$ – унитарная матрица (21), а $G(z)$ – комплексно сопряженная к $H(z)$, $G(z) = \overline{H(z)}$. В общем случае для нахождения фильтров восстановления необходимо найти матрицу, обратную к $H(z)$.

2.14. Примеры N-масштабирующих функций и вейвлетов

В данном параграфе мы построим примеры масштабирующих функций с масштабным коэффициентом $N > 2$ и приведем примеры соответствующих вейвлетов. А именно будет дано построение вейвлетов Хаара, Шеннона, Кантора, вейвлетов с коэффициентом масштабирования вида N^k и неортогональных вейвлетов на основе B-сплайнов.

2.14.1. Вейвлеты Хаара с параметром сжатия N

В этом разделе мы изложим построение вейвлетов *Хаара* с параметром масштабирования N и обсудим смысл коэффициентов вейвлет-разложения.

Рассмотрим функцию Хаара:

$$\varphi(x)=\begin{cases}1, & x\in[0,1)\\ 0, & \text{для других } x\end{cases}.$$

Для любого N легко видеть, что

$$\varphi(x) = \varphi(Nx) + \varphi(Nx-1) + \cdots + \varphi(Nx-N+1) = \\ = \sqrt{N}\sum_{n=0}^{N-1} h_n \varphi(Nx-n), \tag{1}$$

где ненулевые коэффициенты h_n только такие: $h_0 = h_1 = ... = h_{N-1} = 1/\sqrt{N}$. Поэтому функция Хаара является N-масштабирующей для любого целого $N > 1$. Частотная функция легко вычисляется:

$$H_0(\omega) = \frac{1}{N}(1 + e^{-i\omega} + e^{-i2\omega} + \cdots + e^{-i(N-1)\omega}) = \frac{1}{N} e^{-i(N-1)\omega/2} \frac{\sin(N\omega/2)}{\sin(\omega/2)}.$$

Рассмотрим ортогональное N-кратномасштабное разложение пространства $L^2(\mathbf{R})$, порожденное функцией $\varphi(x)$:

$$... \subset V_{-1} \subset V_0 \subset V_1 \subset V_2 \subset ...$$

Построим $N-1$ вейвлетов $\psi^1(x)$, ..., ψ^{N-1}, которые порождают $N-1$ пространств W_0^l, $l = 1, 2, ..., N-1$, в следующем ортогональном разложении:

$$V_1 = V_0 \oplus W_0^1 \oplus ... \oplus W_0^{N-1}.$$

Найдем вейвлет-функции $\psi^1(x)$, ..., $\psi^{N-1}(x)$ из свойств ортогональности подпространств. Пусть $\psi^l(x) \in W_0^l$. Поскольку $W_0^l \subset V_1$, тогда $\psi^l(x)$ раскладывается по базису $\{\varphi_{1,n}(x)\}$ пространства V_1: $\psi^l = \sum_n g_n^l \varphi_{1,n}$. Поскольку $\psi^l \perp V_0$, то для любого n имеем: $(\psi^l, \varphi_{0,n}) = 0$. Функции $\varphi_{0,n}(x)$ также раскладываются по базису $\{\varphi_{1,n}(x)\}$ пространства V_1: $\varphi_{0,n} = \sum_k h_k \varphi_{1,Nn+k}$. Коэффициенты этого разложения для функции $\varphi(x)$ ранее были найдены, $h_0 = h_1 = ... = h_{N-1} = 1/\sqrt{N}$. Тогда условие ортогональности ψ^l к V_0 принимает вид

$$(\psi^l, \varphi_{0,n}) = \left(\sum_k g_k^l \varphi_{1,k}, \frac{1}{\sqrt{N}}\varphi_{1,Nn} + \frac{1}{\sqrt{N}}\varphi_{1,Nn+1} + \cdots + \frac{1}{\sqrt{N}}\varphi_{1,Nn+N-1} \right) = 0.$$

Так как $\{\varphi_{1,n}(x)\}$ – ортонормированный базис, то из последнего равенства получаем:

$$g_{Nn}^l + g_{Nn+1}^l + \cdots + g_{Nn+N-1}^l = 0, \quad \forall n, \quad \forall l = 1, 2, \ldots, N-1.$$

Из условия ортонормированности вейвлетов, $(\psi^l, \psi^s) = \delta_{ls}$, имеем: $\sum_n g_n^l g_n^s = \delta_{ls}$. Итак, мы получили следующую систему уравнений:

$$\begin{cases} g_{Nn}^{l} + g_{Nn+1}^{l} + \cdots + g_{Nn+N-1}^{l} = 0, \quad \forall n, \quad \forall l = 1, 2, \ldots, N-1, \\ \sum\limits_{n} g_{n}^{l} g_{n}^{s} = \delta_{ls}, \quad l, s = 1, 2, \ldots, N-1. \end{cases} \tag{2}$$

Система имеет множество решений. Простейшее решение, соответствующее минимальному набору ненулевых коэффициентов, находится неоднозначно. Решением будет любой набор из $N - 1$ ортонормированных векторов-строк

$$G^{l} = (g_{0}^{l}, g_{1}^{l}, \ldots, g_{N-1}^{l}), \quad l = 1, 2, \ldots, N-1$$

в плоскости в $\mathbf{R}^{N}$, заданной уравнением $x_0 + x_1 + ... + x_{N-1} = 0$. Нахождение решения сводится к построению матрицы порядка N с ортогональными строками, первая строка которой есть вектор

$$h = (h_0,\ h_1,\ \ldots,\ h_{N-1}) = \frac{1}{\sqrt{N}}(1,\ 1,\ \ldots,\ 1).$$

Можно взять, например, следующую матрицу порядка N:

$$A_0 = \begin{pmatrix} 1 & 1 & 1 & 1 & \cdots & 1 & 1 \\ 1 & -1 & 0 & 0 & \cdots & 0 & 0 \\ 1 & 1 & -2 & 0 & \cdots & 0 & 0 \\ 1 & 1 & 1 & -3 & \cdots & 0 & 0 \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ 1 & 1 & 1 & 1 & \cdots & 1 & -N+1 \end{pmatrix}, \tag{3}$$

а затем провести нормировку строк.

Пример 1. Пусть $N = 3$, выберем простейшее нормированное решение системы (2):

$$g_0^1 = \frac{1}{\sqrt{2}}, \quad g_1^1 = -\frac{1}{\sqrt{2}}, \quad g_2^1 = 0 \quad \text{и} \quad g_0^2 = \frac{1}{\sqrt{6}}, \quad g_1^2 = \frac{1}{\sqrt{6}}, \quad g_2^2 = -\frac{\sqrt{2}}{\sqrt{3}},$$

остальные коэффициенты считаем равными нулю. Первому набору коэффициентов соответствует функция

$$\psi^1(x) = \frac{1}{\sqrt{2}}\varphi_{1,0}(x) - \frac{1}{\sqrt{2}}\varphi_{1,1}(x) = \frac{\sqrt{3}}{\sqrt{2}}\varphi(3x) - \frac{\sqrt{3}}{\sqrt{2}}\varphi(3x-1).$$

Второму набору коэффициентов соответствует функция

$$\psi^2(x) = \frac{1}{\sqrt{2}}\varphi(3x) + \frac{1}{\sqrt{2}}\varphi(3x-1) - \sqrt{2}\varphi(3x-2).$$

Графики вейвлетов приведены на рис. 2.14.1.

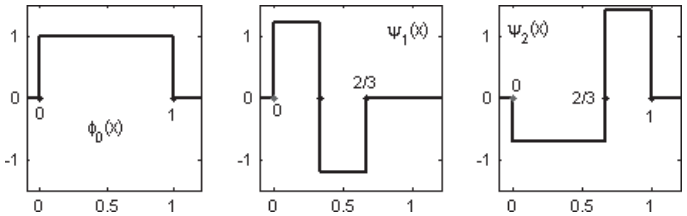


Рис. 2.14.1. Графики функции $\varphi(x)$ и вейвлетов $\psi^1(x)$, $\psi^2(x)$

Разложения сигнала при помощи вейвлетов Хаара. Обсудим смысл вейвлет-разложения сигнала при помощи вейвлетов Хаара. Для простоты пусть $N = 3$. Вейвлет-разложение сигнала $A = \{a_n\}$ производится по формулам:

$$a_{1,k} = \sum_n \overline{h_n}\, a_{n+Nk}, \quad d^l_{1,k} = \sum_n \overline{g^l_n}\, a_{n+Nk}, \; l = 1, 2, \ldots, N-1,$$

где $\{h_n\}_{n\in\mathbf{Z}}$ и $\{g^l_n\}_{n\in\mathbf{Z}}$ – фильтры вейвлетов $\varphi(x)$ и $\psi^l(x)$. В нашем случае фильтры имеют вещественные элементы. Тогда для случая $N = 3$ имеем: $a_{1,k} = (a_{3k} + a_{1+3k} + a_{2+3k})/\sqrt{3}$. Весь сигнал разбивается на тройки подряд идущих элементов. Коэффициенты аппроксимации $cA_1 = \{a_{1,k}\}$ получаются усреднением последовательных троек первоначальных элементов массива $\{a_n\}$.

Теперь рассмотрим детализирующие коэффициенты $cD^1_1 = \{d^1_k\}$, соответствующие вейвлету $\psi^1(x)$, $d^1_{1,k} = (a_{3k} - a_{1+3k})/\sqrt{2}$. Коэффициенты $cD^1_1 = \{d^1_k\}$ выражаются через разности двух первых элементов в последовательных тройках первоначальных элементов массива $\{a_n\}$. Детализирующие коэффициенты $cD^2_1 = \{d^2_k\}$, соответствующие вейвлету $\psi^2(x)$, выражаются через разности среднего значения двух первых элементов и третьего элемента в последовательных тройках первоначальных элементов массива $\{a_n\}$: $d^1_{1,k} = (a_{3k} + a_{1+3k} - 2a_{2+3k})/\sqrt{6}$.

Неоднозначность выбора вейвлетов. Как уже отмечалось, решение системы (2) находится неоднозначно. Если найдено некоторое решение, например в виде нормированных строк матрицы (3), то любое другое может быть получено умножением матрицы (3) слева на ортогональную матрицу вида

$$S = \begin{pmatrix} 1 & 0 & 0 & \cdots & 0 \\ 0 & S_1^1 & S_2^1 & \cdots & S_{N-1}^1 \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ 0 & S_1^{N-1} & S_2^{N-1} & \cdots & S_{N-1}^{N-1} \end{pmatrix}. \qquad (4)$$

Это соответствует ортогональному вращению векторов решений $G^l = (g_0^l, g_1^l, ..., g_{N-1}^l)$, $l = 1, 2, ... , N-1$ в плоскости $x_0 + x_1 + ... + x_{N-1} = 0$.

Например, для случая $N = 3$ можно выбрать матрицу вида

$$S = \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos t & \sin t \\ 0 & -\sin t & \cos t \end{pmatrix}.$$

Тогда получаем:

$$SA_0 = \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos t & \sin t \\ 0 & -\sin t & \cos t \end{pmatrix} \begin{pmatrix} 1/\sqrt{3} & 1/\sqrt{3} & 1/\sqrt{3} \\ 1/\sqrt{2} & -1/\sqrt{2} & 0 \\ 1/\sqrt{6} & 1/\sqrt{6} & -2/\sqrt{6} \end{pmatrix} =$$

$$= \begin{pmatrix} 1/\sqrt{3} & 1/\sqrt{3} & 1/\sqrt{3} \\ \cos t/\sqrt{2} + \sin t/\sqrt{6} & -\cos t/\sqrt{2} + \sin t/\sqrt{6} & -2\sin t/\sqrt{6} \\ -\sin t/\sqrt{2} + \cos t/\sqrt{6} & \sin t/\sqrt{2} + \cos t/\sqrt{6} & -2\cos t/\sqrt{6} \end{pmatrix}.$$

Соответствующие вейвлеты имеют выражения:

$$\psi_t^1(x) = \frac{1}{\sqrt{2}}((\sqrt{3}\cos(t) + \sin(t))\varphi(3x) +$$

$$+(-\sqrt{3}\cos(t) + \sin(t))\varphi(3x-1) - 2\sin(t)\varphi(3x-2)),$$

$$\psi_t^2(x) = \frac{1}{\sqrt{2}}((-\sqrt{3}\sin(t) + \cos(t))\varphi(3x) +$$

$$+(\sqrt{3}\sin(t) + \cos(t))\varphi(3x-1) - 2\cos(t)\varphi(3x-2)).$$

2.14.2. Вейвлеты Шеннона с параметром сжатия *N*

В этом параграфе построим *N*-кратные аналоги хорошо известных для $N = 2$ масштабирующих функций и вейвлетов *Шеннона*.

Известно, что функция $\varphi(x) = \sin\pi x/\pi x$ является масштабирующей с коэффициентом сжатия 2. Покажем ее *N*-масштабируемость для любого целого $N > 1$.

Теорема 1. *Функция* $\varphi(x) = \sin\pi x/\pi x$ *является N-масштабирующей функцией для любого целого* $N > 1$. *Масштабирующее соотношение для данной функции имеет вид:*

$$\varphi(x) = \sqrt{N}\sum_{n\in Z} h_n \frac{\sin\pi(Nx-n)}{\pi(Nx-n)}, \tag{5}$$

где

$$h_n = \frac{\sqrt{N}}{\pi n}\sin\left(\frac{\pi n}{N}\right). \tag{6}$$

Доказательство. Напомним, что функция $f(x) \in L^2(\mathbf{R})$ называется функцией с ограниченной шириной полосы, если ее преобразование Фурье $\hat{f}(\omega)$ равно нулю вне некоторой полосы частот $[-\Omega, \Omega]$. По теореме Шеннона–Котельникова функция с ограниченной шириной полосы $[-\Omega, \Omega]$ может быть представлена в виде

$$f(x) = \sum_{n\in Z} f\left(\frac{\pi}{\Omega}n\right)\frac{\sin(\Omega x - \pi n)}{\Omega x - \pi n}.$$

Функция $\varphi(x) = \sin\pi x/\pi x$ имеет преобразование Фурье $\hat{\varphi}(\omega)$ с носителем на промежутке $[-\pi, \pi]$, $\hat{\varphi}(\omega) = \chi_{[-\pi,\pi]}(\omega)$. Следовательно, носитель лежит и на большем промежутке $[-\pi N, \pi N]$. Поэтому для нашей функции $\varphi(x) = \sin\pi x/\pi x$ можно положить $\Omega = \pi N$, и тогда

$$\varphi(x) = \sum_{n\in Z}\varphi\left(\frac{\pi}{\pi N}n\right)\frac{\sin(\pi Nx - \pi n)}{\pi Nx - \pi n} = \sum_{n\in Z}\varphi\left(\frac{n}{N}\right)\frac{\sin\pi(Nx-n)}{\pi(Nx-n)} =$$

$$= \sqrt{N}\sum_{n\in Z} h_n\varphi(Nx-n).$$

Рассмотрим масштабирующее соотношение в частотной области:

$$\hat{\varphi}(\omega) = H_0\left(\frac{\omega}{N}\right)\hat{\varphi}\left(\frac{\omega}{N}\right).$$

Поскольку $\hat{\varphi}(\omega/N)=1$ на промежутке $[-\pi N, \pi N]$ и нуль в противном случае, то $H_0(\omega/N)=\hat{\varphi}(\omega)$ на $[-\pi N, \pi N]$, или $H_0(\omega)=\hat{\varphi}(N\omega)$ на промежутке $[-\pi, \pi]$. Поэтому

$$H_0(\omega)=\begin{cases}1 & \omega\in[-\pi/N, \pi/N],\\ 0 & \omega\in[-\pi, -\pi/N)\cup(\pi/N, \pi].\end{cases} \tag{7}$$

Вне промежутка $[-\pi, \pi]$ функция $H_0(\omega)$ продолжается периодически.

Построим последовательность уточняющих пространств V_j кратномасштабного разложения пространства $L^2(\mathbf{R})$. Пространство V_0, порожденное сдвигами $\varphi(x-n)$, $n\in\mathbf{Z}$, состоит из всех интегрируемых с квадратом функций $f(x)$, для которых преобразование Фурье имеет носитель на промежутке $[-\pi, \pi]$. Образуем последовательность пространств V_j, каждое из которых является масштабированной версией V_0. Например, пространство V_1 состоит из всех интегрируемых с квадратом функций вида $f(Nx)$, где $f(x)\in V_0$. Поскольку $F[f(Nx)]=\hat{f}(\omega/N)/N$, то преобразования Фурье функций из V_1 имеют носитель на промежутке $[-N\pi, N\pi]$. Поэтому $V_0\subset V_1$. Пространство V_j состоит из функций, преобразования Фурье которых имеют носитель на промежутке $[N^j\pi, N^j\pi]$.

Найдем вейвлеты $\psi^1(x)$, $\psi^2(x)$, ..., $\psi^{N-1}(x)$. Для этого найдем их частотные функции $H_k(\omega)$ из унитарности матрицы (20) из раздела 2.13.3. Для $k=1, 2, ..., N-1$ положим (рис. 2.14.2):

$$H_k(\omega)=\begin{cases}1 & |\omega|\in[k\pi/N, (k+1)\pi/N],\\ 0 & \text{для остальных } \omega \text{ промежутка } [-\pi, \pi].\end{cases} \tag{8}$$

Вейвлет $\psi^k(x)\in W_0^k$ определяется своим преобразованием Фурье из формулы

$$\hat{\psi}^k(\omega)=H_k\left(\frac{\omega}{N}\right)\hat{\varphi}\left(\frac{\omega}{N}\right)=\begin{cases}1 & k\pi\le|\omega|\le(k+1)\pi,\\ 0 & \text{при остальных } \omega.\end{cases} \tag{9}$$

Тогда для $k=1, 2, ..., N-1$ мы имеем:

$$\psi^k(x)=\frac{1}{2\pi}\int_{-\infty}^{\infty}\hat{\psi}^k(\omega)e^{i\omega x}d\omega=\frac{1}{2\pi}\int_{-(k+1)\pi}^{-k\pi}e^{i\omega x}d\omega+\frac{1}{2\pi}\int_{k\pi}^{(k+1)\pi}e^{i\omega x}d\omega=$$

$$=\frac{\sin(k+1)\pi x-\sin k\pi x}{\pi x}.$$

Данные функции назовем вейвлетами Шеннона с параметром масштабирования N.

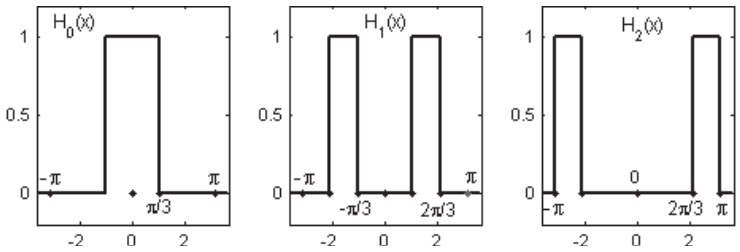


Рис. 2.14.2. Графики частотных функций
$H_0(\omega)$ и $H_1(\omega)$ и $H_2(\omega)$ для $N = 3$

Напомним, что коэффициенты $\{h_n\}$ масштабирующего фильтра вейвлетов Шеннона находятся по формуле (6). Найдем коэффициенты фильтров вейвлетов. Поскольку $\sqrt{N}H_k(\omega) = \sum_{n\in\mathbf{Z}} g_n^k e^{-in\omega}$ – разложение в ряд Фурье, то для $k = 1, 2, ..., N-1$:

$$g_n^k = \sqrt{N}\frac{1}{2\pi}\int_{-\pi}^{\pi} H_k(\omega)e^{i\omega n}\,d\omega = \sqrt{N}\frac{1}{2\pi}\left(\int_{-(k+1)\pi/N}^{-k\pi/N} e^{i\omega n}\,d\omega + \int_{k\pi/N}^{(k+1)\pi/N} e^{i\omega n}\,d\omega\right) =$$

$$= \frac{\sqrt{N}}{2\pi}2\mathrm{Re}\left(\int_{k\pi/N}^{(k+1)\pi/N} e^{i\omega n}\,d\omega\right) = \frac{\sqrt{N}}{\pi}\mathrm{Re}\frac{e^{i(k+1)\pi/N} - e^{ik\pi/N}}{in} =$$

$$= \sqrt{N}\frac{\sin((k+1)\pi/N) - \sin(k\pi/N)}{n\pi}.$$

Мы видим, что коэффициенты фильтров вейвлетов Шеннона убывают как n^{-1}. Это медленное убывание коэффициентов фильтра делает вейвлет Шеннона менее удобным в практических применениях.

Замечание 1. По построению вейвлеты Шеннона соответствуют идеальным фильтрам $H_0(\omega)$, $H_1(\omega)$, ..., $H_{N-1}(\omega)$. Вследствие разрывности этих функций масштабирующая функция $\varphi(x)$ и вейвлеты $\psi^k(x)$ убывают очень медленно. Поэтому вейвлеты Шеннона слабо локализованы в пространстве. Вейвлет Хаара есть противоположный крайний случай, когда $\varphi(x)$ является разрывной по пространственной переменной x.

Замечание 2. Сглаживая разрывные функции $H_0(\omega)$, $H_1(\omega)$, ..., $H_{N-1}(\omega)$ вейвлетов Шеннона, в работе [Po2] получены N-аналоги вейвлетов Мейера.

2.14.3. Вырожденные масштабирующие функции и вейвлеты Кантора

Масштабирующее соотношение

$$\varphi(x) = \frac{3}{2}(\varphi(3x) + \varphi(3x-2))$$

определяет [J03], [DJ] в качестве масштабирующей функции сингулярную меру, сосредоточенную в точках канторова множества на отрезке [0, 1]. Будем называть ее функцией Кантора. На следующем рисунке изображены несколько последовательных приближений к $\varphi(x)$ итерационного метода, изложенного в разделе 2.2.2.

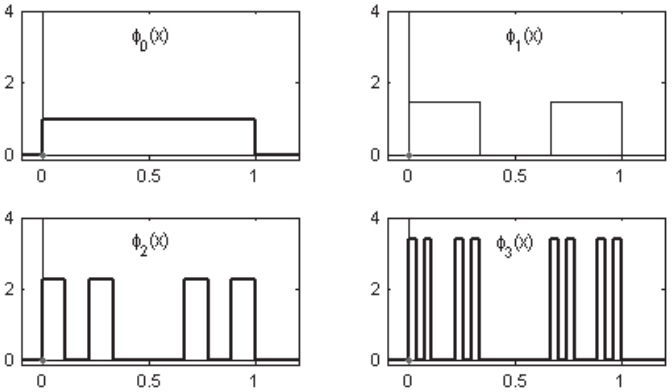


Рис. 2.14.3. Приближения масштабирующей функции Кантора

Интеграл по промежутку [0, 1] от каждого приближения $\varphi_N(x)$ равен единице. Поэтому в пределе получается сингулярная мера, сосредоточенная в точках канторова множества на отрезке [0, 1]. В пределе будет функция $\varphi(x)$, которая даже локально не интегрируема. Несмотря на сложную природу функции $\varphi(x)$, все остальные характеристики имеют простой смысл. Фильтр коэффициентов $\{h_n\}$ состоит из двух ненулевых элементов: $h_0 = \sqrt{3}/2$, $h_2 = \sqrt{3}/2$. Эти коэффициенты удовлетворяют свойству $h_0 + h_2 = \sqrt{3}$. Частотная функция имеет вид

$$H_0(\omega) = (1 + e^{-i2\omega})/2 = e^{-i\omega}(e^{i\omega} + e^{-i\omega})/2 = e^{-i\omega}\cos(\omega). \qquad (10)$$

Тогда в соответствии с методом построения $\hat{\varphi}(\omega)$ через бесконечное произведение получаем:

$$\hat{\varphi}(\omega)=\prod_{n=1}^{\infty}e^{-i\omega/3^n}\cos\left(\frac{\omega}{3^n}\right)=e^{-i\omega/2}\prod_{n=1}^{\infty}\cos\left(\frac{\omega}{3^n}\right). \qquad (11)$$

Как было показано в теореме 1 раздела 2.2.2, данное бесконечное произведение сходится поточечно на всей числовой прямой и равномерно на компактных подмножествах. Однако функция $\hat{\varphi}(\omega)$ не принадлежит $L^2(\mathbf{R})$, поскольку соответствующая ей функция $\varphi(x)$ является сингулярной мерой, сосредоточенной в точках канторова множества на отрезке [0, 1]. Такая функция $\varphi(x)$ не является даже локально интегрируемой. На рис. 2.14.4 изображено приближение $p_8(\omega)$ функции $\hat{\varphi}(\omega)$, состоящее из 8 сомножителей бесконечного произведения на промежутке $[0, 200\pi]$. Рисунок показывает слабое убывание функции $\hat{\varphi}(\omega)$.

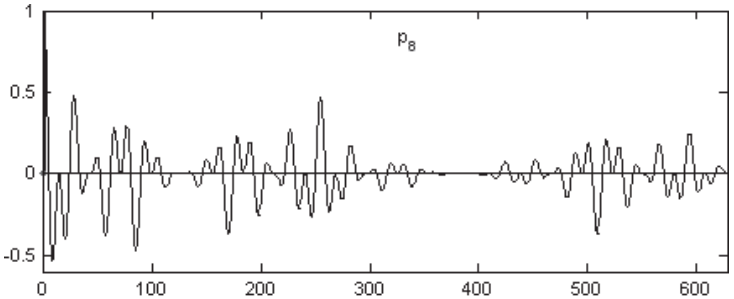


Рис. 2.14.4. График частичного произведения $p_8(\omega)$

Приведенные выше конструкции допускают обобщение на случай любого нечетного $N > 3$. Мы будем исходить из процедуры построения *множества Кантора*. Промежуток [0, 1] разделим на N равных частей и выбросим открытый интервал длины $1/N$, расположенный посередине [0, 1]. Получим $N - 1$ промежутков, каждый длины $1/N$. На втором этапе каждый из оставшихся промежутков разделим на N равных частей и выбросим открытый интервал длины $1/N^2$. Получим $(N-1)^2$ промежутков, каждый длины $1/N^2$. Продолжая эту процедуру до бесконечности, получаем канторово множество $K(N)$ на [0, 1].

Вычислим суммарную длину всех выкинутых интервалов:

$$\frac{1}{N}+(N-1)\frac{1}{N^2}+(N-1)^2\frac{1}{N^3}+\cdots=1.$$

Таким образом, мера канторова множества $K(N)$ на [0, 1] равна нулю.

Масштабирующее соотношение записываем аналогично случаю $N = 3$. Если $N = 2k + 1$, то в соответствии с процедурой построения множества $K(N)$, пропуская среднее слагаемое $\varphi(Nx - k)$, получаем масштабирующее соотношение в виде:

$$\varphi(x) = \frac{N}{N-1}(\varphi(Nx) + \varphi(Nx-1) + \cdots + \varphi(Nx-k+1) +$$

$$+ \varphi(Nx-k-1) + \cdots + \varphi(Nx-N+1)).$$

Фильтр коэффициентов $\{h_n\}$ состоит из $2k$ ненулевых элементов:

$$h_0 = h_1 = \ldots = h_{k-1} = h_{k+1} = \ldots = h_{N-1} = \frac{\sqrt{N}}{N-1}. \qquad (12)$$

Эти коэффициенты удовлетворяют свойству $\sum_n h_n = \sqrt{N}$. Частотная функция имеет вид:

$$H_0(\omega) = \frac{1}{N-1}(1 + e^{-i\omega} + \cdots + e^{-i(k-1)\omega} + e^{-i(k+1)\omega} + \cdots + e^{-i(N-1)\omega}) =$$

$$= \frac{2}{N-1} e^{-ik\omega}(\cos\omega + \cos 2\omega + \cdots + \cos k\omega).$$

Заметим, что сумма косинусов легко вычисляется [АСЧ], и мы получаем:

$$H_0(\omega) = \frac{2}{N-1} e^{ik\omega} \frac{\cos(k+1)\omega/2 \cdot \sin k\omega/2}{\sin\omega}. \qquad (13)$$

Тогда функция $\hat{\varphi}(\omega)$ находится через бесконечное произведение:

$$\hat{\varphi}(\omega) = \prod_{n=1}^{\infty} H_0\left(\frac{\omega}{N^n}\right).$$

Замечание 3. Хотя мы предполагали, что коэффициент масштабирования является нечетным, $N = 2k + 1$, легко видеть, что все конструкции имеют обобщения для любого N. Во-первых, можно опускать одно слагаемое на любом месте:

$$\varphi(x) = \frac{N}{N-1}(\varphi(Nx) + \varphi(Nx-1) + \cdots + \varphi(Nx-p+1) +$$

$$+ \varphi(Nx-p-1) + \cdots + \varphi(Nx-N+1)),$$

где $p = 1, 2, ..., N - 1$. Во-вторых, можно опускать несколько слагаемых на разных местах.

Вейвлеты Кантора. Построение вейвлетов *Кантора* $\psi^1(x)$, ..., $\psi^{N-1}(x)$ полностью повторяет построение вейвлетов Хаара. Коэффициенты фильтров $\{g_n^l\}$ находятся из системы уравнений

$$\begin{cases} g^l_{Nn} + g^l_{Nn+1} + \cdots + g^l_{Nn+k-1} + g^l_{Nn+k+1} + \cdots + g^l_{Nn+N-1} = 0, \ \forall l = 1, 2, \ldots, N-1, \forall n, \\ \sum\limits_n g^l_n g^s_n = \delta_{ls}, \quad \forall l, s = 1, 2, \ldots, N-1. \end{cases} \tag{15}$$

Система имеет множество решений. Простейшее решение, соответствующее минимальному набору ненулевых коэффициентов, находится неоднозначно. Решением будет любой набор из $N-1$ ортонормированных векторов-строк

$$G^l = (g^l_0, g^l_1, \ldots g^l_{N-1}), \ l = 1, 2, \ldots, N-1$$

в плоскости в $\mathbf{R}^N$, заданной уравнением $x_0 + \ldots + x_{k-1} + x_{k+1} + \ldots + x_{N-1} = 0$. Нахождение решения сводится к построению матрицы порядка N с ортонормированными строками, первая строка которой есть вектор

$$h = (h_0, h_1, \ldots, h_{N-1}) = \frac{\sqrt{N}}{N-1}(1, 1, \ldots, 1, 0, 1, \ldots, 1),$$

у которого на k-м месте стоит нуль.

Пример 2. Пусть $N = 3$. Тогда масштабирующее соотношение имеет вид

$$\varphi(x) = \frac{3}{2}(\varphi(3x) + \varphi(3x-2)).$$

Фильтр коэффициентов $\{h_n\}$ состоит из двух ненулевых элементов: $h_0 = \sqrt{3}/2$, $h_2 = \sqrt{3}/2$. Найдем вейвлеты $\psi^1(x)$ и $\psi^2(x)$. Выберем простейшее решение системы (15):

$$g^1_0 = 0, \quad g^1_1 = 1, \quad g^1_2 = 0 \quad \text{и} \quad g^2_0 = \frac{1}{\sqrt{2}}, \quad g^2_1 = 0, \quad g^2_2 = -\frac{1}{\sqrt{2}};$$

остальные коэффициенты считаем равными нулю. Первому набору коэффициентов соответствует функция

$$\psi^1(x) = \sqrt{3}\varphi(3x - 1).$$

Второму набору коэффициентов соответствует функция

$$\psi^2(x) = \frac{\sqrt{3}}{\sqrt{2}}\varphi(3x) - \frac{\sqrt{3}}{\sqrt{2}}\varphi(3x-2).$$

Частотные функции, соответствующие вейвлетам $\psi^1(x)$ и $\psi^2(x)$, имеют вид:

$$H_1(\omega) = \frac{1}{\sqrt{3}}e^{-i\omega}, \quad H_2(\omega) = \frac{1}{\sqrt{6}}(1 - e^{-i2\omega}) = \frac{e^{-i\omega}}{\sqrt{6}}2i\sin\omega.$$

Разложения сигнала при помощи вейвлетов Кантора. Несмотря на то что функция Кантора $\varphi(x)$ и соответствующие вейвлеты имеют непонятную функциональную трактовку, использование их для вейвлет-разложения сигнала не представляет трудностей. Для простоты продемонстрируем схему вейвлет-разложения при $N = 3$. Вейвлет-разложение сигнала $A = \{a_n\}$ производится по формулам:

$$a_{1,k} = \sum_n \overline{h}_n a_{n+Nk}, \quad d^l_{1,k} = \sum_n \overline{g^l_n}\, a_{n+Nk}, \; l = 1, 2, \ldots, N-1l,$$

где $\{h_n\}_{n\in\mathbf{Z}}$ и $\{g^l_n\}_{n\in\mathbf{Z}}$ – фильтры вейвлетов $\varphi(x)$ и $\psi^l(x)$. В нашем случае фильтры имеют вещественные элементы, и сумма состоит не более чем из трех слагаемых. Для коэффициентов аппроксимации $cA_1 = \{a_{1,k}\}$ имеем: $a_{1,k} = \sqrt{3}(a_{3k} + a_{2+3k})/2$. Поэтому весь сигнал разбивается на тройки подряд идущих элементов. Коэффициенты аппроксимации $cA_1 = \{a_{1,k}\}$ выражаются через средние двух крайних элементов в последовательных тройках первоначальных коэффициентов $\{a_n\}$.

Теперь рассмотрим детализирующие коэффициенты $cD^1_1 = \{d^1_k\}$, соответствующие вейвлету $\psi^1(x)$. Легко видеть, что $d^1_{1,k} = a_{1+3k}$ – это выборка среднего элемента в последовательных тройках первоначальных коэффициентов $\{a_n\}$. Детализирующие коэффициенты $cD^2_1 = \{d^2_k\}$, соответствующие вейвлету $\psi^2(x)$, находятся по формуле $d^1_{1,k} = (a_{3k} - a_{2+3k})/\sqrt{2}$ – это разности двух крайних элементов в последовательных тройках первоначальных коэффициентов $\{a_n\}$.

Таким образом, вейвлет-разложение сигнала при помощи вейвлетов Кантора производится по следующей схеме:

- весь сигнал разбивается на группы из N последовательных элементов;
- часть элементов каждой группы усредняется;
- образуются коэффициенты, которые отражают детали операции усреднения;
- производится выборка остальных элементов группы, которые не участвуют в усреднении.

Такой вейвлет-анализ естественно использовать для временных рядов, которые имеют периодичность по отсчетам времени. Например, если сигнал $A = \{a_n\}$ представляет собой дневные объемы продаж, то естественно учесть недельный цикл времени $N = 7$. Представляет интерес выборка объемов продаж в субботу и воскресенье. В рабочие дни объемы продаж можно усреднить, а в детализирующих коэффициентах отразить изменения объемов продаж в рабочие дни каждой недели.

Замечание 4. Как и в случае вейвлетов Хаара, решение системы (15) находится неоднозначно. Если найдено некоторое решение $G^l = (g^l_0, ..., g^l_{N-1})$, $l = 1, 2, ..., N-1$, то его можно подвергнуть ортогональному вращению в плоскости $x_0 + ... + x_{k-1} + x_{k+1} + ... + x_{N-1} = 0$.

2.14.4. Сплайновые масштабирующие функции

В этом разделе мы покажем, что B-сплайны являются масштабирующими функциями для любого N.

Рассмотрим сначала простой пример B-сплайна $\varphi(x)$ степени 1:

$$\varphi(x) = \begin{cases} 1-|x|, & x \in [-1,1], \\ 0 & \text{при остальных } x. \end{cases} \tag{16}$$

Преобразование Фурье данной функции:

$$\hat{\varphi}(\omega) = \left(\frac{\sin\omega/2}{\omega/2}\right)^2 = 2\frac{1-\cos\omega}{\omega^2}.$$

Хорошо известно, что функция $\varphi(x)$ удовлетворяет масштабирующему уравнению с коэффициентом 2. В то же время легко видеть, что для $N = 3$ также имеет место равенство

$$\varphi(x) = {}^1\!/_3\varphi(3x+2) + {}^2\!/_3\varphi(3x+1) + \varphi(2x) + \\ + {}^2\!/_3\varphi(3x-1) + {}^1\!/_3\varphi(3x-2). \tag{17}$$

Следующий рис. 2.14.5 иллюстрирует данное равенство.

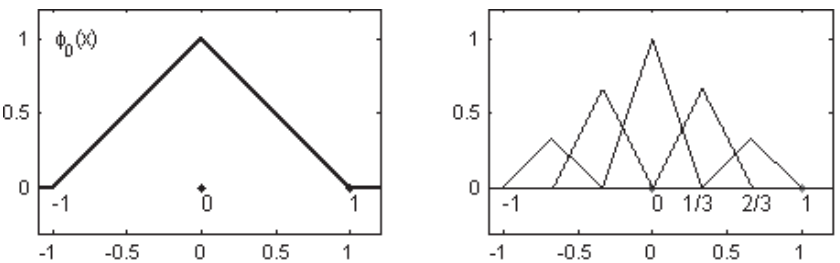


Рис. 2.14.5. Масштабирующее соотношение для B-сплайна степени 1

Также легко видеть, что при любом целом $N > 1$ имеет место следующее N-масштабирующее соотношение:

$$\varphi(x) = \sum_{n=-N+1}^{N-1} \frac{N-|n|}{N}\varphi(Nx-n). \tag{18}$$

Поэтому B-сплайн $\varphi(x)$ степени 1 является масштабирующей функцией для любого N.

Поскольку B-сплайны степени 0 и степени 1 являются N-масштабирующими, то естественно предположить, что это свойство имеет место для любых B-сплайнов. Оказывается, это предположение верно.

N-масштабируемость B-сплайнов. B-сплайн нулевого порядка есть функция Хаара $\varphi_0(x)$ (характеристическая функция промежутка [0, 1)). B-сплайны высших порядков определяются индуктивно через свертку по формуле $\varphi_{p+1} = \varphi_p * \varphi_0$. В дальнейшем удобно будет считать, что для B-сплайна $\varphi_p(x)$ носитель находится на промежутке $[-(p+1)/2, (p+1)/2]$ в случае нечетного p, а в случае четного p – на промежутке $[-p/2, p/2+1]$. Тогда преобразование Фурье B-сплайна $\varphi_p(x)$ имеет вид:

$$\hat{\varphi}_p(\omega) = e^{-iK\omega/2}\left(\frac{\sin(\omega/2)}{\omega/2}\right)^{p+1}, \tag{19}$$

где $K = 0$ – в случае нечетного p и $K = 1$ – в случае четного p.

Теорема 2. *B-сплайны $\varphi_p(x)$ порядка p являются N-масштабирующими функциями для любого целого $N > 1$. При этом правая часть масштабирующего соотношения $\varphi(x) = \sqrt{N}\sum_k h_k \varphi(Nx-k)$ является конечной суммой. Коэффициенты h_k находятся по формуле*

$$h_k = \frac{1}{\sqrt{N}\,N^p}\sum_{\alpha}\frac{(p+1)!}{\alpha_0!\alpha_1!\ldots\alpha_{N-1}!}, \tag{20}$$

где суммирование производится по всем мультииндексам $\alpha = (\alpha_0, \alpha_1, \ldots, \alpha_{N-1})$, удовлетворяющим в случае нечетного p двум условиям:

$$|\alpha| = \alpha_0 + \alpha_1 + \ldots + \alpha_{N-1} = p+1,$$
$$\alpha_1 + 2\alpha_2 + \ldots + (N-1)\alpha_{N-1} - (N-1)(p+1)/2 = k,$$

индекс k меняется от $k = -(N-1)(p+1)/2$ до $k = (N-1)(p+1)/2$.

В случае четного p суммирование в формуле (20) *производится по всем мультииндексам $\alpha = (\alpha_0, \alpha_1, \ldots, \alpha_{N-1})$, удовлетворяющим двум условиям:*

$$|\alpha| = \alpha_0 + \alpha_1 + \ldots + \alpha_{N-1} = p+1,$$
$$\alpha_1 + 2\alpha_2 + \ldots + (N-1)\alpha_{N-1} - (N-1)p/2 = k,$$

индекс k меняется от $k = -(N-1)p/2$ до $k = (N-1)(p/2+1)$.

Доказательство. Рассмотрим сначала случай нечетного p. Тогда носитель $\varphi_p(x)$ находится на промежутке $[-(p+1)/2, (p+1)/2]$ и пре-

образование Фурье имеет вид (19), где $K = 0$. Подставим (19) в масштабирующее соотношение и выразим частотную функцию $H_0(\omega)$:

$$H_0(\omega) = \frac{\hat{\varphi}_p\, N\omega}{\hat{\varphi}_p(\omega)} = \left(\frac{\sin(N\omega/2)}{N\omega/2}\right)^{p+1}\left(\frac{\omega/2}{\sin(\omega/2)}\right)^{p+1} = \left(\frac{1}{N}\frac{\sin(N\omega/2)}{\sin(\omega/2)}\right)^{p+1} =$$

$$= \left(\frac{1}{N}\frac{e^{iN\omega/2} - e^{-iN\omega/2}}{e^{i\omega/2} - e^{-i\omega/2}}\right)^{p+1} = \left(\frac{1}{N}e^{i(N-1)\omega/2}\right)^{p+1}\left(\frac{1-e^{-iN\omega}}{1-e^{-i\omega}}\right)^{p+1} =$$

$$= \frac{1}{N^{p+1}}e^{i\omega(N-1)(p+1)/2}(1 + e^{-i\omega} + \cdots + e^{-i(N-1)\omega})^{p+1}.$$

Поскольку p – нечетное, сомножитель $e^{i(N-1)\omega(p+1)/2}$ является целочисленной степенью числа $e^{i\omega}$. Раскрывая степень $(1 + e^{-i\omega} + \ldots + e^{-i(N-1)\omega})^{p+1}$, мы получаем представление частотной функции $H_0(\omega)$ в виде конечной суммы по степеням переменной $e^{-i\omega}$, то есть в виде $H_0(\omega) = \sqrt{N^{-1}}\sum_k h_k e^{-ik\omega}$. Коэффициенты разложения полученной функции $H_0(\omega)$, взятые с коэффициентом $\sqrt{N}$, дают нам требуемые коэффициенты h_k масштабирующего соотношения. При этом суммирование производится от $k = -(N-1)(p+1)/2$ до $k = (N-1)(p+1)/2$. Вычислим эти коэффициенты. Для этого используем известный аналог формулы бинома Ньютона

$$(z_0 + z_1 + \cdots + z_k)^m = \sum_{|\alpha|=m}\frac{m!}{\alpha_0!\ldots\alpha_k!}z_0^{\alpha_0}\cdots z_k^{\alpha_k}.$$

Тогда

$$(1 + e^{-i\omega} + \cdots + z^{-i(N-1)\omega})^{p+1} = \sum_{k=0}^{(N-1)(p+1)}\sum_{\substack{|\alpha|=p+1\\ \alpha_1+2\alpha_2+\cdots+(N-1)\alpha_{N-1}=k}}\frac{(p+1)!}{\alpha_0!\alpha_1!\ldots\alpha_{N-1}!}e^{-ik\omega},$$

$$H_0(\omega) = \frac{1}{N^{n+1}}e^{i(N-1)\omega(p+1)/2}(1 + e^{-i\omega} + \cdots + z^{-i(N-1)\omega})^{p+1} =$$

$$= \frac{1}{N^{p+1}}\sum_{k=-(N-1)(p+1)/2}^{(N-1)(p+1)/2}\sum_{\substack{|\alpha|=p+1\\ \alpha_1+2\alpha_2+\cdots+(N-1)\alpha_{N-1}=k+(N-1)(p+1)/2}}\frac{(p+1)!}{\alpha_0!\alpha_1!\ldots\alpha_{N-1}!}e^{-ik\omega}.$$

Из последнего выражения следует утверждение теоремы. Во втором случае, когда p – четное, доказательство проводится точно так же. Разница только в том, что носитель $\varphi_p(x)$ находится на промежутке $[-p/2, p/2 + 1]$ и преобразование Фурье функции $\varphi_p(x)$ имеет фазовый множитель $e^{-i\omega/2}$ в соответствии с формулой (19).

Замечание 5. Из доказательства теоремы 2 следует, что *частотная функция B*-сплайна как *N*-масштабирующей функции имеет вид:

- для нечетного p:

$$H_0(\omega) = \frac{1}{N^{p+1}} e^{i\omega(N-1)(p+1)/2} (1 + e^{-i\omega} + \cdots + e^{-i(N-1)\omega})^{p+1}; \quad (21)$$

- для четного p:

$$H_0(\omega) = \frac{1}{N^{p+1}} e^{i\omega(N-1)p/2} (1 + e^{-i\omega} + \cdots + e^{-i(N-1)\omega})^{p+1}. \quad (22)$$

Отметим также, что длина фильтра ненулевых коэффициентов B-сплайна как N-масштабирующей функции равна $(N - 1)(p + 1) + 1$.

2.14.5. Вейвлеты на основе B-сплайнов

Предположим, что $\varphi(x)$ – B-сплайновая масштабирующая функция. Она является N-масштабирующей, и ее частотная функция $H_0(\omega)$ является полиномиальной (относительно $z = e^{-i\omega}$). Однако $\varphi(x)$ не определяет ортогонального кратномасштабного анализа. Поэтому вопрос о построении соответствующих вейвлетов для масштабирующей функции $\varphi(x)$ является достаточно сложным. В этом разделе покажем, как можно построить *сплайновые вейвлеты* $\psi^1(x)$, ..., $\psi^{N-1}(x)$ с полиномиальными частотными функциями, которые обеспечивают N-канальное разложение сигнала с возможностью его точного восстановления дуальными вейвлетами $\tilde{\varphi}(x)$, $\tilde{\psi}^1(x)$, ..., $\tilde{\psi}^{N-1}(x)$. Сначала проведем несколько общих построений.

Пусть $\varphi(x)$ – B-сплайновая масштабирующая функция степени p и $H_0(\omega)$ – ее частотная функция. Пусть $H_1(\omega)$, ..., $H_{N-1}(\omega)$ – искомые частотные (полиномиальные) функции фильтров N-канального разложения. Согласно теореме 4 раздела 2.13.5, для точного восстановления необходимо, чтобы была невырожденной матрица частотных функций вида

$$H(z) = \begin{pmatrix} H_0(z) & H_0(\rho z) & \cdots & H_0(\rho^{N-1} z) \\ H_1(z) & H_1(\rho z) & \cdots & H_1(\rho^{N-1} z) \\ \cdots & \cdots & \cdots & \cdots \\ H_{N-1}(z) & H_{N-1}(\rho z) & \cdots & H_{N-1}(\rho^{N-1} z) \end{pmatrix}, \quad (23)$$

где $z = e^{-i\omega}$ и $\rho = e^{-i2\pi/N}$ (мы предполагаем, что сомножитель $1/\sqrt{N}$ включен в частотные функции, как в формуле 5 раздела 2.13.1). В этой матрице можно считать, что первая строка известна. Остальные строки должны быть найдены, причем так, чтобы матрица $H(z)$ была бы невырожденной.

Матрица (23) имеет очень специальный вид. Поэтому трудно получить простые условия на многочлены $H_1(z)$, ..., $H_{N-1}(z)$ для обеспечения невырожденности $H(z)$.

Легко видеть, что от специального вида матрицы $H(z)$ можно избавиться следующим образом. Каждую функцию $H_l(z)$ можно представить в виде суммы

$$H_l(z) = A_{l,0}(z^N) + zA_{l,1}(z^N) + ... + z^{N-1}A_{l,N-1}(z^N). \quad (24)$$

Тогда

$$H_l(\rho^s z) = A_{l,0}(z^N) + \rho^s zA_{l,1}(z^N) + ... + (\rho^s z)^{N-1}A_{l,N-1}(z^N).$$

Мы получили матрицу $A(z^N) = (A_{l,k}(z^N))$, которая называется *полифазной*. Ее элементы не зависят от ρ, и матрица частотных функций (23) может быть представлена в виде произведения следующих матриц:

$$H(z) = A(z^N)R(z). \quad (25)$$

Специфика матрицы $H(z)$ теперь заключена в матрице $R(z)$ вида:

$$R(z) = \begin{pmatrix} 1 & 1 & \cdots & 1 \\ z & \rho z & \cdots & \rho^{N-1} z \\ \cdots & \cdots & \cdots & \cdots \\ z^{N-1} & \rho^{N-1} z^{N-1} & \cdots & \rho^{(N-1)^2} z^{N-1} \end{pmatrix}, \quad (26)$$

а матрица $A(z^N)$ является уже обычной невырожденной матрицей, элементы которой являются многочленами по переменной $w = z^N$. Причем первая строка матрицы $A(w)$ известна, она выражается через заданную частотную функцию $H_0(z)$. Выбирая остальные строки так, чтобы матрица $A(w)$ была невырожденной, мы можем построить по формуле (25) матрицу частотных функций $H(z)$ и вместе с ней частотные функции $H_1(z)$,... , $H_{N-1}(z)$ вейвлетов, следовательно, и сами вейвлеты $\psi^1(x)$, ..., $\psi^{N-1}(x)$.

Замечание 6. Многочлены $A_{l,k}(z^N)$ в формуле (24) называются *полифазными компонентами* функции $H_l(z)$, а сама формула (24) называется [Va] *полифазным разложением* функции $H_l(z)$. Каждая строка полифазной матрицы $A(z^N)$ – это просто выборка элементов со степенями z^{kN} (то есть кратными N) в многочленах

$$H_l(z), z^{-1}H_l(z), ..., z^{-N+1}H_l(z).$$

Формула для нахождения полифазных компонент может быть представлена в виде преобразования *Фурье* по циклической группе $\mathbf{Z}/N\mathbf{Z} = \{1, \rho, \rho^2, ..., \rho^{N-1}\}$ [BJ]:

$$A_{l,k}(w) = \frac{1}{N}\sum_{z^N = w} z^{-k} H_l(z), \quad l,k = 0,1,\ldots,N-1. \tag{27}$$

Обратное преобразование определяется формулой (24).

Отметим, что матрица $R(z)$ является почти унитарной на единичной окружности $z = e^{-i\omega}$, а именно имеет место равенство $R(z)R^*(z) = N$. Последнее равенство легко следует из того, что $\bar{z} = z^{-1}$, $\bar{\rho} = \rho^{-1}$, и из свойства $1 + \rho + \rho^2 + \ldots + \rho^{N-1} = 0$, которое верно для любого корня $\rho \neq 1$ из единицы степени N. Таким образом,

$$R(z)^{-1} = \frac{1}{N}\begin{pmatrix} 1 & \bar{z} & \cdots & \bar{z}^{N-1} \\ 1 & \bar{\rho}\bar{z} & \cdots & \bar{\rho}^{N-1}\bar{z}^{N-1} \\ \cdots & \cdots & \cdots & \cdots \\ 1 & \bar{\rho}^{N-1}\bar{z} & \cdots & \bar{\rho}^{(N-1)^2}\bar{z}^{N-1} \end{pmatrix}. \tag{28}$$

Построение полифазной матрицы

В предположении, что полиномиальная частотная функция $H_0(z)$ является заданной, можно считать, что первая строка матрицы $A(z^N)$ известна:

$$A_{0,j}(w) = \frac{1}{N}\sum_{z^N = w} z^{-j} H_0(z). \tag{29}$$

Задача заключается в том, чтобы образовать остальные строки. Это мы сделаем позднее, а теперь проанализируем имеющуюся первую строку

$$\alpha(w) = (A_{0,0}(w), A_{0,1}(w), \ldots, A_{0,N-1}(w)). \tag{30}$$

Поскольку каждый элемент этой первой строки является многочленом, то она может быть представлена в виде многочлена некоторой степени $g - 1$ с векторными коэффициентами

$$\alpha(w) = \alpha_0 + \alpha_1 w + \ldots + \alpha_{g-1} w^{g-1}, \tag{31}$$

где $\alpha_0, \alpha_1, \ldots, \alpha_{g-1}$ – g векторов-строк из $\mathbf{C}^N$.

В случае, когда частотная функция $H_0(z)$ соответствует B-сплайну степени p, количество векторных коэффициентов в выражении (31) можно оценить.

Лемма 3. *Если степень B-сплайна удовлетворяет неравенству* $p + 1 \le N$, *то число g векторных коэффициентов в разложении* (31) *первой строки* $\alpha(w) = (A_{0,0}(w), A_{0,1}(w), \ldots, A_{0,N-1}(w))$ *равно* $p + 1$.

Доказательство. Если не учитывать сдвиг носителя B-сплайна, то

$$H_0(z) = \frac{1}{N^{p+1}}(1 + z + \cdots + z^{(N-1)})^{p+1}$$

– это многочлен степени $Np + N - (p + 1)$. Преобразование (29) есть выборка элементов со степенями z^{kN} (то есть кратными N) в полиномах

$$H_0(z),\ z^{-1}H_0(z),\ \ldots,\ z^{-N+1}H_0(z).$$

Если $p + 1 \le N$, то $0 \le N - (p + 1) < N$. Поэтому $Np \le (N - 1)\times$ $\times(p + 1) < N(p + 1)$. Следовательно, многочлен $H_0(z)$ имеет ровно $p + 1$ элементов со степенями, кратными z^N, включая z^0, это элементы $z^0, z^N, z^{2N}, \ldots, z^{pN}$. Легко видеть, что в остальных полиномах $z^{-1}H_0(z), \ldots, z^{-N+1}H_0(z)$ таких степеней не может быть больше.

В случае, когда частотная функция $H_0(z)$ соответствует B-сплайну степени p, векторные коэффициенты в выражении (31) обладают дополнительным интересным свойством.

Теорема 4. ([PS1]) *Пусть* $\varphi(x)$ *– B-сплайн порядка* $p < N$ *и* $H_0(z)$ *– его частотная функция. Если*

$$\alpha(w) = \alpha_0 + \alpha_1 w + \ldots + \alpha_p w^p$$

является разложением первой строки $\alpha(w) = (A_{0,0}(w),\ A_{0,1}(w),\ \ldots,\ A_{0,N-1}(w))$, *состоящей из полифазных компонент* (29) *частотной функции* $H_0(z)$, *тогда сумма векторов-коэффициентов есть вектор с одинаковыми координатами*:

$$\alpha_0 + \alpha_1 + \cdots + \alpha_p = \frac{1}{N}(1, 1, \ldots, 1). \tag{32}$$

Вернемся к разложению (31) первой строки (30), полученной преобразованием Фурье (29) частотной функции $H_0(z)$. Она может быть дополнена до невырожденной матрицы при помощи следующей процедуры (которая является обобщением аналогичных конструкций в ортогональном случае [BJ]). Поскольку векторы α_i в формуле (31) могут быть линейно зависимыми, мы будем рассматривать более общий случай:

$$\alpha(w) = \alpha_0 p_0(w) + \alpha_1 p_1(w) + \ldots + \alpha_{g-1} p_{g-1}(w), \tag{33}$$

где $\alpha_0, \alpha_1, \ldots, \alpha_{g-1}$ – g линейно независимых векторов из $\mathbf{C}^N$ и $p_0(w)$, $p_1(w), \ldots, p_{g-1}(w)$ – некоторые многочлены, ни один из которых не обращается в нуль на единичной окружности $w = e^{it}$.

Теорема 5. *Пусть* $\alpha = (\alpha_0, \alpha_1, \ldots, \alpha_{g-1})$ *– g линейно независимых векторов из* $\mathbf{C}^N$ *и* $p_0(w), p_1(w), \ldots, p_{g-1}(w)$ *– некоторые многочлены, ни один из которых не обращается в нуль на единичной окружности* $w = e^{-it}$. *Тогда существует полиномиальная петля* $A(w)$ *на группе* $GL(N, \mathbf{C})$, *такая что первая строка матрицы* $A(w)$ *есть* $\sum_{i=0}^{g-1} \alpha_i p_i(w)$, *при этом*

степень петли $A(w)$ не превосходит максимальной степени многочленов $p_i(w)$.

Замечание 7. Термин «полиномиальная петля» означает следующее. Для наших целей комплексная матрица $A(w)$ должна быть невырожденной, то есть она должна лежать в группе $GL(N, \mathbf{C})$ всех комплексных невырожденных матриц порядка N. Поскольку переменная $w = e^{-it}$ пробегает единичную окружность S^1, то $A(w) = A(e^{-it})$ будет семейством матриц, зависящих от точки окружности. Геометрически это замкнутая кривая, или петля, на группе $GL(N, \mathbf{C})$. Когда $A(w)$ зависит от переменной w полиноминально, мы говорим, что $A(w)$ – полиномиальная петля.

Доказательство теоремы. Будем использовать индукцию по максимальной степени k многочленов $p_i(w)$. Без ограничения общности можно считать, что сумма $\sum_{i=0}^{g-1} \alpha_i p_i(w)$ упорядочена по возрастанию степеней многочленов $p_i(w)$. Если $k = 0$, то α_0 – это некоторая ненулевая вектор-строка, и мы можем положить $A(w) = A_0$, где A_0 – это невырожденная (постоянная) матрица с первой строкой α_0.

Предположим, что $k > 0$ и что результат был доказан для всех меньших k. Мы можем считать, что $\alpha_{g-1} \neq 0$. Пусть P есть одномерная проекция на вектор α_{g-1} вдоль подпространства, образованного первыми векторами $(\alpha_0, \alpha_1, ..., \alpha_{g-2})$. Тогда

$$\alpha_{g-1}P = \alpha_{g-1}, \quad \alpha_l P = 0 \text{ для } l = 0, 1, ..., g-2.$$

Поскольку наши векторы α_l – это строки, а оператор P представлен матрицей, то действие P на вектор α мы представляем умножением слева вектора на матрицу. Напомним, что оператор проектирования обладает следующими свойствами:

$$P^2 = P, \quad P(1-P) = 0, \quad (1-P)^2 = 1 - P.$$

Теперь определим

$$\begin{aligned}
&\beta(w) = \alpha(w)(1 - P + p_{g-1}(w)^{-1}P) = \\
&= (\alpha_0 p_0(w) + \alpha_1 p_1(w) + ... + \alpha_{g-1} p_{g-1}(w))(1 - P + p_{g-1}(w)^{-1}P) = \\
&= p_0(w)\alpha_0(1-P) + p_0(w)p_{g-1}(w)^{-1}\alpha_0 P + p_1(w)\alpha_1(1-P) + \\
&+ p_1(w)p_{g-1}(z)^{-1}\alpha_1 P + ... + p_{g-1}(w)\alpha_{g-1}(1-P) + p_{g-1}(w)p_{g-1}(w)^{-1}\alpha_{g-1}P = \\
&= \alpha_{g-1} + \alpha_0 p_0(w) + \alpha_1 p_1(w) + ... + \alpha_{g-2} p_{g-2}(w).
\end{aligned}$$

Полином $\beta(w)$ имеет степень, меньшую, чем k, поскольку $\deg(p_{g-2}(w)) < \deg(p_{g-1}(w)) = k$. Следовательно, по индуктивному предположению, существует полиномиальная петля $B(w)$ степени $\deg(p_{g-2}(w))$, такая что первая строка $B(w)$ есть $\beta(w)$. Тогда полагаем:

$$A(w) = B(w)(1 - P + p_{g-1}(w)P).$$

Из полученного выражения для $\beta(w)$ следует, что первая строка $A(w)$ есть $\alpha(w)$. Это заканчивает индукцию и, таким образом, доказательство теоремы 5.

Практически данная теорема используется следующим образом. Если полиномиальная частотная функция $H_0(z)$ является заданной, то первая строка матрицы $A(w)$ известна в виде многочлена с векторными коэффициентами, $\alpha(w) = \sum_{i=0}^{g-1} \alpha_i p_i(w)$. После первого этапа проектирования P_1 на вектор α_{g-1} вдоль остальных векторов (α_0, α_1, ..., α_{g-2}) мы получаем строку $\beta(w)$ в виде многочлена с векторными коэффициентами степени меньше, чем степень $\alpha(w)$, $\beta(w) = \alpha(w)(1 - P_1 + p_{g-1}(w)^{-1}P_1)$. Для дальнейшего понижения степени мы применяем проектирование на вектор α_{g-2} вдоль остальных векторов (α_{g-1}, α_0, α_1, ..., α_{g-3}). В результате мы получаем строку $\gamma(w)$ в виде многочлена с векторными коэффициентами степени меньше, чем степень $\beta(w)$:

$$\gamma(w) = \beta(w)(1 - P_2 + p_{g-2}(w)^{-1}P_2) =$$
$$= \alpha(w)(1 - P_1 + p_{g-1}(w)^{-1}P_1)(1 - P_2 + p_{g-2}(w)^{-1}P_2).$$

Продолжая процедуру понижения степени, мы получим многочлен нулевой степени, то есть вектор ν_0:

$$\nu_0 = \alpha(w)(1 - P_1 + p_{g-1}(w)^{-1}P_1)...(1 - P_{g-1} + p_0(w)^{-1}P_{g-1}),$$

следовательно,

$$\alpha(w) = \nu_0(1 - P_{g-1} + p_0(w)P_{g-1})...(1 - P_1 + p_{g-1}(w)P_1).$$

Дополним этот вектор $\nu_0 \in \mathbf{C}^N$ до числовой невырожденной матрицы A_0. Тогда мы получаем выражение искомой полифазной матрицы через числовую невырожденную матрицу A_0 и серию проекторов:

$$A(w) = A_0(1 - P_{g-1} + p_0(w)P_{g-1})...(1 - P_1 + p_{g-1}(w)P_1). \qquad (34)$$

Поскольку задание числовой невырожденной матрицы A_0 с определенной первой строкой является несложной задачей, можно считать, что задача построения полиномиальной матрицы $A(w)$ решена.

Сформулируем теорему 5 в терминах низкочастотного фильтра $H_0(z)$.

Следствие. *Пусть $H_0(z)$ есть полином и $N \geq 2$. Предположим, что*

$$|H_0(z)|^2 + |H_0(z\rho)|^2 + ... + |H_0(z\rho^{N-1})|^2 \neq 0$$

для всех $z \in S^1$, где $\rho = e^{-i2\pi/N}$. Тогда существуют полиномы $\{H_l(z)$; $l = 1, ..., N-1\}$, такие что объединенная система $\{H_l(z)$; $0 \leq l \leq N-1\}$ удовлетворяет свойству невырожденности матрицы (23) *на единичной окружности $z = e^{-i\omega}$. Кроме того, если степень полинома $H_0(z)$ не*

превосходит K, тогда полиномы H_1, ..., H_{N-1} *могут быть взяты так, что их степень также не превосходит K.*

Замечание 8. Изложенные методы построения системы фильтров вейвлетов разложения $\{H_l(z);\ 0 \le l < N-1\}$ применимы и в ортогональном случае. Для этого должно выполняться условие

$$|H_0(z)|^2 + |H_0(z\rho)|^2 + ... + |H_0(z\rho^{N-1})|^2 = 1$$

на единичной окружности $z = e^{-i\omega}$ и в качестве матрицы $A(w)$ нужно строить унитарную матрицу.

Фильтры разложения и восстановления на основе B-сплайнов

Как известно, частотная функция $H_0(z)$ для B-сплайновой масштабирующей функции является полиномиальной и не определяет ортогонального кратномасштабного анализа. Приведенные выше конструкции позволяют строить неортогональные вейвлеты с полиномиальными частотными функциями и найти дуальные функции, которые обеспечивают точное восстановление сигнала.

Действительно, если задана частотная функция $H_0(z)$, то теорема 5 и ее доказательство позволяют построить матрицу $A(w)$ по формуле (34) и затем, по формуле $H(z) = A(z^N)R(z)$, найти матрицу $H(z)$ частных функций вейвлетов.

При вейвлет-разложении участвуют сопряженные фильтры вейвлетов. Это означает, что в качестве матрицы частотных функций фильтров разложения мы должны взять не матрицу $H(z)$, а комплексно-сопряженную матрицу $\overline{H(z)}$. По теореме 4 из раздела 2.13.5 матрица

$$G(z) = \begin{pmatrix} G_0(z) & G_0(\rho z) & \cdots & G_0(\rho^{N-1} z) \\ G_1(z) & G_1(\rho z) & \cdots & G_1(\rho^{N-1} z) \\ \cdots & \cdots & \cdots & \cdots \\ G_{N-1}(z) & G_{N-1}(\rho z) & \cdots & G_{N-1}(\rho^{N-1} z) \end{pmatrix}$$

фильтров восстановления является транспонированной к обратной матрице $\overline{H(z)}$. Поскольку $H(z) = A(z^N)R(z)$, то в соответствии с формулой $G^t(z)\overline{H(z)} = 1$ теоремы 4 и формулой (25) матрица $G(z)$ может быть найдена из выражения:

$$G^t(z) = \overline{H(z)^{-1}} = \overline{R(z)^{-1}}\,\overline{A(z^N)^{-1}} = \frac{1}{N} R^t(z) A(z^{-N})^{-1},$$

$$G(z) = \frac{1}{N}(A(z^{-N})^{-1})^{t} R(z). \tag{35}$$

Здесь мы использовали вещественность коэффициентов матрицы $A(z^N)$ и то, что на единичной окружности $z = e^{-i\omega}$ выполняется равенство $\bar{z} = z^{-1}$.

Последняя формула дает возможность найти фильтры восстановления (дуальные вейвлеты) как коэффициенты $\{\tilde{g}_n^l\}$ функций $G_l(z) = \frac{1}{\sqrt{N}} \sum_n \tilde{g}_n^l z^n$, $l = 0, 1, \ldots, N-1$, первого столбца матрицы $G(z)$. Дуальная масштабирующая функция $\tilde{\varphi}(x)$ может быть найдена по своему фильтру $\{g_n^0\}$ обычными методами (см. раздел 2.2.2).

После построения двойственной масштабирующей функции $\tilde{\varphi}(x)$ дуальные вейвлеты $\tilde{\psi}^1(x)$, ..., $\tilde{\psi}^{N-1}(x)$ находятся по обычным формулам:

$$\tilde{\psi}^l(x) = \sqrt{N} \sum_n \tilde{g}_n^l \tilde{\varphi}(Nx - n).$$

Покажем на примере эффективность изложенного метода.

Построение неортогональных вейвлетов на основе B-сплайна степени 1

Здесь мы достаточно подробно на примере проиллюстрируем построение неортогональных вейвлетов на основе B-сплайнов. Пусть $N = 3$. Рассмотрим B-сплайн $\varphi(x)$ степени 1:

$$\varphi(x) = \begin{cases} 1 - |x|, & x \in [-1,1], \\ 0 & \text{при остальных } x. \end{cases}$$

Как мы уже ранее отмечали, частотная функция имеет вид

$$H_0(z) = \frac{1}{3^2} z^{-2} (1 + z + z^2)^2 = \frac{1}{9} z^{-2} (1 + 2z + 3z^2 + 2z^3 + z^4).$$

Вычислим полифазную матрицу $A(w)$, $w = z^3$. Для удобства вычислений опустим временно множитель $1/9$ и положим

$$m_0(z) = z^{-2}(1 + z + z^2)^2 = z^{-2} + 2z^{-1} + 3 + 2z^1 + z^2.$$

Поскольку $N = 3$, то $m_0(z) = A_{0,0}(z^3) + zA_{0,1}(z^3) + z^2 A_{0,2}(z^3)$. Нахождение многочленов $A_{0,0}(z^3)$, $A_{0,1}(z^3)$ и $A_{0,2}(z^3)$ – это выборка элементов со степенями z^{3k} в полиномах $m_0(z)$, $z^{-1}m_0(z)$, $z^{-2}m_0(z)$. Поэтому $A_{0,0}(w) = 3$. Далее:

$$z^{-1}m_0(z) = z^{-3} + 2z^{-2} + 3z^{-1} + 2 + z, \quad A_{0,1}(w) = w^{-1} + 2.$$

$z^{-2}m_0(z) = z^{-4} + 2z^{-3} + 3z^{-2} + 2z^{-1} + z, \quad A_{0,2}(w) = 2w^{-1} + 1.$

Раскладываем первую строку по степеням с векторными коэффициентами:

$(A_{0,0}(w), A_{0,1}(w), A_{0,2}(w)) = (3, w^{-1} + 2, 2w^{-1} + 1) = (0, 1, 2)w^{-1} + (3, 2, 1),$

$\alpha_{-1} = (0, 1, 2), \alpha_0 = (3, 2, 1).$

Дополним векторы $\alpha_{-1} = (0, 1, 2)$ и $\alpha_0 = (3, 2, 1)$ до базиса $\mathbf{C}^3$ вектором $\alpha_1 = (0, 0, 1)$. Пусть P есть одномерная проекция на вектор α_{-1} вдоль α_0 и α_1, то есть $\alpha_{-1}P = \alpha_{-1}$ и $\alpha_0 P = 0$, $\alpha_1 P = 0$. Теперь определим

$\beta(w) = \alpha(w)(1 - P + wP) = (\alpha_0 + \alpha_1 w^{-1})(1 - P + wP) = (\alpha_0 + \alpha_1).$

Легко видеть, что $(1 - P + w^{-1}P)$ является обратной матрицей для $(1 - P + wP)$, поэтому

$\alpha(w) = \beta(w)(1 - P + w^{-1}P) = (\alpha_0 + \alpha_1)(1 - P + w^{-1}P),$

где $\alpha_0 + \alpha_1 = (3, 3, 3)$.

Дополняем эту первую строку $\alpha_0 + \alpha_1$ до невырожденной матрицы, например строками $(1, -1, 0)$ и $(1, 1, -2)$. Получим матрицу с ортогональными строками:

$$A_0 = \begin{pmatrix} 3 & 3 & 3 \\ 1 & -1 & 0 \\ 1 & 1 & -2 \end{pmatrix}.$$

Тогда $A(w) = A_0(1 - P + w^{-1}P)$. В базисе $\{\alpha_{-1}, \alpha_0, \alpha_1\}$ операторы проектирования принимают вид

$$P = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix}, \quad 1 - P = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}, \quad 1 - P + w^{-1}P = \begin{pmatrix} w^{-1} & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}.$$

Найдем выражение оператора $(1 - P + w^{-1}P)$ в стандартном базисе e_0, e_1, e_2 пространства $\mathbf{C}^3$. Для этого найдем матрицы B и B^{-1} перехода от одного базиса к другому $\alpha = Be$, $e = B^{-1}\alpha$. Имеем следующие формулы преобразования базисов:

$$\begin{pmatrix} \alpha_{-1} \\ \alpha_0 \\ \alpha_1 \end{pmatrix} = \begin{pmatrix} 0 & 1 & 2 \\ 3 & 2 & 1 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} e_0 \\ e_1 \\ e_2 \end{pmatrix}, \quad \begin{pmatrix} e_0 \\ e_1 \\ e_2 \end{pmatrix} = \begin{pmatrix} -2/3 & 1/3 & 1 \\ 1 & 0 & -2 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} \alpha_0 \\ \alpha_1 \\ \alpha_2 \end{pmatrix}.$$

Пусть $P^j_{\alpha k}$ – матрица оператора $(1 - P + w^{-1}P)$ в базисе $(\alpha_{-1}, \alpha_0, \alpha_1)$, здесь нижний индекс k – номер строки. Тогда матрица оператора $(1 - P + w^{-1}P)$ в стандартном базисе e вычисляется по формуле

$$P_e(w) = B^{-1}P_\alpha B, \quad P_{es}^{p} = (B^{-1})_s^k P_{\alpha k}^j B_j^p.$$

Поэтому

$$P_e(w) = \begin{pmatrix} 1 & 2/3(1-w^{-1}) & 4/3(1-w^{-1}) \\ 0 & w^{-1} & -2(1-w^{-1}) \\ 0 & 0 & 1 \end{pmatrix}.$$

Это позволяет вычислить полиномиальную матрицу $A(w) = A_0P_e(w)$:

$$A(w) = \begin{pmatrix} 3 & 3 & 3 \\ 1 & -1 & 0 \\ 1 & 1 & -2 \end{pmatrix}\begin{pmatrix} 1 & 2/3(1-w^{-1}) & 4/3(1-w^{-1}) \\ 0 & w^{-1} & -2(1-w^{-1}) \\ 0 & 0 & 1 \end{pmatrix} =$$

$$= \begin{pmatrix} 3 & 2+w^{-1} & 1+2w^{-1}) \\ 1 & (2-5w^{-1})/3 & 10(1-w^{-1})/3 \\ 1 & (2+w^{-1})/3 & -2(4-w^{-1})/3 \end{pmatrix}.$$

Найдем матрицу $H(z)$ частотных функций (с учетом опущенного ранее коэффициента 1/9) по формуле $H(z) = \frac{1}{9}A(z^N)R(z)$:

$$\begin{pmatrix} H_0(z) & H_0(\rho z) & H_0(\rho^2 z) \\ H_1(z) & H_1(\rho z) & H_1(\rho^2 z) \\ H_2(z) & H_2(\rho z) & H_2(\rho^2 z) \end{pmatrix} =$$

$$= \frac{1}{9}\begin{pmatrix} 3 & 2+z^{-3} & 1+2z^{-3}) \\ 1 & (2-5z^{-3})/3 & 10(1-z^{-3})/3 \\ 1 & (2+z^{-3})/3 & -2(4-z^{-3})/3 \end{pmatrix}\begin{pmatrix} 1 & 1 & 1 \\ z & \rho z & \rho^2 z \\ z^2 & \rho^2 z^2 & \rho^4 z^2 \end{pmatrix}.$$

Получаем следующие частотные функции и фильтры вейвлетов:

$$H_0(z) = \frac{1}{9}(z^{-2} + 2z^{-1} + 3 + 2z^1 + z^2),$$

$$H_1(z) = \frac{1}{27}(-5z^{-2} - 10z^{-1} + 3 + 2z + 10z^2),$$

$$H_2(z) = \frac{1}{27}(z^{-2} + 2z^{-1} + 3 + 2z - 8z^2).$$

Найдем фильтры восстановления. Следует учесть, что при вейвлет-разложении участвуют сопряженные фильтры вейвлетов. Это

означает, что для нахождения матрицы $G(z)$ мы должны взять матрицу $\overline{H(z)}$, образованную сопряженными функциями $\overline{H_0(z)}$, $\overline{H_1(z)}$ и $\overline{H_2(z)}$. По теореме 4 из раздела 2.13.5 матрица $G(z)$ фильтров восстановления находится обращением матрицы $\overline{H(z)}$, $G(z)^t = \overline{H(z)}^{-1} = 9\overline{R(z)^{-1}A(z^N)^{-1}}$. Элементарные вычисления показывают, что

$$R(z)^{-1} = \frac{1}{3}\begin{pmatrix} 1 & z^{-1} & z^{-2} \\ 1 & \rho^{-1}z^{-1} & \rho^{-2}z^{-2} \\ 1 & \rho^{-2}z^{-1} & \rho^{-4}z^{-2} \end{pmatrix},$$

$$A(z^N)^{-1} = \begin{pmatrix} (3-2z^3)/9 & (1+2z^3)/6 & (-1+2z^3)/6 \\ (-2+3z^3)/9 & -z^3/2 & (4-3z^3)/6 \\ 1/9 & 0 & -1/3 \end{pmatrix}.$$

Вычислим $\overline{G(z)}^t = N^2 R(z)^{-1} A(z^N)^{-1}$. После перемножения выбираем элементы первой строки и получаем следующие функции:

$$\overline{G_0(z)} = \frac{1}{3}(z^{-2} - 2z^{-1} + 3 + 3z^2 - 2z^3),$$

$$\overline{G_1(z)} = \frac{1}{2}(1 - 3z^2 + 2z^3),$$

$$\overline{G_2(z)} = \frac{1}{2}(-2z^{-2} + 4z^{-1} - 1 - 3z^2 + 2z^3).$$

Поскольку коэффициенты этих функций вещественные, то для нахождения $G_0(z)$, $G_1(z)$, $G_2(z)$ достаточно заменить $z = e^{-i\omega}$ на $\bar{z} = z^{-1}$:

$G_0(z) = {}^1/_3(-2z^{-3} + 3z^{-2} + 3 - 2z + z^2)$,
$G_1(z) = {}^1/_2(2z^{-3} - 3z^{-2} + 1)$, $G_2(z) = {}^1/_2(2z^{-3} - 3z^{-2} - 1 + 4z - 2z^2)$.

Учитывая коэффициент $1/\sqrt{3}$ в формулах частотных функций (теорема 4 раздела 2.13.5), получаем следующие фильтры разложения и восстановления (ненулевые элементы).

Фильтры вейвлетов:

$$h_{-2} = \frac{1}{3\sqrt{3}}, h_{-1} = \frac{2}{3\sqrt{3}}, h_0 = \frac{1}{\sqrt{3}}, h_1 = \frac{2}{3\sqrt{3}}, h_2 = \frac{1}{3\sqrt{3}},$$

$$h^1_{-2} = \frac{-5}{9\sqrt{3}}, h^1_{-1} = \frac{-10}{9\sqrt{3}}, h^1_0 = \frac{1}{3\sqrt{3}}, h^1_1 = \frac{2}{9\sqrt{3}}, h^1_2 = \frac{10}{9\sqrt{3}},$$

$$h^2_{-2} = \frac{1}{9\sqrt{3}}, h^2_{-1} = \frac{2}{9\sqrt{3}}, h^2_0 = \frac{1}{3\sqrt{3}}, h^2_1 = \frac{2}{9\sqrt{3}}, h^2_2 = \frac{-8}{9\sqrt{3}}.$$

Фильтры восстановления:

$$\tilde{h}_{-3}=\frac{-2}{\sqrt{3}},\tilde{h}_{-2}=\sqrt{3},\ \tilde{h}_{-1}=0,\tilde{h}_{0}=\sqrt{3},\tilde{h}_{1}=\frac{-2}{\sqrt{3}},\tilde{h}_{-2}=\frac{1}{\sqrt{3}},$$

$$\tilde{g}^{1}_{-3}=\sqrt{3},\tilde{g}^{1}_{-2}=-\frac{3\sqrt{3}}{2},\tilde{g}^{1}_{-1}=0,\tilde{g}^{1}_{0}=\frac{\sqrt{3}}{2},\tilde{g}^{1}_{1}=0,\tilde{g}^{1}_{2}=0,$$

$$\tilde{g}^{2}_{-3}=\sqrt{3},\tilde{g}^{2}_{-2}=-\frac{3\sqrt{3}}{2},\tilde{g}^{2}_{-1}=0,\tilde{g}^{2}_{0}=-\frac{\sqrt{3}}{2},\tilde{g}^{2}_{1}=2\sqrt{3},\tilde{g}^{2}_{2}=-\sqrt{3}.$$

Отметим, что, в отличие от симметричных биортогональных вейвлетов для $N=2$, которые строились в разделе 2.10.6, полученные вейвлеты $\psi^1(x)$ и $\psi^2(x)$ являются несимметричными (рис. 2.14.6).

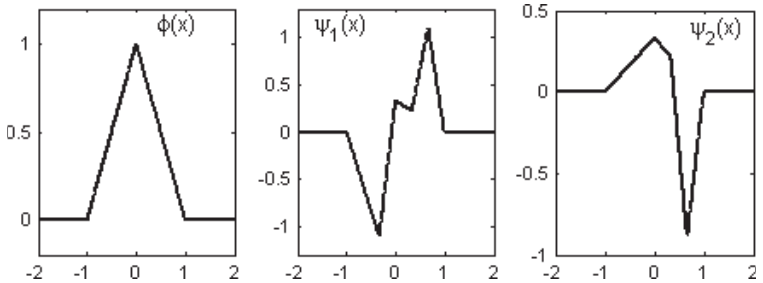


Рис. 2.14.6. Графики функций $\varphi(x)$ и $\psi^1(x)$, $\psi^2(x)$

Дуальная масштабирующая функция $\tilde{\varphi}(x)$ получается разрывной, что показывают ее первые 5 приближений (рис. 2.14.7). Отметим, что для коэффициентов $\tilde{h}_n$ свойство $\sum_n \tilde{h}_n=\sqrt{3}$ выполняется.

Программа на Maple для вычисления фильтров разложения и восстановления для B-сплайна степени 1, $N=3$, $p=1$.

```
> restart:
> with(LinearAlgebra):with(linalg):with(PolynomialTools):
Исходная частотная функция и ее преобразование Фурье
Без учета коэффициента 1/9. Преобразование Фурье сводится
к выбору степеней, кратных 3, в выражениях H_0(z),
z^(-1)*H_0(z), z^(-2)*H_0(z)
> N:=3: p:=1: K:=N^(p+1);
> H0z:=z^(-2)+2*z^(-1)+3+2*z+z^2; A00z:=3;
> H0z1:=expand(H0z*z^(-1)); A01z:=z^(-3)+2;
> H0z2:=expand(H0z*z^(-2)); A02z:=2*z^(-3)+1;
> A00w:=3; A01w:=w^(-1)+2; A02w:=2*w^(-1)+1;
Векторные коэффициенты для строки alpha(w)
```

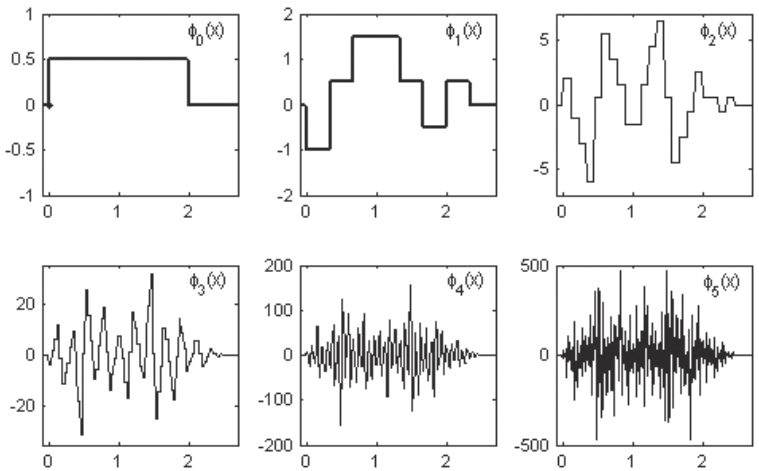


Рис. 2.14.7. Последовательные приближения функции $\tilde{\varphi}(x)$

```
> alphaw:=<A00w | A01w | A02w>;
> al_1:=<0 | 1 | 2>;
> al0:=<3 | 2 | 1>;
Дополняем вектором al1
> al1:=<0 | 0 | 1>;
Удаление степеней. Проектор P1 на вектор al_1 вдоль других
векторов al0 и al1 и оператор Pw1=(1-P1+w^(-1)P1). Первая
матрица перехода B, ее обратная B^(-1) и оператор Pe
в стандартном базисе C3
> Pa:=matrix(3,3,[[w^(-1),0,0],[0,1,0],[0,0,1]]);
> B:= <al_1,al0,al1>;
> B_inv:=inverse(B);
> Pe:=multiply(multiply(B_inv,Pa),B);
> Pe_inv:=inverse(Pe);
Вычисление строки чисел A_str=alpha(w)*(1-P1+wP1)
> A_1str:=simplify(multiply(alphaw,Pe_inv));
Дополняем полученный вектор до матрицы A0 с ортогональными строками
> A0:=<A_1str, <1 | -1 | 0 >, <1 | 1 | -2>>;
Вычисление матрицы A(w)=A(z^3), A(w) = A0*(1-P1+w^(-1)P1) = A0*Pe
> Aw:=simplify(multiply(A0,Pe));
Замена w на z^3 в матрице A
> Az:=simplify(Aw,{w=z^3});
Вспомогательные матрицы R и R_inv. R_inv - это обратная, то есть
транспонированная комплексно-сопряженная и деленная на N
> R:=matrix(3,3,[[1,1,1],[z,rho*z,rho^2*z],[z^2,rho^2*z^2,rho^4*z^2]]);
> R_inv:=matrix(3,3,[[1,z^(-1),z^(-2)],[1,(rho*z)^(-1),
(rho*z)^(-2)],[1,(rho)^(-2)*(z)^(-1),rho^(-4)*z^(-2)]])/N;
```

```
> s:=simplify(multiply(R_inv,R),{1+rho+rho^2=0});
Вычисление матриц фильтров M(z)=A(z^3)*R(z) и H(z)=M(z)/K,
где K - опущенный ранее коэффициент K=N^(p+1)
> M:=simplify(multiply(Az,R)): H:=evalm(M/K);
Частотные функции вейвлетов. Выборка элементов первого столбца.
> H0:=H[1,1]; H1:=H[2,1]; H2:=H[3,1];
Вычисление сопряженной матрицы фильтров восстановления G(z) по формуле
G=R_inv(z)*A(z^N)^(-1). Сначала находим обратную матрицу A(z^N)^(-1)
и учитываем опущенный ранее коэффициент K
> Az_inv:=inverse(Az);
> Gz:=simplify(multiply(R_inv,K*Az_inv));
Проверка G*H=1
> Q:=simplify(multiply(Gz,H),{1+rho+rho^2=0});
Выборка элементов первой строки матрицы Gz -
сопряженные частотные функции фильтров восстановления.
> G00:=Gz[1,1]; G10:=Gz[1,2]; G20:=Gz[1,3];
Переход от сопряженных частотных функций к обычным.
> G0:=sort(expand(subs(z=z^(-1),G00)));
> G1:=sort(expand(subs(z=z^(-1),G10)));
> G2:=sort(expand(subs(z=z^(-1),G20)));
```

Схема лифтинга

При построении фильтров разложения предложенным в этом разделе методом получается, что степени полиномов $H_1(z)$, ..., $H_{N-1}(z)$ не превосходят степени полинома $H_0(z)$. Это не всегда удобно. Отметим также, что при построении симметричных биортогональных вейвлетов в разделе 2.10.7 многочлен $\tilde{H}_0(z)$ мог иметь степень больше, чем степень $H_0(z)$. Так, в примере 1 раздела 2.10.7 многочлен $H_0(z)$ имеет степень 2, а многочлен $\tilde{H}_0(z)$ – степень 4. Для построения полиномов $H_1(z)$, ..., $H_{N-1}(z)$, степени которых больше степени полинома $H_0(z)$, можно использовать следующий способ.

Предположим, что полифазная матрица $A(w)$ найдена. Первая строка $\alpha(w) = (A_{0,0}(w), A_{0,1}(w), ..., A_{0,N-1}(w))$ этой матрицы считается заданной. Для построения полифазной матрицы, у которой первая строка будет та же самая, а другие строки будут содержать многочлены более высоких степеней, достаточно умножить слева эту матрицу $A(w)$ на невырожденную полиномиальную матрицу $B(w)$ вида

$$B(w) = \begin{pmatrix} 1 & 0 & \cdots & 0 \\ B_{1,0}(w) & B_{1,1}(w) & \cdots & B_{1,N-1}(w) \\ \cdots & \cdots & \cdots & \cdots \\ B_{N-1,0}(w) & B_{N-1,1}(w) & \cdots & B_{N-1,N-1}(w) \end{pmatrix}.$$

В качестве такой матрицы можно взять, например, диагональную матрицу с некоторыми многочленами на диагонали, либо нижнетреугольную матрицу вида

$$B(w)=\begin{pmatrix} 1 & 0 & 0 & \cdots & 0 \\ B_{1,0}(w) & 1 & 0 & \cdots & 0 \\ B_{2,0}(w) & B_{2,1}(w) & 1 & \cdots & 0 \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ B_{N-1,0}(w) & B_{N-1,1}(w) & B_{N-1,2}(w) & \cdots & 1 \end{pmatrix},$$

где многочлены $B_{i,j}(w)$ – произвольные.

Этот способ изменения фильтров разложения с целью их адаптации к исследуемому сигналу называется схемой лифтинга.

2.14.6. Кратные коэффициенты масштабирования

Известно очень много примеров масштабирующих функций и вейвлетов для $N = 2$. На их основе можно построить N-масштабирующие функции и *вейвлеты*, когда N является степенью двойки, $N = 2^k$.

Если $\varphi(x)$ есть масштабирующая функция с масштабным коэффициентом N, то легко видеть, что она является масштабирующей функцией с масштабным коэффициентом N^p для любого натурального показателя p. Действительно, если

$$\varphi(x)=\sqrt{N}\sum_{k\in\mathbf{Z}}h_k\varphi(Nx-k),$$

то можно написать масштабирующее соотношение для функции $\varphi(Nx-k)$:

$$\varphi(Nx-k)=\sqrt{N}\sum_{m\in\mathbf{Z}}h_m\varphi(N(Nx-k)-m)=$$

$$=\sqrt{N}\sum_{m\in\mathbf{Z}}h_m\varphi(N^2x-Nk-m)=\sqrt{N}\sum_{n\in\mathbf{Z}}h_{n-Nk}\varphi(N^2x-n)$$

и подставить это выражение в исходное масштабирующее соотношение:

$$\begin{aligned}\varphi(x)&=\sqrt{N}\sum_{k\in\mathbf{Z}}h_k\sqrt{N}\sum_{n\in\mathbf{Z}}h_{n-Nk}\varphi(N^2x-n)=\\&=N\sum_{n\in\mathbf{Z}}\left(\sum_{k\in\mathbf{Z}}h_kh_{n-Nk}\right)\varphi(N^2x-n).\end{aligned}\qquad(36)$$

Мы видим, что масштабирующая функция $\varphi(x)$ с масштабным коэффициентом N удовлетворяет масштабирующему соотношению с масштабным коэффициентом N^2. При этом масштабирующий фильтр состоит из чисел вида

$$h_n^{(2)} = \sum_{k\in\mathbf{Z}} h_k h_{n-Nk}. \tag{37}$$

Продолжая эту процедуру подстановки, получаем требуемое утверждение для любого p. На уровне частотных функций. Если

$$\hat{\varphi}(\omega) = H_0\left(\frac{\omega}{N}\right)\hat{\varphi}\left(\frac{\omega}{N}\right),$$

то, подставляя $\hat{\varphi}(\omega/N) = H_0(\omega/N^2)\,\hat{\varphi}(\omega/N^2)$, получаем:

$$\hat{\varphi}(\omega) = H_0\left(\frac{\omega}{N}\right)H_0\left(\frac{\omega}{N^2}\right)\cdots H_0\left(\frac{\omega}{N^p}\right)\hat{\varphi}\left(\frac{\omega}{N^p}\right) = H_0^{(p)}\left(\frac{\omega}{N^p}\right)\hat{\varphi}\left(\frac{\omega}{N^p}\right),$$

где частотная функция $H_0^{(p)}(\omega)$ масштабирующей функции $\varphi(x)$ с масштабным коэффициентом N^p имеет вид

$$H_0^{(p)}(\omega) = H_0(\omega N^{p-1})H_0(\omega N^{p-2})...H_0(\omega). \tag{38}$$

Замечание 9. Легко показать, что если функция $\varphi(x)$ является масштабирующей с масштабными коэффициентами N_1 и N_2, то она является масштабирующей с масштабным коэффициентом N_1N_2.

Рассмотрим построение вейвлетов с параметром масштабирования $N = 2^p$. Пусть

$$\psi(x) = \sqrt{2}\sum_n g_n\,\varphi(2x-n) \tag{39}$$

– вейвлет, соответствующий функции $\varphi(x)$, и

$$H_1(\omega) = \sqrt{2^{-1}}\sum_n g_n e^{-in\omega} \tag{40}$$

– его частотная функция.

Процедуру построения вейвлетов с параметром масштабирования $N = 2^p$ рассмотрим для случая $N = 4$. Как уже мы отмечали, функция $\varphi(x)$ является 4-масштабирующей функцией. Напишем сначала масштабирующее соотношение для $N = 4$:

$$\varphi(x) = 2\sum_{k\in\mathbf{Z}} h_k^{(2)}\varphi(4x-k), \tag{41}$$

где масштабирующий фильтр состоит из чисел вида:

$$h_k^{(2)} = \sum_{n \in \mathbf{Z}} h_n h_{k-2n}. \qquad (42)$$

Пусть

$$... \subset V_{-1} \subset V_0 \subset V_1 \subset V_2 \subset ...$$

– ортогональное 2-кратномасштабное разложение пространства $L^2(\mathbf{R})$, определенное функцией $\varphi(x)$. Очевидно, что ортогональное 4-кратномасштабное разложение пространства $L^2(\mathbf{R})$ состоит из следующих подпространств:

$$... \subset V_{-2} \subset V_0 \subset V_2 \subset V_4 \subset V_6 \subset ...$$

Пространства вейвлетов W_0^1, W_0^2 и W_0^3 образуют ортогональное разложение

$$V_2 = V_0 \oplus W_0^1 \oplus W_0^2 \oplus W_0^3.$$

Поэтому для нахождения этих пространств воспользуемся ортогональным 2-вейвлет-разложением пространства V_2:

$$V_2 = V_1 \oplus W_1 = V_0 \oplus W_0 \oplus W_1,$$

где два первых пространства имеют одинаковый «размер», порождены целочисленными сдвигами $\varphi(x - n)$, $\psi(x - k)$, а третье пространство – «в два раза больше», оно порождено полуцелыми сдвигами $\psi(2x - k)$. Применим процедуру пакетного вейвлет-разложения к пространству W_1, тогда у нас появляются два новых пространства и ортогональное разложение:

$$W_1 = W_{10} \oplus W_{11}.$$

Пространства W_{10} и W_{11} порождены, соответственно, целочисленными сдвигами функций (пакетные вейвлеты):

$$w_2(x) = \sqrt{2} \sum_{n \in \mathbf{Z}} h_n \psi(2x - n) \quad \text{и} \quad w_3(x) = \sqrt{2} \sum_{k \in \mathbf{Z}} g_k \psi(2x - k).$$

Мы получили ортогональное разложение

$$V_2 = V_0 \oplus W_0^1 \oplus W_0^2 \oplus W_0^3, \qquad (43)$$

где каждое пространство порождено целочисленными сдвигами одной функции. Таким образом:

- пространство V_0 имеет ортонормированный базис $\{\varphi(x - n)\}$;
- пространство $W_0^1 = W_0$ имеет ортонормированный базис $\{\psi(x - n)\}$;

- пространство $W_0^2 = W_{10}$ имеет ортонормированный базис $\{w_2(x-n)\}$;
- пространство $W_0^3 = W_{11}$ имеет ортонормированный базис $\{w_3(x-n)\}$.

Это означает, что мы нашли вейвлеты $\psi^1(x)$, $\psi^2(x)$ и $\psi^3(x)$ для масштабирующей функции $\varphi(x)$:

$$\psi^1(x) = \psi(x), \quad \psi^2(x) = w_2(x), \quad \psi^3(x) = w_3(x), \tag{44}$$

где $w_2(x)$ и $w_3(x)$ – вейвлет-пакетные функции, соответствующие ортогональным 2-вейвлетам $\varphi(x)$ и $\psi(x)$.

Найдем фильтры данных вейвлет-функций $\varphi(x)$, $\psi^1(x)$, $\psi^2(x)$ и $\psi^3(x)$. Как мы уже отмечали,

$$\varphi(x) = 2\sum_{k\in\mathbf{Z}} h_k^{(2)}\varphi(4x-k), \quad \text{где } h_k^{(2)} = \sum_{n\in\mathbf{Z}} h_n h_{k-2n}. \tag{45}$$

Для вейвлета $\psi(x)$ по построению мы имеем:

$$\psi(x) = \sqrt{2}\sum_{n\in\mathbf{Z}} g_n\varphi(2x-n) = 2\sum_{n\in\mathbf{Z}} g_n \sum_k h_k\ \varphi(4x-2n-k) =$$

$$= 2\sum_m \sum_{n\in\mathbf{Z}} g_n h_{m-2n}\varphi(4x-m).$$

Поэтому

$$\psi^1(x) = 2\sum_m g_m^1\varphi(4x-m), \quad \text{где } g_m^1 = \sum_{n\in\mathbf{Z}} g_n h_{m-2n}. \tag{46}$$

Аналогично для функций $\psi^2(x) = w_2(x)$ и $\psi^3(x) = w_3(x)$ имеем:

$$\psi^2(x) = 2\sum_m g_m^2\varphi(4x-m), \quad \text{где } g_m^2 = \sum_{n\in\mathbf{Z}} h_n g_{m-2n}; \tag{47}$$

$$\psi^3(x) = 2\sum_m g_m^3\varphi(4x-m), \quad \text{где } g_m^3 = \sum_{n\in\mathbf{Z}} g_n g_{m-2n}. \tag{48}$$

Из равенств

$$\hat{\varphi}(\omega) = H_0\left(\frac{\omega}{2}\right)\hat{\varphi}\left(\frac{\omega}{2}\right), \qquad \hat{\psi}(\omega) = H_1\left(\frac{\omega}{2}\right)\hat{\varphi}\left(\frac{\omega}{2}\right),$$

$$\hat{w}_2(\omega) = H_0\left(\frac{\omega}{2}\right)\hat{\psi}\left(\frac{\omega}{2}\right), \quad \hat{w}_3(\omega) = H_1\left(\frac{\omega}{2}\right)\hat{\psi}\left(\frac{\omega}{2}\right)$$

сразу следуют выражения частотных функций $H_0^{(2)}(\omega)$, $H_1^{(2)}(\omega)$, $H_2^{(2)}(\omega)$ и $H_3^{(2)}(\omega)$ вейвлетов $\varphi(x)$, $\psi^1(x)$, $\psi^2(x)$ и $\psi^3(x)$:

$$H_0^{(2)}(\omega) = H_0(2\omega)H_0(\omega), \quad H_1^{(2)}(\omega) = H_1(2\omega)H_0(\omega),$$

$$H_2^{(2)}(\omega) = H_0(2\omega)H_1(\omega), \quad H_3^{(2)}(\omega) = H_1(2\omega)H_1(\omega).$$

Пример 3. Рассмотрим вейвлеты с коэффициентом масштабирования 4, порожденные вейвлетами Хаара с коэффициентом 2. Фильтры 2-вейвлетов Хаара имеют коэффициенты $h_0 = h_1 = 1/\sqrt{2}$ и $g_0 = -g_1 = 1/\sqrt{2}$. Тогда

$H_0(z) = {}^1/_2(1 + z)$, $H_1(z) = {}^1/_2(1 - z)$.

Поэтому частотные функции имеют вид:

$$H_0^{(2)}(z) = H_0(z^2)H_0(z) = \frac{1}{4}(1+z^2)(1+z) = \frac{1}{4}(1+z+z^2+z^3),$$

$$H_1^{(2)}(z) = H_1(z^2)H_0(z) = \frac{1}{4}(1-z^2)(1+z) = \frac{1}{4}(1+z-z^2-z^3),$$

$$H_2^{(2)}(z) = H_0(z^2)H_1(z) = \frac{1}{4}(1+z^2)(1-z) = \frac{1}{4}(1-z+z^2-z^3),$$

$$H_3^{(2)}(z) = H_1(z^2)H_1(z) = \frac{1}{4}(1-z^2)(1-z) = \frac{1}{4}(1-z-z^2+z^3).$$

Масштабирующие соотношения для вейвлетов Хаара:

$\varphi(x) = \varphi(4x) + \varphi(4x - 1) + \varphi(4x - 2) + \varphi(4x - 3)$,
$\psi^1(x) = \varphi(4x) + \varphi(4x - 1) - \varphi(4x - 2) - \varphi(4x - 3)$,
$\psi^2(x) = \varphi(4x) - \varphi(4x - 1) + \varphi(4x - 2) - \varphi(4x - 3)$,
$\psi^3(x) = \varphi(4x) - \varphi(4x - 1) - \varphi(4x - 2) + \varphi(4x - 3)$.

Ниже приведены графики (рис. 2.14.8) полученных вейвлет-функций Хаара с параметром масштабирования 4. Отметим, что вейвлеты Хаара с $N = 4$ могут быть построены и другим способом, который изложен в разделе 2.14.1.

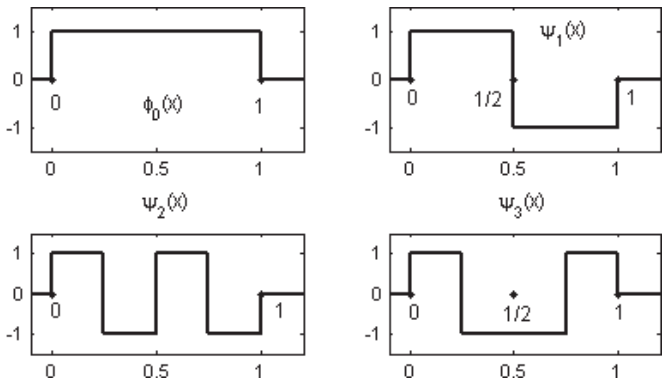


Рис. 2.14.8. Графики вейвлетов Хаара для $N = 4 = 2^2$

2.15. Построение ортогональных вейвлетов с компактным носителем для $N > 2$

В разделах 2.8 и 2.10 изложены методы построения вейвлетов с полиномиальными частотными функциями для $N = 2$. Однако существует и другой способ, основанный на построении полифазных матриц [BJ], [J03] и более приспособленный для коэффициента масштабирования $N > 2$. В разделе 2.14 он использовался для построения неортогональных вейвлетов на основе B-сплайнов. В работе [BJ] показано, как построить полифазную матрицу при известной масштабирующей функции. В данном разделе приведем простой метод построения достаточно большого семейства *ортогональных вейвлетов* с компактными носителями, не предполагая задания масштабирующей функции, то есть построим и масштабирующую функцию $\varphi(x)$, и вейвлеты $\psi^1(x)$, ..., $\psi^{N-1}(x)$, с полиномиальными частотными функциями и с вещественными фильтрами. Схема построения будет следующая.

Сначала мы построим унитарную матрицу

$$H(z) = \begin{pmatrix} H_0(z) & H_0(\rho z) & \cdots & H_0(\rho^{N-1} z) \\ H_1(z) & H_1(\rho z) & \cdots & H_1(\rho^{N-1} z) \\ \cdots & \cdots & \cdots & \cdots \\ H_{N-1}(z) & H_{N-1}(\rho z) & \cdots & H_{N-1}(\rho^{N-1} z) \end{pmatrix} \tag{1}$$

частотных полиномиальных функций $H_0(z)$, $H_1(z)$, ..., $H_{N-1}(z)$, где $z = e^{-i\omega}$ и $\rho = e^{-i2\pi/N}$. Тогда будет выполнено необходимое условие ортогональности $\sum_{s=0}^{N-1} |H_0(\omega + 2\pi sm/N)|^2 = 1$.

На втором шаге мы построим масштабирующую функцию $\varphi(x)$ формулой

$$\hat{\varphi}(\omega) = \prod_{j=1}^{\infty} H_0\left(\frac{\omega}{N^j}\right) \tag{2}$$

и проверим выполнение свойства ортонормированности системы функций $\phi_{0,n}(x) = \phi(x - n)$, $n \in \mathbf{Z}$.

На третьем этапе находим вейвлеты $\psi^1(x)$, ..., $\psi^{N-1}(x)$ по формулам

$$\psi^l(x) = \sqrt{N} \sum_n g_n^l \varphi(Nx - n), \quad l = 1, 2, \ldots, N-1,$$

где g_n^l – коэффициенты многочленов $H_1(z)$, ..., $H_{N-1}(z)$:

$$H_k(\omega)=\frac{1}{\sqrt{N}}\sum\nolimits_n g_n^k e^{-in\omega},\ k=1,2,\ldots,N-1.$$

2.15.1. Условия ортогональности

Пусть $\varphi(x)$ – N-масштабирующая функция с масштабирующим соотношением

$$\varphi(x)=\sqrt{N}\sum_{n\in\mathbf{Z}} h_n\varphi(Nx-n)$$

и частотной функцией

$$H_0(\omega)=\frac{1}{\sqrt{N}}\sum_{n\in\mathbf{Z}} h_n e^{-i\,n\omega}.$$

Для ортонормированности системы функций $\varphi_{0,n}(x)=\varphi(x-n)$, $n\in\mathbf{Z}$, должно выполняться свойство

$$\sum_{n\in Z}\left|\hat{\varphi}(\omega+2\pi n)\right|^2=1 \text{ п. в.} \tag{3}$$

Более простое необходимое условие ортогональности выражается равенством

$$\sum_{s=0}^{N-1}\left|H_0(\omega+2\pi sm/N)\right|^2=1 \text{ п. в.} \tag{4}$$

Данное условие эквивалентно следующему:

$$\sum\nolimits_n h_n\overline{h}_{n-Np}=\delta_{0p}. \tag{5}$$

Имеет место следующее достаточное условие ортонормированности, которое доказывается так же, как и в разделе 2.3.5.

Теорема 1. *Предположим, что частотная функция является тригонометрическим полиномом вида* $H_0(\omega)=\sqrt{N^{-1}}\sum_{n=0}^{L} h_n e^{-in\omega}$, *удовлетворяющим условиям* $\sum_{s=0}^{N-1}\left|H_0(\omega+2\pi sm/N)\right|^2=1$ *и* $H_0(0)=1$, *а функция* $\varphi(x)$ *определена соотношением* $\hat{\varphi}(\omega)=\prod_{j=1}^{\infty} H_0(\omega/N^j)$. *Если собственное значение* 1 *матрицы* $A_{kl}=\sum_{n=0}^{L} h_n\overline{h}_{k-Nl+n}$ *порядка* $2N-1$ *не является кратным, то функции* $\varphi_{0,n}(x)=\varphi(x-n)$ *образуют ортонормированную систему.*

Для практического использования данного условия удобнее взять следующую матрицу B порядка $2L-1$:

$$B_{pq}=A_{p-L,q-L}=\sum_{n=0}^{L} h_n\overline{h}_{p-Nq+n+L(N-1)},\quad 1\le p,\ q\le 2L-1.$$

Матрицу B можно представить в виде произведения матриц, элементы которых состоят из коэффициентов масштабирующего фильтра:

$$B_{pq} = \sum_{n=0}^{L} h_n \overline{h}_{p-Nq+n+L(N-1)} = \left[\begin{array}{c} p+n=m \\ p \le m \le p+L \end{array} \right] = \sum_{m} h_{m-p} \overline{h}_{L(N-1)+m-Nq}. \qquad (7)$$

Пусть

$$F_{p,m} = h_{m-p}, \quad 1 \le p \le 2L-1, \ 1 \le m \le 3L-1,$$

и

$$\tilde{F}_{m,q} = \overline{h}_{L(N-1)+m-Nq}, \quad 1 \le m \le 3L-1, \ 1 \le q \le 2L-1.$$

Тогда $B = F \cdot \tilde{F}$. Матрица F имеет $2L-1$ строк и $3L-1$ столбцов и имеет следующий вид:

$$F = \begin{pmatrix} h_0 & h_1 & h_2 & \cdots & h_L & 0 & 0 & 0 & \cdots & 0 \\ 0 & h_0 & h_1 & \cdots & h_{L-1} & h_L & 0 & 0 & \cdots & 0 \\ 0 & 0 & h_0 & \cdots & h_{L-2} & h_{L-1} & h_L & 0 & \cdots & 0 \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \end{pmatrix}.$$

Матрица $\tilde{F}$ имеет $3L-1$ строк и $2L-1$ столбцов. Вид этой матрицы лучше показать на примере. Пусть $L = 3$ и $N = 3$. Тогда имеем следующий набор возможных ненулевых коэффициентов: h_0, h_1, h_2, h_3, h_4, h_5. Матрица $\tilde{F}$ имеет 8 строк и 5 столбцов, состоит из элементов $\tilde{F}_{m,q} = \overline{h}_{2L+m-3q} = \overline{h}_{6+m-3q}$ и имеет следующий вид:

$$\tilde{F} = \begin{pmatrix} \overline{h}_4 & \overline{h}_1 & 0 & 0 & 0 \\ \overline{h}_5 & \overline{h}_2 & 0 & 0 & 0 \\ 0 & \overline{h}_3 & \overline{h}_0 & 0 & 0 \\ 0 & \overline{h}_4 & \overline{h}_1 & 0 & 0 \\ 0 & \overline{h}_5 & \overline{h}_2 & 0 & 0 \\ 0 & 0 & \overline{h}_3 & \overline{h}_0 & 0 \\ 0 & 0 & \overline{h}_4 & \overline{h}_1 & 0 \\ 0 & 0 & \overline{h}_5 & \overline{h}_2 & 0 \end{pmatrix}.$$

По теореме 1, если характеристический многочлен $P(\lambda) = \det(B - \lambda E)$ имеет число $\lambda = 1$ корнем кратности 1, то система функций $\{\varphi(x-n)\}$ образует ортонормированную систему. Для проверки кратности корня вычисляем значение многочлена $P(\lambda)$ и его первой производной $P'(\lambda)$ при $\lambda = 1$. Если $P(1) = 0$ и $P'(1) \neq 0$, то корень $\lambda = 1$ не кратный.

2.15.2. Построение матрицы частотных функций

В данном разделе мы построим унитарную матрицу (1) частотных функций $H_k(z)$. Эта матрица имеет очень специальный вид. Однако ее можно разложить в произведение двух унитарных (на окружности $|z| = 1$) матриц

$$H(z) = \frac{1}{\sqrt{N}} A(z^N) R(z),$$

где $A(z^N)$ – унитарная матрица, элементами которой являются многочлены от z^N, а матрица $R(z)$ имеет вид

$$R(z) = \begin{pmatrix} 1 & 1 & \cdots & 1 \\ z & \rho z & \cdots & \rho^{N-1} z \\ \cdots & \cdots & \cdots & \cdots \\ z^{N-1} & \rho^{N-1} z^{N-1} & \cdots & \rho^{(N-1)^2} z^{N-1} \end{pmatrix}, \tag{8}$$

где $\rho = e^{-i2\pi/N}$. Матрица $A(w)$ называется полифазной. Каждая строка полифазной матрицы $A(z^N)$ – это просто выборка элементов со степенями z^{kN} (то есть кратными N) в многочленах

$$H_k(z), z^{-1}H_k(z), ..., z^{-N+1}H_k(z).$$

Формула для нахождения полифазных компонент может быть представлена в виде преобразования *Фурье* по циклической группе $\mathbf{Z}/N\mathbf{Z} = \{1, \rho, \rho^2, ..., \rho^{N-1}\}$ [BJ]:

$$A_{k,j}(w) = \frac{1}{N} \sum_{z^N = w} z^{-j} H_k(z). \tag{9}$$

Обратное преобразование определяется формулой

$$H_k(z) = \sum_{j=0}^{N-1} z^j A_{k,j}(z^N). \tag{10}$$

Последнее соотношение определяет требуемое разложение:

$$H(z) = A(z^N) \frac{1}{\sqrt{N}} \begin{pmatrix} 1 & 1 & \cdots & 1 \\ z & \rho z & \cdots & \rho^{N-1} z \\ \cdots & \cdots & \cdots & \cdots \\ z^{N-1} & \rho^{N-1} z^{N-1} & \cdots & \rho^{(N-1)^2} z^{N-1} \end{pmatrix} = \tag{11}$$

$$= \frac{1}{\sqrt{N}} A(z^N) R(z).$$

В этом выражении матрица $A(w)$ является уже произвольной унитарной, при $|w| = 1$, матрицей с полиномиальными элементами. Отметим, что матрица $R(z)/\sqrt{N}$ является унитарной на единичной окружности $z = e^{-i\omega}$, поскольку имеет место равенство $R(z)R^*(z) = N$.

Задавая унитарную полифазную матрицу $A(w)$, мы можем построить матрицу частотных функций $H(z)$ и вместе с ней частотные функции $H_0(z), H_1(z), \ldots, H_{N-1}(z)$ вейвлетов, следовательно, и сами вейвлеты $\varphi(x), \psi^1(x), \ldots, \psi^{N-1}(x)$.

В работе [BJ] дана схема построения полифазной матрицы $A(z^N)$ в предположении, что полиномиальная частотная функция $H_0(z)$ является заданной. Тогда в соответствии с формулой (9) можно считать, что первая строка матрицы $A(z^N)$ известна, и нужно построить остальные строки матрицы. Ниже мы покажем, что построения можно провести и без задания $H_0(z)$.

Схема построения полифазной матрицы. В данном разделе мы приведем простую схему построения большой серии унитарных (при $|w| = 1$) полифазных матриц $A(w)$, элементы которых являются многочленами с вещественными коэффициентами. Это позволяет определить как масштабирующую функцию $\varphi(x)$ с компактным носителем и с коэффициентом масштабирования $N \geq 2$, так и соответствующие ортогональные вейвлеты $\psi^1(x), \ldots, \psi^{N-1}(x)$.

Выберем произвольную ортогональную матрицу A_0 порядка $N > 1$, $A_0 = \{a_{kj}, k, j = 0, 1, \ldots, N-1\}$. Умножим ее на диагональную унитарную матрицу вида $D_p(w) = diag(w^{p_0}, w^{p_1}, \ldots, w^{p_{N-1}})$, где $p = (p_0, p_1, \ldots, p_{N-1})$ – набор целых чисел и $|w| = 1$, а затем – на ортогональную матрицу $B_0 = \{b_{kj}, k, j = 0, 1, \ldots, N-1\}$. В результате получим унитарную матрицу

$$A(w) = A_0 D_p(w) B_0, \tag{12}$$

элементы которой, $A_{kj}(w) = \sum_{s=0}^{N-1} a_{ks} b_{sj} w^{p_s}$, являются многочленами по переменной w с вещественными коэффициентами.

Теперь подставим $w = z^N$, где $z = e^{-i\omega}$. Получаем унитарную (на окружности $|w| = 1$) матрицу $A(z^N)$ с полиномиальными элементами с вещественными коэффициентами. Умножим ее на унитарную матрицу $R(z)/\sqrt{N}$. Тогда, согласно формуле (11), мы получим унитарную (на окружности $|z| = 1$) матрицу $H(z)$ частотных полиномиальных функций $H_0(z), H_1(z), \ldots, H_{N-1}(z)$. Таким образом,

$$H(z) = \frac{1}{\sqrt{N}} A_0 \cdot D_p(z^N) \cdot B_0 \cdot R(z). \tag{13}$$

$$H_k(z)=\frac{1}{\sqrt{N}}\sum_{s,j=0}^{N-1}a_{ks}b_{sj}z^j z^{Np_s},\quad k=0,1,\ldots,N-1. \tag{14}$$

Для того чтобы полученные функции были частотными функциями ортогональных вейвлетов, необходимо, чтобы сумма коэффициентов для $H_0(z)$ была равна единице, а суммы коэффициентов для остальных функций $H_1(z)$, ..., $H_{N-1}(z)$ были равны нулю.

$$\frac{1}{\sqrt{N}}\sum_{s,j=0}^{N-1}a_{0s}b_{sj}=\frac{1}{\sqrt{N}}\sum_{s=0}^{N-1}a_{0s}\left(\sum_{j=0}^{N-1}b_{sj}\right)=1.$$

$$\frac{1}{\sqrt{N}}\sum_{s,j=0}^{N-1}a_{ks}b_{sj}=\frac{1}{\sqrt{N}}\sum_{s=0}^{N-1}a_{ks}\left(\sum_{j=0}^{N-1}b_{sj}\right)=0,\ k=1,2,\ldots,N-1.$$

Эти равенства можно записать в матричном виде:

$$\begin{pmatrix} a_{00} & a_{01} & \cdots & a_{0,N-1} \\ a_{10} & a_{11} & \cdots & a_{1,N-1} \\ \cdots & \cdots & \cdots & \cdots \\ a_{N-1,0} & a_{N-1,1} & \cdots & a_{N-1,N-1} \end{pmatrix}\begin{pmatrix} b_{00}+b_{01}+\cdots+b_{0,N-1} \\ b_{10}+b_{11}+\cdots+b_{1,N-1} \\ \cdots \\ b_{N-1,0}+b_{N-1,1}+\cdots+b_{N-1,N-1} \end{pmatrix}=\begin{pmatrix} \sqrt{N} \\ 0 \\ \cdots \\ 0 \end{pmatrix}. \tag{15}$$

Выбирая различные ортогональные матрицы A_0 и B_0, удовлетворяющие указанному равенству (15), получаем различные частотные функции (14) и по ним определяем вейвлеты.

Для получения достаточно простого класса ортогональных вейвлетов с компактным носителем и параметром масштабирования $N > 1$, возьмем в качестве ортогональной матрицы A_0 следующую матрицу:

$$A_0=\begin{pmatrix} 1/\sqrt{N} & 1/\sqrt{N} & 1/\sqrt{N} & 1/\sqrt{N} & \cdots & 1/\sqrt{N} \\ 1/\sqrt{2} & -1/\sqrt{2} & 0 & 0 & \cdots & 0 \\ 1/\sqrt{6} & 1/\sqrt{6} & -2/\sqrt{6} & 0 & \cdots & 0 \\ 1/\sqrt{12} & 1/\sqrt{12} & 1/\sqrt{12} & -3/\sqrt{12} & \cdots & 0 \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ 1/\sqrt{N(N-1)} & 1/\sqrt{N(N-1} & 1/\sqrt{N(N-1} & 1/\sqrt{N(N-1} & \cdots & -(N-1)/\sqrt{N(N-1} \end{pmatrix}.$$

Она интересна тем, что преобразует вектор единиц $e = (1, 1, ..., 1)$ в вектор $\sqrt{N}e_0 = (\sqrt{N}, 0, ..., 0)$, $A_0e = \sqrt{N}e_0$. Поэтому для выбранной матрицы A_0 из равенства (15) следует, что элементы ортогональной матрицы B_0 должны удовлетворять следующей системе уравнений:

$$\begin{cases} b_{00} + b_{01} + \cdots + b_{0,N-1} = 1 \\ b_{10} + b_{11} + \cdots + b_{1,N-1} = 1 \\ \cdots \quad \cdots \\ b_{N-1,0} + b_{N-1,1} + \cdots + b_{N-1,N-1} = 1 \end{cases} \tag{15}$$

Решением этой системы будет любой набор ортонормированных векторов (строк), координаты которых удовлетворяют уравнению плоскости $x_0 + x_1 + ... + x_{N-1} = 1$ в $\mathbf{R}^N$.

Очевидно, что координаты базисных векторов $e_0 = (1, 0, ..., 0)$, $e_1 = (0, 1, 0, ..., 0)$, ..., $e_{N-1} = (0, ..., 0, 1)$ удовлетворяют этому уравнению. Данное решение соответствует единичной матрице B_0. Любое другое решение получается ортогональным преобразованием (вращением) базисного решения $e_0, e_1, ..., e_{N-1}$ вокруг вектора $e = e_0 + e_1 + ... + e_{N-1}$, то есть в плоскости $x_0 + x_1 + ... + x_{N-1} = 1$. Найдем эти решения. Возьмем вращение вокруг оси e_0:

$$M = \begin{pmatrix} 1 & 0 & 0 & \cdots & 0 \\ 0 & m_1^1 & m_2^1 & \cdots & m_{N-1}^1 \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ 0 & m_1^{N-1} & m_2^{N-1} & \cdots & m_{N-1}^{N-1} \end{pmatrix}. \tag{16}$$

Поскольку $A_0 e = \sqrt{N} e_0$, то вращение вокруг оси e задается матрицей $M_e = A_0^{-1} M A_0$.

Матрица вращения M_e действует умножением на вектор-столбец координат, следовательно, строки матрицы B_0 состоят из координат векторов-столбцов, которые получены из базисных векторов-столбцов $e_0, e_1, ..., e_{N-1}$ действием на них матрицей вращения M_e. Поэтому матрица B_0 является транспонированной к M_e, $B_0 = A_0^{-1} M^T A_0$. Действительно, $M_e e_k$ – это вектор, полученный вращением базисного вектора e_k вокруг оси e. С другой стороны, $M_e e_k$ – это k-й столбец матрицы M_e. Тогда

$$H_M(z) = \frac{1}{\sqrt{N}} A_0 D_p(z^N) M_e^T R(z) = \frac{1}{\sqrt{N}} A_0 D_p(z^N) A_0^{-1} M^T A_0 R(z), \tag{17}$$

где M – произвольная ортогональная матрица вида (16) и $D_p(w)$ – диагональная матрица вида $D_p(w) = diag(w^{p_0}, w^{p_1}, \ldots, w^{p_{N-1}})$.

Формула (17) дает прямой способ построения частотных функций $H_0(z)$, $H_1(z)$, ..., $H_{N-1}(z)$ ортогональных вейвлетов с компактным носителем. Ортогональную матрицу M вида (16) и вектор степеней $p = (p_0, p_1, ..., p_{N-1})$ можно задавать произвольно. При построении час-

тотной функции по формуле (2) нужно учитывать достаточное условие ортогональности, установленное теоремой 1.

2.15.3. Построение матрицы частотных функций в случае $N = 3$

В данном разделе мы покажем на примере параметра масштабирования $N = 3$ эффективность изложенной выше схемы построения матрицы частотных функций в виде (17). Возьмем для примера диагональную матрицу вида $D(w) = \mathrm{diag}\{1, w, 1\}$, $|w| = 1$. В случае $N = 3$ матрица A_0 имеет вид

$$A_0 = \begin{pmatrix} 1/\sqrt{3} & 1/\sqrt{3} & 1/\sqrt{3} \\ 1/\sqrt{2} & -1/\sqrt{2} & 0 \\ 1/\sqrt{6} & 1/\sqrt{6} & -2/\sqrt{6} \end{pmatrix}. \qquad (18)$$

Тогда элементы второй ортогональной матрицы B_0 из (12) должны удовлетворять условиям:

$$b_{00} + b_{01} + b_{02} = 1, \quad b_{10} + b_{11} + b_{12} = 1, \quad b_{20} + b_{21} + b_{22} = 1.$$

Решением будет любой набор ортогональных векторов, координаты которых удовлетворяют уравнению плоскости $x_0 + x_1 + x_2 = 1$. Очевидно, что базисные векторы e_0, e_1, e_2 представляют решение, которому соответствует единичная матрица B_0. Любое другое решение может быть получено вращением базисных векторов e_0, e_1, e_2 в плоскости $x_0 + x_1 + x_2 = 1$. Возьмем вращение вокруг оси Ox:

$$M(t) = \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos t & \sin t \\ 0 & -\sin t & \cos t \end{pmatrix}. \qquad (21)$$

Тогда вращение вокруг оси $e = e_0 + e_1 + e_2$ в плоскости $x_0 + x_1 + x_2 = 1$ задается матрицей:

$$M_e(t) = A_0^{-1} M(t) A_0 =$$

$$= \frac{1}{3}\begin{pmatrix} 1+2\cos t & 1+\sqrt{3}\sin t-\cos t & 1-\sqrt{3}\sin t-\cos t \\ 1-\sqrt{3}\sin t-\cos t & 1+2\cos t & 1+\sqrt{3}\sin t-\cos t \\ 1+\sqrt{3}\sin t-\cos t & 1-\sqrt{3}\sin t-\cos t & 1+2\cos t \end{pmatrix}. \qquad (22)$$

Строки матрицы B_0 состоят из координат векторов, которые получены из базисных векторов-столбцов e_0, e_1, e_2 действием на них матри-

цы $M_e(t)$. Поэтому матрица B_0 является транспонированной к $M_e(t)$. Получаем:

$$H(t,z) = A_0 D(z^N) M_e^T(t) R(z)/\sqrt{3}, \tag{23}$$

где $D(z^N) = \mathrm{diag}\{1, z^N, 1\}$ и матрица $R(z)$ имеет вид:

$$R(z) = \begin{pmatrix} 1 & 1 & 1 \\ z & \rho z & \rho^2 z \\ z^2 & \rho^2 z^2 & \rho^4 z^2 \end{pmatrix}. \tag{24}$$

Перемножая все эти матрицы, получаем:

$$H_0(t,z) = \frac{1}{9}(2 + \cos t - \sqrt{3}\sin t + (2 - 2\cos t)z +$$
$$+ (2 + \cos t + \sqrt{3}\sin t)z^2 + (1 - \cos t + \sqrt{3}\sin t)z^3 + \tag{25}$$
$$+ (1 + 2\cos t)z^4 + (1 - \cos t - \sqrt{3}\sin t)z^5,$$

$$H_1(t,z) = \frac{1}{3\sqrt{6}}((1 + 2\cos t) + (1 - \cos t - \sqrt{3}\sin t)z +$$
$$+ (1 - \cos t + \sqrt{3}\sin t)z^2 - (1 - \cos t + \sqrt{3}\sin t)z^3 - \tag{26}$$
$$- (1 + 2\cos t)z^4 + (-1 + \cos t + \sqrt{3}\sin t)z^5),$$

$$H_2(t,z) = \frac{1}{9\sqrt{2}}((-1 + 4\cos t + 2\sqrt{3}\sin t) - (1 - \cos t + 3\sqrt{3}\sin t)z -$$
$$- (1 + 5\cos t - \sqrt{3}\sin t)z^2 + (1 - \cos t + \sqrt{3}\sin t)z^3 + (1 + 2\cos t)z^4 + \tag{27}$$
$$+ (1 - \cos t - \sqrt{3}\sin t)z^5).$$

Полученные частотные функции $H_0(z)$, $H_1(z)$, $H_2(z)$ позволяют построить вейвлеты $\varphi(x)$, $\psi^1(x)$ и $\psi^2(x)$ способами, изложенными в разделе 2.2.2. Максимальная степень частотной функции $H_0(z)$ равна пяти, поэтому длина носителя L равна 2.5, поскольку она находится из формулы $L(N-1) = \deg(H_0(z))$.

Замечание. Выбирая в формуле (23) другую матрицу $D(z^N)$, мы можем совершенно аналогично построить другие ортогональные вейвлеты с другой длиной носителя.

2.15.4. Примеры масштабирующих функций и вейвлетов для $N = 3$

Вычислим значения коэффициентов полученных частотных функций (25), (26), (27) для ряда значений параметра t и найдем соот-

ветствующие фильтры и вейвлеты $\varphi(x)$, $\psi^1(x)$ и $\psi^2(x)$ итерационным методом. Для примеров рассмотрим следующие значения параметра t: $0, \pi/3$ и $\pi/2$.

Если коэффициенты $\{h_n\}$ фильтра масштабирующей функции $\varphi(x)$ найдены, можно построить функцию $\varphi(x)$ несколькими методами (см. раздел 2.2.2). Будем использовать итерационный метод. Напомним, что масштабирующее уравнение $\varphi(x) = \sqrt{N}\sum_{n\in\mathbf{Z}} h_n \varphi(Nx-n)$ записывается в виде операторного уравнения

$$\varphi(x) = \sum_{n\in\mathbf{Z}} h_n A_N \circ T_n(\varphi(x)) = S(\varphi(x)),$$

где $A_N(f)(x) = \sqrt{N} f(Nx)$ и $T_n(f)(x) = f(x-n)$ – два унитарных оператора и $S = \sum_{n\in\mathbf{Z}} h_n A_N \circ T_n$ – линейная комбинация операторов $A_N \circ T_n$. Масштабирующее уравнение, записанное в виде $\varphi(x) = S(\varphi(x))$, показывает, что масштабирующая функция $\varphi(x)$ является неподвижной точкой оператора S. Поэтому $\varphi(x)$ может быть определена последовательными приближениями

$$\varphi_{n+1}(x) = S(\varphi_n(x)),\ n \to \infty,$$

где в качестве начального приближения можно взять, например, функцию $\varphi_0(x)$ Хаара.

Функции $\psi^1(x)$ и $\psi^2(x)$ определяются формулами $\psi^1(x) = \sqrt{3}\sum_n g_n^1 \varphi(3x-n)$ и $\psi^2(x) = \sqrt{3}\sum_n g_n^2 \varphi(3x-n)$ по известным фильтрам $\{g_n^1\}$ и $\{g_n^2\}$ и функции $\varphi(x)$.

Матричные операции для вычисления фильтров (25), (26), (27) удобно реализовать в системе Maple. Итерации для нахождения $\varphi(x)$ проведем на MATLAB. В рассмотренных ниже примерах для нахождения функции $\varphi(x)$ делаем пять итераций. Тогда если L – длина носителя, то

$$\varphi_5(x) = (\sqrt{3})^5 \sum_{p,q,r,m,n=0}^{L-1} h_p h_q h_r h_m h_n \varphi_0(3^5 x - 3^4 p - 3^3 q - 3^2 r - 3m - n).$$

Аннотированная программа на Maple вычисления частотных функций и фильтров ортогональных вейвлетов для $N = 3$:

```
> restart:
> with(LinearAlgebra):with(linalg):with(PolynomialTools):
Задаем матрицу D(w).Сразу учитываем, что w=z^N.
> DzN:=matrix(3,3,[[1,0,0],[0,z^3,0],[0,0,1]]):
Матрица R(z)/sqrt(3).
```

```
> R:=(1/sqrt(3))*matrix(3,3,[[1,1,1],[z,rho*z,rho^2*z],
[z^2,rho^2*z^2,rho^4*z^2]]):
Задаем матрицу A0. Вычисляем обратную матрицу A0^(-1).
> A0:=matrix(3,3,[[1/sqrt(3),1/sqrt(3),1/sqrt(3)],[1/sqrt(2),
-1/sqrt(2),0],[1/sqrt(6),1/sqrt(6),-2/sqrt(6)]]);
> A0m1:=inverse(A0):
Выбор матрицы вращения вокруг оси Ox.
> Mt:= matrix(3,3,[[1,0,0],[0,cos(t),sin(t)],[0,-sin(t),cos(t)]]);
Матрица вращения в плоскости x0+x1+x2=1.
> Met:=simplify(multiply(A0m1,multiply(Mt,A0)));
Вычисление матрицы B(t).
> Bt:=transpose(Met);
Вычисление матрицы H(t,z).
> H:=simplify(multiply(multiply(A0,multiply(DzN,Bt)),R)):
Выбираем элементы первого столбца - это частотные функции вейвлетов.
> H0:=H[1,1]:
> H1:=H[2,1]:
> H2:=H[3,1]:
Выбираем коэффициенты в порядке возрастания степеней.
> Vt0:=CoefficientList(H0,z);
> Vt1:=CoefficientList(H1,z);
> Vt2:=CoefficientList(H2,z);
Вычисление фильтров h0, g1, g2 для различных значений t.
Учитываем коэффициент 1/sqrt(3) перед Hk.
Вычисление фильтров для значения t=pi/6.
> h0:=sqrt(3)*subs(t=pi/6,Vt0):
> g1:=sqrt(3)*subs(t=pi/6,Vt1):
> g2:=sqrt(3)*subs(t=pi/6,Vt2):
Числовые значения фильтров.
> h0r:=simplify(subs({cos(pi/6)=sqrt(3)/2, sin(pi/6)=1/2},h0));
> g1r:=simplify(subs({cos(pi/6)=sqrt(3)/2, sin(pi/6)=1/2},g1));
> g2r:=simplify(subs({cos(pi/6)=sqrt(3)/2, sin(pi/6)=1/2},g2));
```

Программа на MATLAB вычисления пятой итерации функции $\varphi(x)$. В этой программе в качестве начального приближения используется функция Хаара Ha02(x), которая на промежутке [0, 2] принимает постоянное значение $1/\sqrt{2}$, а вне промежутка [0, 2] – значение 0.

```
function y=Phi(x);
L=length(x);
N=3;
sq=(sqrt(N))^5;
h=[0.1924500898, 0.1924500898,0.7698003590, 0.3849001795, 0.3849001795,
  -0.1924500898];
S=length(h);
```

```
Phi0(1:L)=0;
for k=1:L;
    for p=1:S;
        for q=1:S;
            for r=1:S;
                for m=1:S;
                    for n=1:S;
Phi0(k)=Phi0(k)+h(p)*h(q)*h(r)*h(m)*h(n)*Ha02((N^5)*x(k)- ...
-(N^4)*(p-1)-(N^3)*(q-1)-(N^2)*(r-1)-N*(m-1)-(n-1));
                    end
                end
            end
        end
    end
y(k)=sq*Phi0(k);
end
```

Листинг функции `Ha02(x)`:

```
function y=Ha02(x);
for k=1:length(x);
if x(k) < 0; y(k)=0;
elseif x(k) < 2; y(k)=1/sqrt(2);
else y(k)=0;
end
end
```

Фильтр h_n имеет шесть ненулевых элементов. Поэтому длина носителя функции $\varphi(x)$ равна 2,5. Носитель лежит на промежутке $[0, 3]$, $L = 3$.

Для проверки ортогональности системы функций $\varphi(x - n)$ используем матрицу B, которая строится по элементам фильтра $\{h_0, h_1, h_2, h_3, h_4, h_5\}$, см. формулу (7). Она может быть представлена в виде произведения $B = F \cdot \tilde{F}$, где матрица F содержит $2L - 1 = 5$ строк и $3L - 1 = 8$ столбцов и имеет следующий вид:

$$F = \begin{pmatrix} h_0 & h_1 & h_2 & h_3 & h_4 & h_5 & 0 & 0 \\ 0 & h_0 & h_1 & h_2 & h_3 & h_4 & h_5 & 0 \\ 0 & 0 & h_0 & h_1 & h_2 & h_3 & h_4 & h_5 \\ 0 & 0 & 0 & h_0 & h_1 & h_2 & h_3 & h_4 \\ 0 & 0 & 0 & 0 & h_0 & h_1 & h_2 & h_3 \end{pmatrix}.$$

Матрица $\tilde{F}$ содержит 8 строк и 5 столбцов, состоит из элементов $\tilde{F}_{m,q} = \overline{h}_{2L+m-3q} = \overline{h}_{6+m-3q}$ и имеет следующий вид:

$$\tilde{F} = \begin{pmatrix} \overline{h}_4 & \overline{h}_1 & 0 & 0 & 0 \\ \overline{h}_5 & \overline{h}_2 & 0 & 0 & 0 \\ 0 & \overline{h}_3 & \overline{h}_0 & 0 & 0 \\ 0 & \overline{h}_4 & \overline{h}_1 & 0 & 0 \\ 0 & \overline{h}_5 & \overline{h}_2 & 0 & 0 \\ 0 & 0 & \overline{h}_3 & \overline{h}_0 & 0 \\ 0 & 0 & \overline{h}_4 & \overline{h}_1 & 0 \\ 0 & 0 & \overline{h}_5 & \overline{h}_2 & 0 \end{pmatrix}.$$

По теореме 1, если характеристический многочлен $P(\lambda) = \det(B - \lambda E)$ имеет число $\lambda = 1$ корнем кратности 1, то система функций $\{\varphi(x - n)\}$ образует ортонормированную систему. Для проверки кратности корня вычисляем значение многочлена $P(\lambda)$ и его первой производной $P'(\lambda)$ при $\lambda = 1$. Если $P(1) = 0$ и $P'(1) \neq 0$, то корень $\lambda = 1$ не кратный.

1. Значение параметра $t = 0$. В этом случае частотные функции имеют вид

$$H_0(z) = \frac{1}{3}(1 + z^2 + z^4), \quad H_1(z) = \frac{1}{\sqrt{6}}(1 - z^4), \quad H_2(z) = \frac{1}{3\sqrt{2}}(1 - 2z^2 + z^4).$$

Максимальная степень частотной функции $H_0(z)$ равна четырем, поэтому длина носителя L равна 2, поскольку она находится из формулы $L(N - 1) = \deg(H_0(z))$. Получаем масштабирующее соотношение:

$$\varphi(x) = \varphi(3x) + \varphi(3x - 2) + \varphi(3x - 4).$$

Легко видеть, что масштабирующая функция $\varphi(x)$ является характеристической функцией промежутка $[0, 2)$:

$$\varphi(x) = \begin{cases} 1, & x \in [0,2) \\ 0, & \text{для других } x. \end{cases}$$

Очевидно, что она не определяет ортонормированную систему функций $\{\varphi(x - n), n \in \mathbf{Z}\}$. Ситуацию можно исправить, если вместо $\varphi(x)$ взять функцию $\varphi(2x)$ или если сделать замену $w = z^2$. Тогда

$$H_0(w) = \frac{1}{3}(1 + w + w^2), \quad H_1(w) = \frac{1}{\sqrt{6}}(1 - w^2), \quad H_2(w) = \frac{1}{3\sqrt{2}}(1 - 2w + w^2).$$

В этом случае масштабирующая функция $\varphi(x)$ будет характеристической функцией промежутка $[0, 1)$. Получаются вейвлеты

Хаара. Масштабирующее соотношение и вейвлеты (см. рис. 2.15.1) имеют вид:

$$\varphi(x) = \varphi(3x) + \varphi(3x-1) + \varphi(3x-2),$$

$$\psi^1(x) = \frac{\sqrt{3}}{\sqrt{2}}\varphi(3x) - \frac{\sqrt{3}}{\sqrt{2}}\varphi(3x-2),$$

$$\psi^2(x) = \frac{1}{\sqrt{2}}(\varphi(3x) - 2\varphi(3x-1) + \varphi(3x-2)).$$

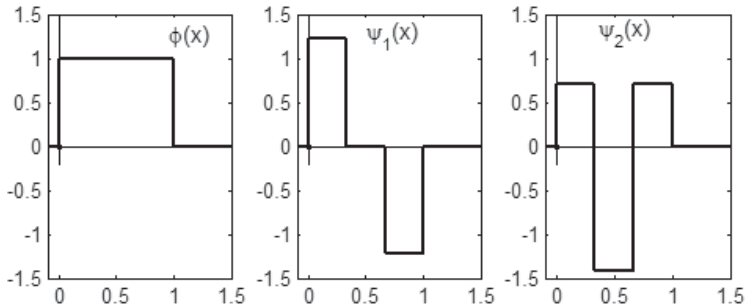


Рис. 2.15.1. Графики функций $\varphi(x)$ и $\psi^1(x)$, $\psi^2(x)$ для $t = 0$

2. Значение параметра $\pi/3$. Фильтры масштабирующей функции $\varphi(x)$ и вейвлетов $\psi^1(x)$ и $\psi^2(x)$ имеют вид:

$$h_0 = \frac{\sqrt{3}}{9}(1, 1, 4, 2, 2, -1),$$

$$g_0 = \frac{\sqrt{2}}{6}(2, -1, 2, -2, -2, 1),$$

$$g_2 = \frac{\sqrt{6}}{18}(4, -5, -2, 2, 2, -1).$$

Для проверки ортогональности системы функций $\varphi(x - n)$ используем характеристический многочлен $P(\lambda) = \det(B - \lambda E)$. В нашем случае $P(1) = 0$ и $P'(1) \neq 0$, поэтому корень $\lambda = 1$ не кратный и система $\{\varphi(x - n)\}$ является ортонормированной.

Масштабирующее соотношение:

$$\varphi(x) = {}^1\!/_3(\varphi(3x) + \varphi(3x-1) + 4\varphi(3x-2) + 2\varphi(3x-3) + 2\varphi(3x-4) - \varphi(3x-5)).$$

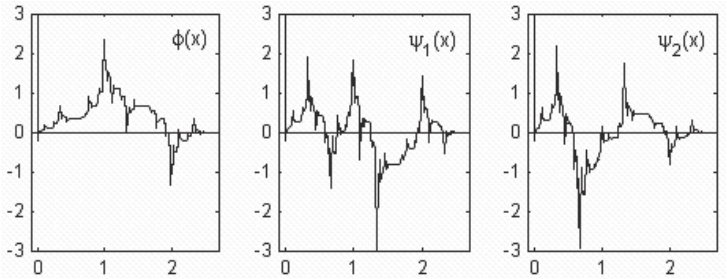


Рис. 2.15.2. Графики функций $\varphi(x)$ и $\psi^1(x)$, $\psi^2(x)$ для $t = \pi/3$

3. Значение параметра $\pi/2$. Фильтры масштабирующей функции $\varphi(x)$ и вейвлетов $\psi^1(x)$ и $\psi^2(x)$ имеют вид:

$$h_0 = \frac{\sqrt{3}}{9}(2-\sqrt{3}, 2, 2+\sqrt{3}, 1+\sqrt{3}, 1, 1-\sqrt{3}),$$

$$g_1 = \frac{\sqrt{6}}{18}(\sqrt{3}, -3+\sqrt{3}, 3+\sqrt{3}, -3-\sqrt{3}, -\sqrt{3}, 3-\sqrt{3}),$$

$$g_2 = \frac{\sqrt{6}}{18}(-1+2\sqrt{3}, -1-3\sqrt{3}, -1+\sqrt{3}, 1+\sqrt{3}, 1, 1-\sqrt{3}).$$

Для проверки ортогональности системы функций $\varphi(x - n)$ используем характеристический многочлен $P(\lambda) = \det(B - \lambda E)$. В нашем случае $P(1) = 0$ и $P'(1) \neq 0$, поэтому корень $\lambda = 1$ не кратный и система $\{\varphi(x - n)\}$ является ортонормированной.

Масштабирующее соотношение:

$$\varphi(x) = {}^1\!/_3((2-\sqrt{3})\varphi(3x) + 2\varphi(3x-1) + (2+\sqrt{3})\varphi(3x-2) + \\ + (1+\sqrt{3})\varphi(3x-3) + \varphi(3x-4) + (1-\sqrt{3})\varphi(3x-5)).$$

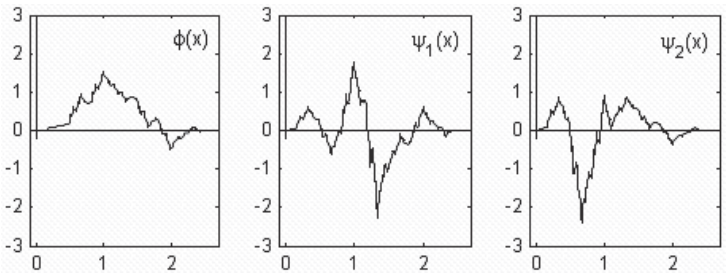


Рис. 2.15.3. Графики функций $\varphi(x)$ и $\psi^1(x)$, $\psi^2(x)$ для $t = \pi/2$

Используя приведенные выше листинги программ, можно вычислить фильтры вейвлетов и найти графики масштабирующей функции $\varphi(x)$ и вейвлетов $\psi^1(x)$ и $\psi^2(x)$ при любых других значениях параметра t.

Упражнение. Построить вейвлеты с компактным носителем для коэффициента масштабирования $N = 2$ по формуле (17), выбирая различные диагональные матрицы $D_k(w)$.

2.16. Многомерные вейвлеты с матричным коэффициентом масштабирования

В данном параграфе мы снова обращаемся к вейвлет-анализу на многомерных пространствах. В разделе 2.11 мы рассмотрели двумерный случай, когда вейвлет-анализ строился на основе одномерного и с одним и тем же коэффициентом масштабирования. Здесь мы рассмотрим более общий случай матричного коэффициента масштабирования. Будут установлены основные формулы теории вейвлетов: условия ортогональности, формулы быстрого вейвлет-преобразования, условия на частотные функции вейвлетов. Изложение подходов и возникающих проблем по многомерным обобщениям дискретного вейвлет-преобразования можно найти в книге [Дб] (глава 10), в статьях [BJ99], [J03], [LWW13] и приведенной там литературе.

Сигналом в многомерном пространстве мы будем называть функцию $f(x)$, зависящую от нескольких переменных $x = (x_1, ..., x_p)$, $p > 1$. Соответственно, дискретный сигнал зависит от нескольких дискретных переменных $n = (n_1, ..., n_p)$, $p > 1$, $n_i \in \mathbf{Z}$, и его естественно обозначить как $a_{n_1,...,n_p}$, или короче: a_n, здесь n – это многомерный индекс. Таким образом, *сигналом в многомерном пространстве*, или просто *многомерным сигналом*, мы будем называть массив действительных, или комплексных, чисел $\{a_n\}$, где индекс n меняется во множестве $\mathbf{Z}^p$ всех наборов из p целых чисел, $n = (n_1, ..., n_p)$, $p \geq 1$. Если $n \in \mathbf{Z}^p$ и $z = (z_1, ..., z_p) \in \mathbf{C}^p$, то символом z^n будем обозначать моном вида $z^n = z_1^{n_1} \cdots z_p^{n_p}$. С каждым многомерным сигналом $\{a_n\}$ ассоциируется следующий степенной ряд (z-преобразование):

$$X(z) = \sum_{n \in \mathbf{Z}^p} a_n z^n = \sum_{n \in \mathbf{Z}^p} a_{n_1,...,n_p} z_1^{n_1} \cdots z_p^{n_p}.$$

Нам потребуются также некоторые свойства многомерного преобразования Фурье. Пусть $f(x) \in L^1(\mathbf{R}^p) \cap L^2(\mathbf{R}^p)$. *Преобразование Фурье* функции $f(x)$ определяется формулой

$$\hat{f}(\omega) = \int_{\mathbf{R}^p} f(x) e^{-i(x,\omega)} d^p x,$$

где $\omega = (\omega_1, \omega_2, ..., \omega_p)$, $x = (x_1, x_2, ..., x_p)$, $(x, \omega) = x_1\omega_1 + x_2\omega_2 + ... + x_p\omega_p$ и $d^p x = dx_1 dx_2 ... dx_p$.

Преобразование Фурье в $\mathbf{R}^p$ обладает аналогичными свойствами, что и в одномерном случае. Напомним некоторые из них (см. раздел 2.12.2).

Сдвиг пространственной переменной:

$$F[f(x-b)](\omega) = e^{-i(b,\omega)} \hat{f}(\omega). \tag{1}$$

Действие линейного оператора на x. Пусть $y = Ax$ – линейный невырожденный оператор в пространстве $\mathbf{R}^p$. Тогда

$$F[f(Ax)](\omega) = \frac{1}{|\det(A)|} \hat{f}((A^t)^{-1}\omega), \tag{2}$$

где $(A^t)^{-1}$ – обратная транспонированная матрица для A.

2.16.1. Масштабирующие функции

Масштабирующее соотношение для случая многомерного пространства $\mathbf{R}^p$ определяется совершенно аналогично классическому одномерному случаю. Вместо коэффициента масштабирования берется невырожденная целочисленная $p \times p$ матрица A, модули собственных чисел которой больше единицы. Степень растяжения характеризуется модулем определителя $N = |\det A|$ матрицы A, поскольку $|\det A|$ – это объем параллелепипеда, который получается из стандартного единичного параллелепипеда $\{e_1, ..., e_p\}$ действием оператора A. Определим p-мерный тор как $T^p = \mathbf{R}^p / 2\pi\mathbf{Z}^p$. Он является прямым произведением $T^p = S^1 \times ... \times S^1$ единичных окружностей $S^1 \subset \mathbf{C}$.

Определение 1. *Функция* $\varphi(x) \in L^2(\mathbf{R}^p)$ *называется A-масштабирующей, если она может быть представлена в виде следующего ряда*:

$$\varphi(x) = \sqrt{|\det A|} \sum_{n \in \mathbf{Z}^p} h_n \varphi(Ax - n), \tag{3}$$

где числа $\{h_n\}$, $n \in \mathbf{Z}^p$, *удовлетворяют условию* $\sum_{n \in \mathbf{Z}^p} |h_n|^2 < \infty$. *Равенство* (3) *называется масштабирующим уравнением. Набор коэффициентов* $\{h_n\}$ *называется масштабирующим фильтром.*

Пусть $\varphi(x)$ – масштабирующая функция. Образуем следующие функции:

$$\varphi_{j,n}(x) = \sqrt{|\det A|^j}\,\varphi(A^j x - n), \quad j \in \mathbf{Z}, n \in \mathbf{Z}^p. \tag{4}$$

Теорема 1. *Для любых $j \in \mathbf{Z}$, $k \in \mathbf{Z}^p$ имеет место следующее разложение:*

$$\varphi_{j-1,k}(x) = \sum_{n \in \mathbf{Z}^p} h_{n-Ak}\varphi_{j,n}(x) = \sum_{n \in \mathbf{Z}^p} h_n \varphi_{j,n+Ak}(x). \tag{5}$$

В частности,

$$\varphi_{0,k}(x) = \sqrt{|\det A|}\sum_{n \in \mathbf{Z}^p} h_{n-Ak}\varphi(Ax - n) \quad \text{и} \quad \varphi_{j-1,0}(x) = \sum_{n \in \mathbf{Z}^p} h_n \varphi_{j,n}(x).$$

Доказательство. Полностью аналогично одномерному случаю, см. раздел 2.2.1.

Сделаем преобразование Фурье масштабирующего соотношения (3). Поскольку $\varphi(Ax - n) = \varphi(A(x - A^{-1}n))$, то

$$F[\varphi(Ax - n)] = \frac{1}{|\det A|} e^{-i(n,(A^t)^{-1}\omega)} F[\varphi]((A^t)^{-1}\omega).$$

Поэтому

$$\hat{\varphi}(\omega) = \sqrt{|\det A|}\sum_{n \in \mathbf{Z}^d} h_n F[\varphi(Ax - n)] = \frac{\sqrt{|\det A|}}{|\det(A)|}\sum_{n \in \mathbf{Z}^d} h_n e^{-i(n,(A^t)^{-1}\omega)}\hat{\varphi}((A^t)^{-1}\omega) =$$

$$= \frac{1}{\sqrt{|\det A|}}\sum_{n \in \mathbf{Z}^d} h_n e^{-i(n,(A^t)^{-1}\omega)}\hat{\varphi}((A^t)^{-1}\omega) = H_0((A^t)^{-1}\omega)\hat{\varphi}((A^t)^{-1}\omega).$$

Окончательно получаем:

$$\hat{\varphi}(\omega) = H_0((A^t)^{-1}\omega)\hat{\varphi}((A^t)^{-1}\omega), \tag{6}$$

где

$$H_0(\omega) = \frac{1}{\sqrt{|\det A|}}\sum_{n \in \mathbf{Z}^p} h_n e^{-i(n,\omega)} \tag{7}$$

– частотная функция масштабирующей функции $\varphi(x)$. Отметим, что она является 2π-периодической по всем переменным ω_i и поэтому может считаться определенной на торе $T^p = \mathbf{R}^p/2\pi\mathbf{Z}^p$.

Если масштабирующая функция $\varphi(x)$ имеет носитель в шаре радиуса L, то масштабирующий фильтр содержит конечное число ненулевых коэффициентов $\{h_n\}$. Это число K ненулевых коэффициентов может быть оценено сверху: $K \le (2LN - 1)^p$.

О масштабирующей функции Хаара

Как известно, в одномерном случае наиболее простой масштабирующей функцией является функция Хаара $\varphi(x) = \chi_{[01)}(x)$ – характеристическая функция интервала $[0, 1)$. В многомерном случае ситуация значительно сложнее. Легко видеть, что характеристическая функция многомерного интервала $[0, 1)^p$ для общей невырожденной целочисленной матрицы A не является масштабирующей. Попробуем построить аналог функции Хаара из масштабирующего соотношения.

Пусть $D = \{d_0, d_1, ..., d_{N-1}\}$ – элементы из $\mathbf{Z}^p$, представляющие все классы фактор-группы $\mathbf{Z}^p/A\mathbf{Z}^p$. Тогда аналогом масштабирующего соотношения Хаара естественно считать следующее уравнение:

$$\varphi(x) = \sum_{s=0}^{N-1} \varphi(Ax - d_s).$$

При некоторых дополнительных условиях [BJ99] на матрицу A решением этого уравнения является характеристическая функция компактного множества $\mathbf{T}(A, D) \subset \mathbf{R}^p$, удовлетворяющего уравнению

$$A\mathbf{T} = \bigcup_{s=0}^{N-1} (\mathbf{T} + d_s).$$

Решение $\varphi(x)$ может быть найдено итерационным методом (см. раздел 2.2.2):

$$\varphi_k(x) = \sum_{s=0}^{N-1} \varphi_{k-1}(Ax - d_s)$$

с начальным приближением $\varphi_0(x)$ – характеристической функцией многомерного промежутка $[0, 1)^p$. Приведем два примера.

Пример 1. Пусть $d = 2$ и $A = \begin{pmatrix} 1 & 2 \\ -2 & 1 \end{pmatrix}$. Тогда $N = \det A = 5$. Эта матрица является поворотом на угол $\alpha = \operatorname{arctg} 2$ по часовой стрелке и растяжением с коэффициентом $\sqrt{5}$. Фактор-группа $\mathbf{Z}^2/A\mathbf{Z}^2$ имеет 5 классов, представленных элементами $(0, 0)$, $e_1 = (1, 0)$, $2e_1 = (2, 0)$, $e_1 - e_2 = (1, -1)$, $2e_1 - e_2 = (2, -1)$, рис. 2.16.1. Здесь все элементы выбраны в одной фундаментальной области. Мы примем другой, более удобный выбор представителей классов: $D = \{0, -e_1, e_1, -e_2, e_2\}$.

Масштабирующее уравнение Хаара для $D = \{0, -e_1, e_1, -e_2, e_2\}$ имеет вид:

$$\varphi(x) = \varphi(Ax) + \varphi(Ax + e_1) + \varphi(Ax - e_1) + \varphi(Ax + e_2) + \varphi(Ax - e_2).$$

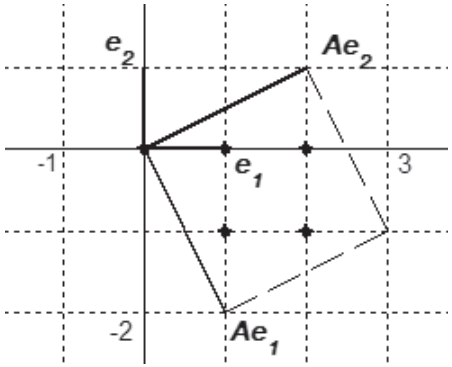


Рис. 2.16.1. Элементы, представляющие классы фактор-группы $\mathbf{Z}^2/A\mathbf{Z}^2$

Подробнее, в координатах:

$$\varphi(x_1, x_2) = \varphi(x_1 + 2x_2, -2x_1 + x_2) + \varphi(x_1 + 2x_2 + 1, -2x_1 + x_2) + \\ + \varphi(x_1 + 2x_2 - 1, -2x_1 + x_2) + \varphi(x_1 + 2x_2, -2x_1 + x_2 + 1) + \\ + \varphi(x_1 + 2x_2, -2x_1 + x_2 - 1).$$

Решением является [BJ99] характеристическая функция $\varphi(x_1, x_2) = \chi_{\mathbf{T}}(x_1, x_2)$ компактного множества $\mathbf{T} \subset \mathbf{R}^2$, которое обладает свойством $\mathbf{T} = \bigcup_{s=0}^{5} A^{-1}(\mathbf{T} + d_s)$, имеет меру 1, замощает $\mathbf{R}^2$ при целочисленных сдвигах и имеет фрактальную границу, рис. 2.16.2.

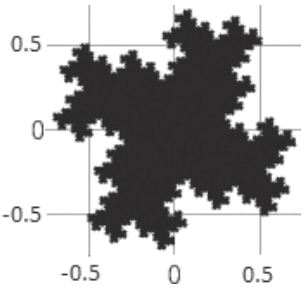


Рис. 2.16.2. Множество $\mathbf{T}$ для $D = \{0, -e_1, e_1, -e_2, e_2\}$

Отметим, что множество $\mathbf{T}$ зависит от выбора элементов, представляющих классы фактор-группы $\mathbf{Z}^2/A\mathbf{Z}^2$.

Рассмотрим вопрос о том, почему получилось такое сложное множество $\mathbf{T}$ и как его можно построить. Сначала приведем несложную процедуру на MATLAB для вычисления итерационных приближений функции Хаара $\varphi(x, y)$. В программе используется функция `Ha01.m` – характеристическая функция промежутка [0, 1). Ее листинг имеется в разделе 2.2.2.

Пусть $\varphi_0(x, y) = \chi_{[01)}(x)\chi_{[01)}(y)$ – характеристическая функция 2-мерного интервала $[0, 1)^2$. Ее реализация в MATLAB:

```
function z=Phi0(x1,x2);
for k=1:length(x1);
    for s=1:length(x2);
        z(k,s)=Ha01(x1(k))*Ha01(x2(s));
    end
end
```

Первое приближение $\varphi_1(x, y)$ находится из масштабирующего соотношения

$$\varphi_1(x_1, x_2) = \varphi_0(x_1 + 2x_2, -2x_1 + x_2) + \varphi_0(x_1 + 2x_2 + 1, -2x_1 + x_2) + \varphi_0(x_1 + 2x_2 - 1, -2x_1 + x_2) + \varphi_0(x_1 + 2x_2, -2x_1 + x_2 + 1) + \varphi_0(x_1 + 2x_2, -2x_1 + x_2 - 1).$$

Листинг для функции $\varphi_1(x_1, x_2)$:

```
function z=Phi1(x1,x2);
for k=1:length(x);
    for s=1:length(y);
z0(k,s)=Phi0(x1(k)+2*x2(s),-2*x1(k)+x2(s));
z1(k,s)=Phi0(x1(k)+2*x2(s)+1,-2*x1(k)+x2(s)); %Сдвиг на -e_1
z2(k,s)=Phi0(x1(k)+2*x2(s)-1,-2*x1(k)+x2(s)); %Сдвиг на e_1
z3(k,s)=Phi0(x1(k)+2*x2(s),-2*x1(k)+x2(s)+1); %Сдвиг на -e_2
z4(k,s)=Phi0(x1(k)+2*x2(s),-2*x1(k)+x2(s)-1); %Сдвиг на e_2
z(k,s)=z0(k,s)+z1(k,s)+z2(k,s)+z3(k,s)+z4(k,s);
    end
end
```

Дальнейшие итерации $\varphi_k(x_1, x_2)$ находятся совершенно аналогично. Их листинги получаются из приведенного выше листинга для $\varphi_1(x, y)$ простой заменой символов `Phi1` на символ `Phi2` и `Phi0` на символ `Phi1` и т. д.

Носитель функции $\varphi_0(x_1, x_2)$ – это единичный квадрат (рис. 2.16.3). Носитель функции $\varphi_1(x) = \varphi_0(Ax) + \varphi_0(Ax + e_1) + \varphi_0(Ax - e_1) + \varphi_0(Ax + e_2) + \varphi_0(Ax - e_2)$ – это 5 квадратов – носителей пяти слагаемых (рис. 2.16.3):

$$\operatorname{supp} \varphi_0(Ax) = \{x \in \mathbf{R}^2; Ax \in [0, 1]\times[0, 1]\} = \{x \in \mathbf{R}^2; x \in A^{-1}([0, 1]\times[0, 1]\},$$

$$\operatorname{supp} \varphi_0(Ax + e_1) = \{x \in \mathbf{R}^2; Ax + e_1 \in [0, 1]\times[0, 1]\} = \{x \in \mathbf{R}^2; x \in A^{-1}([-1, 0]\times[0, 1]\},$$

$$\operatorname{supp} \varphi_0(Ax - e_1) = \{x \in \mathbf{R}^2; Ax - e_1 \in [0, 1]\times[0, 1]\} = \{x \in \mathbf{R}^2; x \in A^{-1}([1, 2]\times[0, 1]\},$$

$$\operatorname{supp} \varphi_0(Ax + e_2) = \{x \in \mathbf{R}^2; Ax + e_2 \in [0, 1]\times[0, 1]\} = \{x \in \mathbf{R}^2; x \in A^{-1}([0, 1]\times[-1, 0]\},$$

$$\operatorname{supp} \varphi_0(Ax - e_2) = \{x \in \mathbf{R}^2; Ax - e_2 \in [0, 1]\times[0, 1]\} = \{x \in \mathbf{R}^2; x \in A^{-1}([0, 1]\times[1, 2]\}.$$

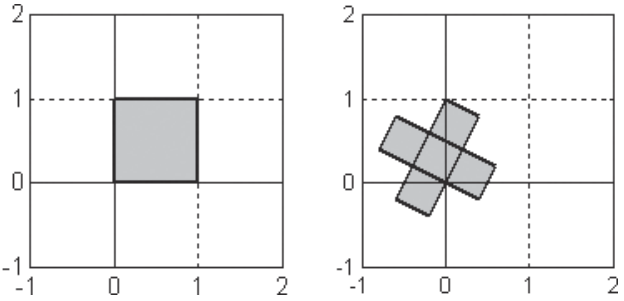


Рис. 2.16.3. Носители функций $\varphi_0(x, y)$ и $\varphi_1(x, y)$

Мы видим, что носитель первой итерации, функции $\varphi_1(x_1, x_2)$, получается в виде объединения пяти множеств, каждое их которых получается из носителя функции $\varphi_0(x_1, x_2)$ действием пяти функций:

$$F_0(x) = A^{-1}(x), \qquad F_1(x) = A^{-1}(x - e_1), \qquad F_2(x) = A^{-1}(x + e_1),$$
$$F_3(x) = A^{-1}(x - e_2), \qquad F_4(x) = A^{-1}(x + e_2).$$

Носитель второй итерации, функции $\varphi_2(x_1, x_2)$, получается в виде объединения пяти множеств, каждое их которых получается из носителя функции $\varphi_1(x_1, x_2)$ действием этих же пяти функций $F_0(x)$, $F_1(x)$, $F_2(x)$, $F_3(x)$, $F_4(x)$. Мы имеем *систему итерируемых функций* (IFS) $\{F_0(x), F_1(x), F_2(x), F_3(x), F_4(x)\}$, действующих на компактных множествах в $\mathbf{R}^2$ с коэффициентами сжатия $1/\sqrt{5}$ (подробнее об итерируемых функциях см. [У]). Носитель масштабирующей функции Хаара $\varphi(x_1, x_2)$ получается в пределе и является фракталом, построенным на основе системы итерируемых функций $\{F_0, F_1, F_2, F_3, F_4\}$.

Для построения таких фрактальных множеств имеется свободно распространяемая программа IFS Builder 3d (http://fractals.nsu.ru/), созданная А. Кравченко и Д. Мехонцевым для построения 3-мерных фрактальных множеств. Воспользуемся этой программой для построения итераций множества **T** – носителя масштабирующей функции Хаара $\varphi(x_1, x_2)$. Поскольку действие матрицы A есть поворот на угол $\alpha = \operatorname{arctg} 2$ по часовой стрелке и растяжение с коэффициентом $\sqrt{5}$, то действие матрицы A^{-1} заключается в повороте на угол $\alpha = \operatorname{arctg} 2 = 63.434948822922010^\circ$ против часовой стрелки и сжатии с коэффициентом $1/\sqrt{5} = 0.447213595499958$. Приведем основной фрагмент программы для IFS Builder 3d для построения итераций фрактального множества **T** – носителя масштабирующей функции Хаара $\varphi(x_1, x_2)$:

```
   //создание квадрата как фрактала
f1 := scale(0.5,0.5,0);
f2 := scale(0.5,0.5,0)*translate(1.0,1.0,0);
f3 := scale(0.5,0.5,0)*translate(1.0,0,0);
f4 := scale(0.5,0.5,0)*translate(0,1.0,0);
set sq = (f1+f2+f3+f4)sq;         //система IFS для квадрата

   //создание фрактала T, система итерируемых функций
F0 := rotate(-63.434948822922010)*
      scale(0.447213595499958,0.447213595499958,0);
F1 := rotate(-63.434948822922010)*
      scale(0.447213595499958,0.447213595499958,0)*translate(1.0,0,0);
F2 := rotate(-63.434948822922010)*
      scale(0.447213595499958,0.447213595499958,0)*translate(-1.0,0,0);
F3 := rotate(-63.434948822922010)*
      scale(0.447213595499958,0.447213595499958,0)*translate(0,1.0,0);
F4 := rotate(-63.434948822922010)*
      scale(0.447213595499958,0.447213595499958,0)*translate(0,-1.0,0);

m:=(F0+F1+F2+F3+F4);
build
//sq;   //квадрат
//(m)sq;    //первая итерация
//m(m) sq;  //вторая итерация
//m(m(m)) sq;    //третья итерация
(m(m(m(m(m(m)))))sq;    //шестая итерация
```

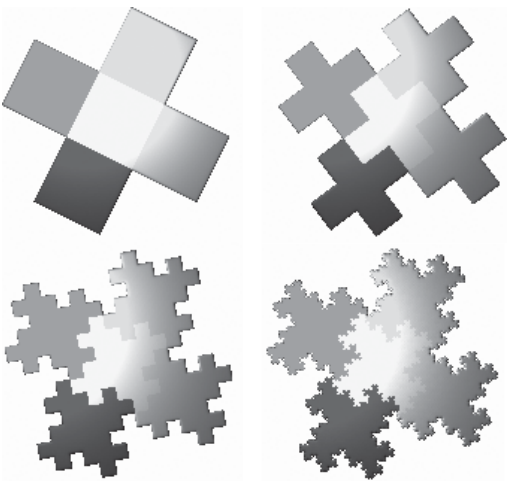


Рис. 2.16.4. Итерации построения фрактального множества $\mathbf{T}$, носители функций $\varphi_1(x, y)$, $\varphi_2(x, y)$, $\varphi_3(x, y)$ и $\varphi_6(x, y)$

Пример 2. Приведем еще один пример, когда $N = \det A = 2$. Рассмотрим так называемую квинкунциальную матрицу и $A = \begin{pmatrix} 1 & -1 \\ 1 & 1 \end{pmatrix}$. Это поворот на 45% против часовой стрелки и растяжение с коэффициентом $\sqrt{2}$. Фактор-группа $\mathbf{Z}^2/A\mathbf{Z}^2$ имеет 2 класса, представленные элементами (0, 0) и $e_2 = (0, 1)$, рис. 2.16.5. Отметим, что такое название матрицы связано с тем, что вектор $e_2 = (0, 1)$ находится в центре квадрата, образованного элементами класса $A\mathbf{Z}^2$, образуя конфигурацию из пяти элементов, рис. 2.16.5.

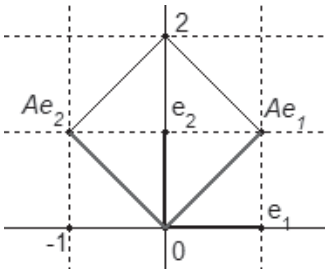


Рис. 2.16.5. Элементы, представляющие классы фактор-группы $\mathbf{Z}^2/A\mathbf{Z}^2$

Масштабирующее уравнение Хаара для $D = \{0, e_2\}$ имеет вид $\varphi(x) = \varphi(Ax) + \varphi(Ax - e_2)$. Подробнее, в координатах:

$$\varphi(x_1, x_2) = \varphi(x_1 - x_2, x_1 + x_2) + \varphi(x_1 - x_2, x_1 + x_2 - 1).$$

Решением является характеристическая функция $\varphi(x_1, x_2) = \chi_{\mathbf{T}}(x_1, x_2)$ компактного множества $\mathbf{T} \subset \mathbf{R}^2$, которое удовлетворяет равенству $\mathbf{T} = A^{-1}(\mathbf{T}) \cup A^{-1}(\mathbf{T} + e_2)$, имеет меру 1, замощает $\mathbf{R}^2$ при целочисленных сдвигах и имеет фрактальную границу, рис. 2.16.6. Это множество принято называть «Twin Dragon» – спаренный дракон (рис. 2.16.6 и 2.16.7).

Рассмотрим вопрос о том, как можно построить множество **T**. Сначала приведем несложную процедуру на MATLAB для вычисления итерационных приближений функции Хаара $\varphi(x, y)$. Пусть $\varphi_0(x, y) = \chi_{[01)}(x) \times \chi_{[01)}(y)$ – характеристическая функция 2-мерного интервала $[0, 1)^2$. Первое приближение $\varphi_1(x, y)$ находится из масштабирующего соотношения

$$\varphi_1(x_1, x_2) = \varphi_0(x_1 - x_2, x_1 + x_2) + \varphi_0(x_1 - x_2, x_1 + x_2 - 1).$$

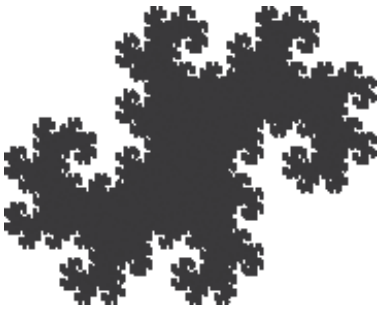


Рис. 2.16.6. Множество **T**
для $D = \{0, e_2\}$ (спаренный дракон)

В листинге для функции $\varphi_1(x_1, x_2)$ используется функция `Phi0` из предыдущего примера:

```
function z=Phi1(x1,x2);
for k=1:length(x);
    for s=1:length(y);
z0(k,s)=Phi0(x1(k)+2*x2(s),-2*x1(k)+x2(s));
z1(k,s)=Phi0(x1(k)+2*x2(s),-2*x1(k)+x2(s)-1); % + сдвиг на e_2
z(k,s)=z0(k,s)+z1(k,s);
    end
end
```

Дальнейшие итерации $\varphi_k(x_1, x_2)$ находятся совершенно аналогично. Их листинги получаются из приведенного выше листинга для $\varphi_1(x, y)$ простой заменой символов `Phi1` на символ `Phi2` и `Phi0` на символ `Phi1` и т. д.

Носитель функции $\varphi_0(x_1, x_2)$ – это единичный квадрат. Носитель функции $\varphi_1(x) = \varphi_0(Ax) + \varphi_0(Ax - e_2)$ – это 2 квадрата – носители двух слагаемых:

$$\operatorname{supp} \varphi_0(Ax) = \{x \in \mathbf{R}^2; Ax \in [0, 1]\times[0, 1]\} =$$
$$= \{x \in \mathbf{R}^2; x \in A^{-1}([0, 1]\times[0, 1]\},$$

$$\operatorname{supp} \varphi_0(Ax - e_2) = \{x \in \mathbf{R}^2; Ax - e_2 \in [0, 1]\times[0, 1]\} =$$
$$= \{x \in \mathbf{R}^2; x \in A^{-1}([0, 1]\times[1, 2]\}.$$

Мы видим, что носитель первой итерации, функции $\varphi_1(x_1, x_2)$, получается в виде объединения двух множеств, каждое их которых получается из носителя функции $\varphi_0(x_1, x_2)$ действием двух функций:

$$F_0(x) = A^{-1}(x), \quad F_1(x) = A^{-1}(x + e_2).$$

Первое множество – это сжатие исходного единичного квадрата с коэффициентом $1/\sqrt{2}$ и поворот квадрата на 45% по часовой стрелке. Второе множество – это сдвиг вверх исходного единичного квадрата на 1, сжатие с коэффициентом $1/\sqrt{2}$ и поворот квадрата на 45% по часовой стрелке. Носитель второй итерации, функции $\varphi_2(x_1, x_2)$, получается в виде объединения двух множеств, каждое из которых получается из носителя функции $\varphi_1(x_1, x_2)$ действием этих же двух функций $F_0(x), F_1(x)$. Мы имеем *систему итерируемых функций* (IFS) $\{F_0(x), F_1(x)\}$, действующих на компактных множествах в $\mathbf{R}^2$ с коэффициентами сжатия $1/\sqrt{2}$. Носитель масштабирующей функции Хаара $\varphi(x_1, x_2)$ получается в пределе и является фракталом, определенным системой итерируемых функций $\{F_0, F_1\}$.

Для построения этого фрактального множества воспользуемся свободно распространяемой программой IFS Builder 3d (http://fractals.nsu.ru/). Построим несколько итераций для построения множества **T** – носителя масштабирующей функции Хаара $\varphi(x_1, x_2)$. Поскольку действие матрицы A есть поворот на угол 45% против часовой стрелки и растяжение с коэффициентом $\sqrt{2}$, то действие матрицы A^{-1} заключается в повороте на угол $\alpha = 45\%$ по часовой стрелке и сжатии с коэффициентом $1/\sqrt{2} = 0.70710678118654752440$. Приведем основной фрагмент программы для IFS Builder 3d для построения итераций фрактального множества **T** (рис. 2.16.7):

```
   //sozdanie drakona
F0 := rotate(45)*scale(0.7071067811865475244,0.7071067811865475244,0);
F1 := rotate(45)*scale(0.7071067811865475244,0.7071067811865475244,0)
*translate(0,1.0,0);
m:=(F0+F1); //система функций для дракона
//set sq_n=sq;           // первая итерация
//set sq_n=(m)sq;        // вторая итерация
//set sq_n=(m(m))sq;     // третья итерация
set sq_n=(m(m(m(m(m(m(m(m(m(m)))))))))sq; //
build F0*sq_n; color(0,0,1);
build F1*sq_n; color(0,1,0);
```

Приведенные два примера показывают, что даже простейшая в одномерном случае масштабирующая функция Хаара перестает быть простой даже в двумерном случае. Ее носитель – фрактальное множество. Поэтому является актуальной задача построения масштабирующей функции в многомерном случае, которая определяла бы кратномасштабное разложение, аналогичное одномерному, то есть на

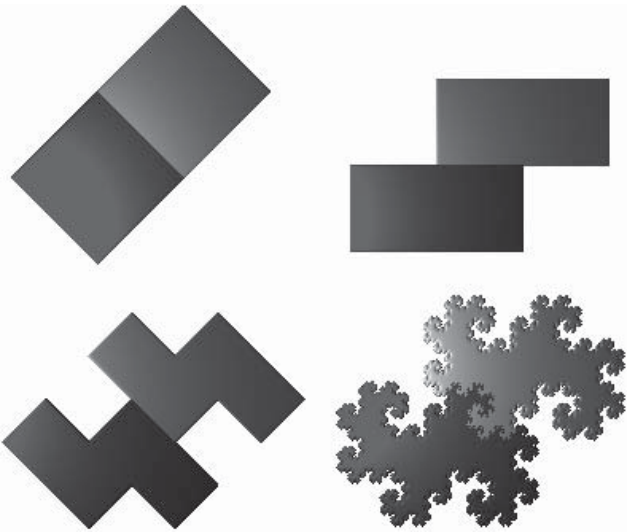


Рис. 2.16.7. Итерации построения спаренного дракона **T**, носители функций $\varphi_1(x, y)$, $\varphi_2(x, y)$, $\varphi_3(x, y)$ и $\varphi_{12}(x, y)$

основе кусочно-постоянных функций на достаточно простых множествах. Обсудим кратко подходы к решению этой задачи.

Для квинкунциальной матрицы $A = \begin{pmatrix} 1 & -1 \\ 1 & 1 \end{pmatrix}$ рассмотрим на плоскости $\mathbf{R}^2$ единичный квадрат $[-1/2, 1/2]\times[-1/2, 1/2]$. Пусть S – группа 8 симметрий квадрата, заданная элементами $S = \{\sigma_i : i = 0, 1, ..., 7\}$, где

$$\sigma_0 = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}, \quad \sigma_1 = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}, \quad \sigma_2 = \begin{pmatrix} 0 & -1 \\ 1 & 0 \end{pmatrix}, \quad \sigma_3 = \begin{pmatrix} -1 & 0 \\ 0 & 1 \end{pmatrix},$$

и $\sigma_i = -\sigma_{i-4}$ для $i = 4, 5, 6, 7$. Пусть R_0 – треугольник с вершинами (0, 0), (1/2, 0), (1/2, 1/2) и пусть $R_i = \sigma_i R_0$, $i = 1, 2, ..., 7$ – остальные треугольники (рис. 2.16.8).

Теперь каждый треугольник R_i можно представить в виде объединения сжатых и сдвинутых треугольников. Например, $R_0 = A^{-1}R_1 \cup \cup A^{-1}(R_6 + e_2)$. Напомним, что невыполнение такого свойства для квадрата и приводило к фрактальным носителям.

Пусть $\phi = \chi_{R_0}$ – характеристическая функция треугольника R_0. Определим векторнозначную функцию

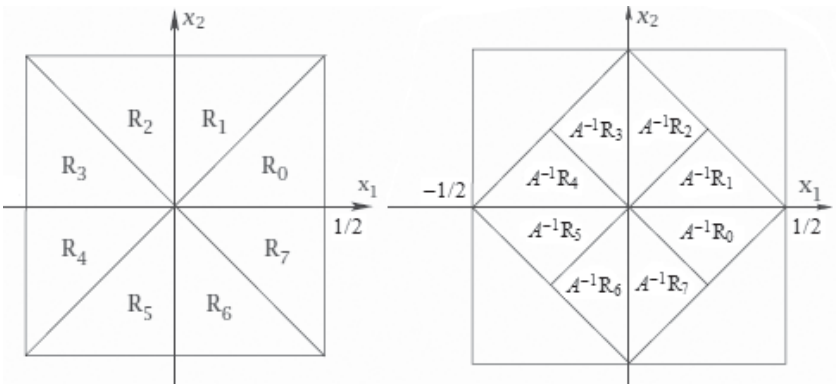


Рис. 2.16.8. Разбиение квадрата на симметричные треугольники

$$\Phi = \begin{pmatrix} D_{\sigma_0}\phi \\ \vdots \\ D_{\sigma_7}\phi \end{pmatrix} = \begin{pmatrix} \phi(\sigma_0^{-1}x) \\ \vdots \\ \phi(\sigma_7^{-1}x) \end{pmatrix} = \begin{pmatrix} \phi^0(x) \\ \vdots \\ \phi^7(x) \end{pmatrix},$$

состоящую из характеристических функций треугольников $R_i = \sigma_i R_0$, $i = 1, 2, ..., 7$. Легко видеть, что

$$\phi(x) = \phi^0(x) = \phi^1(Ax) + \phi^6(Ax - e_2).$$

Применяя операторы $D_{\sigma_i} f(x) = f(\sigma_i^{-1}x)$, $i = 1, 2, ..., 7$, к последнему выражению, мы получаем аналогичные равенства для остальных функций $\phi^i(x)$. Эти равенства в совокупности образуют аналоги масштабирующих соотношений Хаара.

Определим пространство V_0 как замыкание линейной оболочки функций

$$\{\phi(\sigma x - k) : \sigma \in S, k \in \mathbf{Z}^2\}.$$

Отметим, что $|\det \sigma| = 1$ для любого $\sigma \in S$ и V_0 есть подпространство $L^2(\mathbf{R}^2)$, состоящее из функций, которые постоянны на $\mathbf{Z}^2$-переносах треугольников R_i, $i = 0, 1, ..., 7$. Аналог классического кратномасштабного разложения состоит из пространств V_j, порожденных элементами вектор-функций $|\det A|^{j/2}\Phi(A^j x - n)$, $j \in \mathbf{Z}$, $n \in \mathbf{Z}^2$. Тогда $\{V_j : j \in \mathbf{Z}\}$ есть кратномасштабный анализ с векторнозначной масштабирующей функцией Φ.

Положим теперь $\psi(x) = \phi^1(Ax) - \phi^6(Ax - e_2)$. Эта функция приводит нас к искомому аналогу вейвлета Хаара. Это векторнозначная функция

$$\Psi = \begin{pmatrix} D_{\sigma_0}\psi \\ \vdots \\ D_{\sigma_7}\psi \end{pmatrix} = \begin{pmatrix} \psi(\sigma_0^{-1}x) \\ \vdots \\ \psi(\sigma_7^{-1}x) \end{pmatrix} = \begin{pmatrix} \psi^0(x) \\ \vdots \\ \psi^7(x) \end{pmatrix}.$$

Отметим, что система функций $\{|\det A|^{j/2}\psi(A^j x - k) : j \in \mathbf{Z}^2\}$ образует ортонормированный базис $L^2(\mathbf{R}^2)$.

Таким образом, вместо фрактальных аналогов вейвлетов Хаара мы получаем мультивейвлетные многомерные аналоги с достаточно простыми носителями.

Более подробно об этом направлении можно прочитать в статьях [GLLWW1], [GLLWW2] и в обзоре [LWW13].

2.16.2. *A*-кратномасштабное разложение

Дадим здесь определение ортогонального *кратномасштабного* разложения пространства $L^2(\mathbf{R}^p)$ с матричным коэффициентом масштабирования A и установим аналоги условий ортогональности.

Определение 2. *Ортогональным A-кратномасштабным разложением пространства* $L^2(\mathbf{R}^p)$ *называется последовательность замкнутых вложенных друг в друга подпространств* $\ldots \subset V_{-1} \subset V_0 \subset V_1 \subset V_2 \subset \ldots$, *обладающая свойствами:*

1) *замыкание их объединения совпадает с* $L^2(\mathbf{R}^p)$, $\overline{\bigcup_{j\in\mathbf{Z}} V_j} = L^2(\mathbf{R}^p)$;
2) *пересечение всех подпространств – нулевое,* $\bigcap_{j\in\mathbf{Z}} V_j = \{0\}$;
3) $f(x) \in V_j \Leftrightarrow f(Ax) \in V_{j+1}$;
4) *существует такая функция* $\varphi(x) \in V_0$, *называемая масштабирующей, что функции* $\varphi_{0,n}(x) = \varphi(x - n)$, $n \in \mathbf{Z}^p$ *образуют ортонормированный базис пространства* V_0.

Поскольку пространства V_j являются масштабированными версиями пространства V_0, то из свойства 4 сразу вытекает, что функции

$$\varphi_{j,n}(x) = \sqrt{|\det A|^j}\,\varphi(A^j x - n), \quad n \in \mathbf{Z}^p,$$

образуют ортонормированный базис пространства V_j для любого j.

Теорема 2. *Функции* $\varphi_n(x) = \varphi(x - n)$ *образуют ортонормированный базис подпространства* $V_0 \subset L^2(\mathbf{R})$ *тогда и только тогда, когда*

$$\sum_{n\in\mathbf{Z}^p} |\hat{\varphi}(\omega + 2\pi n)|^2 = 1 \text{ п. в.} \tag{8}$$

Доказательство. Заменой переменной любое скалярное произведение (φ_k, φ_n) можно привести к виду (φ, φ_n). Тогда по теореме Планшереля имеем:

$$(\varphi,\varphi_n)=\int\limits_{\mathbf{R}^p}\varphi(x)\overline{\varphi(x-n)}dx=\frac{1}{(2\pi)^p}\int\limits_{\mathbf{R}^p}\hat{\varphi}(\omega)\overline{e^{-in\omega}\hat{\varphi}(\omega)}d\omega=$$

$$=\frac{1}{(2\pi)^p}\int\limits_{\mathbf{R}^p}e^{in\omega}|\hat{\varphi}(\omega)|^2d\omega=\frac{1}{(2\pi)^p}\sum_{k\in\mathbf{Z}^p}\int\limits_{\Omega^p}e^{in\omega}\left|\hat{\varphi}(\omega+2\pi k)\right|^2d\omega=\delta_{0n},$$

где $\Omega^p=[0,\,2\pi]\times\ldots\times[0,\,2\pi]$. Из полноты системы функций $\{e^{-in\omega}\}$ на p-мерном промежутке Ω^p следует то, что интеграл равен δ_{0n} тогда и только тогда, когда сумма под знаком интеграла равна 1 почти всюду, $\sum_{k\in\mathbf{Z}^p}\left|\hat{\varphi}(\omega+2\pi k)\right|^2=1$ *п. в.*

Пример 3. Кратномасштабное разложение Шеннона. Пусть снова $d=2$ и $A=\begin{pmatrix}1 & 2\\ -2 & 1\end{pmatrix}$. Рассмотрим функцию

$$\varphi(x,y)=\frac{\sin\pi x\sin\pi y}{\pi^2xy}.$$

Покажем, что $\varphi(x,\,y)$ определяет ортогональное кратномасштабное разложение. Преобразование Фурье функции $\varphi(x,\,y)$ имеет вид

$$\hat{\varphi}(\omega_1,\omega_2)=\begin{cases}1, & \omega_1,\omega_2\in[-\pi,\pi],\\ 0, & \text{при других } \omega_1,\omega_2.\end{cases}$$

Поэтому необходимое и достаточное условие (8) ортонормированности системы функций $\{\varphi(x-n,\,y-k),\,n,k\in\mathbf{Z}\}$ выполнено.

Очевидно, что пространство V_0, порожденное целочисленными сдвигами $\varphi(x-n,\,y-k)$, состоит из функций $f(x,\,y)$, преобразование Фурье которых имеет носитель в квадрате $[-\pi,\,\pi]^2$. Пространство V_1, порожденное сдвигами $\varphi(x+2y-n,\,-2x+y-k)$, состоит из функций $f(x,\,y)$, преобразование Фурье которых имеет носитель в квадрате $A^t([-\pi,\,\pi]^2)$, который получается из квадрата $[-\pi,\,\pi]^2$ действием матрицы A^t (это следует из формулы (2)). Поскольку коэффициент растяжения матрицы A^t равен $\sqrt{5}$, то имеет место включение $[-\pi,\,\pi]^2\subset A^t([-\pi,\,\pi]^2)$. Поэтому $V_0\subset V_1$. Следовательно, функция $\varphi(x,\,y)=(\sin\pi x\sin\pi y)/(\pi^2xy)$ является A-масштабирующей. Заметим, что данная функция является масштабирующей для любой матрицы A, у которой модули собственных чисел будут больше или равны $\sqrt{2}$. Действительно, в этом случае выполняется включение $[-\pi,\,\pi]^2\subset$ $\subset A^t([-\pi,\,\pi]^2)$ и, следовательно, $V_0\subset V_1$.

Получим необходимое условие ортогональности, использующее функцию $H_0(\omega)$. Рассмотрим элементы $d_s\in\mathbf{Z}^p$, $s=0,\,1,\,\ldots,\,N-1$, представляющие все классы конечной группы $\mathbf{Z}^p/A^t\mathbf{Z}^p$. Напомним, что

$N = |\det A|$. Считаем, что $d_0 = 0$. Тогда целочисленная решетка $\mathbf{Z}^p$ является объединением следующих классов:

$A^t\mathbf{Z}^p, d_1 + A^t\mathbf{Z}^p, ..., d_{N-1} + A^t\mathbf{Z}^p$.

Следующий пример демонстрирует данное разложение.

Пример 4. Пусть $d = 2$ и $A = \begin{pmatrix} 1 & 2 \\ -2 & 1 \end{pmatrix}$. Тогда $N = \det A = 5$. Фактор-группа $\mathbf{Z}^2/A^t\mathbf{Z}^2$ имеет 5 классов, представленных элементами (0, 0), $d_1 = e_2 = (0, 1)$, $d_2 = 2e_2 = (0, 2)$, $d_3 = e_2 - e_1 = (-1, 1)$ и $d_4 = 2e_2 - e_1 = (-1, 2)$, рис. 2.16.9. Здесь все элементы выбраны в одной фундаментальной области. Возможен и другой выбор: $D = \{0, e_1, -e_1, e_2, -e_2\}$.

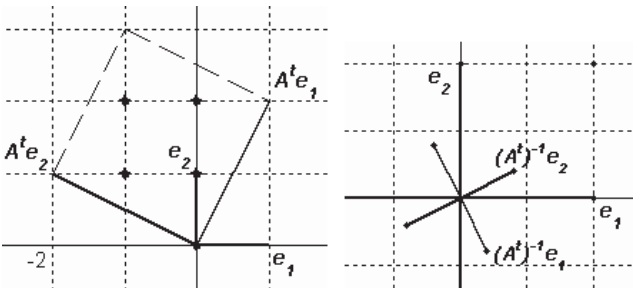


Рис. 2.16.9. Элементы, представляющие классы фактор-группы $\mathbf{Z}^2/A^t\mathbf{Z}^2$, и элементы $(A^t)^{-1}d_s$: 0, $(A^t)^{-1}e_1$, $-(A^t)^{-1}e_1$, $(A^t)^{-1}e_2$, $-(A^t)^{-1}e_1$

Теорема 3. *Если сдвиги* $\varphi_n(x) = \varphi(x - n)$ *масштабирующей функции* $\varphi(x)$ *образуют ортонормированный базис подпространства* V_0, *то частотная функция* $H_0(\omega)$ *обладает следующим свойством*:

$$\sum_{s=0}^{N-1} |H_0(\omega + 2\pi(A^t)^{-1}(d_s))|^2 = 1 \text{ п. в.}, \tag{9}$$

где элементы $d_s \in \mathbf{Z}^p$ *представляют все классы фактор-группы* $\mathbf{Z}^p/A^t\mathbf{Z}^p$.

Доказательство. Подставим в (8) выражение $\hat{\varphi}(\omega) = H_0((A^t)^{-1}\omega) \times \hat{\varphi}((A^t)^{-1}\omega)$:

$$1 = \sum_{k\in\mathbf{Z}^p} |\hat{\varphi}(\omega + 2\pi k)|^2 = \sum_{k\in\mathbf{Z}^p} \left|H_0((A^t)^{-1}(\omega + 2\pi k))\hat{\varphi}((A^t)^{-1}(\omega + 2\pi k))\right|^2 =$$

$$= \sum_{k\in\mathbf{Z}^p} \left|H_0((A^t)^{-1}\omega + 2\pi(A^t)^{-1}k)\hat{\varphi}((A^t)^{-1}\omega + 2\pi(A^t)^{-1}k)\right|^2 = [(A^t)^{-1}\omega = \varsigma] =$$

$$= \sum_{k\in\mathbf{Z}^p} \left|H_0(\varsigma + 2\pi(A^t)^{-1}k)\hat{\varphi}(\varsigma + 2\pi(A^t)^{-1}k)\right|^2.$$

Выберем по одному представителю из каждого класса конечной группы $\mathbf{Z}^p/A^t\mathbf{Z}^p$. Пусть это будут элементы $D = \{d_0 = 0, d_1, ..., d_{N-1}\}$. Тогда целочисленная решетка $\mathbf{Z}^p$ является объединением следующих классов: $A^t\mathbf{Z}^p$, $d_1 + A^t\mathbf{Z}^p$, ..., $d_{N-1} + A^t\mathbf{Z}^p$. Поэтому последняя сумма может быть разбита на N сумм следующим образом:

$$\sum_{k\in\mathbf{Z}^p}\left|H_0(\varsigma+2\pi(A^t)^{-1}k)\hat{\varphi}(\varsigma+2\pi(A^t)^{-1}k)\right|^2 =$$

$$=\sum_{m\in\mathbf{Z}^p}\sum_{s=0}^{N-1}\left|H_0(\varsigma+2\pi(A^t)^{-1}(d_s+A^tm))\hat{\varphi}(\varsigma+2\pi(A^t)^{-1}(d_s+A^tm))\right|^2 =$$

$$=\sum_{m\in\mathbf{Z}^p}\sum_{s=0}^{N-1}\left|H_0(\varsigma+2\pi(A^t)^{-1}(d_s)+2\pi m))\hat{\varphi}(\varsigma+2\pi(A^t)^{-1}(d_s)+2\pi m))\right|^2 =$$

$$=\sum_{s=0}^{N-1}\sum_{m\in\mathbf{Z}^p}\left|H_0(\varsigma+2\pi(A^t)^{-1}(d_s))\right|^2\left|\hat{\varphi}(\varsigma+2\pi(A^t)^{-1}(d_s)+2\pi m))\right|^2 =$$

$$=\sum_{s=0}^{N-1}\left|H_0(\varsigma+2\pi(A^t)^{-1}(d_s))\right|^2=1.$$

В последних равенствах мы использовали 2π-периодичность функции $H_0(\omega)$ и равенство (8).

Так же как и в разделе 2.3.1, показывается, что условие (9) эквивалентно следующему:

$$\sum_n h_n\overline{h}_{n-Ar}=\delta_{0r}. \tag{10}$$

Получим *достаточное условие* ортонормированности для вейвлетов с компактным носителем с конечным числом ненулевых коэффициентов $\{h_n\}$. Формулировка и доказательство вполне аналогичны скалярному случаю (см. раздел 2.3.5 и [Дб, стр. 262]). Необходимое условие (9), (10) предполагается выполненным.

Предположим, что индексы $n = (n_1, ..., n_p)$ меняются в пределах: $0 \le n_1 \le L_1$, ..., $0 \le n_p \le L_p$. Тогда частотная функция является тригонометрическим полиномом вида

$$H_0(\omega)=\frac{1}{\sqrt{N}}\sum_{n=0}^{L}h_ne^{-i(n,\omega)}=\frac{1}{\sqrt{N}}\sum_{n_1,...,n_p=0}^{L_1,...,L_p}h_{n_1,...,n_pn}e^{-i(n_1\omega_1+...+n_p\omega_p)}.$$

В следующей теореме сумма $\sum_{n=0}^{L}$ будет использоваться именно в этом смысле.

Теорема 4. *Предположим, что частотная функция является тригонометрическим полиномом вида* $H_0(\omega)=\sqrt{N^{-1}}\sum_{n=0}^{L}h_ne^{-i(n,\omega)}$, *удовлет-*

воряющим условиям (9) *и* $H_0(0) = 1$, *а функция* $\varphi(x)$ *определена соотношением* $\hat{\varphi}(\omega) = \prod_{j=1}^{\infty} H_0((A^t)^{-j}(\omega))$. *Если собственное значение* 1 *матрицы* $A_{kl} = \sum_{n=0}^{L} h_n \overline{h}_{k-Al+n}$ *не является кратным, то функции* $\varphi_{0,n}(x) = \varphi(x - n)$, $n \in \mathbf{Z}^p$ *образуют ортонормированную систему.*

Доказательство. Очевидно, что система функций $\varphi(x - n)$ образует ортонормированную систему тогда и только тогда, когда $\int_{\mathbf{R}^p} \varphi(x)\overline{\varphi(x-l)}dx = \delta_{0,l}$. Поскольку носитель функции $\varphi(x)$ компактен, то для проверки ортонормированности нужно найти конечный набор чисел $\alpha_l = \int_{\mathbf{R}^p} \varphi(x)\overline{\varphi(x-l)}dx$. В случае ортонормированности массив α с векторным индексом l имеет вид: $\alpha_l = \delta_{0l}$. Используя масштабирующее соотношение $\varphi(x) = \sqrt{N}\sum_{n=0}^{L} h_n \varphi(Ax - n)$, сделаем следующие преобразования:

$$\alpha_l = \int_{\mathbf{R}^p} \varphi(x)\overline{\varphi(x-l)}dx = |\det A| \int_{\mathbf{R}^p} \sum_{n,m} h_n \overline{h_m} \varphi(Ax-n)\overline{\varphi(Ax-Al-m)}dx =$$

$$= \sum_{n,m} h_n \overline{h_m} \int_{\mathbf{R}^p} \varphi(y)\overline{\varphi(y-Al-m+n)}\,dy = \sum_{n,m} h_n \overline{h_m}\,\alpha_{Al+m-n} = \sum_k \left(\sum_{n=0}^{L} h_n \overline{h}_{k-Al+n} \right) \alpha_k.$$

Определим матрицу A (с векторными индексами) по формуле

$$A_{kl} = \sum_{n=0}^{L} h_n \overline{h}_{k-Al+n}, \tag{11}$$

где $h_m = 0$, если $m < 0$ или $m > L$. Тогда последнее равенство $\alpha_l = \sum_k A_{kl}\alpha_k$ говорит о том, что α является собственным вектором матрицы A для собственного значения 1, $A\alpha = \alpha$. Заметим, что 1 всегда является собственным значением A, если выполнено необходимое условие ортонормированности (10). Действительно, определим вектор $\beta_l = \delta_{0,l}$. Легко видеть, что вектор β является собственным для матрицы A:

$$(A\beta)_l = \sum_k A_{kl}\delta_{k,0} = A_{0l} = \sum_n h_n \overline{h}_{n-Al} = \delta_0 = \beta.$$

Если собственное значение 1 является простым, то вектор α должен быть пропорциональным β, $\alpha = c\beta$. Тогда $\int_{\mathbf{R}^p} \varphi(x)\overline{\varphi(x-l)}dx = c\delta_{l,0}$. При нормировке $\int_{\mathbf{R}^p} |\varphi(x)|^2 dx = 1$ мы получаем, что $c = 1$. Система функций $\varphi_{0,n}(x) = \varphi(x - n)$, $n \in \mathbf{Z}^2$, является ортонормированной. Теорема доказана.

Из масштабирующего соотношения $\hat{\varphi}(\omega) = H_0((A^t)^{-1}\omega)\hat{\varphi}((A^t)^{-1}\omega)$ в частотной области получаем (полагая, что функция $\hat{\varphi}(\omega)$ непрерывна и $\varphi(x)$ нормирована соотношением $\hat{\varphi}(0) = 1$):

$$\hat{\varphi}(\omega) = \prod_{j=1}^{\infty} H_0\left(\frac{\omega}{(A^t)^j}\right).$$

Данная формула дает метод для нахождения масштабирующей функции через частотную функцию $H_0(\omega)$.

Теорема 5. *Если для коэффициентов h_n разложения $H_0(0) = \frac{1}{\sqrt{N}}\sum_{n\in\mathbf{Z}^p} h_n e^{-i(n,\omega)}$ выполнено условие $\sum_{n\in\mathbf{Z}^p} |h_n| \cdot \| n \| < \infty$ и $H_0(0) = 1$, тогда бесконечное произведение*

$$\hat{\varphi}(\omega) = \prod_{j=1}^{\infty} H_0\left(\frac{\omega}{(A^t)^j}\right) \tag{12}$$

сходится абсолютно и равномерно на компактных подмножествах из $\mathbf{R}^p$.

Доказательство совершенно аналогично одномерному случаю.

Таким образом, для сходимости бесконечного произведения (12) от коэффициентов $\{h_n\}$ требуется лишь достаточно быстрое убывание. Это условие заведомо выполнено в случае, когда лишь конечное число коэффициентов $\{h_n\}$ отлично от нуля.

2.16.3. Вейвлеты с матрицей масштабирования

Пусть A – некоторая целочисленная $p \times p$ матрица и $N = |\det A| > 1$. Кроме того, предполагается, что модули собственных чисел матрицы A больше единицы. Предположим, что задано ортогональное A-кратномасштабное разложение пространства $L^2(\mathbf{R}^p)$ с масштабирующей функцией $\varphi(x) \in V_0$.

Определение 3. *Функции $\psi^1(x)$, ..., $\psi^{N-1}(x) \in L^2(\mathbf{R}^p)$ называются вейвлетами для A-масштабирующей функции $\varphi(x)$, если их целочисленные сдвиги образуют ортонормированные базисы подпространств W_0^1, W_0^2, ..., W_0^{N-1} в $L^2(\mathbf{R}^p)$, которые вместе с пространством V_0 образуют ортогональное разложение пространства V_1:*

$$V_1 = V_0 \oplus W_0^1 \oplus \ldots \oplus W_0^{N-1}. \tag{13}$$

Для любого j получаем следующее ортогональное разложение:

$$V_{j+1} = V_j \oplus W_j^1 \oplus \ldots \oplus W_j^{N-1}. \tag{14}$$

Функции

$$\psi^l_{j,n}(x) = \sqrt{N^j}\,\psi^l(A^j x - n), \quad n \in \mathbf{Z}^p, \tag{15}$$

образуют ортонормированные базисы пространств W^l_j, l = 1, 2, ..., N – 1. Поскольку $\psi^l(x) \in W^l_0 \subset V_1$, то каждый вейвлет $\psi^l(x)$ раскладывается по базису функций пространства V_1:

$$\psi^l(x) = \sqrt{|\det A|}\sum_{n\in\mathbf{Z}^p} g^l_n \varphi(Ax - n), \quad l = 1, 2, \ldots, N-1. \tag{16}$$

Числа $\{g^l_n\}$ называются фильтрами вейвлетов $\psi^l(x)$. Определим частотные функции вейвлетов $\psi^l(x)$:

$$H_l(\omega) = \frac{1}{\sqrt{N}}\sum_{n\in\mathbf{Z}^p} g^l_n e^{-i(n,\omega)}, \quad l = 1, 2, \ldots, N-1. \tag{17}$$

Для ортогональных вейвлетов $\varphi(x)$, $\psi^1(x)$, ..., $\psi^{N-1}(x)$ функции $H_0(\omega)$, $H_1(\omega)$, ..., $H_{N-1}(\omega)$ обладают свойством унитарности следующей матрицы:

$$H(z) = \frac{1}{\sqrt{N}}\begin{pmatrix} H_0(z) & H_0(\rho_1 z) & \cdots & H_0(\rho_{N-1} z) \\ H_1(z) & H_1(\rho_1 z) & \cdots & H_1(\rho_{N-1} z) \\ \cdots & \cdots & \cdots & \cdots \\ H_{N-1}(z) & H_{N-1}(\rho_1 z) & \cdots & H_{N-1}(\rho_{N-1} z) \end{pmatrix}$$

на торе T^p, $z = (z_1, \ldots, z_p) \in T^p = S^1 \times \ldots \times S^1$ и $\rho_s = e^{-i2\pi(A^t)^{-1}d_s}$ (см. раздел 2.16.5). Из унитарности данной матрицы сразу следует, что коэффициенты вейвлетов $\{h_n\}$ и $\{g^l_n\}$ обладают свойствами:

$$\sum_{n\in\mathbf{Z}^p} |g^l_n|^2 = 1, \quad \sum_{n\in\mathbf{Z}^p} \overline{g^s_n} g^l_{n-Ak} = 0, \quad \forall\, s \neq l,\ s, l = 0, 1, \ldots, N-1, k \in \mathbf{Z}^p,$$

где мы считаем, что $g^0_n = h_n$.

2.16.4. Вейвлет-преобразование

Получим формулы быстрого вейвлет-преобразования. Предположим, что мы знаем приближение $P_j(f)$ функции $f(x)$ в пространстве V_j:

$$P_j(f) = \sum_{n\in\mathbf{Z}^d} a_n \varphi_{j,n}(x),$$

где $a_n = (f, \varphi_{j,n})$. Тогда в соответствии с разложением (14) имеем:

$$P_j(f) = \sum_{k\in\mathbf{Z}^d} a_{1,k}\varphi_{j-1,k}(x) + \sum_{l=1}^{N-1}\sum_{k\in\mathbf{Z}^p} d^l_{1,k}\psi^l_{j-1,k}(x),$$

где $a_{1,k} = (f, \varphi_{j-1,k})$ и $d^l_{1,k} = (f, \psi^l_{j-1,k})$, $l = 1, 2, ..., N-1$. Выразим базисные функции $\varphi_{j-1,k}(x)$ и $\psi^l_{j-1,k}(x)$ в этих формулах через базисные функции $\varphi_{j,n}(x)$ пространства V_j по формулам (5) и (16):

$$\varphi_{j-1,k}(x) = \sum_{n\in\mathbf{Z}^p} h_n \varphi_{j,\,n+Ak}(x),$$

$$\psi^l_{j-1,k}(x) = \sum_{n\in\mathbf{Z}^p} g^l_n \varphi_{j,\,n+Ak}(x), \quad l = 1,\ 2, \ldots,\ N-1.$$

Тогда получаем:

$$a_{1,k} = (f, \varphi_{j-1,k}) = \left(f, \sum_{n\in\mathbf{Z}^p} h_n \varphi_{j,n+Ak} \right) = \sum_{n\in\mathbf{Z}^p} \overline{h_n}\, a_{n+Ak},$$

$$d^l_{1,k} = (f, \psi^l_{j-1,k}) = \left(f, \sum_{n\in\mathbf{Z}^d} g^l_n \varphi_{j,n+Ak} \right) = \sum_{n\in\mathbf{Z}^p} \overline{g^l_n}\, a_{n+Ak},\ l = 1,\ 2,\ \ldots,\ N-1.$$

Таким образом, для сигнала, заданного массивом $A = \{a_n\}$, вейвлет-разложение производится по формулам:

$$a_{1,k} = \sum_{n\in\mathbf{Z}^d} \overline{h_n}\, a_{n+Ak}, \quad d^l_{1,k} = \sum_{n\in\mathbf{Z}^p} \overline{g^l_n}\, a_{n+Ak},\ l = 1,\ 2,\ \ldots,\ N-1. \tag{18}$$

Индекс Ak в сумме справа говорит о проведенной A-децимации, то есть выборке элементов с номерами только из решетки $A\mathbf{Z}^p$. Последние формулы можно записать в виде свертки. Для этого введем следующие коэффициенты: $h^*_n = \overline{h}_{-n}$, $g^{*l}_n = \overline{g^l_{-n}}$. Тогда

$$a_{1,k} = \sum_{n\in\mathbf{Z}^d} h^*_n a_{Ak-n}, \quad d^l_{1,k} = \sum_{n\in\mathbf{Z}^p} g^{*l}_n a_{Ak-n},\ l = 1,\ 2,\ \ldots,\ N-1. \tag{19}$$

Из данных формул следует, что вейвлет-разложение производится сопряженными фильтрами $\{h^*_n\}$, $\{g^{*l}_n\}$ с последующей A-адической децимацией (выбором только элементов с номерами из решетки $A\mathbf{Z}^p$).

Легко видеть, что восстановление массива $A = \{a_n\}$ по коэффициентам вейвлет-разложения $cA_1 = \{a_{1,k}\}$ и $cD^1_1 = \{d^1_{1,k}\}$, ..., $cD^{N-1}_1 = \{d^{N-1}_{1,k}\}$ производится следующим образом:

$$a_n = \sum_{k\in\mathbf{Z}^d} h_{n-Ak} a_{1,k} + \sum_{l=1}^{N-1} \sum_{k\in\mathbf{Z}^p} g^l_{n-Ak} d^l_{1,k}. \tag{20}$$

Последнюю формулу можно также записать в виде свертки, сделав обратную децимацию массивов $cA_1 = \{a_{1,k}\}$ и $cD_1 = (d^1_k, \ldots, \{d^{N-1}_k\}$:

$$\tilde{a}_{1,m} = \begin{cases} a_{1,k}, & \text{если } m = Ak, \\ 0, & \text{если } m \notin A\mathbf{Z}^p \end{cases}, \quad \tilde{d}^l_{1,m} = \begin{cases} d^l_{1,k}, & \text{если } m = Ak, \\ 0, & \text{если } m \notin A\mathbf{Z}^p. \end{cases}$$

Тогда формула (20) принимает вид

$$a_n = \sum_{m\in\mathbf{Z}^p} h_{n-m}\tilde{a}_{1,m} + \sum_{l=1}^{N-1}\sum_{m\in\mathbf{Z}^p} g^l_{n-m}\tilde{d}^l_{1,m}. \qquad (21)$$

2.16.5. Разложение и восстановление

Формулы вейвлет-разложения и восстановления (19)–(20) установлены только для ортогонального случая. Предположим, что разложение сигнала $\{a_k\}$ производится некоторыми (неортогональными) фильтрами $\{h^*_n\}$ и $\{g^{*l}_n\}$, $l = 1, 2, ..., N-1$ по формулам (19), и найдем другие фильтры $\{\tilde{h}_n\}$, $\{\tilde{g}^l_n\}$, $l = 1, 2, ..., N-1$, которые обеспечивают точное восстановление сигнала по формулам типа (21). Задачу удобно решить на уровне формальных степенных рядов. Определим передаточные функции заданных фильтров:

$$H_0(z) = \sum_{n\in\mathbf{Z}^p} h^*_n z^n, \quad H_l(z) = \sum_{n\in\mathbf{Z}^p} g^{*l}_n z^n, \quad l = 1, 2, \ldots, N-1.$$

Как известно, действие фильтров $\{h^*_n\}$ и $\{g^{*l}_n\}$ на сигнал $\{a_k\}$ заключается в умножении соответствующей функции $X(z)$ на $H_0(z)$ и $H_l(z)$:

$X_0(z) = H_0(z)X(z)$, $X_l = H_l(z)X(z)$, $l = 1, 2, ..., N-1$.

При вейвлет-разложении (19) необходимо еще провести A-децимацию, то есть выборку элементов с номерами из решетки $A\mathbf{Z}^p$. На уровне формальных степенных рядов для этого достаточно удалить все слагаемые со степенями, отличными от z^{Ak}: $X(z) \to X_A(z^A) = X_A(w)$.

Пусть $e_1, e_2, ..., e_p$ – стандартный базис пространства $\mathbf{R}^p$. Если A – матрица с целыми элементами и $Ae_s = A^1_s e_1 + ... + A^p_s e_p$ – образы базисных векторов, то определим следующие мономы:

$$w_1 = z^{Ae_1} = z_1^{A^1_1} z_2^{A^2_1} \cdots z_p^{A^p_1}, \ \ldots, \ w_p = z^{Ae_p} = z_1^{A^1_p} z_2^{A^2_p} \cdots z_p^{A^p_p}.$$

Такой набор переменных $w = (w_1, ..., w_p)$ будем обозначать символом $w = z^A$. Например, в случае матрицы $A = \begin{pmatrix} 1 & 2 \\ -2 & 1 \end{pmatrix}$, рассматриваемой в приведенных выше примерах, $z^A = (z_1 z_2^{-2}, z_1^2 z_2) = (w_1, w_2)$.

Отметим, что для любого целочисленного набора k имеет место равенство $z^{Ak} = w^k$, действительно:

$$z^{Ak} = z^{k_1 Ae_1 + ... + k_p Ae_p} = (z^{Ae_1})^{k_1} ... (z^{Ae_p})^{k_p} = w_1^{k_1} ... w_p^{k_p} = w^k.$$

После A-децимации должен остаться многочлен $X_A(w)$ только от этих переменных w. Коэффициенты полученного ряда $X_A(w)$ дают требуемую A-децимацию.

Для транспонированной матрицы A^t рассмотрим снова решетку $A^t\mathbf{Z}^p$, порожденную векторами $A^t e_s$, $s = 1, 2, ..., p$, и конечную коммутативную группу $\mathbf{Z}^p/A^t\mathbf{Z}^p$ порядка $N = |\det A|$. Пусть $d_s \in \mathbf{Z}^p$, $s = 0, 1, ..., N - 1$ – все элементы, представляющие классы группы $\mathbf{Z}^p/A^t\mathbf{Z}^p$. Считаем, что $d_0 = 0$, а остальные элементы – минимальной нормы в каждом классе. Тогда целочисленная решетка $\mathbf{Z}^p$ является объединением следующих классов: $A^t\mathbf{Z}^p$, $d_1 + A^t\mathbf{Z}^p$, ..., $d_{N-1} + A^t\mathbf{Z}^p$. Для каждого для $s = 0, 1, ..., N - 1$ рассмотрим вектор-столбцы координат векторов $\delta_s = (A^t)^{-1}d_s \in \mathbf{R}^p$:

$$(A^t)^{-1}d_s = (\delta_{1s}, ..., \delta_{ps})^t.$$

Определим числа $\rho_{rs} = e^{-i2\pi\delta rs}$ и для $s = 0, 1, ..., N - 1$ зададим вектор

$$\rho_s = e^{-i2\pi(A^t)^{-1}d_s} = (e^{-i2\pi\delta_{1s}}, \ldots, e^{-i2\pi\delta_{ps}}). \qquad (22)$$

Определим умножение векторов ρ_s и $z = (z_1, ..., z_p)$ покоординатно:

$$\rho_s z = e^{-i2\pi(A^t)^{-1}d_s} z = (e^{-i2\pi\delta_{1s}} z_1, \ldots, e^{-i2\pi\delta_{ps}} z_p).$$

Тогда если $n = (n_1, ..., n_p) \in \mathbf{Z}^p$, то для монома $z^n = z_1^{n_1} \cdots z_p^{n_p}$ имеем следующую формулу умножения:

$$(\rho_s z)^n = e^{-i2\pi(d_s, A^{-1})} z^n.$$

Действительно,

$$(\rho_s z)^n = (e^{-i2\pi\delta_{1s}} z_1)^{n_1} \cdots (e^{-i2\pi\delta_{ps}} z_p)^{n_p} = e^{-i2\pi(\delta_{1s}n_1 + \ldots + \delta_{ps}n_p)}(z_1^{n_1} \cdots z_p^{n_p}) =$$

$$= e^{-i2\pi((A^t)^{-1}d_s, n)} z^n = e^{-i2\pi(d_s, A^{-1}n)} z^n.$$

Отсюда, в частности, следует, что для элемента $An \in A\mathbf{Z}^p$ выполняются равенства

$$\rho_s^{An} = 1 \quad \text{и} \quad (\rho_s z)^{An} = z^{An}.$$

Проведем удаление степеней. Рассмотрим следующие степенные ряды:

$$X(z) = \sum_{n \in \mathbf{Z}^p} (a_{An} z^{An} + a_{d_1+An} z^{d_1+An} + \cdots + a_{d_{N-1}+An} z^{d_{N-1}+An}),$$

$$X(\rho_1 z) = \sum_{n \in \mathbf{Z}^p} (a_{An} z^{An} + a_{d_1+An} e^{-i2\pi(d_1, A^{-1}d_1)} z^{d_1+An} + \cdots + a_{d_{N-1}+An} e^{-i2\pi(d_1, A^{-1}d_{N-1})} z^{d_{N-1}+An}),$$

$$X(\rho_2 z) = \sum_{n \in \mathbf{Z}^p} (a_{An} z^{An} + a_{d_1+An} e^{-i2\pi(d_2, A^{-1}d_1)} z^{d_1+An} + \cdots + a_{d_{N-1}+An} e^{-i2\pi(d_2, A^{-1}d_{N-1})} z^{d_{N-1}+An}),$$

$\ldots$

$$X(\rho_{N-1} z) = \sum_{n \in \mathbf{Z}^p} (a_{An} z^{An} + a_{d_1+An} e^{-i2\pi(d_{N-1}, A^{-1}d_1)} z^{d_1+An} + \cdots + a_{d_{N-1}+An} e^{-i2\pi(d_{N-1}, A^{-1}d_{N-1})} z^{d_{N-1}+An}).$$

Проведем усреднение этих рядов:

$$\frac{1}{N}\sum_{s=0}^{N-1}X(\rho_s z)=\sum_{n\in\mathbf{Z}^p}a_{An}z^{An}+$$

$$+\frac{1}{N}\sum_{n\in\mathbf{Z}^p}a_{d_1+An}z^{d_1+An}(1+e^{-i2\pi(d_1,A^{-1}d_1)}+\ldots+e^{-i2\pi(d_{N-1},A^{-1}d_1)})+$$

$$\ldots$$

$$+\frac{1}{N}\sum_{n\in\mathbf{Z}^p}a_{d_{N-1}+An}z^{d_{N-1}+An}(1+e^{-i2\pi(d_1,A^{-1}d_{N-1})}+\ldots+e^{-i2\pi(d_{N-1},A^{-1}d_{N-1})})=$$

$$=\sum_{n\in\mathbf{Z}^p}a_{An}z^{An}=\sum_{n\in\mathbf{Z}^p}a_{An}w^n=X_A(w).$$

Покажем, что выражения в скобках равны нулю:

$$1+e^{-i2\pi(d_1,A^{-1}d_s)}+e^{-i2\pi(d_2,A^{-1}d_s)}+\ldots+e^{-i2\pi(d_{N-1},A^{-1}d_s)}=0. \qquad (23)$$

Для каждого фиксированного элемента d_s рассмотрим числа на единичной окружности:

$$1, e^{-i2\pi(d_1,A^{-1}d_s)}, e^{-i2\pi(d_2,A^{-1}d_s)}, \ldots, e^{-i2\pi(d_{N-1},A^{-1}d_s)}.$$

Легко видеть, что они зависят не от элементов $d_1, \ldots, d_{N-1}$, а от классов $A^t\mathbf{Z}^p$, $d_1+A^t\mathbf{Z}^p$, ..., $d_{N-1}+A^t\mathbf{Z}^p$, которые они представляют. Действительно, $\forall k\in\mathbf{Z}^p$ имеем:

$$e^{-i2\pi(d_1+A^tk,A^{-1}d_s)}=e^{-i2\pi(d_1,A^{-1}d_s)}e^{-i2\pi(A^tk,A^{-1}d_s)}=$$

$$=e^{-i2\pi(d_1,A^{-1}d_s)}e^{-i2\pi(k,d_s)}=e^{-i2\pi(d_1,A^{-1}d_s)}.$$

Данные числа образуют группу порядка N, поскольку классы $A^t\mathbf{Z}^p$, $d_1+A^t\mathbf{Z}^p$, ..., $d_{N-1}+A^t\mathbf{Z}^p$ образуют группу $\mathbf{Z}^p/A^t\mathbf{Z}^p$. Например,

$$e^{-i2\pi(d_1,A^{-1}d_s)}e^{-i2\pi(d_2,A^{-1}d_s)}=e^{-i2\pi(d_1+d_2,A^{-1}d_s)},$$

$d_1+d_2\in\mathbf{Z}^p/A^t\mathbf{Z}^p$. Поэтому указанный набор чисел представляет собой группу порядка N чисел на единичной окружности S^1. Следовательно, их сумма равна нулю. Таким образом, мы получили формулу децимации. Сформулируем полученный результат в виде теоремы.

Теорема 6. *Если $X(z)$ – формальный степенной ряд, соответствующий сигналу $\{a_k\}$, то коэффициенты следующего ряда*

$$X_A(w)=\frac{1}{N}(X(z)+X(\rho_1 z)+X(\rho_2 z)+\cdots+X(\rho_{N-1}z)) \qquad (24)$$

относительно переменных $w=z^A$ дают A-децимацию сигнала $\{a_k\}$.

Таким образом, на уровне степенных рядов вейвлет-разложение производится по формулам:

$$X_0(w)=\frac{1}{N}\sum_{s=0}^{N-1}H_0(\rho_s z)X(\rho_s z),$$

$$X_l(w)=\frac{1}{N}\sum_{s=0}^{N-1}H_l(\rho_s z)X(\rho_s z),\quad l=1,2,\ldots,N-1.$$

Образуем матрицу

$$H(z)=\frac{1}{\sqrt{N}}\begin{pmatrix} H_0(z) & H_0(\rho_1 z) & \cdots & H_0(\rho_{N-1}z)\\ H_1(z) & H_1(\rho_1 z) & \cdots & H_1(\rho_{N-1}z)\\ \cdots & \cdots & \cdots & \cdots\\ H_{N-1}(z) & H_{N-1}(\rho_1 z) & \cdots & H_{N-1}(\rho_{N-1}z)\end{pmatrix}. \qquad (25)$$

Тогда разложение производится по формуле

$$X(z)\to\begin{pmatrix} X_0(w)\\ X_1(w)\\ \cdots\\ X_{N-1}(w)\end{pmatrix}=\frac{1}{N}\begin{pmatrix} H_0(z) & H_0(\rho_1 z) & \cdots & H_0(\rho_{N-1}z)\\ H_1(z) & H_1(\rho_1 z) & \cdots & H_1(\rho_{N-1}z)\\ \cdots & \cdots & \cdots & \cdots\\ H_{N-1}(z) & H_{N-1}(\rho_1 z) & \cdots & H_{N-1}(\rho_{N-1}z)\end{pmatrix}\begin{pmatrix} X(z)\\ X(\rho_1 z)\\ \cdots\\ X(\rho_{N-1}z)\end{pmatrix}.$$

Из равенства

$$X(z)=\sum_{n\in\mathbf{Z}^p}(a_{An}z^{An}+a_{d_1+An}z^{d_1+An}+\cdots+a_{d_{N-1}+An}z^{d_{N-1}+An})=$$

$$=\sum_n(a_{An}z^{An}+z^{d_1}a_{d_1+An}z^{An}+\cdots+z^{d_{N-1}}a_{d_{N-1}+An}z^{An})=$$

$$=\sum_n a_{An}z^{An}+z^{d_1}\sum_n a_{d_1+An}z^{An}+\cdots+z^{d_{N-1}}\sum_n a_{d_{N-1}+An}z^{An}=$$

$$=B_0(w)+z^{d_1}B_1(w)+\cdots+z^{d_{N-1}}B_{N-1}(w),$$

где $w=z^A$, мы видим, что ряд $X(z)$ выражается через слагаемые $B_0(z^A)$, ..., $B_{N-1}(z^A)$, которые называются полифазными компонентами. Эти полифазные слагаемые $B_j(w)$ можно выделить по формуле (24), примененной к $\sqrt{2}$. Для матрицы $H(z)$ фильтров (25) определим полифазную матрицу $B(w)$ по формуле

$$B_{i,j}(w)=\frac{1}{N}\sum_{s=0}^{N-1}(\rho_s z)^{-d_j}H_i(\rho_s z).$$

Легко видеть, что сумма справа зависит от $w=z^A$. Обратное преобразование определяется формулой

$$H_i(z)=\sum_{j=0}^{N-1}B_{i,j}(z^A)z^{d_j},\quad z\in T^p=\mathbf{R}^p/2\pi\mathbf{Z}^p. \qquad (26)$$

Тогда последнее соотношение (24) может быть представлено в матричном виде:

$$H(z)=B(z^A)\begin{pmatrix}1 & 1 & \cdots & 1\\ z^{d_1} & (\rho_1 z)^{d_1} & \cdots & (\rho_{N-1}z)^{d_1}\\ \cdots & \cdots & \cdots & \cdots\\ z^{d_{N-1}} & (\rho_1 z)^{d_{N-1}} & \cdots & (\rho_{N-1}z)^{d_{N-1}}\end{pmatrix}=B(z^A)R(z), \qquad (27)$$

где $B(z^A)$ – определенная выше полифазная матрица. Отметим, что матрица $R(z)/\sqrt{N}$ является унитарной на торе $T^p = S^1\times\ldots\times S^1$. Действительно, для $z = (z_1, ..., z_p) \in T^p$, используя равенство $\bar{z} = z^{-1} = (z_1^{-1}, ..., z_p^{-1})$, получаем элементы произведения $NR(z)R^*(z)$:

$$(\rho_0 z)^{d_i}(\overline{\rho_0 z})^{d_j}+(\rho_1 z)^{d_i}(\overline{\rho_1 z})^{d_j}+\ldots+(\rho_{N-1}z)^{d_i}(\overline{\rho_{N-1}z})^{d_j}=$$
$$=z^{d_i-d_j}+(\rho_1 z)^{d_i-d_j}+\ldots+(\rho_{N-1}z)^{d_i-d_j}=z^{d_i-d_j}(1+\rho_1^{d_i-d_j}+\ldots+\rho_{N-1}^{d_i-d_j})=N\delta_{ij}.$$

Последнее следует из того, что если $c = d_i - d_j$, то $\rho_s^{d_i-d_j}=e^{-i2\pi(d_s,A^{-1}c)}$ и числа $1, e^{-i2\pi(d_1,A^{-1}c)}, e^{-i2\pi(d_2,A^{-1}c)},\ldots, e^{-i2\pi(d_{N-1},A^{-1}c)}$ образуют при $i\neq j$ конечную нетривиальную группу чисел на единичной окружности. Поэтому

$$1+e^{-i2\pi(d_1,A^{-1}c)}+e^{-i2\pi(d_2,A^{-1}c)}+\ldots+e^{-i2\pi(d_{N-1},A^{-1}c)}=0.$$

В выражении (27) матрица $B(w)$ является уже произвольной невырожденной матрицей с полиномиальными элементами. Специфика матрицы $H(z)$ отражена теперь в матрице $R(z)$. Задавая $B(w)$, мы можем построить матрицу $H(z)$ и вместе с ней частотные функции $H_1(\omega), ..., H_{N-1}(\omega)$ вейвлетов, следовательно, и сами вейвлеты $\psi^1(x), ..., \psi^{N-1}(x)$.

Восстановление сигнала производим другими фильтрами $G_l(z)=\sum_n \tilde{g}_n^l z^n$, $l = 0, 1, 2, ..., N-1$, по формуле (21). На уровне степенных рядов обратная децимация эквивалентна замене степенного ряда $X(w)$ по степеням переменной w на ряд $X(z^A)$ по степеням переменной z, $X(w)\to X(z^A)$. Поэтому восстановление на уровне степенных рядов делается по формуле

$$\{X_0(w), X_1(w), \ldots, X_{N-1}(w)\}\to\{X_0(z^A), X_1(z^A), \ldots, X_{N-1}(z^A)\}\to$$
$$\to\sum_{l=0}^{N-1}G_l(z)X_l(z^A)=\frac{1}{N}\sum_{l=0}^{N-1}G_l(z)\sum_{s=0}^{N-1}H_l(\rho_s z)X(\rho_s z)=$$
$$=\frac{1}{N}\sum_{s=0}^{N-1}\left(\sum_{l=0}^{N-1}G_l(z)H_l(\rho_s z)\right)X(\rho_s z)=X(z).$$

Для точного восстановления достаточно, чтобы выполнялись равенства

$$\sum_{l=0}^{N-1} G_l(z)H_l(z) = N \quad (\text{при } s=0),$$

$$\sum_{l=0}^{N-1} G_l(z)H_l(\rho_s z) = 0 \quad \text{для } s = 1, 2, \ldots, N-1.$$

Данные условия точного восстановления удобно выразить через следующие матрицы фильтров разложения и восстановления в виде:

$$\begin{pmatrix} G_0(z) & G_1(z) & \cdots & G_{N-1}(z) \\ G_0(\rho z) & G_1(\rho z) & \cdots & G_{N-1}(\rho z) \\ \cdots & \cdots & \cdots & \cdots \\ G_0(\rho^{N-1}z) & G_1(\rho^{N-1}z) & \cdots & G_{N-1}(\rho^{N-1}z) \end{pmatrix} \begin{pmatrix} H_0(z) & H_0(\rho z) & \cdots & H_0(\rho^{N-1}z) \\ H_1(z) & H_1(\rho z) & \cdots & H_1(\rho^{N-1}z) \\ \cdots & \cdots & \cdots & \cdots \\ H_{N-1}(z) & H_{N-1}(\rho z) & \cdots & H_{N-1}(\rho^{N-1}z) \end{pmatrix} = N.$$

Мы получили следующий факт.

Теорема 7. *Если матрица $H(z)$ фильтров разложения невырождена при $z \in T^p$, то возможно точное восстановление сигнала фильтрами $G_l(z)$, $l = 0, 1, 2, ..., N-1$, матрица которых*

$$G(z) = \frac{1}{\sqrt{N}} \begin{pmatrix} G_0(z) & G_0(\rho z) & \cdots & G_0(\rho^{N-1}z) \\ G_1(z) & G_1(\rho z) & \cdots & G_1(\rho^{N-1}z) \\ \cdots & \cdots & \cdots & \cdots \\ G_{N-1}(z) & G_{N-1}(\rho z) & \cdots & G_{N-1}(\rho^{N-1}z) \end{pmatrix} \qquad (28)$$

является транспонированной к обратной матрице $H(z)$ исходных фильтров.

Замечание. В ортогональном случае $H(z)$ – унитарная (на торе, $z \in T^p = S^1 \times \ldots \times S^1$) матрица, а $G(z)$ – комплексно-сопряженная к $H(z)$, $G(z) = \overline{H(z)}$. В общем случае для нахождения фильтров восстановления необходимо найти матрицу, обратную к $H(z)$.

2.16.6. Построение вейвлетов с матрицей масштабирования

В разделе 2.15 предложен метод построения большой серии ортогональных вейвлетов с компактным носителем в одномерном случае. Распространим этот метод на случай многомерных вейвлетов с матричным коэффициентом масштабирования с полиномиальными частотными функциями и с вещественными фильтрами. Схема построения следующая.

Сначала находится унитарная матрица

$$H(z)=\begin{pmatrix} H_0(z) & H_0(\rho_1 z) & \cdots & H_0(\rho_{N-1}z) \\ H_1(z) & H_1(\rho_1 z) & \cdots & H_1(\rho_{N-1}z) \\ \cdots & \cdots & \cdots & \cdots \\ H_{N-1}(z) & H_{N-1}(\rho\, z) & \cdots & H_{N-1}(\rho_{N-1}z) \end{pmatrix} \tag{29}$$

частотных полиномиальных функций, где $H_0(z)$ – частотная функция (7) для масштабирующей функции $\varphi(x)$ и $H_1(z)$, ..., $H_{N-1}(z)$ – для вейвлет-функций, где $z=(z_1,\ldots,z_p)=(e^{-i\omega_1},\ldots,e^{-i\omega_p})$, $\rho_s=e^{-i2\pi(A^t)^{-1}d_s}=(e^{-i2\pi\delta_{1s}},\ldots,e^{-i2\pi\delta_{ps}})$, d_0, d_1, ..., d_{N-1} – элементы из $\mathbf{Z}^p$, представляющие все классы факторгруппы $\mathbf{Z}^p/A^t\mathbf{Z}^p$ и $(A^t)^{-1}d_s=(\delta_{1s},\ldots,\delta_{ps})$.

Затем находится масштабирующая функция $\varphi(x)$ по формуле (12) и проверяется выполнение достаточного условия ортонормированности (теорема 4) системы функций $\varphi_{0,n}(x)=\varphi(x-n)$, $n\in\mathbf{Z}^p$. На последнем этапе находятся вейвлет-функции $\psi^1(x)$, ... , $\psi^{N-1}(x)$ по формулам $\psi^l(x)=\sqrt{N}\sum_{n\in\mathbf{Z}^p} g_n^l\varphi(Ax-n)$, $l=1, 2, \ldots N-1$, где g_n^l – коэффициенты многочленов $H_1(z)$, ..., $H_{N-1}(z)$:

$$H_l(\omega)=\frac{1}{\sqrt{N}}\sum_{n\in\mathbf{Z}^p} g_n^l e^{-i(n,\omega)},\quad l=1,2,\ldots,N-1.$$

Матрица (29) имеет очень специальный вид. Однако ее можно разложить (см. раздел 2.15) в произведение двух унитарных (на торе $|z_1|=1$, ..., $|z_p|=1$) матриц

$$H(z)=B(z^A)R(z), \tag{30}$$

где $B(z^A)$ – унитарная матрица, элементами которой являются многочлены от z^A, а матрица $R(z)$ имеет вид

$$R(z)=\frac{1}{\sqrt{N}}\begin{pmatrix} 1 & 1 & \cdots & 1 \\ z^{d_1} & (\rho_1 z)^{d_1} & \cdots & (\rho_{N-1}z)^{d_1} \\ \cdots & \cdots & \cdots & \cdots \\ z^{d_{N-1}} & (\rho_1 z)^{d_{N-1}} & \cdots & (\rho_{N-1}z)^{d_{N-1}} \end{pmatrix}. \tag{31}$$

Напомним определение мономов z^A. Пусть e_1, ..., e_p – стандартный базис пространства $\mathbf{R}^p$. Если A – матрица с целочисленными элементами и $Ae_1=A_1^1e_1+\ldots+A_1^pe_p$, ..., $Ae_p=A_p^1e_1+\ldots+A_p^pe_p$, то определим следующие мономы:

$$w_1=z^{Ae_1}=z_1^{A_1^1}\cdots z_p^{A_1^p},\ \ldots,\ w_p=z^{Ae_p}=z_1^{A_p^1}\cdots z_p^{A_p^p}.$$

Такой набор переменных $w = (w_1, ..., w_p)$ будем обозначать символом $w = z^A$. Отметим, что для любого целочисленного набора $k = (k_1, ..., k_p)$ имеет место равенство $z^{Ak} = w^k$, действительно,

$$z^{Ak} = z^{k_1 Ae_1 + \cdots + k_p Ae_p} = (z^{Ae_1})^{k_1} \cdots (z^{Ae_p})^{k_p} = w_1^{k_1} \cdots w_p^{k_p} = w^k.$$

Матрица $B(w)$ в выражении (30) называется полифазной. Она является уже произвольной унитарной, при $|w_1| = 1, ..., |w_p| = 1$, матрицей с полиномиальными элементами. Отметим, что матрица $R(z)$ является унитарной на торе $T^p = S^1 \times ... \times S^1$, а именно имеет место равенство $R(z)R^*(z) = E$.

Задавая унитарную полифазную матрицу $B(w)$, мы можем построить по формуле (30) матрицу частотных функций $H(z)$ и вместе с ней частотные функции $H_0(z)$, $H_1(z)$, ..., $H_{N-1}(z)$ вейвлетов, следовательно, и сами вейвлеты $\varphi(x)$, $\psi^1(x)$, ..., $\psi^{N-1}(x)$.

Схема построения полифазной матрицы

Приведем простую схему построения унитарной (при $|w_1| = 1, ..., |w_p| = 1$) матрицы $B(w)$, элементы которой являются многочленами с вещественными коэффициентами.

Выберем произвольную ортогональную матрицу B_0 порядка $N > 1$, $B_0 = \{b_{kj},\ k, j = 0, 1, ... N - 1\}$. Умножим ее на диагональную унитарную матрицу вида $D_u(w) = diag(w^{u_0}, w^{u_1}, \ldots, w^{u_{N-1}})$, где $u_0, ..., u_{N-1}$ – целочисленные векторы. Затем умножим на ортогональную матрицу $C_0 = \{c_{kj},\ k, j = 0, 1, ... N - 1\}$. В результате получим унитарную матрицу

$$B(w) = D_u(w) \cdot B_0 \cdot C_0, \tag{32}$$

элементы которой, $B_{kj}(w) = \sum_{s=0}^{N-1} b_{ks} c_{sj} w^{p_k}$, являются многочленами по переменной w с вещественными коэффициентами.

Теперь подставим $w = z^A$, где $z = (e^{-i\omega_1}, \ldots, e^{-i\omega_p})$. Получаем унитарную матрицу $B(z^A)$ с полиномиальными элементами с вещественными коэффициентами. Умножим ее на унитарную матрицу $R(z)$. Тогда, согласно формуле (30), мы получим унитарную (на торе) матрицу $H(z)$ частотных полиномиальных функций $H_0(z)$, $H_1(z)$, ..., $H_{N-1}(z)$. Таким образом,

$$H(z) = D_p(z^A) \cdot B_0 \cdot C_0 \cdot R(z). \tag{33}$$

Для того чтобы полученные функции были частотными функциями ортогональных вейвлетов, необходимо, чтобы сумма коэффициентов для $H_0(z)$ была равна единице, а суммы коэффициентов для остальных функций $H_1(z)$, ..., $H_{N-1}(z)$ были равны нулю. Учитывая, что

умножение на диагональную матрицу $D_p(z^A)$ не влияет на сумму коэффициентов, получаем:

$$\frac{1}{\sqrt{N}}\sum_{s,j=0}^{N-1} b_{0s}c_{sj} = \frac{1}{\sqrt{N}}\sum_{s=0}^{N-1} b_{0s}\left(\sum_{j=0}^{N-1} c_{sj}\right) = 1.$$

$$\frac{1}{\sqrt{N}}\sum_{s,j=0}^{N-1} b_{ks}c_{sj} = \frac{1}{\sqrt{N}}\sum_{s=0}^{N-1} b_{ks}\left(\sum_{j=0}^{N-1} c_{sj}\right) = 0,\ k = 1, 2, \ldots,\ N-1.$$

Эти равенства можно записать в матричном виде:

$$\begin{pmatrix} b_{00} & b_{01} & \cdots & b_{0,N-1} \\ b_{10} & b_{11} & \cdots & b_{1,N-1} \\ \cdots & \cdots & \cdots & \cdots \\ b_{N-1,0} & b_{N-1,1} & \cdots & b_{N-1,N-1} \end{pmatrix} \begin{pmatrix} c_{00} + c_{01} + \cdots + c_{0,N-1} \\ c_{10} + c_{11} + \cdots + c_{1,N-1} \\ \cdots \\ c_{N-1,0} + c_{N-1,1} + \cdots + c_{N-1,N-1} \end{pmatrix} = \begin{pmatrix} \sqrt{N} \\ 0 \\ \cdots \\ 0 \end{pmatrix}. \qquad (34)$$

Выбирая различные ортогональные матрицы B_0 и C_0, удовлетворяющие указанному равенству (34), получаем по формуле (33) матрицу частотных функций.

Для получения достаточно простого класса ортогональных вейвлетов с компактным носителем и с матричным коэффициентом масштабирования возьмем в качестве ортогональной матрицы B_0 следующую матрицу:

$$B_0 = \begin{pmatrix} 1/\sqrt{N} & 1/\sqrt{N} & 1/\sqrt{N} & 1/\sqrt{N} & \cdots & 1/\sqrt{N} \\ 1/\sqrt{2} & -1/\sqrt{2} & 0 & 0 & \cdots & 0 \\ 1/\sqrt{6} & 1/\sqrt{6} & -2/\sqrt{6} & 0 & \cdots & 0 \\ 1/\sqrt{12} & 1/\sqrt{12} & 1/\sqrt{12} & -3/\sqrt{12} & \cdots & 0 \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ 1/\sqrt{N(N-1)} & 1/\sqrt{N(N-1} & 1/\sqrt{N(N-1} & 1/\sqrt{N(N-1} & \cdots & -(N-1)/\sqrt{N(N-1} \end{pmatrix}.$$

В разделе 2.15 показано, что тогда в качестве матрицы C_0 можно взять матрицу вида $C_0 = B_0^{-1} M^T B_0$, где M – ортогональная матрица вида

$$M = \begin{pmatrix} 1 & 0 & 0 & \cdots & 0 \\ 0 & m_1^1 & m_2^1 & \cdots & m_{N-1}^1 \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ 0 & m_1^{N-1} & m_2^{N-1} & \cdots & m_{N-1}^{N-1} \end{pmatrix}. \qquad (35)$$

Получаем:

$$H_M(z) = D_p(z^N) B_0 M_e^T R(z) = D_p(z^N) M^T B_0 R(z). \qquad (36)$$

Формула (36) дает прямой способ построения частотных функций $H_0(z)$, $H_1(z)$, ..., $H_{N-1}(z)$ ортогональных вейвлетов с компактным носителем. Ортогональную матрицу M вида (35) и вектор степеней $u = (u_0, u_1, ..., u_{N-1})$ можно задавать произвольно. При построении частотной функции по формуле (12) нужно учитывать достаточное условие ортогональности, установленное теоремой 1.

Замечание. Для построения полифазной матрицы при помощи матриц $D_p(w)$, B_0 и C_0 можно также использовать формулу $B(w) = B_0 \cdot D_u(w) \cdot C_0$. В этом случае мы получаем матрицу частотных функций в виде: $H_M(z) = B_0 D_u(z^N) B_0^T M^T B_0 R(z)$. Это приводит к другой серии ортогональных вейвлетов с компактным носителем.

Примеры построения матрицы частотных функций

В данном разделе мы покажем на примерах эффективность изложенной выше схемы построения матрицы частотных функций в виде (36). Для масштабирующей функции Хаара мы построим частотные функции вейвлетов двумя способами. Сначала – самый простой случай, а затем – чуть сложнее.

Пример 5. Фрактальные вейвлеты Хаара. Пусть $A = \begin{pmatrix} 1 & 2 \\ -2 & 1 \end{pmatrix}$. Тогда $N = \det A = 5$. Эта матрица является поворотом на угол $\alpha = \operatorname{arctg} 2$ по часовой стрелке и растяжением с коэффициентом $\sqrt{5}$. Фактор-группа $\mathbf{Z}^2/A^t\mathbf{Z}^2$ имеет 5 классов, представленных элементами (вектор-столбцами):

$d_0 = (0, 0)^t$, $d_1 = e_1 = (1, 0)^t$, $d_2 = -e_1 = (-1, 0)^t$,
$d_3 = e_2 = (0, 1)^t$, $d_4 = -e_2 = (0, -1)^t$.

Найдем матрицу $R(z)$. Для каждого $s = 0, 1, ..., 4$ рассмотрим вектор-столбцы координат векторов, $(A^t)^{-1} d_s = (\delta_{1s}, \delta_{2s})^t$. Для выбранной матрицы A имеем $(A^t)^{-1} = \frac{1}{5}\begin{pmatrix} 1 & 2 \\ -2 & 1 \end{pmatrix}$. Поэтому векторы $\rho_s = e^{-i2\pi(A^t)^{-1}d_s} = (e^{-i2\pi\delta_{1s}}, e^{-i2\pi\delta_{2s}})$ в нашем случае имеют выражения:

$$\rho_0 = e^{-i2\pi(A^t)^{-1}d_0} = (1,1), \quad \rho_1 = e^{-i2\pi(A^t)^{-1}d_1} = (e^{-i2\pi/5}, e^{i4\pi/5}),$$
$$\rho_2 = e^{-i2\pi(A^t)^{-1}d_2} = (e^{i2\pi/5}, e^{-i4\pi/5}), \quad \rho_3 = e^{-i2\pi(A^t)^{-1}d_3} = (e^{-i4\pi/5}, e^{-i2\pi/5}),$$
$$\rho_4 = e^{-i2\pi(A^t)^{-1}d_4} = (e^{i4\pi/5}, e^{i2\pi/5}).$$

Умножение векторов ρ_s и $z = (z_1, z_2)$ определяется покоординатно, $\rho_s z = e^{-i2\pi(A^t)^{-1}d_s} z = (e^{-i2\pi\delta_{1s}} z_1, e^{-i2\pi\delta_{2s}} z_2)$. Тогда если $n = (n_1, n_2) \in \mathbf{Z}^2$,

то для монома $z^n = z_1^{n_1} z_2^{n_2}$ имеем следующую формулу умножения: $(\rho_s z)^n = e^{-i2\pi(d_s, A^{-1}n)} z^n$. В случае выбранной матрицы $z^A = (z_1 z_2^{-2}, z_1^2 z_2) = (w_1, w_2)$. Тогда матрица $R(z)$ принимает вид

$$R(z) = \frac{1}{\sqrt{5}} \begin{pmatrix} 1 & 1 & 1 & 1 & 1 \\ z_1 & e^{-i2\pi/5} z_1 & e^{i2\pi/5} z_1 & e^{-i4\pi/5} z_1 & e^{i4\pi/5} z_1 \\ z_1^{-1} & e^{i2\pi/5} z_1^{-1} & e^{-i2\pi/5} z_1^{-1} & e^{i4\pi/5} z_1^{-1} & e^{-i4\pi/5} z_1^{-1} \\ z_2 & e^{i4\pi/5} z_2 & e^{-i4\pi/5} z_2 & e^{-i2\pi/5} z_2 & e^{i2\pi/5} z_2 \\ z_2^{-1} & e^{-i4\pi/5} z_2^{-1} & e^{i4\pi/5} z_2^{-1} & e^{-i2\pi/5} z_2^{-1} & e^{i2\pi/5} z_2^{-1} \end{pmatrix}. \tag{37}$$

Диагональную матрицу $D(w)$ возьмем, для этого примера, в виде $D(w) = \mathrm{diag}\{1, 1, 1, 1, 1\} = 1$. Выберем матрицу B_0 пятого порядка вида

$$B_0 = \begin{pmatrix} 1/\sqrt{5} & 1/\sqrt{5} & 1/\sqrt{5} & 1/\sqrt{5} & 1/\sqrt{5} \\ 1/\sqrt{2} & -1/\sqrt{2} & 0 & 0 & 0 \\ 1/\sqrt{6} & 1/\sqrt{6} & -2/\sqrt{6} & 0 & 0 \\ 1/\sqrt{12} & 1/\sqrt{12} & 1/\sqrt{12} & -3/\sqrt{12} & 0 \\ 1/\sqrt{20} & 1/\sqrt{20} & 1/\sqrt{20} & 1/\sqrt{20} & -4/\sqrt{20} \end{pmatrix}.$$

Будем также считать, что $M(t) = 1$. Получаем:

$$H(t, z) = B_0 R(z) = B_0 R(z), \tag{38}$$

где матрица $R(z)$ имеет вид (37). Перемножая эти матрицы, находим частотные функции $H_0(z)$, $H_1(z)$, $H_2(z)$, $H_3(z)$, $H_4(z)$:

$$H_0(\omega_1, \omega_2) = \frac{1}{5}(1 + e^{-i\omega_1} + e^{i\omega_1} + e^{-i\omega_2} + e^{i\omega_2}),$$

$$H_1(\omega_1, \omega_2) = \frac{1}{\sqrt{10}}(1 - e^{-i\omega_1}),$$

$$H_2(\omega_1, \omega_2) = \frac{1}{\sqrt{30}}(1 + e^{-i\omega_1} - 2e^{i\omega_1}),$$

$$H_3(\omega_1, \omega_2) = \frac{1}{\sqrt{60}}(1 + e^{-i\omega_1} + e^{i\omega_1} - 3e^{-i\omega_2}),$$

$$H_4(\omega_1, \omega_2) = \frac{1}{10}(1 + e^{-i\omega_1} + e^{i\omega_1} + e^{-i\omega_2} - 4e^{i\omega_2}),$$

здесь мы сделали замены $z_1 = e^{-i\omega_1}$ и $z_2 = e^{-i\omega_2}$. Тогда имеется всего 5 ненулевых коэффициентов масштабирующего уравнения:

$$h_{(-1,0)} = h_{(0,-1)} = h_{(0,0)} = h_{(1,0)} = h_{(0,1)} = 1/\sqrt{5}.$$

Поэтому масштабирующее уравнение имеет вид

$$\varphi(x_1, x_2) = \varphi(x_1 + 2x_2, -2x_1 + x_2) + \varphi(x_1 + 2x_2 + 1, -2x_1 + x_2) + \\ + \varphi(x_1 + 2x_2 - 1, -2x_1 + x_2) + \varphi(x_1 + 2x_2, -2x_1 + x_2 + 1) + \\ + \varphi(x_1 + 2x_2, -2x_1 + x_2 - 1)$$

и соответствует следующим векторам решетки $D = \{0, -e_1, e_1, -e_2, e_2\}$. Такая масштабирующая функция типа Хаара $\varphi(x, y)$ является характеристической функцией компактного множества $\mathbf{T} \subset \mathbf{R}^2$, которое удовлетворяет уравнению $A\mathbf{T} = \bigcup_{s=0}^{5} (\mathbf{T} + d_s)$, имеет меру 1, замощает $\mathbf{R}^2$ при целочисленных сдвигах и имеет фрактальную границу, рис. 2.16.6.

Вейвлет, соответствующий первой функции $H_1(\omega_1, \omega_2)$, имеет вид:

$$\psi^1(x_1, x_2) = \frac{1}{\sqrt{2}}\varphi(x_1 + 2x_2, -2x_1 + x_2) - \frac{1}{\sqrt{2}}\varphi(x_1 + 2x_2 - 1, -2x_1 + x_2).$$

Он принимает значение $1/\sqrt{2}$ на центральной $A^{-1}\mathbf{T}$ и $-1/\sqrt{2}$ – на правой верхней части $A^{-1}(\mathbf{T} + e_1)$ множества $\mathbf{T}$, рис. 2.16.10.

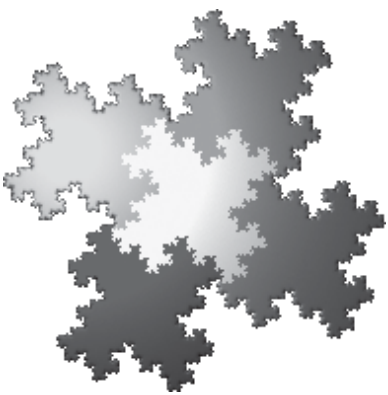


Рис. 2.16.10. Множество $\mathbf{T}$ для $D = \{0, -e_1, e_1, -e_2, e_2\}$

Пример 6. Пусть снова $A = \begin{pmatrix} 1 & 2 \\ -2 & 1 \end{pmatrix}$, $N = \det A = 5$. В этом примере диагональную матрицу $D(w)$ возьмем в виде $D(w) = \mathrm{diag}\{1, w_1, 1, w_2, 1\}$. Выберем матрицу B_0 пятого порядка вида, как в примере 5. В качестве $M(t)$ возьмем вращение вокруг оси Ox, например в плоскости e_1, e_2:

$$M(t)=\begin{pmatrix}1 & 0 & 0 & 0 & 0\\ 0 & \cos t & \sin t & 0 & 0\\ 0 & -\sin t & \cos t & 0 & 0\\ 0 & 0 & 0 & 1 & 0\\ 0 & 0 & 0 & 0 & 1\end{pmatrix}.$$

Получаем:

$$H(t, z) = D(z^A)B_0M_e^T(t)R(z) = D(z^A)M^T(t)B_0R(z),$$

где $D(z^A) = \mathrm{diag}\{1, z_1z_2^{-2}, 1, z_1^2z_2, 1\}$ и матрица $R(z)$ имеет вид (37). Перемножая все эти матрицы, находим частотные функции $H_0(z)$, $H_1(z)$, $H_2(z)$, $H_3(z)$, $H_4(z)$:

$$H_0(\omega_1,\omega_2)=\frac{1}{5}(1+e^{-i\omega_1}+e^{i\omega_1}+e^{-i\omega_2}+e^{i\omega_2}),$$

$$H_1(\omega_1,\omega_2)=\frac{1}{3\sqrt{10}}(-2\sqrt{3}\sin te^{i2\omega_2}+(3\cos t+\sqrt{3}\sin t)e^{-i\omega_1}e^{i2\omega_2}+$$

$$+(-3\cos t+\sqrt{3}\sin t)e^{-i2\omega_1}e^{i2\omega_2}),$$

$$H_1(\omega_1,\omega_2)=\frac{1}{3\sqrt{10}}(-2\sqrt{3}\sin te^{i2\omega_2}+(3\cos t+\sqrt{3}\sin t)e^{-i(\omega_1-2\omega_2)}+$$

$$+(-3\cos t+\sqrt{3}\sin t)e^{-i(2\omega_1-2\omega_2)}),$$

$$H_2(\omega_1,\omega_2)=\frac{1}{3\sqrt{10}}(-2\sqrt{3}\cos te^{i\omega_1}+(-3\sin t+\sqrt{3}\cos t)+(3\sin t+\sqrt{3}\cos t)e^{-i\omega_1}),$$

$$H_3(\omega_1,\omega_2)=\frac{1}{2\sqrt{15}}(e^{-i\omega_1}e^{-i\omega_2}+e^{-i2\omega_1}e^{-i\omega_2}+e^{-i3\omega_1}e^{-i\omega_2}-3e^{-i2\omega_1}e^{-i2\omega_2}),$$

$$H_3(\omega_1,\omega_2)=\frac{1}{2\sqrt{15}}(e^{-i(\omega_1+\omega_2)}+e^{-i(2\omega_1+\omega_2)}+e^{-i(3\omega_1+\omega_2)}-3e^{-i(2\omega_1+2\omega_2)}),$$

$$H_4(\omega_1,\omega_2)=\frac{1}{10}(e^{i\omega_1}-4e^{i\omega_2}+1+e^{-i\omega_1}+e^{-i\omega_2}).$$

Получилась такая же масштабирующая функция Хаара, как и в примере 5, но вейвлеты отличаются от тех, что получены в примере 5.

2.17. Гармонические вейвлеты

В данном разделе мы рассмотрим так называемые «гармонические вейвлеты», введенные в работе Ньюланда [Newl] в 1993 г. Они являются комплексными аналогами вейвлетов Шеннона–Котельникова. На их основе можно определить периодические гармонические вейв-

леты для вейвлет-разложения непрерывных периодических функций $f(x)$ с периодом $T = 1$. Если функция задана своей оцифровкой $y_r = f(r/N)$, $r = 0, 1, ..., N - 1$, для некоторого N, то в этом случае мы получаем конечное число коэффициентов вейвлет-разложения, которые могут быть получены из дискретного преобразования Фурье массива $\{f(r/N)\}$.

2.17.1. Гармонические вейвлеты на R

Рассмотрим две вещественные функции $\psi_e(x)$ и $\psi_o(x)$, преобразования Фурье которых имеют вид, соответственно:

$$\Psi_e(\omega) = \begin{cases} 1/2 & \text{для } \omega \in [-4\pi,-2\pi) \cup [2\pi,4\pi), \\ 0 & \text{для остальных } \omega, \end{cases}$$

$$\Psi_o(\omega) = \begin{cases} i/2 & \text{для } \omega \in [-4\pi,-2\pi), \\ -i/2 & \text{для } \omega \in [2\pi,4\pi), \\ 0 & \text{для остальных } x\ \omega. \end{cases}$$

Тогда

$$\psi_e(x) = \frac{\sin 4\pi x - \sin 2\pi x}{2\pi x} \quad \text{и} \quad \psi_o(x) = -\frac{\cos 4\pi x - \cos 2\pi x}{2\pi x}.$$

Образуем из этих функций $\psi_e(x)$ и $\psi_o(x)$ следующую комплексную функцию:

$$\psi(x) = \psi_e(x) + i\psi_o(x), \quad \psi(x) = \frac{e^{i4\pi x} - e^{i2\pi x}}{i2\pi x}, \tag{1}$$

которую будем называть *гармоническим вейвлетом*. Ее преобразование Фурье:

$$\Psi(\omega) = \begin{cases} 1 & \text{для } \omega \in [2\pi,4\pi), \\ 0 & \text{для остальных } \omega. \end{cases}$$

В соответствии с общей схемой рассмотрим функции $\psi_{j,k}(x) = \sqrt{2^j}\psi(2^j x - k)$. В явном виде:

$$\psi_{jk}(x) = \sqrt{2^j}\,\frac{e^{i4\pi(2^j x-k)} - e^{i2\pi(2^j x-k)}}{i2\pi(2^j x-k)}.$$

$$\hat{\psi}_{jk}(\omega) = \frac{1}{\sqrt{2^j}} e^{-i\omega k/2^j} \hat{\psi}\left(\frac{\omega}{2^j}\right) = \frac{1}{\sqrt{2^j}} e^{-i\omega k/2^j} \Psi\left(\frac{\omega}{2^j}\right).$$

Здесь $j, k \in \mathbf{N} \cup \{0\}$.

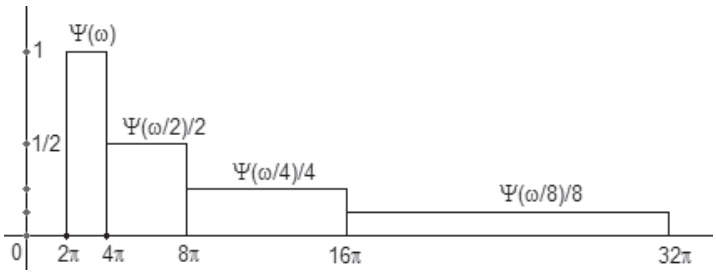


Рис. 2.17.1. Графики образов Фурье функций $\psi(x)$, $\psi(2x)$, $\psi(4x)$ и $\psi(8x)$

Аналогично определим масштабирующую функцию $\varphi(x)$. Сначала рассмотрим две функции в частотной и, соответственно, в реальной областях:

$$\Phi_e(\omega)=\begin{cases}1/2 & \text{для } \omega\in[-2\pi,2\pi),\\ 0 & \text{для остальных } \omega,\end{cases}\qquad \varphi_e(x)=\frac{\sin 2\pi x}{2\pi x},$$

$$\Phi_o(\omega)=\begin{cases}i/2 & \text{для } \omega\in[-2\pi,0)\\ -i/2 & \text{для } \omega\in(0,2\pi],\\ 0 & \text{для остальных } \omega,\end{cases}\qquad \varphi_o(x)=-\frac{\cos 2\pi x-1}{2\pi x}.$$

Составим из них комплексную функцию $\Phi(\omega)=\Phi_e(\omega)+i\Phi_o(\omega)$. Она является преобразованием Фурье функции $\varphi(x)=\varphi_e(x)+i\varphi_o(x)$. Легко видеть, что

$$\Phi(\omega)=\begin{cases}1 & \text{для } \omega\in[0,2\pi)\\ 0 & \text{для остальных } \omega\end{cases}\quad\text{и}\quad \varphi(x)=\frac{e^{i2\pi x}-1}{i2\pi x}. \qquad (2)$$

Данная функция отличается от вейвлетов Шеннона–Котельникова сдвигом на π в частотной области.

Для построения ортогонального кратномасштабного разложения $L^2(\mathbf{R})$ нам нужно определить начальное пространство V_0. Очевидно, что функции $\{\varphi(x-n)\}_{n\in\mathbf{Z}}$ образуют ортонормированную систему – это сразу следует из вида $\Phi(\omega)$ и общего условия ортонормированности (3) из раздела 2.3.1. Однако $\Phi(\omega)$ не является непрерывной в окрестности $\omega=0$, следовательно, свойство $\overline{\bigcup_{j\in\mathbf{Z}}V_j}=L^2(\mathbf{R})$ кратномасштабного разложения не будет обеспечено, если мы в качестве V_0 возьмем пространство, порожденное сдвигами $\{\varphi(x-n)\}_{n\in\mathbf{Z}}$ масштабирующей функции $\varphi(x)$. Чтобы такой неприятности не было,

в случае гармонических вейвлетов в качестве V_0 нужно брать пространство, порожденное функциями $\varphi(x-n)$ и $\overline{\varphi(x-k)}$:

$$V_0 = \mathbf{R}\{\varphi(x-n), \overline{\varphi(x-k)},\ n,k \in \mathbf{Z}\}.$$

Отметим, что сдвиги $\varphi(x-n)$ и $\overline{\varphi(x-k)}$ образуют ортонормированную систему функций. Ортонормированность системы $\{\overline{\varphi(x-k)}, k \in \mathbf{Z}\}$ показывается совершенно аналогично. Легко видеть также, что функции $\varphi(x-n)$ и $\overline{\varphi(x-k)}$ являются ортогональными. Действительно, достаточно доказать ортогональность $\varphi(x-n)$ и $\overline{\varphi(x)}$ для любого $n \in \mathbf{Z}$. Имеем:

$$(\varphi(x-n), \overline{\varphi(x)}) = \frac{1}{2\pi}(F[\varphi(x-n)], F[\overline{\varphi(x)}]) =$$

$$= \frac{1}{2\pi}(e^{-in\omega}F[\varphi], \overline{F[\varphi](-\omega)}]) = \frac{1}{2\pi}\int_{-\infty}^{\infty} e^{-in\omega}\Phi(\omega)\Phi(-\omega)d\omega = 0,$$

поскольку носители функций $\Phi(\omega)$ и $\Phi(-\omega)$ – это промежутки $[0, 2\pi]$ и $[-2\pi, 0]$ соответственно.

Остальные пространства V_j являются, как обычно, масштабированными версиями V_0. Легко видеть, что подпространство V_j, порожденное функциями $\varphi_{jk}(x)$ и $\overline{\varphi_{jk}(x)}$, ортогонально пространству вейвлетов W_j, порожденному функциями $\psi_{jk}(x)$ и $\overline{\psi_{jk}(x)}$.

Таким образом, имеет место ортогональное кратномасштабное разложение пространства $L^2(\mathbf{R})$. Функции $\psi_{jk}(x)$ и $\overline{\psi_{jk}(x)}$, $j, k \in \mathbf{Z}$, образуют ортонормированный базис $L^2(\mathbf{R})$. Вейвлет-разложение функции $f(x) \in L^2(\mathbf{R})$ производится обычным образом.

$$f(x) = \sum_{j\in\mathbf{Z}}\sum_{k\in\mathbf{Z}}(a_{j,k}\psi_{jk}(x) + \tilde{a}_{j,k}\overline{\psi_{jk}(x)}),$$

где $a_{j,k} = \int_{\mathbf{R}} f(x)\overline{\psi_{jk}(x)}\,dx$ и $\tilde{a}_{j,k} = \int_{\mathbf{R}} f(x)\psi_{jk}(x)\,dx$. Еще один вариант разложения с использованием пространства аппроксимаций V_0:

$$f(x) = \sum_{j\in\mathbf{Z}}(a_{\varphi,k}\varphi(x-k) + \tilde{a}_{\varphi,k}\overline{\varphi(x-k)}) +$$
$$+ \sum_{j=0}^{\infty}\sum_{k\in\mathbf{Z}}(a_{j,k}\psi_{jk}(x) + \tilde{a}_{j,k}\overline{\psi_{jk}(x)}), \qquad (3)$$

где $a_{\varphi,k} = \int_{\mathbf{R}} f(x)\overline{\varphi(x-k)}\,dx$ и $\tilde{a}_{\varphi,k} = \int_{\mathbf{R}} f(x)\varphi(x-k)dx$.

В работе [Newl] доказано равенство Парсеваля для разложения по данной системе функций.

$$\sum_{j\in\mathbf{Z}}\sum_{k\in\mathbf{Z}}(|a_{j,k}|^2+|\tilde{a}_{j,k}|^2)=\int_{\mathbf{R}}|f(x)|^2dx.$$

Вычисления коэффициентов по указанным выше формулам численно не являются эффективными, и следующий **альтернативный** подход лучше.

2.17.2. Периодические гармонические вейвлеты

Реально цифровые сигналы $\{f_r\}$ задаются конечным набором значений. Можно сделать его бесконечным периодическим продолжением и считать, что он является оцифровкой периодической функции $f(x)$ с периодом, например, 1. Такая функция строится несложно:

$$f(x)=\sum_{m=0}^{N-1}F_m e^{i2\pi mx}, \tag{4}$$

где

$$F_m=\frac{1}{N}\sum_{r=0}^{N-1}f_r e^{-i2\pi m\frac{r}{N}} \tag{5}$$

– дискретное преобразование Фурье массива $\{f_r\}$ длины N. В этом случае выполняется равенство $f_r = f(r/N)$, $r = 0, 1, ..., N - 1$ в соответствии с обратным дискретным преобразованием Фурье:

$$f_r=\sum_{m=0}^{N-1}F_m e^{i2\pi m\frac{r}{N}}.$$

Несмотря на то что мы имеем для массива $\{f_r\}$ дискретное преобразование Фурье $\{F_m\}$ и представление функции $f(x)$ тригонометрическим полиномом, представляет интерес получить для нее аналог вейвлет-разложения. Для этого нам потребуются периодические аналоги вейвлетов. Следуя [Newl], проведем следующий эвристический вывод формул для периодических вейвлетов.

Рассмотрим сначала *периодизацию* масштабирующей функции $\varphi(x)$ по формуле

$$\varphi_c(x)=\sum_{m=-\infty}^{+\infty}\varphi(x-m).$$

Применим прямое и обратное преобразование Фурье для $\varphi_c(x)$ и воспользуемся формулой $F[\varphi(x - m)] = e^{-i\omega m}\Phi(\omega)$:

$$\varphi_c(x) = F^{-1}[F[\varphi_c(x)]] = \frac{1}{2\pi}\sum_{m=-\infty}^{+\infty}\int_{-\infty}^{\infty}\Phi(\omega)e^{-i\omega m}e^{i\omega x}d\omega.$$

Используем формулу:

$$\frac{1}{2\pi}\sum_{m=-\infty}^{+\infty}e^{-i\omega m} = \sum_{n=-\infty}^{+\infty}\delta(\omega - 2\pi n).$$

Тогда, поскольку $\Phi(\omega)$ есть 1 на $[0, 2\pi)$, получаем:

$$\varphi_c(x) = \sum_{n=-\infty}^{+\infty}\int_{-\infty}^{\infty}\Phi(\omega)\delta(\omega - 2\pi n)e^{i\omega x}d\omega = \int_{-\infty}^{\infty}\Phi(\omega)e^{i\omega x}\delta(\omega)d\omega = 1.$$

Таким образом, периодизация $\varphi_c(x)$ масштабирующей функции – это функция, тождественно равная 1, $\varphi_c(x) = 1$.

Теперь проведем периодизацию вейвлетов $\psi_{j,k}(x) = \sqrt{2^j}\psi(2^j x - k)$:

$$\psi_{jk}^{per}(x) = \sqrt{2^j}\sum_{m=-\infty}^{+\infty}\psi(2^j(x-m) - k).$$

Преобразование Фурье функции $\sqrt{2^j}\psi(2^j(x-m)-k) = \sqrt{2^j}\psi(2^j(x - -m - k/2^j)$:

$$\hat{\psi}_{jk}(\omega) = \frac{1}{\sqrt{2^j}}e^{-i\omega m}e^{-i\omega k/2^j}\Psi\left(\frac{\omega}{2^j}\right).$$

Поэтому, используя прямое и обратное преобразования Фурье, получаем:

$$\psi_{jk}^{per}(x) = \frac{1}{2\pi}\sum_{m=-\infty}^{+\infty}\frac{1}{\sqrt{2^j}}\int_{-\infty}^{\infty}e^{-i\omega m}e^{-i\omega k/2^j}\Psi\left(\frac{\omega}{2^j}\right)e^{i\omega x}d\omega,$$

Применяя формулу $\frac{1}{2\pi}\sum_{m=-\infty}^{+\infty}e^{-i\omega m} = \sum_{n=-\infty}^{+\infty}\delta(\omega - 2\pi n)$, имеем:

$$\psi_{jk}^{per}(x) = \frac{1}{\sqrt{2^j}}\sum_{n=-\infty}^{+\infty}\int_{-\infty}^{\infty}\delta(\omega - 2\pi n)e^{-i\omega k/2^j}\Psi\left(\frac{\omega}{2^j}\right)e^{i\omega x}d\omega.$$

Поскольку $\psi(\omega/2^j) = 1$ только на интервале $[2\pi 2^j, 2\pi 2^{j+1})$, то сумма будет состоять из конечного числа членов, когда δ-функция $\delta(\omega - 2\pi n)$ принимает «ненулевые значения» на интервале $[2\pi 2^j, 2\pi 2^{j+1})$, то есть для $n = 2^j, 2^j + 1, ..., 2^{j+1} - 1$, или для $\omega = 2\pi 2^j, 2\pi(2^j + 1), ..., 2\pi(2^{j+1} - 1)$. Получаем:

$$\psi_{jk}^{per}(x) = \frac{1}{\sqrt{2^j}}\sum_{n=2^j}^{2^{j+1}-1}e^{i2\pi n(x-k/2^j)}. \qquad (6)$$

При этом легко видеть, что k меняется от 0 до $2^j - 1$, затем функции повторяются. Легко видеть, что система функций $\psi_{jk}^{per}(x)$, $j \in \mathbf{N} \cup \{0\}$, $k = 0, ..., 2^j - 1$, является ортонормированной на промежутке [0, 1]. Для получения базиса нужно добавить сопряженные функции и константу.

Вейвлет-базис *периодических вейвлетов* пространства $L^2[0, 1]$ состоит из функций:

$$\{\varphi_c(x) = 1, \psi_{jk}^{per}(x), \overline{\psi_{jk}^{per}(x)};\ j \in \mathbf{N} \cup \{0\}, k = 0, \ldots, 2^j - 1\}.$$

Замечание. Он получается по формулам (6) из обычного тригонометрического базиса функций $\{e^{i2\pi nx}, n \in \mathbf{Z}\}$.

Приведем несколько первых функций системы $\psi_{jk}^{per}(x)$.

Если $j = 0$, то имеется одна вейвлет-функция:

$$\psi_{0,0}^{per}(x) = e^{i2\pi x}.$$

Если $j = 1$, то имеются две вейвлет-функции:

$$\psi_{1,0}^{per}(x) = \frac{1}{\sqrt{2}} \sum_{n=2}^{3} e^{i2\pi nx} = \frac{1}{\sqrt{2}} (e^{i2\pi 2x} + e^{i2\pi 3x});$$

$$\psi_{1,1}^{per}(x) = \frac{1}{\sqrt{2}} \sum_{n=2}^{3} e^{i2\pi n(x-1/2)} = \frac{1}{\sqrt{2}} (e^{i2\pi (2x-1)} + e^{i2\pi (3x-3/2)}) =$$

$$= \frac{1}{\sqrt{2}} (e^{i2\pi 2x} + e^{i2\pi (3x-1/2)}).$$

При $k = 2$ мы получаем $\psi_{1,2}^{per}(x) = \psi_{1,0}^{per}(x)$:

$$\psi_{1,2}^{per}(x) = \frac{1}{\sqrt{2}} \sum_{n=2}^{3} e^{i2\pi n(x-2/2)} = \frac{1}{\sqrt{2}} (e^{i2\pi (2x-2)} + e^{i2\pi (3x-3)}) =$$

$$= \frac{1}{\sqrt{2}} (e^{i2\pi 2x} + e^{i2\pi 3x}) = \psi_{1,0}^{per}(x).$$

Если $j = 2$, то имеются 4 вейвлет-функции, которые легко выписываются.

2.17.3. Дискретное гармоническое вейвлет-разложение

Пусть $f(x)$ – вещественная непрерывная периодическая с периодом $T = 1$ функция, заданная на промежутке [0, 1] своими значениями $f_r = f(r/N)$, $r = 0, 1, ..., N - 1$, при условии $N = 2^p$. Оказывается, что ее вейвлет-коэффициенты по периодическим вейвлетам тесно связаны с элементами массива дискретного преобразования Фурье. Мы

предполагаем, что функция $f(x)$ восстанавливается по значениям $f_r = f(r/N)$ следующим образом:

$$f(x) = \sum_{m=0}^{N-1} F_m e^{i2\pi m x},$$

где

$$F_m = \frac{1}{N} \sum_{r=0}^{N-1} f_r e^{-i2\pi m \frac{r}{N}}$$

– дискретное преобразование Фурье. В этом случае выполняется равенство $f_r = f(r/N)$, $r = 0, 1,..., N - 1$ в соответствии с обратным дискретным преобразованием Фурье:

$$f_r = \sum_{m=0}^{N-1} F_m e^{i2\pi m \frac{r}{N}}.$$

Напомним, что в случае вещественного массива $\{f_r\}$ имеют место свойства: $F_{N-m} = \overline{F_m}$ и коэффициенты F_0 и $F_{N/2}$ вещественны. Поскольку оцифровка $f(r/N)$ позволяет точно восстановить $f(x)$, то у функции $f(x)$ отсутствуют частоты выше частоты Найквиста πN.

Найдем вейвлет-коэффициенты выбранной выше функции $f(x) = \sum_{m=0}^{N-1} F_m e^{i2\pi m x}$. Базис вейвлетов пространства $L^2[0, 1]$ состоит из функций:

$$\varphi_c(x) = 1,$$

$$\psi_{jk}^{per}(x) = \frac{1}{\sqrt{2^j}} \sum_{n=2^j}^{2^{j+1}-1} e^{i2\pi n(x-k/2^j)},$$

$$\overline{\psi_{jk}^{per}(x)} = \frac{1}{\sqrt{2^j}} \sum_{n=2^j}^{2^{j+1}-1} e^{-i2\pi n(x-k/2^j)}.$$

Тогда вейвлет-коэффициенты вычисляются по формулам:

$$a_0 = (f,1) = \int_0^1 f(x)dx;$$

$$a_{jk} = (f, \psi_{jk}^{per}) = \frac{1}{2^{j/2}} \int_0^1 f(x) \sum_{n=2^j}^{2^{j+1}-1} e^{-i2\pi n(x-k/2^j)} dx =$$

$$= \frac{1}{2^{j/2}} \sum_{m=0}^{N-1} F_m \sum_{n=2^j}^{2^{j+1}-1} \int_0^1 e^{i2\pi m x} e^{-i2\pi n(x-k/2^j)} dx =$$

$$= \frac{1}{2^{j/2}} \sum_{n=2^j}^{2^{j+1}-1} e^{i2\pi nk/2^j} \sum_{m=0}^{N-1} F_m \int_0^1 e^{i2\pi m x} e^{-i2\pi n x} dx = \frac{1}{2^{j/2}} \sum_{n=2^j}^{2^{j+1}-1} e^{i2\pi nk/2^j} F_n.$$

Поскольку функция $f(x)$ является тригонометрическим полиномом степени меньше, чем $N = 2^p$, а функции $\psi_{jk}^{per}(x)$ – это сумма функций $e^{i2\pi n(x-k/2^j)}$, где $n \geq 2^j$, то при $j \geq p$ вейвлет-коэффициенты обращаются в нуль, $a_{jk} = (f, \psi_{jk}^{per}) = 0$.

Если функция вещественная, то

$$\tilde{a}_{jk} = (f, \overline{\psi_{jk}^{per}}) = \frac{1}{2^{j/2}} \sum_{n=2^j}^{2^{j+1}-1} e^{-i2\pi nk/2^j} F_{N-n} = \frac{1}{2^{j/2}} \sum_{n=2^j}^{2^{j+1}-1} e^{-i2\pi nk/2^j} \overline{F_n} = \overline{a_{jk}}.$$

Поскольку k меняется от 0 до $2^j - 1$, то имеет смысл переобозначить коэффициенты: $a_{j,k} \equiv a_{2^j+k}$. Тогда формулы принимают вид:

$$a_{2^j+k} = \frac{1}{\sqrt{2^j}} \sum_{s=0}^{2^j-1} F_{2^j+s} e^{i2\pi(2^j+s)\frac{k}{2^j}} = \frac{1}{\sqrt{2^j}} \sum_{s=0}^{2^j-1} F_{2^j+s} e^{i2\pi s\frac{k}{2^j}}, \tag{7}$$

$$\tilde{a}_{2^j+k} = \frac{1}{\sqrt{2^j}} \sum_{s=0}^{2^j-1} F_{N-(2^j+s)} e^{-i2\pi s\frac{k}{2^j}}. \tag{8}$$

Коэффициенты $a_{2^j}, a_{2^j+1}, \ldots, a_{2^{j+1}-1}$ – это, с точностью до коэффициента $\sqrt{2^j}$, обратное преобразование Фурье массива $F_{2^j}, F_{2^j+1}, \ldots, F_{2^{j+1}-1}$.

Как уже отмечалось, $a_{j,k} = 0$ при $j \geq p$. Поэтому вейвлет-разложение выбранной нами функции $f(x)$ должно быть конечным: для $j = 0, \ldots p-1$ и $k = 0, \ldots 2^j - 1$,

$$f(x) = a_0 \varphi_c(x) + \sum_{j=0}^{p-1} \sum_{k=0}^{2^j} (a_{j,k} \psi_{jk}^{per}(x) + \tilde{a}_{j,k} \overline{\psi_{jk}^{per}(x)}).$$

В работе Ньюланда [Newl] показано, что в действительности вейвлет-разложение функции $f(x)$ будет только до уровня $p - 2$, но с дополнительным слагаемым:

$$f(x) = a_0 + \sum_{j=0}^{p-2} \sum_{k=0}^{2^j} (a_{2^j+k} \psi_{jk}^{per}(x) + \tilde{a}_{2^j+k} \overline{\psi_{jk}^{per}(x)}) + a_{N/2} \psi_{N/2}(x),$$

где $\psi_{N/2}(x)$ – особый вейвлет, имеющий вид $\psi_{N/2}(x) = e^{i\pi Nx}$. Тогда мы получаем следующее распределение вейвлет-коэффициентов и соответствующих коэффициентов Фурье массива $\{f_r\}$ (табл. 2.17.1).

Равенство Парсеваля в этом случае имеет вид [Newl]:

$$|a_0|^2 + \sum_{j=0}^{p-2} \sum_{k=0}^{2^j-1} (|a_{2^j+k}|^2 + |\tilde{a}_{2^j+k}|^2) + |a_{N/2}|^2 = \frac{1}{N} \sum_{r=0}^{N-1} |f_r|^2.$$

Учитывая, что коэффициенты $\tilde{a}_{2^j}, \tilde{a}_{2^j+1}, \ldots, \tilde{a}_{2^{j+1}-1}$ выражаются как преобразование Фурье элементов $F_{N-2^j}, F_{N-(2^j+1)}, \ldots, F_{N-(2^{j+1}-1)}$, имеет смысл их переобозначить:

Таблица 2.17.1. Распределение вейвлет-коэффициентов по уровням

Уровень разложения	Вейвлет-коэффициенты	Используемые коэффициенты Фурье
0	$a_0 = F_0$	F_0
0	$a_1 = F_1$ и $\tilde{a}_1 = F_{N-1}$	F_1
1	a_2, a_3 и $\tilde{a}_2, \tilde{a}_3$	F_2, F_3 и F_{N-2}, F_{N-3}
2	a_4, a_5, a_6, a_7 и $\tilde{a}_4, \tilde{a}_5, \tilde{a}_6, \tilde{a}_7$	F_4, F_5, F_6, F_7 и $F_{N-4}, F_{N-5}, F_{N-6}, F_{N-7}$
.....	.....	.....
j	$a_{2^j}, a_{2^j+1}, \ldots, a_{2^{j+1}-1}$ и $\tilde{a}_{2^j}, \tilde{a}_{2^j+1}, \ldots, \tilde{a}_{2^{j+1}-1}$	$F_{2^j}, F_{2^j+1}, \ldots, F_{2^{j+1}-1}$ и $F_{N-2^j}, F_{N-(2^j+1)}, \ldots, F_{N-(2^{j+1}-1)}$
.....	.....	.....
$p-2$	$a_{N/4}, a_{N/4+1}, \ldots, a_{N/2-1}$ и $\tilde{a}_{N/4}, \tilde{a}_{N/4+1}, \ldots, \tilde{a}_{N/2-1}$	$F_{N/4}, F_{N/4+1}, \ldots, F_{N/2-1}$ и $F_{N-N/4}$, $F_{N-(N/4+1)}, \ldots, F_{N-(N/2-1)}$
$p-1$	$a_{N/2} = F_{N/2}$	$F_{N/2}$

$a_{N-(2^j+k)} = \tilde{a}_{2^j+k}$.

Пример. Если для уровня разложения $j = 3$, $N = 16$, первоначально был такой набор коэффициентов:

$a_0, a_1, a_2, a_3, a_4, a_5, a_6, a_7, a_8, \tilde{a}_7, \tilde{a}_6, \tilde{a}_5, \tilde{a}_4, \tilde{a}_3, \tilde{a}_2, \tilde{a}_1,$

то после переобозначения он будет таким:

$a_0, a_1, a_2, a_3, a_4, a_5, a_6, a_7, a_8, a_9, a_{10}, a_{11}, a_{12}, a_{13}, a_{14}, a_{15}$.

При этом $a_{N-k} = \overline{a_k}$, коэффициенты a_0 и $a_{N/2}$ вещественны.

Замечание. Примеры практического использования дискретного гармонического вейвлет-разложения см. в работах [Newl], [Newl-2], [Cat10], [Cat08], [Cat03], [Cat-Kud08], [ККМ] и [КОГЭ].

2.18. Мультивейвлеты

При построении вейвлетов участвуют две скалярные функции: масштабирующая функция $\varphi(x)$ и вейвлет $\psi(x)$. Для анализа изображений важны такие свойства вейвлетов как: компактный носитель, ортогональность, симметричность и нулевые моменты. В скалярном случае невозможно добиться одновременно выполнения этих свойств. Но это возможно для мультивейвлетов, то есть вейвлетов, образуемых вектор-функциями $\Phi(x) = (\varphi_1(x), \ldots, \varphi_r(x))$ и $\Psi(x) = (\psi_1(x), \ldots, \psi_r(x))$. Мультивейвлеты впервые были построены в работах Алперта и Хербе [Alp], [Her]. Ортогональные мультивейвлеты построены в работе

[DGHM]. Подробное изложение теории можно найти в диссертации В.Стрелы [StV] и в книге Кейнерта [Kei]. Основные положения мультивейвлетов совершенно аналогичны скалярному случаю. В данном разделе мы рассмотрим основные элементы этой теории. Отметим, что мультивейвлеты используются для построения базисов в пространстве функций $L^2(\mathbf{R})$ а не в пространстве вектор-функций.

Масштабирующее уравнение имеет вид:

$$\Phi(x) = \sqrt{2}\sum_{n\in\mathbf{Z}} H_n \Phi(2x-n), \tag{1}$$

где H_n – матрицы порядка r и $\Phi(x)$ рассматривается как вектор-столбец. Набор $\{H_n\}$ будем называть матричным фильтром масштабирующей функции.

Пространство V_0 порождено целочисленными сдвигами функций $\{\varphi_l(x-n), l = 1, ..., r, n \in \mathbf{Z}\}$. Будем обозначать кратко $\{\Phi(x-n)\}$ такой набор функций. Другие пространства V_j кратномасштабного разложения пространства $L^2(\mathbf{R})$ определяются также аналогично скалярному случаю. Они порождены функциями $\Phi_{j,n}(x) = \sqrt{2^j}\,\Phi(2^j x - n)$, $n, j \in \mathbf{Z}$.

Масштабирующая функция $\Phi(x) = (\varphi_1(x), ..., \varphi_r(x))$ называется ортогональной, если

$$(\varphi_p(x-n), \varphi_q(x-k))_{L^2} = \delta_{pq}\delta_{nk}, \quad \forall p, q = 1, \ldots, r, \ \forall n, k \in \mathbf{Z}.$$

Пример 1. Пусть $\Phi(x) = (\varphi_1(x), \varphi_2(x))$, где

$$\varphi_1(x) = \begin{cases} 1, & x \in [0,1) \\ 0, & \text{для других } x \end{cases}, \qquad \varphi_2(x) = \begin{cases} \sqrt{3}(2x-1), & x \in [0,1) \\ 0, & \text{для других } x \end{cases},$$

Тогда легко видеть, что

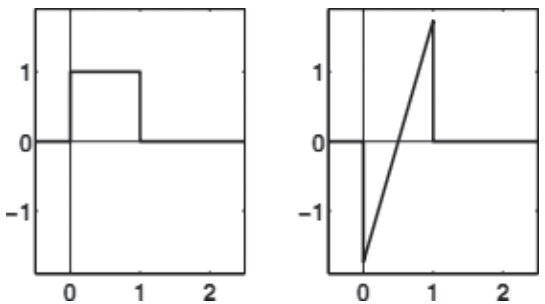


Рис. 2.18.1. Графики мультивейвлетов
$\Phi(x) = (\varphi_1(x), \varphi_2(x))$

$$\varphi_1(x) = \varphi_1(2x) + \varphi_1(2x-1),$$

$$\varphi_2(x) = -\frac{\sqrt{3}}{2}\varphi_1(2x) + \frac{1}{2}\varphi_2(2x) + \frac{\sqrt{3}}{2}\varphi_1(2x-1) + \frac{1}{2}\varphi_2(2x-1).$$

Вектор-функция $\Phi(x) = (\varphi_1(x), \varphi_2(x))$ является масштабирующей со следующими матричными коэффициентами:

$$H_0 = \frac{1}{2\sqrt{2}}\begin{pmatrix} 2 & 0 \\ -\sqrt{3} & 1 \end{pmatrix}, \qquad H_1 = \frac{1}{2\sqrt{2}}\begin{pmatrix} 2 & 0 \\ \sqrt{3} & 1 \end{pmatrix}.$$

Ортогональность легко проверяется прямыми вычислениями.

Пример 2. Пусть $\Phi(x) = (\varphi_1(x), \varphi_2(x))$, где

$$\varphi_1(x) = \begin{cases} 3x^2 - 2x^3 & x \in [0,1] \\ 3(2-x^2) - 2(2-x)^3 & x \in [1,2] \\ 0 & x \notin [0,2] \end{cases}$$

$$\varphi_2(x) = \begin{cases} x^3 - x^2 & x \in [0,1] \\ (2-x^2) - (2-x)^3 & x \in [1,2] \\ 0 & x \notin [0,2] \end{cases}$$

– эрмитовы кубические сплайны, обладающие свойствами: $\varphi_1(1) = 1$, $\varphi_1'(1) = 0$, $\varphi_2(1) = 0$, $\varphi_2'(1) = 1$.

Масштабирующие соотношения имеют вид:

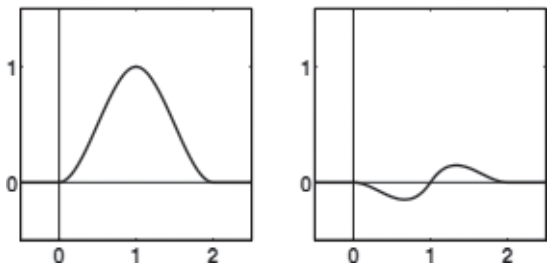


Рис. 2.18.2. Графики эрмитовых кубических сплайнов $\varphi_1(x)$ и $\varphi_2(x)$

$$\varphi_1(x) = \frac{1}{2}\varphi_1(2x) + \frac{3}{4}\varphi_2(2x) + \varphi_1(2x-1) + \frac{1}{2}\varphi_1(2x-2) - \frac{3}{4}\varphi_2(2x-2),$$

$$\varphi_2(x) = -\frac{1}{8}\varphi_1(2x) - \frac{1}{8}\varphi_2(2x) + \frac{1}{2}\varphi_2(2x-1) + \frac{1}{8}\varphi_1(2x-2) - \frac{1}{8}\varphi_2(2x-2).$$

Вектор-функция $\Phi(x) = (\varphi_1(x), \varphi_2(x))$ является масштабирующей со следующими матричными коэффициентами:

$$H_0 = \frac{1}{8\sqrt{2}}\begin{pmatrix} 4 & 6 \\ -1 & -1 \end{pmatrix}, \quad H_1 = \frac{1}{8\sqrt{2}}\begin{pmatrix} 8 & 0 \\ 0 & 4 \end{pmatrix}, \quad H_2 = \frac{1}{8\sqrt{2}}\begin{pmatrix} 4 & -6 \\ 1 & -1 \end{pmatrix}.$$

Это неортогональная масштабирующая вектор-функция $\Phi(x)$.

Пример 3. Симметричная ортогональная масштабирующая функция DGHM (Donovan–Geronimo–Hardin–Massopust). Это первый нетривиальный пример ортогональной масштабирующей функции $\Phi(x) = (\varphi_1(x), \varphi_2(x))$. Определяется своими коэффициентами:

$$H_0 = \frac{1}{20\sqrt{2}}\begin{pmatrix} 12 & 16\sqrt{2} \\ -\sqrt{2} & -6 \end{pmatrix}, \quad H_1 = \frac{1}{20\sqrt{2}}\begin{pmatrix} 12 & 0 \\ 9\sqrt{2} & 20 \end{pmatrix},$$

$$H_2 = \frac{1}{20\sqrt{2}}\begin{pmatrix} 0 & 0 \\ 9\sqrt{2} & -6 \end{pmatrix}, \quad H_3 = \frac{1}{20\sqrt{2}}\begin{pmatrix} 0 & 0 \\ -\sqrt{2} & 0 \end{pmatrix}.$$

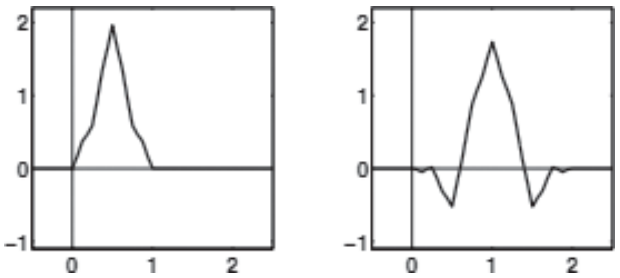


Рис. 2.18.3. Графики компонент масштабирующей функции $\Phi(x)$ DGHM

Пусть $\Phi(x)$ – масштабирующая функция. Образуем следующие функции:

$$\Phi_{j,n}(x) = \sqrt{2^j}\,\Phi(2^j x - n), j, n \in \mathbf{Z}. \tag{2}$$

Предложение 1. *Если* $\Phi(x)$ *– масштабирующая функция с масштабирующим фильтром* $\{H_n, n \in \mathbf{Z}\}$, *то для любых* $j, k \in \mathbf{Z}$ *имеет место следующее разложение*:

$$\Phi_{j-1,k}(x) = \sum_n H_{n-2k}\Phi_{j,n}(x) = \sum_n H_n \Phi_{j,n+2k}(x). \tag{3}$$

В частности,

$$\Phi_{0,k}(x) = \sqrt{2}\sum_n H_{n-2k}\Phi(2x-n) = \sum_n H_{n-2k}\Phi_{1,n}(x),$$

$$\Phi_{j-1,0}(x) = \sum_n H_n \Phi_{j,n}(x).$$

Доказательство.

$$\Phi_{j-1,k}(x) = \sqrt{2^{j-1}}\Phi(2^{j-1}x-k) = \sqrt{2}\sqrt{2^{j-1}}\sum_m H_m \Phi(2(2^{j-1}x-k)-m) =$$

$$= \sqrt{2^j}\sum_m H_m \Phi(2^j x - 2k - m) = \sqrt{2^j}\sum_m H_m \Phi_{j,m+2k}(x) = \sum_n H_{n-2k}\Phi_{j,n}(x).$$

Частотная функция $H(\omega)$ масштабирующей функции $\Phi(x) = (\varphi_1(x), ..., \varphi_r(x))$ определяется равенством

$$H(\omega) = \frac{1}{\sqrt{2}}\sum_{n\in\mathbf{Z}} H_n e^{-in\omega}. \qquad (4)$$

Пусть $\hat{\Phi}(\omega) = (\hat{\varphi}_1(\omega),\ldots,\hat{\varphi}_r(\omega))$ – преобразование Фурье масштабирующей функции $\Phi(x)$, тогда легко видеть, что имеет место равенство

$$\hat{\Phi}(\omega) = H\left(\frac{\omega}{2}\right)\hat{\Phi}\left(\frac{\omega}{2}\right). \qquad (5)$$

Соотношение (5) также называется *масштабирующим уравнением*. Имеет место следующее необходимое условие ортогональности:

$$\sum_n H_n H^*_{n-2k} = \delta_{0k} Id,$$

где Id – единичная матрица, а «звездочка» означает сопряженную матрицу. Эквивалентное условие ортогональности:

$$|H(\omega)|^2 + |H(\omega+\pi)|^2 = Id \text{ п. в.} \qquad (6)$$

В последнем выражении имеется в виду, что $|H|^2 = HH^*$.

Кратномасштабный и ортогональный кратномасштабный анализ определяется совершенно аналогично скалярному случаю.

Построение вейвлетов

Масштабирующее уравнение для вейвлетов $\Psi(x) = (\psi_1(x), ..., \psi_r(x))$ имеет также матричный вид:

$$\Psi(x) = \sqrt{2}\sum_{n\in\mathbf{Z}} G_n \Phi(2x-n),$$

где G_n – матрицы порядка r и $\Psi(x)$ рассматривается как вектор-столбец. Набор $\{G_n\}$ будем называть матричным фильтром вейвлета

$\Psi(x)$. Частотная функция $G(\omega)$ вейвлета $\Psi(x) = (\psi_1(x), ..., \psi_r(x))$ определяется равенством

$$G(\omega) = \frac{1}{\sqrt{2}} \sum_{n \in \mathbf{Z}} G_n e^{-i n \omega}. \tag{7}$$

Пусть $\hat{\Psi}(\omega) = (\hat{\psi}_1(\omega), \ldots, \hat{\psi}_r(\omega))$ – преобразование Фурье масштабирующей функции $\Phi(x)$, тогда легко видеть, что имеет место равенство

$$\hat{\Psi}(\omega) = G\left(\frac{\omega}{2}\right) \hat{\Phi}\left(\frac{\omega}{2}\right). \tag{8}$$

Ортогональность $\Phi(x)$ и $\Psi(x)$ выражается равенствами:

$$\sum_n H_n H^*_{n-2k} = \delta_{0k} Id,$$

$$\sum_n G_n G^*_{n-2k} = \delta_{0k} Id,$$

$$\sum_n H_n G^*_{n-2k} = \sum_n G_n H^*_{n-2k} = 0,$$

или эквивалентно

$$|H(\omega)|^2 + |H(\omega + \pi)|^2 = Id,$$

$$|G(\omega)|^2 + |G(\omega + \pi)|^2 = Id,$$

$$G(\omega)H(\omega)^* + G(\omega + \pi)H(\omega + \pi)^* = 0.$$

Это также эквивалентно унитарности следующей блочной матрицы:

$$\begin{pmatrix} H(\omega) & H(\omega+\pi) \\ G(\omega) & G(\omega+\pi) \end{pmatrix}.$$

Из приведенных условий ортогональности можно найти частотную функцию $G(\omega)$ вейвлета и сам вейвлет $\Psi(x) = (\psi_1(x), ..., \psi_r(x))$. Как и в скалярном случае, в качестве $G(\omega)$ можно взять $G(\omega) = -e^{i\omega}H(\omega + \pi)^*$. Тогда

$$\begin{pmatrix} H(\omega) & H(\omega+\pi) \\ -e^{i\omega}H(\omega+\pi)^* & e^{i\omega}H(\omega)^* \end{pmatrix} \begin{pmatrix} H(\omega)^* & -e^{-i\omega}H(\omega+\pi) \\ H(\omega+\pi)^* & e^{-i\omega}H(\omega) \end{pmatrix} = \begin{pmatrix} Id & 0 \\ 0 & Id \end{pmatrix}.$$

В дальнейшем мы будем считать, что мультимасштабирующая функция $\Phi(x)$ имеет компактный носитель. Тогда число матричных коэффициентов $H_n(\omega)$ и частотная функция $H(\omega)$ является матрицей тригонометрических полиномов. Напомним, что мультивейвлеты

строились с целью получить «хорошие» вейвлеты с компактным носителем.

Операторы проектирования

Пусть $V_j, j \in \mathbf{Z}$ – подпространство, порожденное функциями

$\Phi_{j,n}(x) = \sqrt{2^j}\,\Phi(2^j x - n), j, n \in \mathbf{Z}.$

Ортогональная проекция функции $f(x)$ из $L^2(\mathbf{R})$ на пространство V_j определяется формулой

$$P_j(f) = \sum_{n \in \mathbf{Z}} (f, \Phi_{j,n})\Phi_{j,n}(x),$$

где $(f, \Phi_{j,n})$ – вектор-строка скалярных произведений $f(x)$ с функциями $\varphi_{lj,n}(x)$, а $\Phi_{j,n}(x)$ – вектор столбец функций $\varphi_{lj,n}(x)$, $l = 1, ..., r$. Проекция $P_j(f)$ является аппроксимацией $f(x)$ степени разрешения 2^{-j}. Разность между аппроксимациями степени 2^{-j} и $2^{-(j+1)}$

$$Q_j(f) = P_{j+1}(f) - P_j(f)$$

представляет собой детали степени разрешения 2^{-j}. Эта разность лежит в V_{j+1} и ортогональна к $V_j \in V_{j+1}$, то есть она является ортогональной проекцией на ортогональное дополнение W_j к пространству V_j до V_{j+1},

$$V_{j+1} = V_j \oplus W_j.$$

Как и в скалярном случае ортогонального кратномасштабного разложения, имеем:

$$L^2(\mathrm{R}) = \oplus_{j=-\infty}^{\infty} W_j.$$

Ортогональная проекция $Q_j(f)$ функции $f(x)$ на пространство вейвлетов W_j определяется аналогичной формулой

$$Q_j(f) = \sum_{n \in \mathbf{Z}} (f, \Psi_{j,n})\Psi_{j,n}(x),$$

где $(f, \Psi_{j,n})$ – вектор-строка скалярных произведений $f(x)$ с вейвлет-функциями $\psi_{l,j,n}(x)$, а $\Psi_{j,n}(x)$ – вектор-столбец функций $\psi_{l,j,n}(x)$, $l = 1, ..., r$. Тогда вейвлет-разложение функции $f(x)$ выражается формулой

$$f = \sum_{j=-\infty}^{\infty} Q_j(f).$$

Следующая формула

$$f \approx P_j(f) = P_{j-N}(f) + \sum_{k=1}^{N} Q_{j-k}(f)$$

лежит в основе дискретного вейвлет-преобразования. Явные формулы разложения и восстановления находятся аналогично скалярному случаю.

Быстрое вейвлет-преобразование

Для вейвлет-преобразования функции $f(x)$ необходимо вычислить серию коэффициентов $\{cA_N, cD_N, cD_{N-1}, ..., cD_1\}$ (в этом разделе коэффициенты считаются вектор-столбцами). Каждый вектор-столбец коэффициентов находится интегрированием:

$$a_{j-N,k} = (f, \Phi_{j-N,k}) = \int_R f(x)\overline{\Phi_{j-N,k}(x)}dx, \tag{9}$$

$$d_{j-m,k} = (f, \Psi_{j-m,k}) = \int_R f(x)\overline{\Psi_{j-m,k}(x)}dx, \quad m = 1, 2, \ldots, N. \tag{10}$$

Возникает проблема вычисления большого количества интегралов с необходимой точностью. Быстрое вейвлет-преобразование, предложенное Малла [Ма], позволяет решить эту проблему. Алгоритм Малла дает возможность вычислять коэффициенты вейвлет-разложения без интегрирования, используя алгебраические операции на основе свертки. Предположим, что мы знаем разложение функции $f(x)$ для некоторого уровня разрешения j, то есть знаем коэффициенты $cA_0 = \{a_{j,n}\}$. Проблема заключается в том, чтобы сделать первый шаг разложения

$$P_j(f) = \sum_{k\in Z} a^T_{j-1,k}\Phi_{j-1,k}(x) + \sum_{k\in Z} d^T_{j-1,k}\Psi_{j-1,k}(x)$$

без интегрирования (символ T обозначает транспонирование).

Напомним масштабирующее соотношение для $\Phi(x)$ и выражение вейвлета $\Psi(x)$:

$$\Phi(x) = \sum_{n\in \mathbf{Z}} H_n\Phi_{1,n}(x), \qquad \Psi(x) = \sum_n G_n\Phi_{1,n}(x),$$

где $\{H_n\}$ – масштабирующий (матричный) фильтр функции $\Phi(x)$ и $\{H_n\}$ – фильтр вейвлета $\Psi(x)$. Из этих соотношений сразу вытекают следующие (см. предложение 1 этого раздела):

$$\Phi_{j-1,k}(x) = \sum_n H_n\Phi_{j,n+2k}(x), \qquad \Psi_{j-1,k}(x) = \sum_n G_n\Phi_{j,n+2k}(x).$$

Последние равенства выражают базисные функции $\Phi_{j-1,k}(x)$ и $\Psi_{j-1,k}(x)$ через базисные функции $\Phi_{j,k}(x)$ пространства V_j. Тогда:

$$a_{j-1,k} = (f, \Phi_{j-1,k}) = \left(f, \sum_n H_n \Phi_{j,n+2k} \right) = \sum_n \overline{H_n}\, a_{j,n+2k},$$

$$d_{j-1,k} = (f, \Psi_{j-1,k}) = \left(f, \sum_n G_n \Phi_{j,n+2k} \right) = \sum_n \overline{G_n}\, a_{j,n+2k}.$$

Получили формулы

$$a_{j-1,k} = \sum_n \overline{H_n} a_{j,n+2k}, \qquad d_{j-1,k} = \sum_n \overline{G_n} a_{j,n+2k}, \tag{11}$$

которые выражают коэффициенты первого уровня разложения через начальные коэффициенты при помощи алгебраических операций умножения и суммирования. Индекс $2k$ в сумме справа говорит о проведенной децимации. Совершенно аналогично эти формулы можно представить в виде свертки и последующей децимации.

Ясно, что процедуру быстрого вейвлет-преобразования (11) можно повторить для набора коэффициентов $cA_1 = \{a_{j-1,n}\}$ и таким образом найти все коэффициенты $\{cA_N, cD_N, cD_{N-1}, ..., cD_1\}$ вейвлет-разложения глубины N.

Восстановление

Рассмотрим вопрос о восстановлении начального приближения $P_j(f)$ по набору коэффициентов $\{cA_N, cD_N, cD_{N-1}, ..., cD_1\}$ вейвлет-разложения (в этом разделе коэффициенты считаются вектор-строками). Достаточно описать восстановление cA_0 по коэффициентам $\{cA_1, cD_1\}$. Коэффициенты $cA_0 = \{a_{j,n}\}$ есть коэффициенты разложения

$$P_j(f) = \sum_{n \in \mathbf{Z}} a_{j,n} \Phi_{j,n}(x) \tag{12}$$

по базису $\Phi_{j,n}(x) = \sqrt{2^j}\, \Phi(2^j x - n)$ пространства V_j. Коэффициенты $cA_1 = \{a_{j-1,n}\}$ и $cD_1 = \{d_{j-1,n}\}$ есть коэффициенты разложения

$$P_j(f) = \sum_{k \in Z} a_{j-1,k} \Phi_{j-1,k}(x) + \sum_{k \in Z} d_{j-1,k} \Psi_{j-1\,k}(x) \tag{13}$$

по базису $\Phi_{j-1,k}(x)$ и $\Psi_{j-1,k}(x)$ пространства V_j, соответствующего разложению $V_j = V_{j-1} \oplus W_{j-1}$. Поэтому для нахождения коэффициентов $cA_0 = \{a_{j,n}\}$ достаточно от разложения (13) перейти к разложению вида (12). Это легко сделать с использованием следующих формул (см. предложение 1 этого раздела):

$$\Phi_{j-1,k}(x) = \sum_n H_{n-2k} \Phi_{j,n}(x), \qquad \Psi_{j-1,k}(x) = \sum_n G_{n-2k} \Phi_{j,n}(x).$$

А именно

$$P_j(f) = \sum_{k\in Z} a_{j-1,k}\Phi_{j-1,k}(x) + \sum_{k\in Z} d_{j-1,k}\Psi_{j-1,k}(x) =$$

$$= \sum_k a_{j-1,k} \sum_n H_{n-2k}\,\Phi_{j,n}(x) + \sum_k d_{j-1,k} \sum_n G_{n-2k}\,\Phi_{j,n}(x) =$$

$$= \sum_n \left(\sum_k (a_{j-1,k}H_{n-2k} + d_{j-1,k}G_{n-2k}) \right) \Phi_{j,n}(x).$$

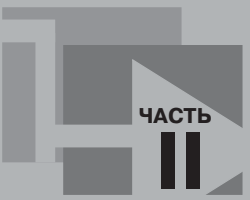
Поэтому

$$a_{j,n} = \sum_k (a_{j-1,k}H_{n-2k} + d_{j-1,k}G_{n-2k}). \qquad (14)$$

Формула (14) дает нам требуемое *восстановление* коэффициентов $cA_0 = \{a_{j,n}\}$ по коэффициентам $\{cA_1, cD_1\}$. Последнюю формулу можно записать в виде свертки, сделав обратную децимацию массивов $cA_1 = \{a_{j-1,n}\}$ и $cD_1 = \{d_{j-1,n}\}$. Отметим, что в последней формуле векторы коэффициентов рассматриваются как вектор-строки.

Вывод. Вейвлет-разложение производится сопряженными фильтрами $H_n^* = \overline{H}_{-n}$ и $G_n^* = \overline{G}_{-n}$ вейвлетов $\Phi(x)$ и $\Psi(x)$ с последующей децимацией, а восстановление производится фильтрами $\{H_n\}$ и $\{G_n\}$ вейвлетов $\Phi(x)$ и $\Psi(x)$ с предварительной обратной децимацией.

ВЕЙВЛЕТЫ В MATLAB



Система компьютерной математики MATLAB наиболее популярна среди инженеров и математиков, занимающихся прикладными разработками. Кроме того, именно в MATLAB вейвлеты представлены наиболее полно. В данной части книги рассматривается работа с вейвлетами по анализу сигналов и изображений в системе MATLAB. Сначала изучаются команды MATLAB Wavelet Toolbox. Для облегчения исследования сигнала при помощи вейвлетов в MATLAB создан комплекс графических оболочек для визуализации исходных данных и результатов вейвлет-анализа. Этот комплекс называется главным меню, или графическим интерфейсом пользователя (GUI). В последней главе достаточно подробно рассматривается работа с вейвлетами в среде GUI MATLAB Wavelet Toolbox.

Функции вейвлет-анализа в MATLAB



Данная глава посвящена описанию работы с вейвлетами в MATLAB в командном режиме. Дается описание основных функций системы MATLAB, связанных с вейвлетами и их использованием. Показано, как можно получить значения и графики основных типов вейвлетов, как найти масштабирующие фильтры и фильтры разложения и восстановления вейвлетов. Рассмотрены возможности Wavelet Toolbox MATLAB для анализа сигналов, очистки от шума, сжатия. Рассмотрен пример вейвлет-анализа кардиосигнала.

В 2012 году изменился внешний вид MATLAB и, соответственно, вид пакета расширения Wavelet Toolbox MATLAB. Введены некоторые новые функции, расширяющие возможности вейвлет-анализа в MATLAB. Существенно изменилась справочная система пакета Wavelet Toolbox MATLAB. Она стала более структурированой (рис. 3.0.1), однако стали недоступными непосредственно статьи, которые описывают теоретические основы вейвлет-анализа (User's Guide). Такие статьи можно теперь найти при изучении справки по какой-либо функции MATLAB. А в основном окне Help Wavelet Toolbox MATLAB остались только справка по функциям вейвлет-анализа в MATLAB и примеры.

Дадим краткое описание справочной системы Help MATLAB и Wavelet Toolbox.

На верхней панели Help MATLAB имеются стрелки навигации, кнопка настройки **Preferences**, панель вкладок с кнопкой **+** для задания новой вкладки и несколько возможностей выбора типа отображения основного поля. Ниже идет строка поиска. Еще ниже – адресная строка справки с двумя кнопкам: **Home** – для выхода в основное меню справки и кнопка **Table of Contents** – для отображения содержания активной вкладки. Основное окно справки имеет три вкладки:

Getting Started, **Examples** и **Release notes**, назначение которых не требует пояснений (рис. 3.0.1, 3.0.2).

Справка Wavelet Toolbox содержит описание функций пакета, сгруппированных в 6 разделов:

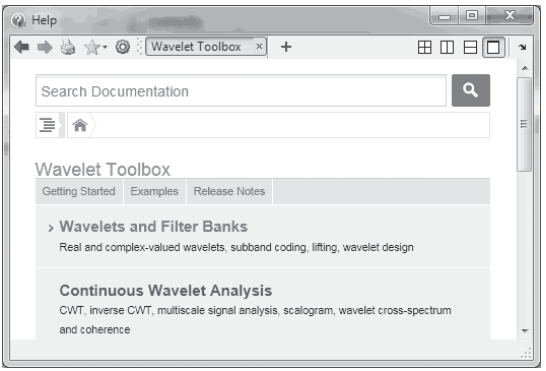


Рис. 3.0.1. Разделы справки Wavelet Toolbox MATLAB

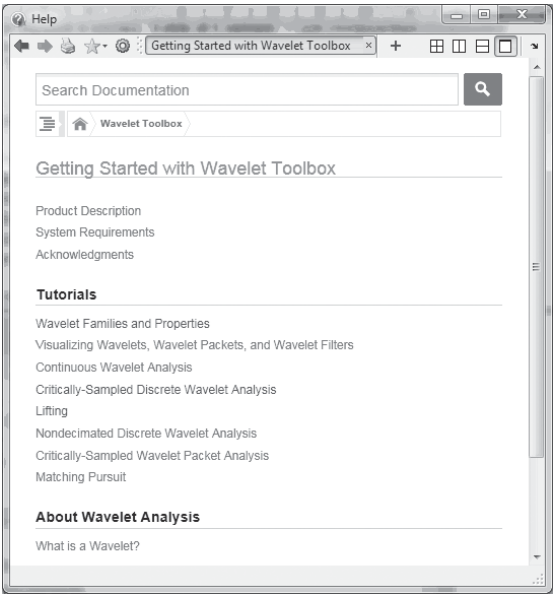


Рис. 3.0.2. Вкладка **Getting Started** справки Wavelet Toolbox MATLAB

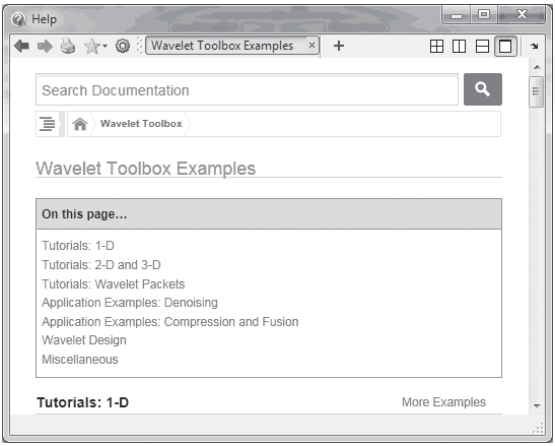


Рис. 3.0.3. Вкладка **Examples** справки Wavelet Toolbox MATLAB

- Вейвлеты и банки фильтров в системе MATLAB (Wavelets and Filter Banks);
- Непрерывное вейвлет-преобразование (Continuous Wavelet Analysis);
- Дискретный вейвлет-анализ (Discrete Wavelet Analysis);
- Вейвлет-пакеты (Wavelet Packets Analysis);
- Удаление шума (Denoising);
- Сжатие сигнала (Compression).

В данной главе мы подробно рассмотрим указанные разделы.

3.1. Вейвлеты и банки фильтров в системе MATLAB

В данном параграфе мы рассмотрим семейства вейвлетов, используемые в MATLAB, графики вейвлет-функций φ и ψ, вещественные и комплексные вейвлеты, ортогональные и биортогональные банки фильтров, лифтинг и построение вейвлетов.

3.1.1. Вещественные и комплексные вейвлеты

В этом разделе мы рассмотрим семейства вейвлетов, используемые в MatLab, графики вейвлет-функций φ и ψ и некоторые другие свойства. При знакомстве с вейвлетами удобно пользоваться главным меню пакета расширения Wavelet Toolbox (User GUI), которое вызывается по команде wavemenu. В командном режиме для получения

первоначальной информации о вейвлетах можно использовать следующие функции.

Информация о вейвлетах:

- waveinfo – информация о вейвлетах;
- waveletfamilies – семейства вейвлетов и члены семейств;
- wavefun – вейвлет и масштабирующая функция;
- wavefun2 – вейвлет и масштабирующая функция 2D;
- bswfun – биортогональные вейвлеты и масштабирующие функции;
- centfrq – центральная частота вейвлета;
- intwave – первообразная вейвлет-функции psi (ψ);
- scal2frq – пересчет масштаба в частоту;
- wavsupport – носитель вейвлета.

Специальные вейвлеты:

- cgauwavf – комплексный гауссов вейвлет;
- cmorwavf – комплексный вейвлет Морле;
- fbspwavf – комплексный частотный B-сплайновый вейвлет;
- gauswavf – гауссов вейвлет;
- mexihat – вейвлет «мексиканская шляпа»;
- meyer – вейвлет Мейера;
- meyeraux – вспомогательная функция вейвлета Мейера;
- morlet – вейвлет Морле;
- shanwavf – комплексный вейвлет Шеннона.

Дополнительные возможности управления вейвлетами:

- wavemenu – главное волновое вейвлет-меню, GUI;
- wavemngr – менеджер вейвлетов (Wavelet manager);
- wavedemo – демонстрационные примеры использования вейвлет-анализа в MATLAB (Wavelet Toolbox software examples);
- wtbxmngr – отображает текущую версию Wavelet Toolbox и позволяет установить параметры Wavelet Toolbox (Wavelet Toolbox manager).

Информация о вейвлетах

Рассмотрим подробнее функции первой группы, которые позволяют получить полезную информацию о вейвлетах.

1. Функция waveinfo – информация о вейвлетах.

Синтаксис:

waveinfo,
waveinfo('*wname*').

Функция **waveinfo** выдает следующую классификацию всех вейвлетов, используемых в Wavelet toolbox.

1. Предвейвлеты. Это гауссовы вейвлеты (gaus), вейвлеты Морле и мексиканская шляпа (mexihat). Они имеют минимальные свойства:

- функция φ не существует;
- анализ неортогональный;
- функция ψ имеет некомпактный носитель;
- восстановление не гарантируется.

Возможно непрерывное разложение.

Основные хорошие свойства: симметрия, ψ имеет явное выражение.

Основные трудности: быстрые алгоритмы и реконструкция невозможны.

2. Бесконечные регулярные вейвлеты Мейера (meyr) и дискретные вейвлеты Мейера (dmey). Свойства вейвлетов Мейера (meyr):

- функция φ существует и анализ ортогональный;
- функции ψ и φ не имеют аналитического выражения;
- функции ψ и φ не имеют компактного носителя.

Возможный анализ: непрерывное разложение, дискретное преобразование.

Основные хорошие свойства: симметрия, бесконечная дифференцируемость.

Основные трудности: быстрые алгоритмы недоступны.

Дискретные вейвлеты Мейера (dmey) – это аппроксимации конечной импульсной характеристики (FIR) к вейвлетам Мейера.

3. Ортогональные вейвлеты с компактным носителем. Это вейвлеты Добеши (dbN), симлеты (symN), койфлеты (coifN). Общие свойства:

- функция φ существует и анализ ортогональный;
- функции ψ и φ имеют компактный носитель;
- функция ψ имеет некоторое число нулевых моментов.

Возможный анализ: непрерывное разложение, дискретное преобразование с использованием быстрых алгоритмов.

Основные хорошие свойства: компактный носитель, обращающиеся в нуль моменты, FIR-фильтры.

Основные трудности: слабая регулярность.

Свойства симметрии:

- Добеши dbN несимметричны;
- симлеты symN почти симметричны;
- койфлеты coifN почти симметричны, и функция φ, как и ψ, имеет несколько нулевых моментов.

4. Биортогональные пары вейвлетов с компактным носителем. Это *B*-сплайновые биортогональные вейвлеты (biorNr.Nd и rbioNr. Nd).

Свойства:

- функция φ существует и анализ биортогональный;
- все функции $\tilde{\varphi}$, $\tilde{\psi}$ и φ, ψ для декомпозиции и реконструкции имеют компактный носитель;
- функции $\tilde{\varphi}$ и $\tilde{\psi}$ для разложения имеют нулевые моменты;
- функции ψ и φ для восстановления обладают некоторой регулярностью.

Возможный анализ: непрерывное разложение, дискретное преобразование с использованием быстрых алгоритмов.

Основные хорошие свойства: симметрия, FIR-фильтры.

Основные трудности: отсутствие ортогональности.

5. Комплексные вейвлеты. Это комплексные гауссовы вейвлеты (cgauN), комплексные вейвлеты Морле (cmorFb-Fc), комплексные вейвлеты Шеннона (shanFb-Fc), комплексные частотные B-сплайновые вейвлеты (fbspM-Fb-Fc). Минимальные свойства:

- функция φ не существует;
- анализ неортогональный;
- функция ψ с некомпактным носителем;
- свойство реконструкции не обеспечено.

Возможный анализ: комплексное непрерывное разложение. Основные хорошие свойства: симметрия, ψ имеет явное выражение. Основные трудности: быстрые алгоритмы и реконструкция невозможны.

Функция **waveinfo(**'*wname*'**)** дает более подробную информацию о семействе вейвлетов с именем *wname*.

2. Функция waveletfamilies – семейства вейвлетов и члены семейств. Употребляется в трех видах:

- waveletfamilies – отображает имена всех семейств вейвлетов, доступных в MATLAB (табл. 3.1.1);
- waveletfamilies('n') – отображает имена всех семейств вейвлетов, доступных в MATLAB, с именами вейвлетов в семействе;
- waveletfamilies('a') – отображает имена всех семейств вейвлетов, доступных в MATLAB, с именами вейвлетов в семействе и их свойствами.

Перечень всех семейств вейвлетов в MATLAB содержится в следующей таблице.

Таблица 3.1.1. Семейства вейвлетов в MATLAB

Краткое имя	Имя семейства вейвлетов
Haar	Вейвлеты Хаара
Db	Вейвлеты Добеши
Sym	Симлеты
Coif	Койфлеты
Bior	Биортогональные вейвлеты
Rbio	Дуальные биортогональные вейвлеты
Meyr	Вейвлеты Мейера
Dmey	Дискретные аппроксимации вейвлетов Мейера
Gaus	Гауссовы вейвлеты
Mexh	Вейвлет «мексиканская шляпа»
Morl	Вейвлеты Морле
Cgau	Комплексные гауссовы вейвлеты
Shan	Вейвлеты Шеннона
Fbsp	Частотные *B*-сплайновые вейвлеты
Cmor	Комплексные вейвлеты Морле

2. Функция wavefun – вейвлет-функции φ и ψ. Как известно, для вейвлетов с компактным носителем нет аналитических выражений для функций φ и ψ. Данная утилита позволяет найти приближения этих функций на достаточно мелкой сетке, которые можно использовать, например, для построения их графиков.

Для ортогональных вейвлетов и вейвлетов Майера. Функция записывается в виде:

[phi,psi,X] = wavefun('*wname*',N),

Входные данные:

- *wname* – имя вейвлета;
- N – число итераций вычисления, определяющее точность аппроксимации, шаг сетки $1/2^N$.

Выходные данные:

- X – массив значений независимой переменной x. Он состоит из точек, равномерно расположенных на носителе вейвлета с шагом 1/2N. Если носитель имеет длину S, то длина массива X равна S*2N + 1, то есть концевые точки носителя включаются;
- phi – значения функции φ(x), вычисленные в точках массива X;
- psi – значения функции ψ(*x*), вычисленные в точках массива X.

Для биортогоналных вейвлетов. Функция записывается в виде:

[phi1,psi1, phi2, psi2,X] = wavefun('*wname*',N),

Выходные данные:

- X – массив значений независимой переменной x;
- phi1 – значения функции $\tilde{\varphi}(x)$ для разложения в точках массива X;
- psi1 – значения функции $\tilde{\psi}(x)$ для разложения в точках массива X;
- phi2 – значения функции $\varphi(x)$ для восстановления в точках массива X;
- psi2 – значения функции $\psi(x)$ для восстановления в точках массива X.

Для вейвлетов без масштабирующей функции (Morlet, Mexican Hat, Gaussian или комплексных вейвлетов):

[psi,X] = wavefun ('*wname*',N)

wavefun ('*wname*') эквивалентна wavefun ('*wname*',8).

Алгоритм. Для нахождения функций φ и ψ используется обратное вейвлет-преобразование с одного уровня (см. описание команды wrcoef).

Рассмотрим сначала масштабирующую функцию φ. Поскольку $\varphi = \varphi_{00}$, то имеют место следующие свойства ортонормированности вейвлет-базиса:

$$(\varphi, \varphi_{0n}) = \delta_{0n}, \quad (\varphi, \psi_{jk}) = 0, \quad \forall j > 0, \quad \forall n, k.$$

Эти равенства можно рассматривать как вейвлет-разложение $\varphi(x)$. Все детализирующие коэффициенты – нулевые, аппроксимирующие – тоже нулевые, кроме одного значения, равного 1 при $n = 0$. Поэтому можно применять алгоритм **upcoef** восстановления для значений нахождения $\varphi(x)$. Поскольку в этом алгоритме используется процедура удвоения числа коэффициентов (обратная децимация), то сетка значений переменной x имеет шаг $1/2^N$, где N – число уровней восстановления.

Функция ψ находится аналогично.

Пример 1. Графики вейвлетов Добеши db4.

```
[phi,psi,x] = wavefun('db4',10);
subplot(121); plot(x,phi); title('y=\phi(x)');
subplot(122); plot(x,psi); title('y=\psi(x)');
```

Пример 2. Аппроксимации симлета sym4.

```
for i = 1:10
    [phi,psi,x] = wavefun('sym4',i);
    plot(x,psi);  hold on
end
```

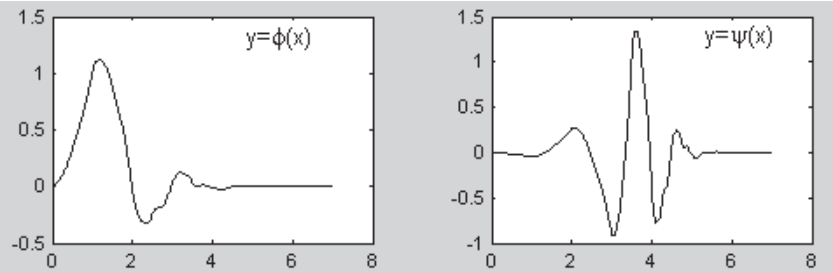


Рис. 3.1.1. Вейвлеты Добеши db4

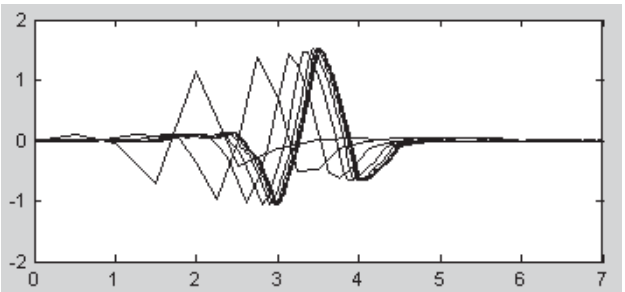


Рис. 3.1.2. Аппроксимации симлета sym4

Пример 3. Биортогональные вейвлеты типа 4 – 4.

```
[phi1,psi1, phi2, psi2,X] = wavefun('bior4.4',10);
subplot(221); plot(X, phi1); title('y=\phi1(x)');
subplot(222); plot(X, psi1); title('y=\psi1(x)');
subplot(223); plot(X, phi2); title('y=\phi2(x)');
subplot(224); plot(X, psi2); title('y=\psi2(x)');
```

3. Функция wavefun2 – вейвлет и масштабирующая функция в размерности 2. Для ортогонального вейвлета 'wname' wavefun2 возвращает масштабирующую функцию и три вейвлета, которые получаются из тензорного произведения масштабирующей функции и вейвлета (см. раздел 2.11).

4. Функция bswfun – биортогональные вейвлеты и масштабирующие функции.

Синтаксис:

[PHIS,PSIS,PHIA,PSIA,XVAL] =
bswfun(LoD,HiD,LoR,HiR,ITER)

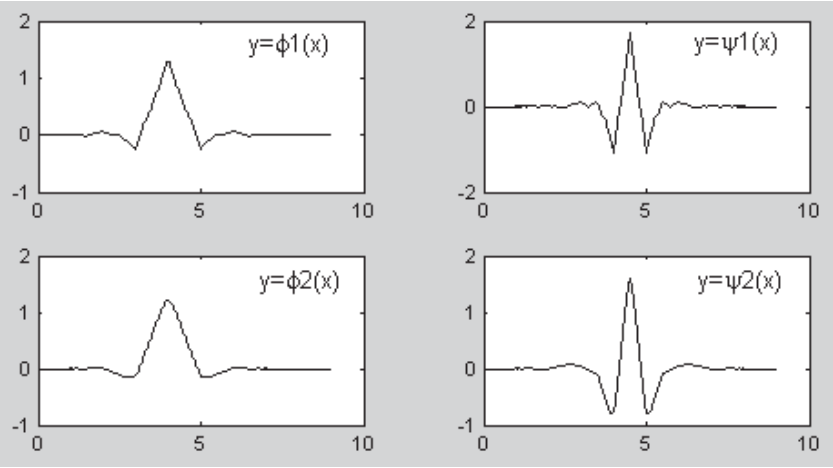


Рис. 3.1.3. Биортогональные вейвлеты $\tilde{\varphi}$, $\tilde{\psi}$ и φ, ψ

Функция возвращает приближения на сетке XVAL двух пар, (PHIA,PSIA) и (PHIS,PSIS), состоящих из масштабирующей функции и вейвлета, ассоциированных с двумя парами фильтров (LoD,HiD), (LoR,HiR). Подробнее о биортогональных вейвлетах см. в разделе 2.10.

5. Функция centfrq – центральная частота вейвлета. Находит *основную частоту* FREQ в герцах функции ψ вейвлета по имени '*wname*'.

Синтаксис:

FREQ = centfrq('*wname*'), FREQ = centfrq('*wname*',N),
[FREQ,X,RECFREQ] = centfrq('*wname*',N,'plot').

Здесь N есть число итераций, выполняемых функцией wavefun при вычислении вейвлета (по умолчанию N = 8).

Функция [FREQ,X,RECFREQ]=centfrq('*wname*',N,'plot') находит дополнительно гармонику RECFREQ, соответствующую основной частоте в точках массива независимой переменной X. Строит графики вейвлет-функции ψ и основной гармоники RECFREQ. Основная частота FREQ соответствует максимуму модуля дискретного преобразования Фурье fft вейвлета.

Пример 4.

```
wname = 'db2'; iter = 8; cfreq = centfrq(wname,8,'plot')
cfreq =
    0.6667
```

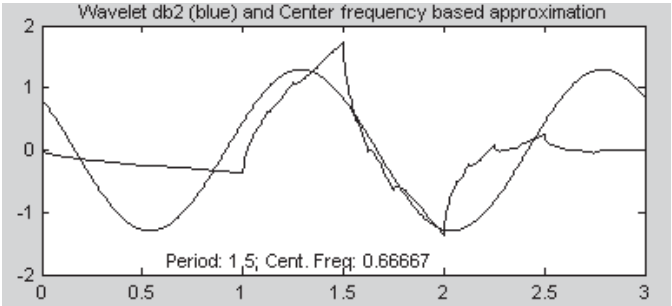


Рис. 3.1.4. Вейвлет Добеши db2
и его центральная частота

В данном случае период T равен 1.5, а частота $F = 1/T = 0.66667$. Гармоника $\sin(\omega x)$, соответствующая основной частоте, имеет вид $\sin(2\pi x/T) = \sin(1.33333\pi x)$.

6. Функция scal2frq – пересчет масштаба в частоту. Находит псевдочасту в герцах, соответствующую заданному масштабу A, вейвлету 'wname' и шагу выборки DELTA.

Синтаксис:

F = scal2frq(A,'wname',DELTA)
scal2frq(A,'wname')

По вейвлету 'wname' вычисляется центральная частота F_c, затем применяется формула $F_a = \dfrac{F_c}{a\,\Delta}$. Если мы ассоциируем с вейвлетом $\psi(x)$ частоту F_c, то для масштабированной функции $\psi(ax)$ соответствует частота F_c/a. Кроме того, центральная частота вейвлета соответствует шагу выборки $\Delta = 1$, при другом шаге выборки Δ анализируемого массива частоту также следует взять с учетом Δ.

7. Функция intwave – первообразная вейвлет-функции ψ. Вычисляет *первообразную* функции ψ как интеграл функции ψ вейвлета '*wname*' от $-\infty$ до переменной точки x.

Синтаксис:

[integ,X] = intwave('*wname*',N),
[integ,X] = intwave('*wname*').

Параметры:

- N – количество итераций (по умолчанию N = 8);
- X – массив значений переменной x. Он состоит из точек, равномерно расположенных на носителе вейвлета с шагом $1/2^N$. Если

носитель имеет длину S, то длина массива X равна $S*2^N+1$, то есть концевые точки носителя включаются;

- integ – значения первообразной функции ψ в точках $x \in X$. Функция ψ аппроксимируется на $S*2^N+1$ точках сетки X. Выходной массив integ является действительным или комплексным в зависимости от типа вейвлета.

Для биортогональных вейвлетов функция [intdec,x,intrec] = = intwave('*wname*',N) вычисляет интегралы intdec и intrec вейвлет-функций разложения $\tilde{\psi}(x)$ и восстановления $\psi(x)$.

[integ,X] = intwave('*wname*') эквивалентно [integ,X] = intwave('*wname*',8).

Пример 5.

```
[phi,psi,xval] = wavefun('db4',7);
subplot(211); plot(xval,psi); title('Wavelet');
[integ,xval] = intwave('db4',7);
subplot(212); plot(xval,integ);
title(['Wavelet integrals over [-Inf x] for each value of xval']);
```

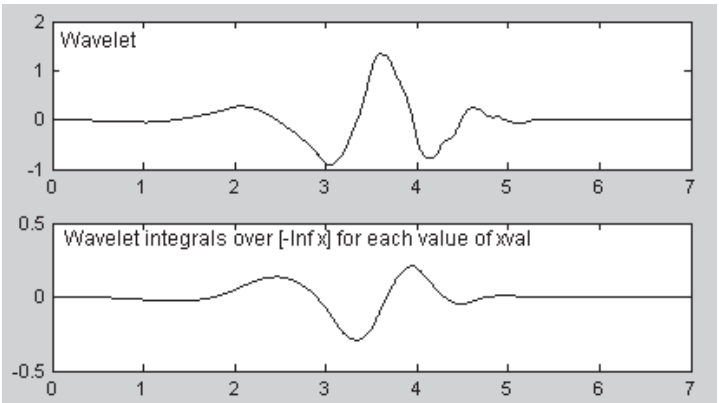


Рис. 3.1.5. Вейвлет Добеши db4 и его первообразная

8. Функция wavsupport – носитель вейвлета.
Синтаксис:

[LB,UB] = wavsupport(wname)

Возвращает нижнюю границу LB и верхнюю границу UB носителя вейвлета wname. Для ортогональных и биортогональных вейвле-

тов нижние и верхние границы равны –0.5*(LF – 1) и 0.5*(LF – 1), где LF – длина фильтра вейвлета.

Специальные вейвлеты

В данном разделе мы обсудим функции, которые позволяют вычислить значения масштабирующей функции и вейвлета для некоторых специальных типов вейвлетов. Все они имеют, вообще говоря, бесконечный носитель, но вычисляются на так называемом эффективном носителе, вне которого значениями функций можно пренебречь.

1. Функция meyer – вейвлет Мейера. Вычисляет значения масштабирующей функции и вейвлет-функции Майера на *n* точках регулярной сетки *x* на интервале [*a*, *b*].

Синтаксис:

[phi,psi,x] = meyer(a,b,n),
[phi,x] = meyer(a,b,n,'phi'), [psi,x] = meyer(a,b,n,'psi').

Число *n* должно быть степенью двойки. Эти функции имеют эффективный носитель на [–8, 8]. Вторая строчка соответствует случаю, когда требуется только одна функция.

Пример 6. Вычисление вейвлет-функций Майера на промежутке [–8, 8] и построение графика.

```
a = -8; b = 8; n = 1024;
[phi,psi,x] = meyer(a,b,n);
subplot(211); plot(x,psi);  title('Meyer wavelet')
subplot(212); plot(x,phi);  title('Meyer scaling function')
```

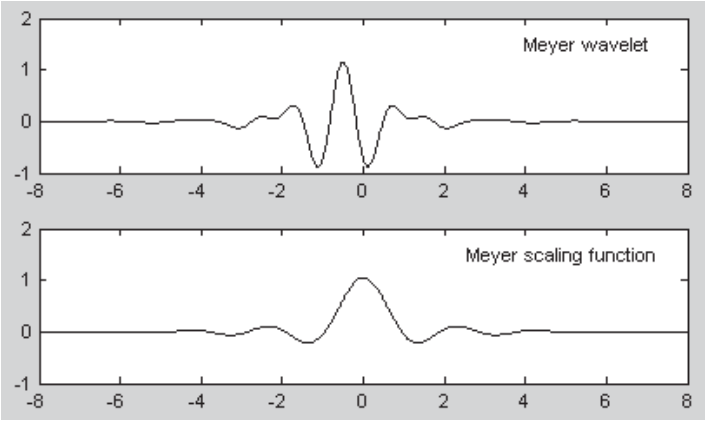


Рис. 3.1.6. Вейвлет-функции Мейера

2. Команда meyeraux – вспомогательная функция вейвлета Мейера.

Синтаксис:

Y = meyeraux(X).

Утилита Y = meyeraux(X) вычисляет значения на векторе или матрице X вспомогательной функции, используемой при определении вейвлета Майера. В MatLab эта функция имеет вид:

$\nu(x) = x^4(35 - 84x + 70x^2 - 20x^3), \quad x \in [0, 1].$

Ниже дано описание функций вейвлетов, для задания которых требуются дополнительные параметры, по сравнению с общей командой wavefun.

3. Функция gauswavf – гауссов вейвлет.

Синтаксис:

[psi,x] = gauswavf(a,b,n,p).

Эта команда вычисляет значения p-й производной функции Гаусса

$$F(x) = C_p e^{-x^2}$$

на n точках регулярной сетки интервала $[a, b]$. Константа C_p выбирается такой, чтобы L_2-норма p-й производной функции F была равна 1. На выходе получается функция psi, вычисленная на сетке x. Эти вейвлеты имеют эффективный носитель на [–5, 5].

[psi,x] = gauswavf(a,b,n) эквивалентно [psi,x] = gauswavf(a,b,n,1).

4. Функция mexihat – вейвлет «мексиканская шляпа».

Синтаксис:

[psi,x] = mexihat(a,b,n).

Функция вычисляет значения вейвлета «*мексиканская шляпа*»

$$\psi(x) = \left(\frac{2}{\sqrt{3}}\pi^{-1/4}\right)(1 - x^2)e^{-x^2/2}$$

на n точках регулярной сетки интервала $[a, b]$. На выходе получается функция psi, вычисленная на сетке x. Этот вейвлет имеет эффективный носитель на [–5, 5]. Функция ψ пропорциональна второй производной функции Гаусса.

5. Функция morlet – вейвлет Морле.

Синтаксис:

[psi,X] = morlet(a,b,n).

Функция вычисляет значения *вейвлета Морле* $\psi(x) = e^{-x^2/2}\cos(5x)$ на *n* точках регулярной сетки на промежутке [*a*, *b*]. На выходе получаются функция psi, вычисленная на сетке X, и сама сетка X. Эффективный носитель вейвлета Морле есть промежуток [–4, 4].

Пример 7. Вейвлет Морле на промежутке [–4, 4].

```
lb = -4; ub = 4; n = 1000;
[psi,x] = morlet(lb,ub,n);
plot(x,psi), title('Morlet wavelet')
```

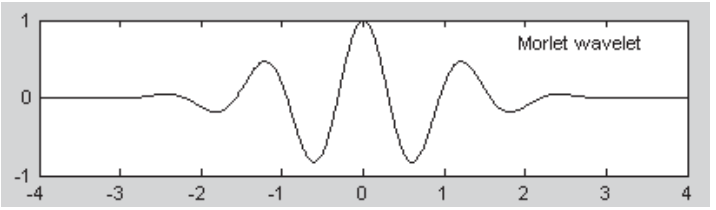


Рис. 3.1.7. Вейвлет Морле

6. Функция cgauwavf – комплексный гауссов вейвлет.
Синтаксис:

[psi,X] = cgauwavf(a,b,n,p).

Функция вычисляет значения *p*-й производной комплексной функции Гаусса

$$\psi(x) = C_p e^{-ix} e^{-x^2}$$

на *n* точках регулярной сетки интервала [*a*, *b*]. Константа C_p выбирается такой, чтобы L_2-норма *p*-й производной функции *F* была равна 1. На выходе получается функция psi, вычисленная на сетке X. Эти вейвлеты имеют эффективный носитель на [–5, 5].

[psi,X] = cgauwavf(a,b,n) эквивалентно [psi,X] = cgauwavf (a,b,n,1).

7. Функция cmorwavf – комплексный вейвлет Морле.
Синтаксис:

[psi,X] = cmorwavf(a,b,n,B,C).

Функция вычисляет значения комплексного вейвлета Морле, определенного шириной полосы частот *B*, центральной частотой *C* и выражением

$$\psi(x) = \frac{1}{\sqrt{\pi B}} e^{i2\pi Cx} e^{-x^2/B}$$

на n точках регулярной сетки на интервале $[a, b]$. На выходе получается функция psi, вычисленная на сетке X.

Пример 8. Вычисление комплексного вейвлета Морле cmor1-1.5 на промежутке $[-8, 8]$ и построение графиков действительной и мнимой частей.

```
fb = 1; fc = 1.5; a = -8; b = 8; n = 1000;
[psi,x] = cmorwavf(a,b,n,fb,fc);
subplot(211); plot(x,real(psi));  title('Complex Morlet wavelet cmor1-1.5')
xlabel('Real part') ; grid
subplot(212); plot(x,imag(psi)); xlabel('Imaginary part'); grid
```

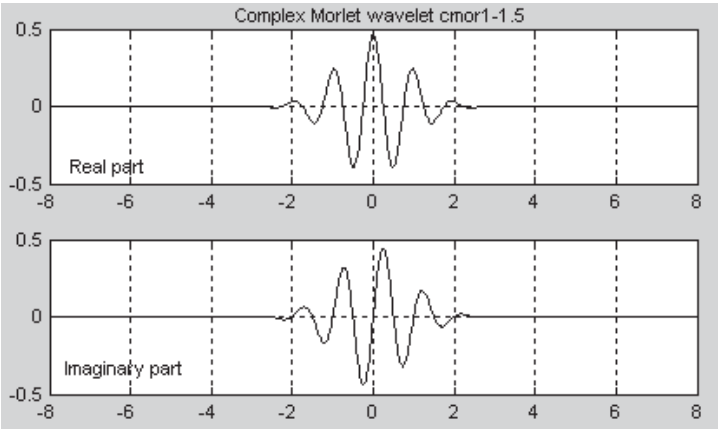


Рис. 3.1.8. Комплексный вейвлет Морле

8. Функция fbspwavf – комплексный частотный B-сплайновый вейвлет.

Синтаксис:

[psi,X] = fbspwavf(a,b,n,M,B,C).

Функция вычисляет значения комплексного частотного B-сплайнового вейвлета, определенного параметром M (M – целое, ≥ 1), шириной полосы частот B, центральной частотой C и выражением

$$\psi(x) = \sqrt{B}\left(\frac{\sin Bx/M}{Bx}\right)^{M} e^{i2\pi Cx}$$

на n точках регулярной сетки на интервале $[a, b]$. Параметры B и C должны быть такими, что $C > B/2 > 0$. На выходе получается функция psi, вычисленная на сетке X.

Пример 9. Вычисление комплексного B-сплайнового вейвлета fbsp 2-1-0.5 на промежутке [–10, 10] и построение графиков действительной и мнимой частей.

```
m = 2; fb = 1; fc = 0.5; a = -10; b = 10; n = 1000; [psi,x] =
fbspwavf(a,b,n,m,fb,fc);
subplot(211); plot(x,real(psi)); title('Complex Frequency B-Spline
wavelet fbsp2-1-0.5')
xlabel('Real part'), grid
subplot(212); plot(x,imag(psi)); xlabel('Imaginary part'), grid
```

9. Функция shanwavf – комплексный вейвлет Шеннона.

Синтаксис:

[psi,X] = shanwavf(a,b,n,B,C).

Функция вычисляет значения комплексного вейвлета Шеннона, определенного шириной полосы частот B, центральной частотой C и выражением

$$\psi(x) = \sqrt{B}\,\frac{\sin Bx}{Bx}\,e^{i2\pi Cx}$$

на n точках регулярной сетки на интервале $[a, b]$. Параметры B и C должны быть такими, что $C > B/2 > 0$. На выходе получается функция psi, вычисленная на сетке X.

Пример 10. Вычисление комплексного вейвлета Шеннона shan1-1.5.

```
fb = 1; fc = 1.5; a = -10; b = 10; n = 1000;
[psi,x] = shanwavf(a,b,n,fb,fc);
subplot(211); plot(x,real(psi)); title('Complex Shannon wavelet shan1-1.5')
xlabel('Real part')
subplot(212); plot(x,imag(psi)); xlabel('Imaginary part')
```

Дополнительные возможности управления вейвлетами:

- wtbxmngr – отображает текущую версию пакета Wavelet Toolbox и позволяет установить некоторые параметры Wavelet Toolbox;
- wavedemo – вызов демонстрационных примеров использования вейвлет-анализа в MATLAB;
- wavemngr – менеджер вейвлетов. Позволяет добавлять, удалять, восстанавливать или загружать вейвлеты;
- wavemenu – главное вейвлет-меню, GUI (ему посвящена последняя глава книги).

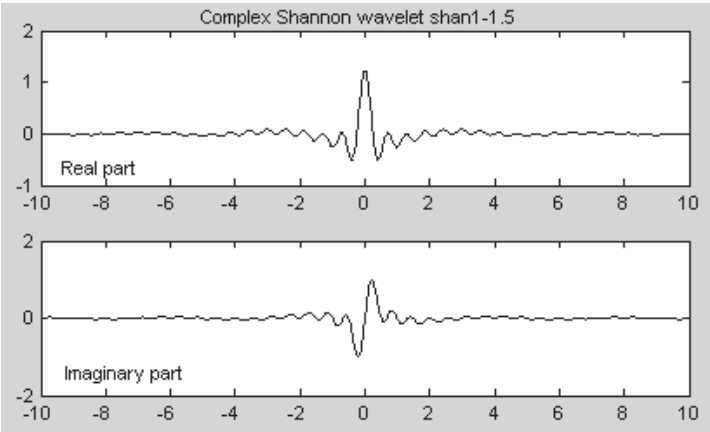


Рис. 3.1.9. Комплексный вейвлет Шеннона

3.1.2. Ортогональные и биортогнальные банки фильтров

С каждым вейвлетом связано несколько *фильтров*, которые используются в практических применениях. Это, во-первых, *масштабирующий фильтр* H, состоящий из коэффициентов $\{h_n\}$, $n \in \mathbf{Z}$, участвующих в основном уравнении кратномасштабного анализа: $\varphi(x) = \sqrt{2}\sum_n h_n \varphi(2x-n)$, – и фильтр вейвлета $g_n = (-1)^n \overline{h}_{N-n}$ для равенства $\psi(x) = \sqrt{2}\sum_n g_n \varphi(2x-n)$. При этом в ортогональном случае для восстановления используются фильтры $\{h_n\}$ и $\{g_n\}$ вейвлетов $\varphi(x)$ и $\psi(x)$, а для вейвлет-разложения – сопряженные (транспонированные) фильтры $h_n^* = \overline{h_{-n}}$ и $g_n^* = \overline{g_{-n}}$.

Масштабирующий фильтр $\{h_n\}$ называется низкочастотным и обозначается далее Lo_R, а фильтр $\{g_n\}$ – высокочастотным и обозначается Hi_R. Сопряженные фильтры разложения обозначаются Lo_D и Hi_D. Таким образом, Lo_D $= h_n^* = \overline{h_{-n}}$ и Hi_D $= g_n^* = \overline{g_{-n}}$.

В биортогональном случае для разложения сигнала на низкочастотную и высокочастотную составляющие используются два фильтра разложения Lo_D и Hi_D. При реконструкции сигнала используются еще два фильтра Lo_R и Hi_R. В данном разделе рассматриваются функции, которые позволяют получить эти фильтры вейвлетов:

- **dbwavf**, **dbaux** – вейвлет-фильтр Добеши;
- **symwavf**, **symaux** – фильтр симлета;
- **coifwavf** – фильтр койфлета;
- **biorwavf** – фильтры биортогонального сплайнового вейвлета;
- **rbiowavf** – обратные биортогональные сплайновые фильтры;
- **qmf** – квадратурный зеркальный фильтр;
- **wrev** – перестановки элементов вектора в обратном порядке;
- **wfilters** – фильтры вейвлета;
- **orthfilt** – набор фильтров разложения и восстановления ортогонального вейвлета;
- **biorfilt** – набор фильтров биортогонального вейвлета.

Кроме того, можно использовать некоторые функции предыдущего раздела 3.2.1 и функцию Y = wrev(X) перестановки элементов вектора X в обратном порядке. Рассмотрим сначала функции, которые вычисляют масштабирующий фильтр W вейвлета, а затем функции для нахождения фильтров разложения и восстановления.

Масштабирующие фильтры

Это масштабирующие фильтры W вейвлетов, то есть коэффициенты масштабирующего уравнения в виде $\varphi(x) = 2\sum_{n\in Z} w_n \varphi(2x-n)$, где φ – масштабирующая функция вейвлета. Отметим, что они отличаются от классических коэффициентов $\{h_n\}$ масштабирующего уравнения $\varphi(x) = \sqrt{2}\sum_{n\in Z} h_n \varphi(2x-n)$. В частности, $w_n = 1/\sqrt{2}\, h_n$.

1. Функция dbwavf – вейвлет-фильтр Добеши.
Синтаксис:

W = dbwavf('dbN')

Функция находит масштабирующий фильтр W вейвлета Добеши 'dbN', где N = 1, 2, ..., 45. Функция W = dbaux(N,SUMW) вычисляет масштабирующий фильтр W вейвлета Добеши 'dbN' так, что sum(W) = SUMW.

2. Функция symwavf – фильтр симлета. Симлеты – это почти симметричные вейвлеты Добеши. Они характеризуются минимальностью фазы частотной функции $H_0(\omega)$.

Синтаксис:

W = symwavf('symN')

Функция находит масштабирующий фильтр W симлета, заданного строкой 'symN'. Возможные значения N: 2, 3, ..., 45. Функция

W = symaux (N,SUMW) вычисляет масштабирующий фильтр W симлета порядка N так, что sum(W) = SUMW.

3. Функция coifwavf – фильтр койфлета.

Синтаксис:

W = coifwavf('coifN').

Функция coifwavf находит масштабирующий фильтр W вейвлета Coiflet, заданного строкой 'coifN'. Возможные значения N: 1, 2, 3, 4, или 5.

4. Функция biorwavf – фильтры биортогонального сплайнового вейвлета.

Синтаксис:

[RW,DW] = biorwavf('biorNr.Nd')

Биортогональные вейвлеты порождены двумя масштабирующими функциями $\tilde{\varphi}(x)$ и $\varphi(x)$ с компактным носителем. Эти функции $\tilde{\varphi}(x)$ и $\varphi(x)$ имеют разные свойства регулярности. Обычно функция $\varphi(x)$ более регулярна (B-сплайн), чем $\tilde{\varphi}(x)$. Поэтому для разложения используются вейвлеты, порожденные функцией $\tilde{\varphi}(x)$, а для восстановления – функцией $\varphi(x)$. Соответствующие фильтры масштабирующих функций $\tilde{\varphi}(x)$ и $\varphi(x)$ называются масштабирующими фильтрами разложения и восстановления.

Функция [RW,DW] = biorwavf('biorNr.Nd') находит два масштабирующих фильтра:

- RW – фильтр вейвлета $\varphi(x)$, используемого для восстановления;
- DW – фильтр вейвлета $\tilde{\varphi}(x)$, используемого для разложения.

Тип биортогональных вейвлетов задается строкой вида biorNr.Nd, где значения параметров Nr и Nd могут быть такими:

Таблица 3.1.2. Параметры биортогональных вейвлетов

Первый параметр Nr	Второй параметр Nd
Nr = 1	Nd = 1, 3, 5
Nr = 2	Nd = 2, 4, 6, 8
Nr = 3	Nd = 1, 3, 5, 7
Nr = 4	Nd = 4
Nr = 5	Nd = 5
Nr = 6	Nd = 8

5. Функция rbiowavf – обратные биортогональные сплайновые масштабирующие фильтры.

Синтаксис:

[RW,DW]= rbiowavf('rbiorNr.Nd')

В отличие от предыдущей команды, в данном случае вейвлеты $\tilde{\varphi}(x)$ и $\varphi(x)$ меняются ролями: для разложения используется $\varphi(x)$, а для восстановления – $\tilde{\varphi}(x)$. Значения параметров Nr и Nd – те же самые.

Фильтры разложения и восстановления

Из масштабирующих фильтров вейвлетов можно получить еще фильтры, используемые при разложении сигнала и при его восстановлении. Напомним, что для разложения используются сопряженные (транспонированные) фильтры, а для восстановления – фильтры вейвлетов $\varphi(x)$ и $\psi(x)$.

6. Функция orthfilt – набор фильтров разложения и восстановления ортогонального вейвлета.

Синтаксис:

[Lo_D,Hi_D,Lo_R,Hi_R] = orthfilt(W).

Вычисляет четыре фильтра, соответствующие масштабирующему фильтру W ортогонального вейвлета:

- Lo_D – *низкочастотный фильтр разложения*;
- Hi_D – *высокочастотный фильтр разложения*;
- Lo_R – *низкочастотный фильтр восстановления*;
- Hi_R – *высокочастотный фильтр восстановления*.

Алгоритм. Пусть $\{w_n\}$ – масштабирующий фильтр вейвлета,

$$\varphi(x) = 2\sum_{n\in Z} w_n \varphi(2x-n),$$

где φ – масштабирующая функция ортогонального вейвлета. Если φ имеет компактный носитель длины N, то фильтр $\{w_n\}$ имеет $N+1$ ненулевых членов. Сумма элементов фильтра равна 1, l^2-норма равна $1/\sqrt{2}$ и $w_n = 1/\sqrt{2}\, h_n$.

Фильтры разложения и восстановления вычисляются через $\{w_n\}$ следующим образом:

- Lo_R – низкочастотный фильтр восстановления $\{h_n\}$, формула для его вычисления: Lo_R=W/norm(W);

- Hi_R – высокочастотный фильтр восстановления находится из формулы $g_n = (-1)^n \overline{h}_{N-n}$ как квадратурный зеркальный фильтр для Lo_R: Hi_R=qmf(Lo_R);
- Lo_D – низкочастотный фильтр разложения, находится как сопряженный к Lo_ R, то есть перестановкой членов в обратном порядке: Lo_D = wrev(Lo_R);
- Hi_D – высокочастотный фильтр разложения: Hi_D= = wrev(Hi_R) , то есть Hi_ D берется в обратном порядке к Hi_R.

Замечание. Перед использованием функции масштабирующий фильтр W должен быть загружен. Правильное применение этой команды:

```
load sym5; w = sym5;
[Lo_D,Hi_D,Lo_R,Hi_R] = orthfilt(w);
```

Неправильное использование этой команды:

```
[Lo_D,Hi_D,Lo_R,Hi_R] = orthfilt('sym5');
```

7. Функция biorfilt – набор фильтров биортогонального вейвлета.

Синтаксис:

[Lo_D,Hi_D,Lo_R,Hi_R] = biorfilt(DW,RW),
[Lo_D1,Hi_D1,Lo_R1,Hi_R1,Lo_D2,Hi_D2,Lo_R2,Hi_R2] = biorfilt(DW,RW,'8').

Функция вычисляет либо четыре, либо восемь фильтров, ассоциированных с биортогональными вейвлетами. Первая функция [Lo_D,Hi_D,Lo_R,Hi_R] = biorfilt(DW,RW) вычисляет четыре фильтра биортогонального вейвлета с масштабирующими фильтрами DW и RW. Вторая команда вычисляет восемь фильтров, из которых первые четыре связаны с масштабирующим фильтром декомпозиции DW функции $\tilde{\varphi}(x)$, а следующие четыре – с фильтром реконструкции RW функции $\varphi(x)$.

8. Функция wfilters – фильтры вейвлета.

Синтаксис:

[Lo_D,Hi_D,Lo_R,Hi_R] = wfilters('*wname*'),
[F1,F2] = wfilters('*wname*','*type*').

Первая функция [Lo_D,Hi_D,Lo_R,Hi_R] = wfilters('*wname*') вычисляет фильтры разложения и восстановления ортогонального или

биортогонального вейвлета *wname*. Возможные значения *wname* перечислены в таблице.

Таблица 3.1.3. Фильтры вейвлетов

Семейство вейвлетов	Вейвлеты
Daubechies	'db1' or 'haar', 'db2', ... ,'db10', ... , 'db45'
Coiflets	'coif1', ... , 'coif5'
Symlets	'sym2', ... , 'sym8', ... ,'sym45'
Discrete Meyer	'dmey'
Biortogonal	'bior1.1', 'bior1.3', 'bior1.5' 'bior2.2', 'bior2.4', 'bior2.6', 'bior2.8' 'bior3.1', 'bior3.3', 'bior3.5', 'bior3.7','bior3.9', 'bior4.4', 'bior5.5', 'bior6.8'
Reverse Biortogonal	'rbio1.1', 'rbio1.3', 'rbio1.5' 'rbio2.2', 'rbio2.4', 'rbio2.6', 'rbio2.8' 'rbio3.1', 'rbio3.3', 'rbio3.5', 'rbio3.7', 'rbio3.9', 'rbio4.4', 'rbio5.5', 'rbio6.8'

Вторая функция [F1,F2] = wfilters('*wname*','*type*') вычисляет следующие фильтры:

Таблица 3.1.4. Типы фильтров

Фильтры	Тип фильтров	Параметр
Lo_D и Hi_D	Фильтры разложения	If '*type*' = 'd'
Lo_R и Hi_R	Фильтры восстановления	If '*type*' = 'r'
Lo_D и Lo_R	Низкочастотные фильтры	If '*type*' = 'l'
Hi_D и Hi_R	Высокочастотные фильтры	If '*type*' = 'h'

9. Функция qmf – квадратурный зеркальный фильтр. Два фильтра F_0 и F_1 называются *квадратурными* зеркальными фильтрами (QMF), если для любого сигнала x конечной энергии выполняется равенство l^2-норм:

$$\|y_0\|^2 + \|y_1\|^2 = \|x\|^2,$$

где y_0 – фильтрованный сигнал с применением децимации, то есть если $x_0 = F_0(x)$, то $y_0(n) = x_0(2n)$, аналогично определяется y_1, $y_1(n) = x_1(2n)$, где $x_1 = F_1(x)$.

Синтаксис:

Y = qmf(X,P), Y = qmf(X)

В первом случае функция Y = qmf(X,P) меняет знак у членов с четными индексами и переставляет в обратном порядке коэффициенты

фильтра X, то есть $Y(k) = (-1)^k X(2N + 1 - k)$, для $k = 1, 2, \ldots, 2N$, если P – четное. Если P – нечетное, делается то же самое, только знаки меняются у нечетных элементов.

Y = qmf(X) эквивалентно Y = qmf(X,0).

3.1.3. Построение вейвлетов. Лифтинг

В системе MATLAB предлагается несколько способов построения новых вейвлетов: это построение вейвлетов «по образцу» для непрерывного вейвлет-преобразования и схема лифтинга. В пакете Wavelet Toolbox имеются функция **pat2cwav** для построения вейвлетов «по образцу» и следующие *функции лифтинга*:

- **addlift** – добавление основных или дуальных элементарных шагов схемы лифтинга;
- **displs** – отображение схемы лифтинга;
- **lsinfo** – информация о схеме лифтинга;
- **filt2ls** – преобразование четверки фильтров в схему лифтинга LS;
- **liftfilt** – применение элементарных шагов лифтинга на четверке фильтров;
- **ls2filt** – преобразование схемы лифтинга LS в четверку фильтров;
- **liftwave** – нахождение схемы лифтинга LS, связанной с вейвлетом;
- **wave2lp** – полиномы Лорана, ассоциированные с вейвлетом;
- **wavenames** – имена вейвлетов, доступные для LWT;
- **laurmat** – конструктор матриц Лорана;
- **laurpoly** – конструктор полиномов Лорана.

Для прямого и обратного вейвлет-преобразования лифтинга (LWT) используются функции:

- **lwt** – 1D-вейвлет-преобразование лифтинга;
- **lwt2** – 2D-вейвлет-преобразование лифтинга;
- **lwtcoef** – извлечение или восстанавление вейвлет-коэффициентов LWT;
- **lwtcoef2** – извлечение или восстанавление вейвлет-коэффициентов 2D LWT;
- **ilwt** – обратное 1D-вейвлет-преобразование лифтинга;
- **ilwt2** – обратное 2D-вейвлет-преобразование лифтинга.

Подробнее о лифтинге см. [M] и документацию MATLAB.

Построение вейвлетов «по образцу»

Рассмотрим возможность создания *новых допустимых вейвлетов* для непрерывного вейвлет-преобразования. Это можно сделать как из командной строки MATLAB при помощи функции **pat2cwav**, так и при помощи главного меню пакета Wavelet Toolbox. Здесь мы рассмотрим проектирование из командной строки.

Иногда бывает так, что анализируемый сигнал содержит достаточно характерные всплески, например комплекс QRS кардиосигнала, или другой всплеск, характеризующий сердечную патологию. Для обнаружения таких характерных колебаний при помощи непрерывного вейвлет-преобразования желательно использовать вейвлет, который имеет вид этого характерного всплеска. Однако имеются некоторые проблемы. Вейвлет для непрерывного вейвлет-преобразования должен иметь нулевой интеграл, а характерная волна, которую мы хотим обнаружить, может иметь ненулевой интеграл. Для решения этой проблемы Wavelet Toolbox предлагает функцию pat2cwav, которая находит достаточно регулярное приближение с нулевым интегралом к указанной волне методом наименьших квадратов.

Функция pat2cwav – нахождение вейвлета по образцу.

Синтаксис:

```
[PSI,XVAL,NC] = pat2cwav(YPAT,METHOD,POLDEGREE,REG
ULARITY)
```

Описание. Данная функция вычисляет допустимый вейвлет нормы, равной 1, для CWT (заданный массивами XVAL и PSI), приближенный к образцу, определенному вектором YPAT.

Значения аргумента x для образца устанавливаются как равномерно распределенные на отрезке [0, 1]:

```
xval = linspace(0,1,length(YPAT))
```

Константа NC является такой, что NC*PSI аппроксимирует YPAT на интервале [0, 1] по методу наименьших квадратов, используя:

- полином степени POLDEGREE, когда METHOD равен 'polynomial';
- проекцию на пространство функций, ортогональных к константам, когда METHOD равен 'othconst'.

Параметр REGULARITY определяет граничные условия в точках 0 и 1. Допустимые значения: 'continuous', 'differentiable' и 'none'.

Когда METHOD равен 'polynomial', то:

- если REGULARITY равен 'continuous', то POLDEGREE должен быть ≥ 3;

- если REGULARITY равен 'differentiable', то POLDEGREE должен быть ≥ 5.

Пример 1. Рассмотрим кардиосигнал. Легко видеть, что комплекс QRS кардиосигнала является типичным всплеском (типы комплексов QRS зависят от выбора отведения и от состояния пациента). Поэтому при вейвлет-анализе естественно использовать вейвлет, который является хорошим приближением к выбранному типу комплекса QRS. Сначала загрузим фрагмент кардиосигнала и выберем комплекс QRS.

```
   % Программа для просмотра ЭКГ
load SmCh4.txt
   %Выборка фрагментов
S1=SmCh4(601:1800);
S=SmCh4(1001:1140);
N1=length(S1); % Длина данных
N=length(S);
   %Графики фрагментов
subplot 121; plot(S1); axis([0 N1 min(S1) max(S1)]);
subplot 122; plot(S); axis([0 N min(S) max(S)]);
save SmQRS.txt -ascii S %Сохранение выбранного комплекса
```

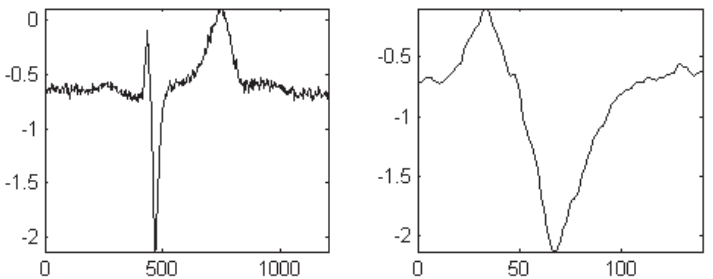


Рис. 3.1.10. Кардиосигнал и его комплекс QRS

Выбранный комплекс QRS рассматриваем в качестве образца. Загружаем этот образец и применяем команду pat2cwav для нахождения адаптированного вейвлета. В конце работы сохраняем результаты в mat-файле WavSmQRS.mat.

```
   % Загрузка образца
load SmQRS.txt;
   % Сдвиг, для того чтобы начало и конец волны были около нуля
Y=SmQRS + 0.7;
```

```
[psi_p,xval_p,nc_p] = pat2cwav(Y, 'polynomial',10, 'continuous') ;
[psi_o,xval_o,nc_o] = pat2cwav(Y, 'orthconst',2, 'continuous') ;
subplot 121; plot(xval_p,Y,xval_p,nc_p*psi_p,'.')
title('polynomial')
subplot 122; plot(xval_o,Y,xval_o,nc_o*psi_o,'.')
title('orthconst')
savefile = 'WavSmQRS.mat';
save(savefile, 'psi_p','xval_p','nc_p','psi_o','xval_o','nc_o')
savefile = 'Patt_SmQRS.mat';
save(savefile, 'Y')
```

Полученные вейвлеты вместе с оригинальными образцами изображены на рис. 3.1.11. Рисунок справа показывает, что метод 'othconst' при значении параметров poldegree=2 и regularity = 'continuous' гораздо точнее отражает особенности образца.

Предложенная схема построения адаптированного вейвлета может успешно применяться в диагностических целях для автоматического обнаружения характерных колебаний кардиосигнала методами вейвлет-анализа cwt.

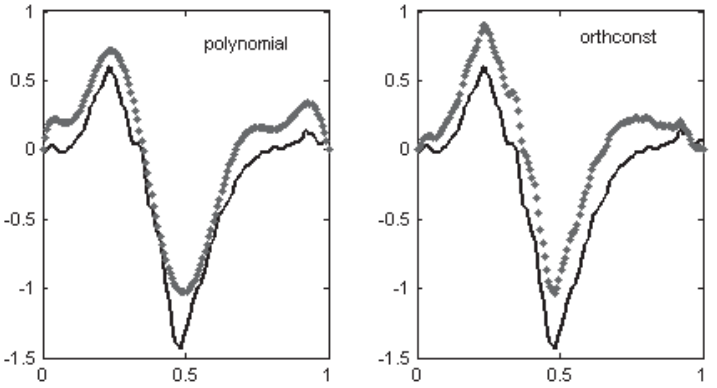


Рис. 3.1.11. Вейвлет,
адаптированный к комплексу QRS кардиосигнала

Схема лифтинга

Рассмотрим *методы лифтинга* проектирования новых вейвлетов. Сначала напомним кратко теоретические основы *лифтинга* (улучшения).

Пусть разложение сигнала производится вейвлетами $\varphi(x)$ и $\psi(x)$, а восстановление – вейвлетами $\tilde{\varphi}(x)$ и $\tilde{\psi}(x)$, вообще говоря, биортого-

нальными. Пусть $H(z)$ и $G(z)$ – частотные функции вейвлетов $\varphi(x)$ и $\psi(x)$ и $\tilde{H}(z)$ и $\tilde{G}(z)$ – частотные функции вейвлетов $\tilde{\varphi}(x)$ и $\tilde{\psi}(x)$. Как известно, при вейвлет-разложении сигнала используются транспонированные фильтры $h_n^* = \overline{h}_{-n}$ и $g_n^* = \overline{g}_{-n}$. Поэтому их частотные функции будут сопряженными исходным: $\overline{H(\omega)}$ и $\overline{G(\omega)}$. Для точного восстановления сигнала фильтрами $\tilde{H}(z)$ и $\tilde{G}(z)$ должны выполняться следующие условия (с учетом $\sqrt{2}$ в частотных функциях):

$$\left.\begin{aligned}\tilde{H}(z)\overline{H}(z)+\tilde{G}(z)\overline{G}(z)=1\\ \tilde{H}(z)\overline{H}(-z)+\tilde{G}(z)\overline{G}(-z)=0\end{aligned}\right\}$$

при $z = e^{-i\omega}$. Считая, что коэффициенты фильтров $H(z)$ и $G(z)$ являются вещественными, последнюю систему можно переписать в виде:

$$\left.\begin{aligned}\tilde{H}(z)H(z^{-1})+\tilde{G}(z)G(z^{-1})=1\\ \tilde{H}(z)H(-z^{-1})+\tilde{G}(z)G(-z^{-1})=0\end{aligned}\right\}. \qquad (1)$$

Предположим, что известна некоторая четверка биортогональных фильтров ($H(z)$, $G(z)$, $\tilde{H}(z)$, $\tilde{G}(z)$), тогда легко видеть, что условию точного восстановления (1) также удовлетворяет следующая четверка ($H^N(z)$, $G(z)$, $\tilde{H}(z)$, $\tilde{G}^N(z)$):

$$H^N(z) = H(z) - G(z)P(z^{-2});$$
$$\tilde{G}^N(z) = \tilde{G}(z) + \tilde{H}(z)P(z^2),$$

где $P(z^2)$ – любой полином Лорана (то есть включающий и отрицательные степени). Переход от биортогональной четверки фильтров ($H(z)$, $G(z)$, $\tilde{H}_0(z)$, $\tilde{G}_0(z)$) к четверке биортогональных фильтров ($H^N(z)$, $G(z)$, $\tilde{H}(z)$, $\tilde{G}^N(z)$) называется основным элементарным шагом лифтинга (улучшения).

Также легко проверить, что условию точного восстановления (1) удовлетворяет следующая четверка ($H(z)$, $G^N(z)$, $\tilde{H}^N(z)$, $\tilde{G}(z)$):

$$G^N(z) = G(z) - H(z)Q(z^2);$$
$$\tilde{H}^N(z) = \tilde{H}(z) + \tilde{G}(z)Q(z^{-2}),$$

где $Q(z^2)$ – любой полином Лорана.

Переход от биортогональной четверки фильтров ($H(z)$, $G(z)$, $\tilde{H}(z)$, $\tilde{G}(z)$) к четверке биортогональных фильтров ($H(z)$, $G^N(z)$, $\tilde{H}^N(z)$, $\tilde{G}(z)$) называется дуальным элементарным шагом лифтинга.

Схема лифтинга в Wavelet Toolbox – это последовательность элементарных шагов и нормализации с целью улучшения свойств вейвлетов.

Алгебраическая природа схемы лифтинга достаточно проста. Действительно, условие точного восстановления в матричной форме выглядит следующим образом (см. разделы 2.13.5 и 2.14.5):

$$\begin{pmatrix} \tilde{H}(z) & \tilde{G}(z) \\ \tilde{H}(-z) & \tilde{G}(-z) \end{pmatrix} \begin{pmatrix} \overline{H(z)} & \overline{H(-z)} \\ \overline{G(z)} & \overline{G(-z)} \end{pmatrix} = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}.$$

Это условие не нарушится, если мы между этими матрицами вставим две взаимно обратные матрицы. В качестве таких матриц выберем следующие:

$$L = \begin{pmatrix} 1 & P(z^2) \\ 0 & 1 \end{pmatrix}, \quad L^{-1} = \begin{pmatrix} 1 & -P(z^2) \\ 0 & 1 \end{pmatrix},$$

где $P(z^2)$ – любой полином Лорана. Тогда

$$\begin{pmatrix} \tilde{H}(z) & \tilde{G}(z) \\ \tilde{H}(-z) & \tilde{G}(-z) \end{pmatrix} \begin{pmatrix} 1 & P(z^2) \\ 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & -P(z^2) \\ 0 & 1 \end{pmatrix} \begin{pmatrix} \overline{H(z)} & \overline{H(-z)} \\ \overline{G(z)} & \overline{G(-z)} \end{pmatrix} = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}.$$

Поскольку $z = e^{-i\omega}$, то $\overline{z} = e^{i\omega} = z^{-1}$ и $\overline{P(z^{-2})} = P(z^2)$. Поэтому новые матрицы частотных функций будут иметь вид:

$$\begin{pmatrix} 1 & -P(z^{-2}) \\ 0 & 1 \end{pmatrix} \begin{pmatrix} H(z) & H(-z) \\ G(z) & G(-z) \end{pmatrix} =$$

$$= \begin{pmatrix} H(z) - P(z^{-2})G(z) & H(-z) - P((-z)^{-2})G(-z) \\ G(z) & G(-z) \end{pmatrix},$$

$$\begin{pmatrix} \tilde{H}(z) & \tilde{G}(z) \\ \tilde{H}(-z) & \tilde{G}(-z) \end{pmatrix} \begin{pmatrix} 1 & P(z^2) \\ 0 & 1 \end{pmatrix} = \begin{pmatrix} \tilde{H}(z) & \tilde{H}(z)P(z^2) + \tilde{G}(z) \\ \tilde{H}(-z) & \tilde{H}(-z)P((-z)^2) + \tilde{G}(-z) \end{pmatrix}.$$

Для дуального лифтинга нужно использовать матрицы

$$L = \begin{pmatrix} 1 & 0 \\ Q(z^{-2}) & 1 \end{pmatrix}, \quad L^{-1} = \begin{pmatrix} 1 & 0 \\ -Q(z^{-2}) & 1 \end{pmatrix},$$

где $Q(z^2)$ – любой полином Лорана.

В пакете Wavelet Toolbox содержатся пять групп новых *функций лифтинга*:

1. Структура схемы лифтинга.
   - **addlift** – добавление основных или дуальных элементарных шагов схемы лифтинга;
   - **displs** – отображение схемы лифтинга;
   - **lsinfo** – информация о схеме лифтинга.

2. Биортогональные четверки фильтров.
   - **bswfun** – вычисление и график биортогональных функций масштабирования и вейвлетов;
   - **filt2ls** – преобразование четверки фильтров в схему лифтинга LS;
   - **liftfilt** – применение элементарных шагов лифтинга на четверке фильтров;
   - **ls2filt** – преобразование схемы лифтинга LS в четверку фильтров.
3. Ортогональные или биортогнальные вейвлеты (true) и «ленивые» (lazy) вейвлеты.
   - **liftwave** – нахождение схемы лифтинга LS, связанной с вейвлетом;
   - **wave2lp** – полиномы Лорана, ассоциированные с вейвлетом;
   - **wavenames** – имена вейвлетов, доступные для LWT.
4. Прямое и обратное вейвлет-преобразование лифтинга (LWT).
   - **lwt** – 1D-вейвлет-преобразование лифтинга;
   - **lwt2** – 2D-вейвлет-преобразование лифтинга;
   - **lwtcoef** – извлечение или восстанавление вейвлет-коэффициентов LWT;
   - **lwtcoef2** – извлечение или восстанавление вейвлет-коэффициентов 2D LWT;
   - **ilwt** – обратное 1D-вейвлет-преобразование лифтинга;
   - **ilwt2** – обратное 2D-вейвлет-преобразование лифтинга.
5. Полиномы Лорана и матрицы.
   - **laurmat** – конструктор класса LM матриц Лорана;
   - **laurpoly** – конструктор класса LP полиномов Лорана.

Подробнее о лифтинге см. [M] и документацию MATLAB.

3.2. Непрерывное вейвлет-преобразование cwt

Используются следующие функции:

- **cwt** – непрерывное одномерное вейвлет-преобразование;
- **cwtext** – непрерывное одномерное вейвлет-преобразование с параметрами расширения массива;
- **cwtft** – непрерывное вейвлет-преобразование, использующее FFT-алгоритм;
- **icwtft** – обратное непрерывное вейвлет-преобразование;

- **icwtlin** – обратное непрерывное вейвлет-преобразование для линейно распределенных масштабов;
- **cwtftinfo** – информация о допустимых вейвлетах для CWT на основе FFT;
- **conofinf** – конус влияния;
- **localmax** – определение цепочек локальных максимумов коэффициентов;
- **wcoher** – вейвлет-когерентность;
- **wscalogram** – скалограмма непрерывного вейвлет-преобразования.

Для непрерывного вейвлет-преобразования используются также упомянутые ранее функции pat2cwav, wavemenu и wavemngr.

1. Функция cwt – непрерывное одномерное вейвлет-преобразование. Команда cwt выполняет *непрерывное вейвлет-преобразование* одномерного сигнала. Пусть $x(t)$ – сигнал и $\psi(t)$ – вейвлет. При непрерывном преобразовании вейвлет-коэффициенты сигнала x, соответствующие масштабному коэффициенту a и положению b, определяются формулой

$$C_{a,b} = W[f](a,b) = \frac{1}{\sqrt{a}} \int_{-\infty}^{\infty} s(t)\,\overline{\psi\left(\frac{t-b}{a}\right)}\,dt.$$

Если сигнал $x(t)$ является дискретным, заданным выборкой значений $\{x_k\}$, то в качестве $x(t)$ можно взять кусочно-постоянную функцию, принимающую значения x_k, на промежутках $t \in [k\Delta t, (k+1)\Delta t)$, тогда $x_k = x(k\Delta t)$. Параметр сдвига b пробегает значения с шагом Δt, то есть $b = n\Delta t$, $n = 1{:}\,lx$, $lx = \text{length}(x)$. Полагаем, что значения масштабного параметра a также дискретны и составляют вектор, который далее обозначается **scales**.

Основное отличие команды **cwt** от дискретного многуровневого вейвлет-преобразования **wavedec** заключается в том, что в cwt можно брать любые значения масштабного коэффициента a, тогда как в **wavedec** используются степени двойки.

Для команды **cwt** предусмотрена возможность графического цветного изображения вейвлет-коэффициентов сигнала x и аналогичного изображения скалограммы сигнала (рис. 3.2.1). *Скалограмма* – это изображение (то есть двумерный массив), которое отражает процентное содержание энергии каждого коэффициента.

Синтаксис cwt:

```
coefs = cwt(x,scales,'wname')
coefs = cwt(x,scales,'wname','plot')
```

```
coefs = cwt(x,scales,'wname','coloration')
[coefs,sgram] = cwt(x,scales,'wname','scal')
[coefs,sgram] = cwt(x,scales,'wname','scalCNT')
coefs = cwt(x,scales,'wname','coloration',xlim)
```

Описание.

Функция **coefs = cwt(x,scales,'wname')** вычисляет непрерывные вейвлет-коэффициенты вещественного вектора сигнала x в положительных масштабах scales, используя вейвлет 'wname'. Сигнал x действителен, но вейвлет может быть как действительным, так и комплексным.

Выходной массив coefs – это матрица *la* на *lx*, где *la* – длина вектора масштабов и *lx* – длина входного сигнала x. Выходной массив coefs – действительная или комплексная матрица, в зависимости от вейвлета. Для каждого данного значения масштабного коэффициента *a* в пределах вектора scales вейвлет-коэффициенты C[a,n] вычисляются для сдвигов по времени $b = n\Delta t$, $n = 1{:}lx$ и сохраняются в виде вектора-строки coefs(i,:), соответствующей *i*-му значению параметра *a* = scales(i).

- **coefs = cwt(x,scales,'wname','plot')** дополнительно изображает непрерывные вейвлет-коэффициенты, используя окраску по умолчанию 'absglb'.
- **coefs = cwt(x,scales,'wname','coloration')** использует для изображения заданную окраску.
- **[coefs,sgram] = cwt(x,scales,'wname','scal')** отображает исходный сигнал и маштабированный образ скалограммы **sgram**.
- **[coefs,sgram] = cwt(x,scales,'wname','scalCNT')** отображает исходный сигнал и контурные линии уровня скалограммы **sgram**.
- **coefs = cwt(x,scales,'wname','coloration',xlim)** окрашивает коэффициенты, используя **'coloration'** и часть массива коэффициентов, определенную вектором **xlim**, который указывает пределы части массива по аргументу x, **xlim** есть вектор [x1 x2], где $1 \le x1 < x2 \le \text{length}(x)$.

Окраска. Результат **coefs** операции **cwt** является матрицей *la* на *lx*, она изображается на плоскости в виде матрицы, состоящей из ячеек в количестве *la* – по вертикали и *lx* – по горизонтали. Каждая ячейка содержит элемент матрицы коэффициентов и окрашивается в соответствии со значением своего коэффициента. Способ окраски определяется параметром **coloration**, значения которого приведены ниже.

Окраска – это метод, используемый для масштабирования значений коэффициентов для их изображения в цвете. Каждый коэффициент делится на получающееся значение окраски:

- 'lvl' – использует максимальное значение в каждом масштабе;
- 'glb' – использует максимальное значение во всех масштабах;
- 'abslvl' или 'lvlabs' – использует максимальное абсолютное значение в каждом масштабе;
- 'absglb' или 'glbabs' – использует максимальное абсолютное значение во всех масштабах;
- 'scal' – производит масштабированный образ скалограммы;
- 'scalCNT' – производит контурное представление скалограммы.

Пример 1. Загрузим кардиосигнал из текстового форматированного файла SmCh4.txt и выберем фрагмент сигнала длиной 4096 отсчетов. Сделаем его непрерывное вейвлет-разложение с гауссовым вейвлетом gaus1, выбрав значения масштабного коэффициента от 1 до 200 с шагом 2.

```
load SmCh4.txt;
SV2=SmCh4(601:4696);
w='gaus1';
Coef = cwt(SV2,1:2:200,w,'plot');
title('Scalogram')
ylabel('Scale')
[Coef,Scal] = cwt(SV2,1:2:200,w,'scal');
```

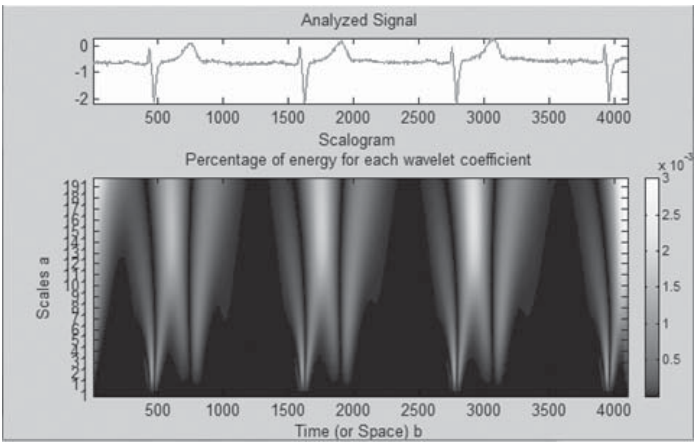


Рис. 3.2.1. График кардиосигнала и его скалограмма

Комплекс QRS кардиограммы хорошо проявляется на высоких частотах (при малых a), волны P, T и U также хорошо видны на спектрограмме на малых частотах (при больших a).

Комплексный непрерывный анализ. В MATLAB возможны следующие типы *комплексных* вейвлетов:

1. Комплексный гауссов вейвлет cgau – это p-я производная комплексной функции Гаусса $\psi(x) = C_p e^{-ix} e^{-x^2}$. Нужно задавать параметр p.
2. Комплексный вейвлет Морле cmor с шириной полосы частот B и центральной частотой C. Параметры записываются в виде B-C. Например, cmor1-1.5 – комплексный вейвлет Морле шириной полосы частот B = 1 и центральной частотой C = 1.5.
3. Комплексный частотный B-сплайновый вейвлет fbsp, определенный параметром M, шириной полосы частот B, центральной частотой C. Параметры записываются в виде M-B-C. Например, fbsp 2-1-0.5.
4. Вейвлет Шеннона shan, определенный шириной полосы частот B, центральной частотой C. Параметры записываются в виде B-C. Например, shan1-1.5.

Комплексный непрерывный анализ используется точно так же. Команда **cwt** вычисляет матрицу, содержащую комплексные коэффициенты в различных масштабах. Например, команда

```
x = cwt(SV2,1:2:64,'cgau4','plot');
```

вычисляет комплексную матрицу x размера 32 на 3700, каждая строка которой соответствует отдельному масштабу. Для изображения коэффициентов необходимы два графика-изображения. Это либо графики вещественных и мнимых частей, либо модулей и углов. Команда MATLAB `x=cwt(SV2,1:2:64,'cgau4','plot');` воспроизводит все четыре изображения.

Алгоритм CWT. Пусть сигнал $x(t)$ задан выборкой значений $\{x_k\}$. В качестве $x(t)$ мы берем кусочно-постоянную функцию, принимающую значения x_k, на промежутках $t \in [k\Delta t, (k+1)\Delta t)$, тогда $x_k = x(k\Delta t)$. Параметр сдвига b пробегает значения с шагом Δt, то есть $b = n\Delta t$, $n = 1{:}\, lx$, $lx = \text{length}(x)$.

Пусть $\psi_0(t)$ – вейвлет с компактным носителем на промежутке $[0, N]$ и $\Psi(t)$ – его первообразная. Поскольку аргумент t пробегает значения с шагом Δt, то разумно использовать вейвлет $\tilde{\psi}(t) = \psi_0(t/\Delta t)$ с носителем на промежутке $[0, N\Delta t]$. Носитель вейвлета $\tilde{\psi}\left(\dfrac{t-b}{a}\right)$ на-

ходится на промежутке $[b,\ b + aN\Delta t]$. Нахождение коэффициентов при $b = n\Delta t$:

$$C_{a,b} = \frac{1}{\sqrt{a}}\int_{-\infty}^{\infty} x(t)\,\overline{\tilde{\psi}\left(\frac{t-b}{a}\right)}dt = \frac{1}{\sqrt{a}}\int_{n\Delta t}^{n\Delta t + aN\Delta t} x(t)\,\overline{\tilde{\psi}\left(\frac{t-b}{a}\right)}dt =$$

$$= \frac{1}{\sqrt{a}}\sum_{k\geq n}\int_{k\Delta t}^{(k+1)\Delta t} x(t)\,\overline{\tilde{\psi}\left(\frac{t-b}{a}\right)}dt = \frac{1}{\sqrt{a}}\sum_{k=n}^{k=n+[aN]+1} x_k \int_{k\Delta t}^{(k+1)\Delta t}\overline{\tilde{\psi}\left(\frac{t-b}{a}\right)}dt.$$

Вычислим последний интеграл, сделав несколько замен переменных:

$$\int_{k\Delta t}^{(k+1)\Delta t}\tilde{\psi}\left(\frac{t-b}{a}\right)dt = a\Delta t\int_{\frac{k}{a}-\frac{b}{a\Delta t}}^{\frac{k+1}{a}-\frac{b}{a\Delta t}}\psi_0(y)dy = a\Delta t\left(\Psi\left(\frac{k+1}{a}-\frac{b}{a\Delta t}\right)-\Psi\left(\frac{k}{a}-\frac{b}{a\Delta t}\right)\right),$$

где $\Psi(t)$ – первообразная вейвлета $\psi_0(t)$. Таким образом,

$$C_{a,b} = \frac{1}{\sqrt{a}}\sum_{k\geq n} x_k \int_{k\Delta t}^{(k+1)\Delta t}\overline{\tilde{\psi}\left(\frac{t-b}{a}\right)}dt = \sqrt{a}\,\Delta x\sum_{k=n}^{k=n+[aN]+1} x_k\,\overline{c_k(a,b)},$$

где для $b = n\Delta t$ коэффициенты c_k имеют вид:

$$c_k(a,b) = \Psi\left(\frac{k+1}{a}-\frac{b}{a\Delta t}\right)-\Psi\left(\frac{k}{a}-\frac{b}{a\Delta t}\right) = \Psi\left(\frac{k+1-n}{a}\right)-\Psi\left(\frac{k-n}{a}\right).$$

В случае, когда вейвлет $\psi_0(t)$ берется из MATLAB, его первообразная $\Psi(t)$ может быть найдена при помощи команды intwave.

2. Функция cwtext – вещественное или комплексное непрерывное вейвлет-преобразование с использованием параметров продолжения. В теории вейвлетов сигналы и вейвлеты определены на всей числовой прямой **R**. Если сигнал конечной длины (или задан на конечном промежутке), то стандартный способ сделать его бесконечным (определенным на **R**) заключается в том, чтобы продолжить его нулями. Именно такой способ продолжения используется в команде **cwt**. Возможные разрывы на концах приводят к нежелательным искажениям вблизи граничных точек. Избежать этого можно выбором других способов продолжения. Функция **cwtext** позволяет задавать несколько других способов продолжения массива.

Синтаксис.

```
COEFS = cwtext(S,SCALES,'wname')
COEFS = cwtext(S,SCALES,'wname',PropName1,ProVal1,...)
COEFS = cwtext(...,'PlotMode',PLOTMODE)
```

Описание.

Функция COEFS = cwtext(S,SCALES,'wname') не отличается от cwt, с параметрами расширения по умолчанию 'zpd' и 'b'.

COEFS = cwtext(S,SCALES,'wname',PropName1,ProVal1,...) – дополнительно указываются параметры расширения анализируемого массива S. Допустимые значения PropName и ProVal указаны в табл. 3.2.1.

Таблица 3.2.1. Дополнительные параметры

Propname – имя свойства	Proval – значение свойства
'ExtMode'	'zpd' (заполнение нулями) 'sp0' (гладкое расширение степени 0) 'sp1' (гладкое расширение степени 1)
'ExtSide'	'l' (или 'u') – расширение влево (или вверх) 'r' (or 'd') – расширение вправо (или вниз) 'b' – расширение в обе стороны
'ExtLen'	Длина расширения
'PlotMode'	'lvl', 'glb', 'abslvl' или 'lvlabs', 'absglb' или 'glbabs'
'Xlim'	[x1 x2] , где 1 <= x1 < x2 <= length(s)

Примеры правильного использования функции:

```
c = cwtext(s,1:32,'cgau4');
c = cwtext(s,[3 18 12.9 7 1.5],'db2');
c = cwtext(s,1:32,'sym2','plotMode','lvl');
c = cwtext(s,1:64,'sym4','plotMode','abslvl','xlim',[100 400]);
c = cwtext(s,1:64,'sym4','plotMode','lvl','extMode','sp0');
```

Пример 2. Рассмотрим тестовый сигнал wcantor – канторова лестница. Сделаем его непрерывное вейвлет-преобазование и изобразим графически коэффициенты разложения (рис. 3.2.2 и 3.2.2):

```
load wcantor
cwt(wcantor,(1:256),'mexh','absglb');
colormap(pink(4))
```

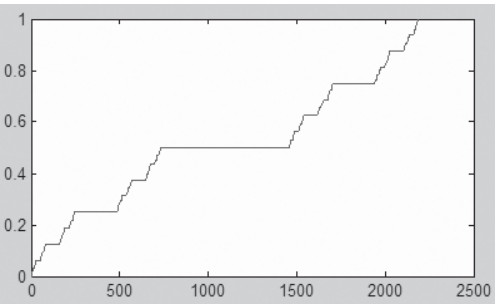


Рис. 3.2.2. Канторова лестница

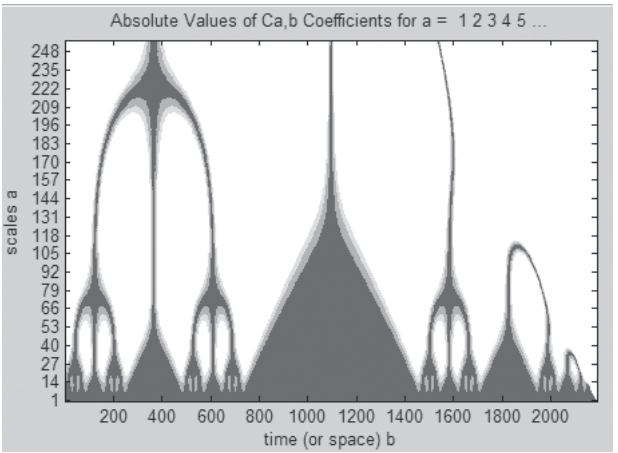


Рис. 3.2.3. Коэффициенты разложения канторовой лестницы

Справа хорошо видны искажения, вызванные разрывностью канторовой лестницы в точке 1, поскольку по умолчанию используется расширение массива нулями. Зададим продолжение по непрерывности константой как 'sp0':

```
figure;
cwtext(wcantor,(1:256),'mexh','extmode','sp0','extLen',2000,…
                                    'plotMode','absglb');
colormap(pink(4))
```

Получаем вполне симметричное изображение (рис. 3.2.4).

3. Функция wscalogram – скалограмма непрерывного вейвлет-преобразования.

Синтаксис.

SC = wscalogram(TYPEPLOT,COEFS)
SC = wscalogram(TYPEPLOT,COEFS,'PropName1',PropVal1,...)

Описание.

Функция SC = wscalogram(TYPEPLOT,COEFS) вычисляет скалограмму SC, которая представляет процентное содержание энергии каждого коэффициента. COEFS – матрица коэффициентов непрерывного вейвлет-преобразования. *Скалограмма* получается вычислениями по формуле

S = abs(coefs.*coefs); SC = 100*S./sum(S(:)).

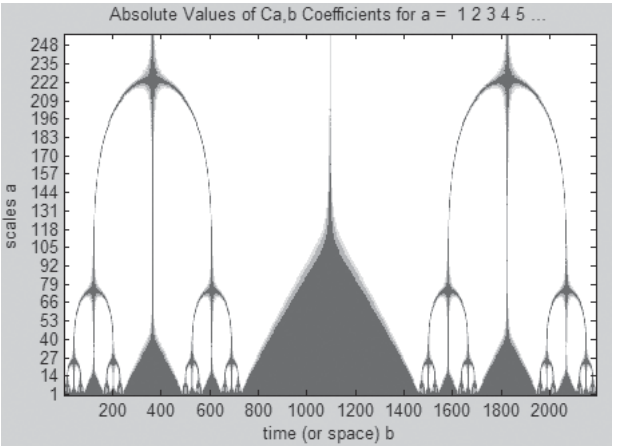


Рис. 3.2.4. Коэффициенты разложения канторовой лестницы для 'sp0'

Если TYPEPLOT равен 'image', отоображается масштабированный образ скалограммы. Если TYPEPLOT = 'contour', отоображается контурный образ скалограммы.

SC = wscalogram(TYPEPLOT,COEFS,'PropName1',PropVal1,...) – с дополнительными аргументами. Допустимый выбор PropName:

- 'scales' – масштабы, используемые для CWT;
- 'ydata' – сигнал, используемый для CWT;
- 'xdata' – значения аргумента x, соответствующие значениям сигнала;
- 'power' – положительное вещественное число. По умолчанию = 0.

4. Функция cwtft – непрерывное вейвлет-преобразование, использующее дискретное преобразование Фурье FFT.

Синтаксис.

```
cwtstruct = cwtft(sig)
cwtstruct = cwtft(sig,Name,Value)
cwtstruct = cwtft(...,'plot')
```

Описание.

Функция **cwtstruct = cwtft(sig)** производит непрерывное вейвлет-преобразование (CWT) одномерного сигнала sig. Функция cwtft использует алгорит FFT дискретного преобразования Фурье. Сигнал sig может быть вектором, структурой или массивом ячеек. Если пери-

од дискретизации сигнала не равен 1, то можно ввести период дискретизации вместе с сигналом sig в массив ячеек или в структуру и получить корректные результаты. Если sig есть массив ячеек, sig{1} – это сам сигнал и sig{2} – период дискретизации. Если sig есть структура, то поле sig.val содержит сигнал и sig.period содержит период дискретизации. По умолчанию cwtft использует аналитический вейвлет Morlet и автоматическое определение числа масштабов. Могут быть использованы также следующие вейвлеты:

- 'dog' – производная пороядка m вейвлета Гаусса для положительного четного m. По умолчанию m = 2;
- 'morlex' – неаналитический вейвлет Морле;
- 'morl0' – неаналитический вейвлет Морле с нулевым средним;
- 'mexh' – мексиканская шляпа. Вторая производная гауссова вейвлета;
- 'paul' – вейвлет Пауля.

cwtstruct = cwtft(sig,Name,Value) производит непрерывное вейвлет-преобразование (CWT) одномерного сигнала sig с дополнительными аргументами Name,Value – это имя вейвлета и вектор масштабов.

cwtstruct = cwtft(…,'plot') – дополнительно изображает графические результаты операции CWT.

Выходной массив cwtstruct представляет собой структуру с шестью полями:

- dt – период выборки входного сигнала;
- cfs – матрица коэффициентов CWT. cwtstruct.cfs – это матрица nb на N, где nb – это число различных масштабов и N – длина входного сигнала;
- meanSIG – среднее анализируемого сигнала;
- omega – вектор угловых частот;
- scales – вектор масштабов, для которых вычисляется CWT. Длина cwtstruct.scales равна числу строк cwtstruct.cfs;
- wav – анализирующий вейвлет.

Обоснование метода. Теорема Планшереля позволяет выразить коэффициенты непрерывного вейвлет-преобразования через интеграл от образов Фурье:

$$C_{a,b} = \frac{1}{\sqrt{a}} \int_{-\infty}^{\infty} x(t)\,\overline{\psi\left(\frac{t-b}{a}\right)}dt = \left(x(t), \frac{1}{\sqrt{a}}\psi\left(\frac{t-b}{a}\right)\right)_{L^2} =$$

$$= \frac{1}{2\pi}(\hat{x}(\omega), \sqrt{a}\,\hat{\psi}(a\omega)e^{-i\omega b})_{L^2} = \frac{1}{2\pi}\int_{-\infty}^{\infty} \hat{x}(\omega)\,\overline{\sqrt{a}\,\hat{\psi}(a\omega)}\,e^{i\omega b}d\omega,$$

где $\hat{x}(\omega)$ и $\sqrt{a}\,\hat{\psi}(a\omega)e^{-i\omega b}$ – преобразования Фурье функций $x(t)$ и $\frac{1}{\sqrt{a}}\psi\left(\frac{t-b}{a}\right)$. Получим дискретный аналог этой формулы. Пусть $\psi(t)$ – анализирующий вейвлет. Определим функцию $\tilde{\psi}_a(t) = \frac{1}{\sqrt{a}}\overline{\tilde{\psi}\left(\frac{-t}{a}\right)}$. Тогда

$$C_{a,b} = \int_{-\infty}^{\infty} x(t)\,\tilde{\psi}_a(b-t)\,dt$$

– это свертка функций. Для нахождения дискретного аналога данной формулы предположим, что сигнал $x(t)$ задан конечной выборкой значений $\{x_n = x(n\Delta t)\}$, $n = 0, 1, ..., N-1$. Считаем, что $x(t) = 0$ вне промежутка $[0, N-1]$. Тогда дискретный аналог вышеприведенной формулы CWT выражается в виде свертки следующим образом:

$$W_a(m) = \sum_{n=0}^{N-1} x(n\Delta t)\,\tilde{\psi}_a((m-n)\Delta t)\Delta t = \Delta t \sum_{n=0}^{N-1} x_n\,\tilde{\psi}_{a,m-n},$$

где $b = m\Delta t$. Для выполнения CWT нужно вычислить свертку для каждого значения параметра сдвига b и для каждого масштаба a. Однако если эти две конечные последовательности периодически (с периодом N) распространить на все значения $n \in \mathbf{N}$, то дискретное преобразование Фурье свертки двух последовательностей можно выразить в виде произведения дискретных преобразования Фурье этих последовательностей:

$$X(\omega) = \Delta t \sum_n x_n e^{-i\omega n\Delta t} \quad \text{и} \quad \Psi_a(\omega) = \Delta t \sum_m \tilde{\psi}_{a,m} e^{-i\omega m\Delta t}.$$

Это периодические функции с периодом $2\pi/\Delta t$. Поскольку на основном промежутке $[0, 2\pi/\Delta t]$

$$\hat{x}(\omega) = \int_{-\infty}^{\infty} x(t)e^{-i\omega t}\,dt \approx \sum_{n=-\infty}^{\infty} x(n\Delta t)e^{-i\omega n\Delta t}\Delta t = X(\omega),$$

то и $\Psi_a(\omega) = \sqrt{a}\,\overline{\hat{\psi}(a\omega)}$. Если $\hat{W}_a(\omega) = \Delta t \sum_n W_a(m)e^{-i\omega m\Delta t}$ – преобразование Фурье последовательности $W_a(m)$, то

$$\hat{W}_a(\omega) = X(\omega)\,\Psi_a(\omega) = \sqrt{a}\,\hat{x}(\omega)\,\overline{\hat{\psi}(a\omega)}.$$

Функция $\hat{W}_a(\omega)$ является $(2\pi/\Delta t)$-периодической функцией. Поскольку $\hat{W}_a(\omega) = \Delta t \sum_m W_a(m)e^{-i\omega m\Delta t}$ – ряд Фурье, то формула обращения выглядит следующим образом:

$$\Delta t\, W_a(m) = \frac{\Delta t}{2\pi} \int_0^{2\pi/\Delta t} \hat{W}_a(\omega) e^{i\omega m \Delta t} d\omega.$$

Заменяя приближенно интеграл суммой Римана, получаем:

$$W_a(m) = \frac{1}{2\pi} \int_0^{2\pi/\Delta t} \hat{W}_a(\omega) e^{i\omega m \Delta t} d\omega \cong \frac{1}{2\pi} \sum_{k=0}^{N-1} \hat{W}_a\left(\frac{2\pi k}{N\Delta t}\right) e^{i\frac{2\pi k}{N\Delta t} m \Delta t} \frac{2\pi}{N\Delta t} =$$

$$= \frac{1}{N\Delta t} \sum_{k=0}^{N-1} \hat{W}_a\left(\frac{2\pi k}{N\Delta t}\right) e^{i\frac{2\pi k}{N} m} = \frac{\sqrt{a}}{N\Delta t} \sum_{k=0}^{N-1} \hat{x}\left(\frac{2\pi k}{N\Delta t}\right) \overline{\hat{\psi}\left(a\frac{2\pi k}{N\Delta t}\right)} e^{i 2\pi k m / N}.$$

Последняя формула лежит в основе непрерывного вейвлет-преобразования через прямое и обратное дискретное преобразование Фурье.

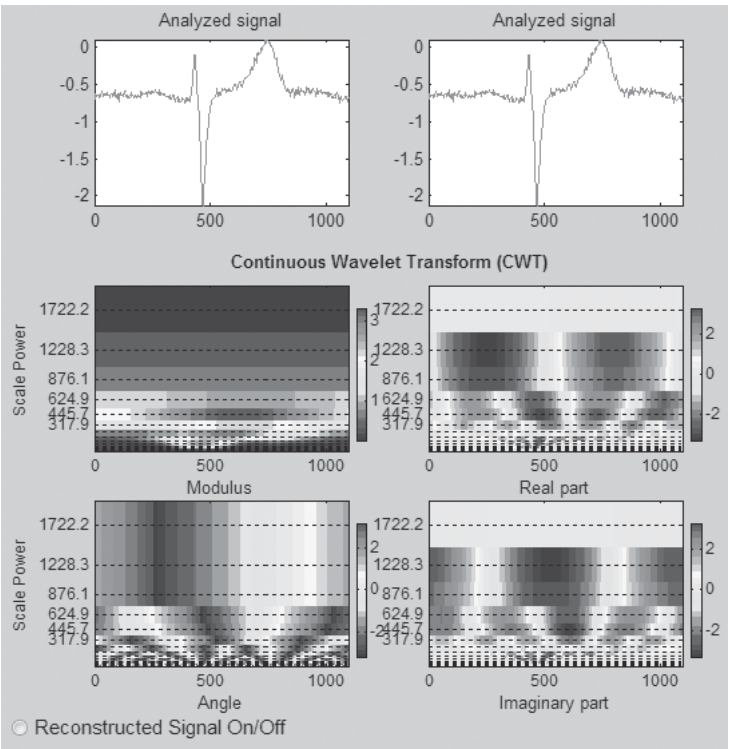


Рис. 3.2.5. Непрерывное вейвлет-преобразование, использующее FFT

Алгоритм cwtft:

- выполнение дискретного преобразования Фурье (DFT) сигнала с использованием функции fft;
- выполнение дискретного преобразования Фурье DFT-анализирующего вейвлета в соответствующих частотах. Масштабирование DFT-вейвлета в различных масштабах;
- выполнение произведения DFT-сигнала и DFT-вейвлета во всех масштабах;
- обратное DFT для получения CWT-коэффициентов.

5. Функция icwtft – обратное непрерывное вейвлет-преобразование.

Синтакис:

```
xrec = icwtft(cwtstruct)
xrec = icwtft(cwtstruct,'plot')
xrec = icwtft(cwtstruct,'signal',SIG,'plot')
```

Описание.

Функция xrec = icwtft (cwtstruct) производит обратное непрерывное преобразование коффициентов, содержащихся в поле cfs структурного массива cwtstruct.

xrec = icwtft(cwtstruct,'plot') – изображает восстановленный сигнал вместе с коэффициентами.

xrec = icwtft(cwtstruct,'signal',SIG,'plot') – отличается от предыдущей команды радиокнопкой слева внизу (рис. 3.2.2).

Отметим, что функция **icwtft** вычисляет обратное непрерывное преобразование, выполненное cwtft с логарифмически распределенными масштабами. Если масштабы распределены линейно, то нужно использовать следующую функцию icwtlin.

6. Функция icwtlin – обратное непрерывное вейвлет-преобразование (CWT) для линейно распределенных масштабов. Данная функция вычисляет обратное непрерывное преобразование, выполненное cwtft с линейно распределенными масштабами. Если масштабы распределены логарифмически, то нужно использовать предыдущую функцию icwtft.

Синтакис:

```
xrec = icwtlin(cwtstruct)
xrec = icwtlin(wav,meanSIG,cfs,scales,dt)
xrec = icwtin(...,'plot')
```

7. Функция cwtftinfo – информация о допустимых вейвлетах для CWT на основе FFT. Выводит информацию о допустимых вейвлетах

для непрерывного вейвлет-преобразования на основе дискретного преобразования Фурье.

8. Функция conofinf – конус влияния. Если $\psi(t)$ – допустимый вейвлет с эффективным носителем $[-B, B]$, тогда функция $\psi_{u,s}(t) = \frac{1}{\sqrt{s}}\psi\left(\frac{t-u}{s}\right)$ будет иметь носитель на промежутке $[u - sB, u + sB]$. *Конус влияния* есть эффективный носитель вейвлета в данной позиции и в данном масштабе, то есть носитель $[u - sB, u + sB]$ функции $\psi_{u,s}(t)$.

Синтаксис:

```
[cone,PL,PR,PLmin,PRmax]=
= conofinf(wname,scales,LenSig,COIval,'plot');
```

Входные параметры:

- wname – имя вейвлета;
- scales – вектор масштабов;
- LenSig – длина сигнала;
- COIval – смещение;
- 'plot' – изображение конуса влияния. Кроме линий конуса, изображаются также две линии. Левая синяя линия имеет тот же наклон, но пересекает ось масштабов в максимальной точке. Правая красная линия имеет тот же наклон, но проходит через точку $[x, s]$, где x – длина сигнала, а s – максимальный масштаб.

Выходные параметры:

- cone – конус влияния. Это матрица из нулей и единиц, число строк которой равно количеству заданных масштабов, а количество столбцов равно длине сигнала;
- PL – вектор из двух элементов, первый элемент – это коэффициент наклона левой части конуса, а второй элемент – это точка пересечения с осью масштабов;
- PR – вектор из двух элементов, первый элемент – это коэффициент наклона правой части конуса, а второй элемент – это точка пересечения с осью масштабов;
- PLmin – вектор из двух элементов, первый элемент – это коэффициент наклона левой части конуса, а второй элемент – это точка пересечения левой ветви при максимальном значении масштаба (эта левая прямая изображается синим при задании параметра 'plot');
- PRmax – вектор из двух элементов, первый элемент – это коэффициент наклона правой части конуса, а второй элемент –

это точка пересечения правой ветви конуса при максимальном значении масштаба (эта правая прямая изображается красным при задании параметра 'plot').

9. Функция localmax – определение цепочек локальных максимумов коэффициентов.

Синтакис:

```
[lmaxima,indices]= localmax(inputmatrix)
[lmaxima,indices]= localmax(inputmatrix,initrow)
[lmaxima,indices]= localmax(inputmatrix,initrow,regflag)
```

Входные аргументы:

- inputmatrix –матрица вещественных или комплексных чисел. Обычно это матрица коэффициентов непрерывного вейвлет-преобразования. localmax находит максимумы абсолютных значений;
- initrow – номер начальной строки для нахождения цепочек локальных максимумов. Алгоритм формирования цепочки начинается в initrow с уменьшением индекса строки на 1 до тех пор, пока не будет достигнута первая строка матрицы. Определяя initrow, можно исключить строки из алгоритма формирования цепочки;
- regflag – флаг регуляризации (значения: true или false). Если regflag устанавлен как true (по умолчанию), то строка входной матрицы, соответствующая номеру initrow, заменяется коэффициентами аппроксимации 5-го уровня, полученными вейвлетом sym4.

Выходные паременные:

- lmaxima – матрица локальных максимумов. Имеет тот же размер, что и входная матрица. Имеет ненулевые элементы только в точках локальных максимумов. Пусть R – номер строки матрицы lmaxima. Значение в точке локального максимума строки с номером R определяется следующим образом:
  - если R > initRow, тогда значение lmaxima в точке локального максимума равно 1;
  - если R = initRow, тогда значение lmaxima в точке локального максимума есть индекс столбца в строке R (то есть значение второй координаты точки максимума);
  - если R < initRow, тогда значение lmaxima в точке локального максимума строки R равно индексу столбца ближайшего локального максимума в строке R + 1. При этом если ло-

кальный максимум в строке R лежит между двумя локальными максимумами строки R + 1, то значение в локальном максимуме строки R является наибольшим индексом столбца строки R + 1.

- indices – столбец линейных индексов ненулевых значений lmaxima. Для преобразования линейных индексов в матричные индексы (строка, столбец) можно использовать функцию ind2sub.

10. Функция wcoher – вейвлет-когерентность (согласованность).
Синтакис:

WCOH = wcoher(Sig1,Sig2,Scales,wname)

Позволяет найти и проанализировать когерентность матриц вейвлет-коэффициентов двух сигналов, полученных при непрерывном вейвлет-преобразовании. Напомним общую формулу для вычисления вейвлет-когерентности двух сигналов x и y:

$$Wcoh = \frac{S(\overline{C_x(a,b)}C_y(a,b))}{\sqrt{S(|C_x(a,b)|^2)}\sqrt{S(|C_y(a,b)|^2)}},$$

где $C_x(a,b)$ и $C_y(a,b)$ – вейвлет-коэффициенты для масштаба a и сдвига $b\Delta t$ и S – некоторый оператор.

3.3. Дискретный вейвлет-анализ

3.3.1. Анализ одномерных сигналов

В данном разделе рассмотрим вейвлет-анализ одномерных сигналов. Используются следующие функции.

Одноуровневое разложение:

- **dwt** – одноуровневое дискретное одномерное вейвлет-преобразование;
- **idwt** – обратное одноуровневое дискретное одномерное вейвлет-преобразование.

Многоуровневое разложение:

- **wavedec** – многоуровневое одномерное вейвлет-разложение;
- **waverec** – многоуровневое одномерное вейвлет-восстановление;
- **upcoef** – прямое восстановление из коэффициентов;
- **appcoef** – вычисляет аппроксимирующие коэффициенты 1D-сигнала;

- **detcoef** – вычисление детализирующие коэффициенты 1D-сигнала;
- **wrcoef** – восстановление сигнала по одной ветви вейвлет-коэффициентов;
- **upwlev** – одноуровневая реконструкция одномерного вейвлет-разложения;
- **ndwt** – недецимированное 1D-вейвлет-преобразование;
- **indwt** – обратное недецимированное 1D-вейвлет-преобразование;
- **wmaxlev** – максимальный уровень вейвлет-разложения.

Дискретные вейвлет-преобразования стационарного сигнала:

- **swt** – дискретное стационарное вейвлет-преобразование;
- **iswt** – обратное стационарное вейвлет-преобразование.

Моделирование броуновского движения:

- **wfbm** – синтез дробного броуновского движения;
- **wfbmesti** – оценка параметра дробного броуновского движения.

Характеристики сигнала и его приближения:

- **wenergy** – энергия 1D-вейвлет или вейвлет-пакетных коэффициентов;
- **wvarchg** – поиск точек изменения дисперсии сигнала;
- **measerr** – оценки качества приближения.

Для одномерного вейвлет-анализа сигнала создан графический интерфейс пользователя wavemenu и можно также использовать общие функции:

- **dwtmode** – мода расширения дискретного вейвлет-преобразования;
- **dyaddown** – диадическая децимация;
- **dyadup** – диадическая обратная децимация;
- **wrev** – перестановки элементов вектора в обратном порядке;
- **wkeep** – извлечение части вектора или матрицы.

Мода расширения сигнала

В теории вейвлетов сигналы и вейвлеты определены на всей числовой прямой **R**. В случае оцифрованного сигнала он представляется в виде бесконечной в обе стороны последовательности. Реально же сигнал всегда конечной длины. Стандартный способ сделать его бесконечным заключается в том, чтобы продолжить его нулями. Однако при этом могут возникнуть нежелательные граничные эффекты вследствие разрыва значений в концевых точках сигнала. Поэтому

используются и другие *методы* (*моды*) *расширения* значений сигнала (табл. 3.3.1). Для этих целей – функция dwtmode.

Таблица 3.3.1. Моды расширения

Параметр mode	Мода расширения
'sym' или 'symh'	Симметричное расширение относительно точки, расположенной на расстоянии $^1/_2$ от концевой точки, в этом случае концевая точка «повторяется дважды» (default mode)
'symw'	Симметричное расширение относительно граничной точки
'asym' или 'asymh'	Симметричное продолжение относительно точки, расположенной на расстоянии $^1/_2$ от концевой точки ($\boldsymbol{t} = \boldsymbol{n} + {}^1/_2$), с отражением по оси значений ***Ox*** относительно значения $\boldsymbol{x} = 0$, в этом случае концевая точка «повторяется», а значения меняют знак
'asymw'	Симметричное продолжение относительно концевой точки ($\boldsymbol{t} = \boldsymbol{n}$) с отражением по оси значений ***Ox*** относительно значения на конце $\boldsymbol{x} = \boldsymbol{x}_n$
'zpd'	Дополнение нулями
'spd' или 'sp1'	Гладкое продолжение, то есть линейное продолжение в направлении первой производной
'sp0'	Непрерывное продолжение (постоянное продолжение первого значения – влево и последнего значения – направо)
'ppd' или 'per'	Периодическое продолжение значений

Синтаксис функции **dwtmode**:

```
ST = dwtmode
ST = dwtmode('status')
dwtmode('mode')
```

Описание.

Функция ST = dwtmode или ST = dwtmode('status') выводит на экран и присваивает параметру ST текущую моду.

Функция dwtmode('mode') устанавливает новую моду расширения.

Метод, используемый по умолчанию, загружается из файла DWTMODE.DEF, если он существует. В противном случае используется DWTMODE.CFG (toolbox/wavelet/wavelet directory).

Функция dwtmode('*save*',MODE) сохраняет метод MODE как новый, используемый по умолчанию метод в DWTMODE.DEF (в текущей директории).

dwtmode('*save*') эквивалентно
dwtmode('*save*',CURRENTMODE).

В следующей сессии используется по умолчанию сохраненный ранее метод. Способ расширения, используемый для IDWT, должен быть тот же самый, что и при DWT, чтобы гарантировать точную реконструкцию.

Пример 1. Применяем команду dwtmode. В результате мы получаем на экране информацию о текущей моде расширения:

```
dwtmode
****************************************
**  DWT Extension Mode: Symmetrization  **
****************************************
```

Изменение существующей моды на периодическую:

```
dwtmode('per')
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!  WARNING: Change DWT Extension Mode  !
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!

****************************************
**  DWT Extension Mode: Periodization  **
****************************************
```

Одноуровневое дискретное одномерное вейвлет-преобразование

В одноуровневом дискретном одномерном вейвлет-преобразовании используются следующие функции:

- dwt – одноуровневое дискретное одномерное вейвлет-преобразование;
- idwt – обратное одноуровневое дискретное одномерное вейвлет-преобразование.

1. Дискретное вейвлет-преобразование dwt. Функция вычисляет аппроксимирующие коэффициенты *cA* и детализирующие коэффициенты *cD* первого уровня вейвлет-преобразования сигнала, заданного вектором *s*. Коэффициенты получаются сверткой сигнала *s* с фильтром разложения нижних частот Lo_D для аппроксимации и с фильтром разложения высоких частот Hi_D для детализации, а затем сопровождаются двоичной децимацией. Схема метода:

Синтаксис:

[cA,cD] = dwt(s,'wname'),
[cA,cD] = dwt(s,'wname', 'mode', MODE),
[cA,cD] = dwt(s,Lo_D,Hi_D),
[cA,cD] = dwt(*s*,Lo_D,Hi_D, 'mode', MODE).

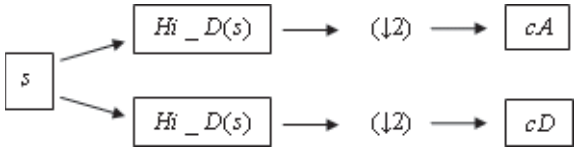


Рис. 3.3.1. Схема вейвлет-преобразования

Описание.

Функция [cA,cD] = dwt(*s*,'*wname*') вычисляет *аппроксимирующие коэффициенты* cA и *детализирующие коэффициенты* cD вектора *s*. Строка *wname* – имя вейвлета.

Функция [cA,cD] = dwt(*s*,Lo_D,Hi_D) находит вейвлет-разложение, как выше, используя вместо имени вейвлета его низкочастотный фильтр разложения Lo_D и высокочастотный фильтр разложения Hi_D.

Параметр MODE определяет метод расширения данных для вейвлет-разложения. Например, [cA,cD] = dwt(*s*,'db2', 'mode', 'sym'), см. функцию dwtmode и табл. 3.3.1.

Пример 2. Зададим сигнал *s*. Найдем его разложение, используя вейвлет Хаара, а затем – Добеши db4.

```
n=1:20; s=sin(0.5*n);
[ca1,cd1] = dwt(s,'haar');
%Графики.
subplot(311); plot(s); title('Original signal');
subplot(323); plot(ca1); title('Approx. coef. for haar');
subplot(324); plot(cd1); title('Detail coef. for haar');
[ca2,cd2] = dwt(s,'db4');
subplot(325); plot(ca2); title('Approx. coef. for db2');
subplot(326); plot(cd2); title('Detail coef. for db2');
```

На рис. 3.3.2 видна существенная разница коэффициентов, полученных с использованием вейвлетов Хаара и Добеши.

2. Обратное дискретное вейвлет-преобразование idwt. Функция осуществляет обратное одноуровневое дискретное одномерное вейвлет-преобразование. Другими словами, по известным аппроксимирующим коэффициентам *cA* и детализирующим коэффициентам *cD* первого уровня команда idwt вычисляет значения сигнала *s*, используя либо имя вейвлета, либо его фильтры реконструкции.

Восстановление производится в обратном порядке. В коэффициенты *cA* и *cD* добавляются нули на нечетные места, и далее производится свертка полученных массивов с фильтрами реконструкции

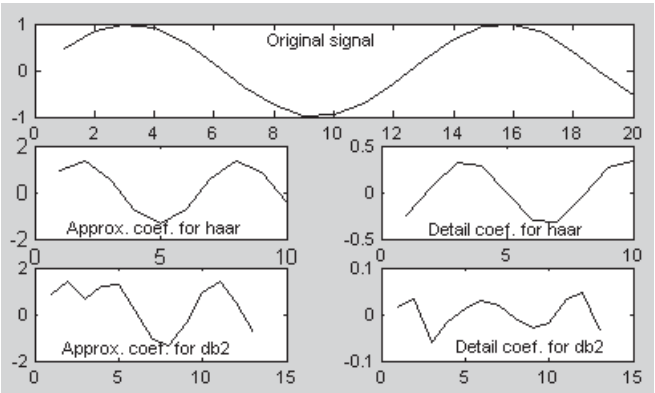


Рис. 3.3.2. Коэффициенты разложения

Lo_R и Hi_R. Функция idwt осуществляет элементарный шаг реконструкции.

Синтаксис:

```
s = idwt(cA,cD,'wname')
s = idwt(cA,cD,Lo_R,Hi_R)
s = idwt(cA,cD,'wname',L)
s = idwt(cA,cD,Lo_R,Hi_R,L)
s = idwt(...,'mode',MODE)
s = idwt(...,'mode',MODE)
s = idwt(cA,[],...)
s = idwt([],cD,...)
```

Входящие параметры:

- 'wname' – имя вейвлета;
- Lo_R – низкочастотный фильтр восстановления;
- Hi_R – высокочастотный фильтр восстановления данного вейвлета;
- L – числовой параметр, если он установлен, то в качестве *s* берется центральная часть длины L массива, полученного по команде idwt(cA,cD,'wname'). L должно быть меньше, чем длина s;
- MODE – определяет метод расширения при вейвлет-преобразовании.

Функция *s* = idwt(cA,cD,'wname') выполняет одноуровневую реконструкцию из коэффициентов аппроксимирующих и детализирующих коэффициентов cA и cD с использованием вейвлета 'wname'.

Функция *s* = idwt(cA,cD,Lo_R,Hi_R) выполняет одноуровневую реконструкцию из коэффициентов аппроксимирующих и детализирующих коэффициентов cA и cD с использованием фильтров реконструкции: Lo_R – низкочастотный и Hi_R – высокочастотный фильтры реконструкции (Lo_R и Hi_R должны быть одинаковой длины).

Замечание 1. О длине реконструированного вектора. Пусть la – длина cA (равна длине cD) и lf – длина фильтров Lo_R и Hi_R, тогда length(s) = ls, где ls = 2*la, если мода расширения для DWT установлена как периодизация 'per'. Для других мод расширения длина восстановленного сигнала ls = 2*la – lf + 2.

Функция *s* = idwt(cA,cD,'wname',L) or X= idwt(cA,cD,Lo_R, Hi_R,L) возвращает центральную часть длины L вектора, полученного по команде idwt(cA,cD,'wname'). L должно быть меньше, чем ls.

Функция *s* = idwt(...,'mode',MODE) выполняет одноуровневую реконструкцию с использованием задаваемой моды расширения MODE.

Функция *s* = idwt(cA,[], ...) возвращает вектор *s* одноуровневого восстановления только по аппроксимирующим коэффициентам cA.

Функция *s* = idwt([],cD, ...) возвращает вектор *s* одноуровневого восстановления только по детализирующим коэффициентам cD.

Пример 3. Предположим, что сигнал *s* – тот, что был рассмотрен в предыдущем примере. Вычислены его коэффициенты ca1, cd1 с использованием вейвлета Хаара и коэффициенты ca2, cd2 с использованием вейвлета Добеши db4. Восстановим этот сигнал функцией idwt и сравним с исходным.

```
ss = idwt(ca2,cd2,'db4');
err = norm(s-ss);
subplot(212); plot([s;ss]');
```

Сравнение показывает, что восстановленный сигнал совпадает с исходным с точностью до err = 4.0459e-012.

Многоуровневый одномерный вейвлет-анализ

В данном параграфе рассмотрим разложение сигналов на несколько уровней. В многоуровневом дискретном одномерном вейвлет-анализе участвуют следующие функции:

- **wavedec** – многоуровневое одномерное вейвлет-разложение;
- **waverec** – многоуровневое одномерное вейвлет-восстановление;
- **appcoef** – вычисляет аппроксимирующие коэффициенты 1D-сигнала;

- **detcoef** – вычисление детализирующие коэффициенты 1D-сигнала;
- **upwlev** – одноуровневая реконструкция одномерного вейвлет-разложения;
- **wrcoef** – восстановление сигнала по одной ветви вейвлет-коэффициентов;
- **upcoef** – прямое восстановление из коэффициентов;
- **ndwt** – недецимированное 1D-вейвлет-преобразование;
- **indwt** – обратное недецимированное 1D-вейвлет-преобразование;
- **wmaxlev** – максимальный уровень вейвлет-разложения.

Схема многоуровневого одномерного вейвлет-разложения. Сначала по массиву значений сигнала *s* вычисляются два набора коэффициентов: коэффициенты аппроксимации cA_1 и детализирующие коэффициенты cD_1. Эти векторы получаются сверткой сигнала *s* с фильтром нижних частот Lo_D для аппроксимации и с фильтром высоких частот Hi_D для детализации, а затем сопровождаются двоичной децимацией. На следующем шаге раскладываются коэффициенты аппроксимации cA_1 на две части с использованием той же самой схемы, заменяя *s* на cA_1 и производя cA_2 и cD_2 и т. д., по схеме на рис. 3.3.3:

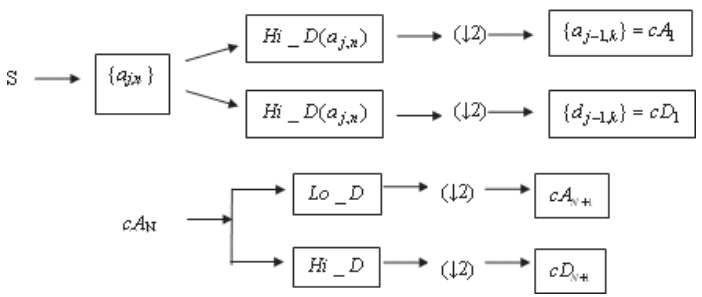


Рис. 3.3.3. Схема разложения

Удобно графически изображать результаты разложения

$$s = cA_0 \rightarrow \{cA_1, cD_1\} \rightarrow \{cA_2, cD_2, cD_1\} \rightarrow ... \rightarrow \{cA_N, cD_N, cD_{N-1}, ..., cD_1\}$$

в виде дерева (рис. 3.3.4).

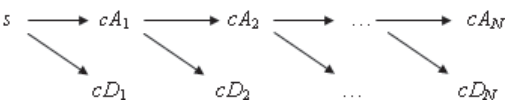


Рис. 3.3.4. Дерево разложения

Вейвлет-разложение сигнала *s*, проведенное до уровня *N*, является вектором, который имеет следующую структуру:

[cA_N, cD_N, ..., cD_1].

1. Функция wavedec – многоуровневое одномерное вейвлет-разложение. Команда **wavedec** производит вейвлет-разложение сигнала *s* до уровня (глубины) *N*, то есть вычисляет аппроксимирующие коэффициенты cA_N и детализирующие коэффициенты cD_1, ..., cD_N.

Синтаксис функции wavedec:

[C,L] = wavedec(s,N,'wname'),
[C,L] = wavedec(s,N,Lo_D,Hi_D).

Входящие аргументы:

- s – одномерный сигнал;
- N – уровень разложения;
- 'wname' – имя используемого вейвлета;
- Lo_D и Hi_D – его фильтры декомпозиции нижних и верхних частот.

Выходные параметры:

- вектор C имеет следующую структуру: C = [cA_N, cD_N, ..., cD_1];
- вектор L учета длин коэффициентов вектора C имеет вид:

  L = [length (cA_N), length (cD_N), ..., length (cD_1), length (s)].

  Его общая длина равна $N + 2$.

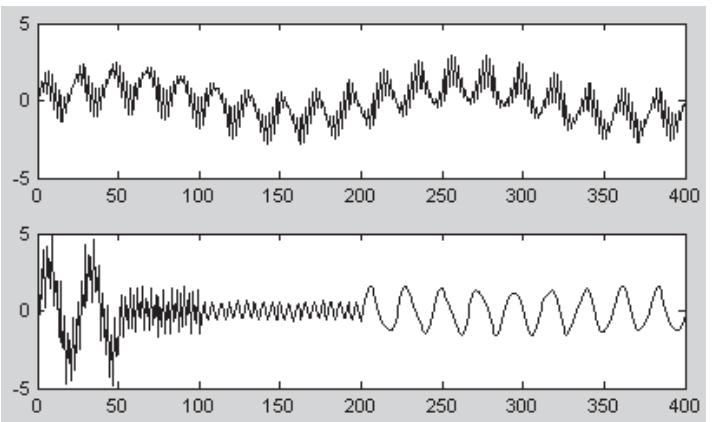


Рис. 3.3.5. Разложение до уровня 3

Пример 4. Загружаем одномерный тестовый сигнал sumsin и выполняем декомпозицию до уровня 3 с использованием вейвлета Добеши db1.

```
load sumsin; s = sumsin(1:400);
[c,l] = wavedec(s,3,'db1');
subplot 211; plot(s); subplot 212; plot(c);
```

На нижнем рисунке хорошо видно, что выходной вектор C имеет ту же длину, что и исходный сигнал, и составлен из нескольких различных компонент cA_3, cD_3, cD_2, cD_1. При этом компонента cA_3 (первая часть вектора C длины 50) более других напоминает исходный сигнал.

2. Функция waverec – многоуровневое одномерное восстановление. Она производит восстановление сигнала s, используя полученные ранее векторы многоуровневого разложения [C, L]. При восстановлении указывается имя используемого вейвлета или фильтры восстановления Lo_R, Hi_R. Схема реконструкции, как и для idwt, обратная к схеме разложения.

Функция waverec используется в виде:

s = waverec(C,L,'*wname*'), s = waverec(C,L,Lo_R,Hi_R),

где C, L – векторы многоуровневого разложения, 'wname' – имя вейвлета и Lo_R, Hi_R – его фильтры восстановления.

Команда s = waverec(C,L,'wname') эквивалентна s = appcoef(C,L, 'wname',0).

3. Функция appcoef – вычисление аппроксимирующих коэффициентов. Команда appcoef вычисляет аппроксимирующие коэффициенты одномерного сигнала задаваемого уровня N, используя полученные ранее векторы многоуровневого разложения [C, L]. Если найдены векторы C и L разложения сигнала до уровня N0, то N0 = = length(L) – 2 и вектор C имеет структуру C = [A(N0), D(N0), ..., D(1)]. Если N = N0, тогда производится простое извлечение A(N0), если N < N0, тогда аппроксимирующие коэффициенты вычисляются с использованием обратного вейвлет-преобразования.

Синтаксис:

A = appcoef(C,L,'wname',N)
A = appcoef(C,L,'wname')
A = appcoef(C,L,Lo_R,Hi_R)
A = appcoef(C,L,Lo_R,Hi_R,N).

Функция A = appcoef(C,L,'wname', N) вычисляет аппроксимирующие коэффициенты уровня N, $0 \le N \le N0$, используя (уже получен-

ную) структуру вейвлет-разложения [C, L] (см. wavedec) до уровня N0, 'wname' – имя вейвлета.

Функция A = appcoef(C,L,'wname') извлекает аппроксимирующие коэффициенты последнего уровня N0 = length(L) – 2.

Вместо имени вейвлета можно использовать его фильтры:

A = appcoef(C,L,Lo_R,Hi_R) или A = appcoef(C,L,Lo_R,Hi_R,N).

Пример 5. Предположим, что уже загружен сигнал s = leleccum(1:3920).

```
[c,l] = wavedec(s,4,'db1');
ca3 = appcoef(c,l,'db1',3);
```

4. Функция detcoef – вычисление детализирующих коэффициентов. Команда detcoef находит детализирующие коэффициенты одномерного сигнала задаваемого уровня N, $0 \le N \le N0 = length(L)-2$, используя полученные ранее векторы многоуровневого разложения [C, L] уровня разложения N0. Вектор C имеет структуру C = [A(N0), D(N0), ..., D(1)].

Синтаксис:

D = detcoef(C,L),
D = detcoef(C,L,N),
DCELL = detcoef(C,L,N,'cells'),
[D1, ... ,Dp] = detcoef(C,L,[N(1), ... ,N(p)]).

Функция D = detcoef(C,L) извлекает детализирующие коэффициенты последнего уровня N0.

Функция D = detcoef(C,L,N) извлекает детализирующие коэффициенты уровня N.

Функция DCELL = detcoef(C,L,N,'cells') применяется, когда N – вектор целых чисел, такой что $1 \le N(j) \le N0$. Функция извлекает массив ячеек коэффициентов, в котором DCELL{j} содержит коэффициенты уровня N(j). Если length(N) > 1, то DCELL = detcoef(C,L,N) эквивалентна DCELL = detcoef(C,L,N,'cells').

Если N – вектор целых чисел, такой что $1 \le N(j) \le N0$, то функция DCELL = detcoef(C,L,N,'cells') извлекает массив ячеек коэффициентов, в котором DCELL{j} содержит коэффициенты уровня N(j). Если length(N) > 1, то DCELL = detcoef(C,L,N) эквивалентна DCELL = detcoef(C,L,N,'cells'). Кроме того,

DCELL = detcoef(C,L,'cells') эквивалентна
DCELL = detcoef(C,L,[1: N0]).

Функция [D1, ... ,Dp] = detcoef(C,L,[N(1), ... ,N(p)]) извлекает детализирующие коэффициенты указанных уровней [N(1), ... ,N(p)].

Пример 6. Сигнал s = leleccum раскладываем до уровня 3, а затем извлекаем детализирующие коэффициенты уровней 1, 2, 3.

```
load leleccum; s = leleccum(1:3920); %загружаем тестовый сигнал leleccum
[c,l] = wavedec(s,3,'db1');
[cd1,cd2,cd3] = detcoef(c,l,[1 2 3]);
```

5. Функция upwlev – одноуровневая реконструкция одномерного вейвлет-разложения. Если заданы векторы вейвлет-разложения [C, L] уровня разложения N0, то команда upwlev находит векторы вейвлет-разложения [NC, NL] уровня разложения N0 – 1, то есть на один уровень меньше. Кроме того, извлекаются также последние аппроксимирующие коэффициенты cA, то есть уровня N0.

Синтаксис:

[NC,NL,cA] = upwlev(C,L,'wname'),
[NC,NL,cA] = upwlev(C,L,Lo_R,Hi_R),

где [C, L] – векторы вейвлет-разложения [C, L] уровня разложения N0, 'wname' – имя вейвлета, [NC,NL] – векторы вейвлет-разложения уровня разложения N0 – 1. Вместо вейвлета можно использовать фильтры реконструкции Lo_R и Hi_R.

Пример 7.

```
load sumsin; s = sumsin(1:1000); %загружаем тестовый сигнал sumsin
[c,l] = wavedec(s,3,'db1');
subplot(311); plot(s);  title('Original signal s.');
subplot(312); plot(c);  title('Wavelet decomposition structure, level 3')
[nc,nl] = upwlev(c,l,'db1');
subplot(313); plot(nc);  title('Wavelet decomposition structure, level 2')
```

На рис. 3.3.6 хорошо видно, что выходной вектор теперь содержит три группы коэффициентов cA_2, cD_2, cD_1.

6. Функция wrcoef – восстановление сигнала по одной ветви вейвлет-коэффициентов. Если известны векторы вейвлет-разложения [C, L] уровня разложения N_0, то команда wrcoef восстанавливает сигнал по одной ветви вейвлет-коэффициентов на уровне разложения $N \le N_0$.

Синтаксис:

X = wrcoef('type',C,L,'wname',N),
X = wrcoef('type',C,L,'wname')

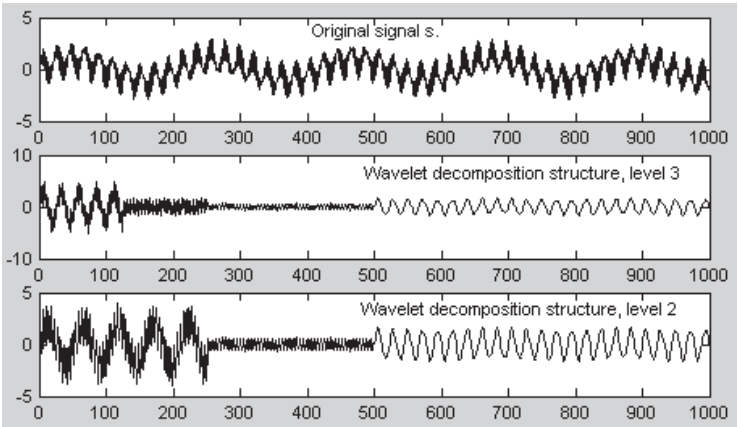


Рис. 3.3.6. Одноуровневая реконструкция

X = wrcoef('type',C,L,Lo_R,Hi_R,N)
X = wrcoef('type',C,L,Lo_R,Hi_R).

Здесь 'type' определяет тип ветви:

- 'type' = 'a' – восстановление из аппроксимирующих коэффициентов;
- 'type' = 'd' – восстановление из детализирующих коэффициентов.

Если N не указано, то имеется в виду максимальный уровень N_0.

Пример 8. В продолжение предыдущего примера найдем аппроксимацию 4-го уровня (то есть сигнал, восстановленный только по коэффициентам аппроксимации):

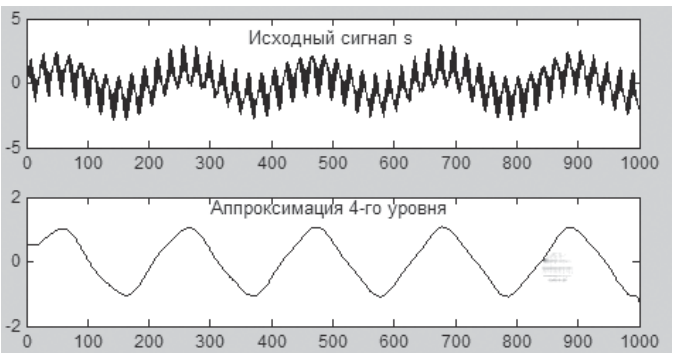


Рис. 3.3.7. Аппроксимация 4-го уровня

```
load sumsin; s = sumsin;
[c,l] = wavedec(s,4,'sym4');
a5 = wrcoef('a',c,l,'sym4',4);
subplot(211); plot(s);  title('Исходный сигнал s');
subplot(212); plot(a5);  title('Аппроксимация 4-го уровня')
```

7. Функция upcoef – прямое восстановление из вейвлет-коэффициентов. Функция upcoef восстанавливает вектор Y из коэффициентов X заданного уровня N, применяя N раз процедуру обратного вейвлет-преобразования. При этом если X считается детализирующими коэффициентами (O = 'd'), то аппроксимирующие коэффициенты берутся нулевыми, и наоборот.

Синтаксис:

Y = upcoef(O,X,'wname',N)
Y = upcoef(O,X,'wname',N,L)
Y = upcoef(O,X,Lo_R,Hi_R,N)
Y= upcoef(O,X,Lo_R,Hi_R,N,L)
Y = upcoef(O,X,'wname'')
Y = upcoef(O,X,'wname'',1)
Y = upcoef(O,X,Lo_R,Hi_R)
Y = upcoef(O,X,Lo_R,Hi_R,1)

Если O = 'a', то коэффициенты X считаются аппроксимирующими.
Если O = 'd', коэффициенты X считаются детализирующими.

Функция Y = upcoef(O,X,'wname',N,L) вычисляет N-шаговую реконструкцию коэффициентов вектора X и выбирает центральную часть результата длины L.

Y = upcoef(O,X,'wname') эквивалентно Y = upcoef(O,X,'wname',1).

Вместо вейвлета можно использовать фильтры реконструкции Lo_R и Hi_R.

Y = upcoef(O,X,Lo_R,Hi_R,N) или
Y = upcoef(O,X,Lo_R,Hi_R,N,L).

Y = upcoef(O,X,Lo_R,Hi_R) эквивалентно
Y = upcoef(O,X,Lo_R,Hi_R,1).

Пример 9. Восстанавливаем сигнал, исходя из единственного коэффициента по вейвлету Добеши db6 вверх на уровни от 1 до 3. Вейвлет db6 имеет носитель на промежутке [0, 10] длины 10. Поэтому при свертке исходного сигнала (в данном случае состоящего из одного ненулевого значения, равного 1) с фильтрами реконструкции Lo_R и Hi_R мы получим в результате сигнал длины 10. Далее длина резуль-

тата (существенная часть) будет удваиваться, как это и происходит при обратном вейвлет-преобразовании. Обратите внимание, что при реконструкции X как аппроксимирующих коэффициентов из одиночного сигнала получается сигнал, близкий к φ-функции вейвлета Добеши db6.

Устанавливаем текущей моду «zero-padding» (см. dwtmode), то есть дополняем вектор cfs = [1] нулевыми значениями.

```
cfs = [1]; essup = 10;
for i=1:3
rec = upcoef('a',cfs,'db6',i);
ax = subplot(3,1,i),h = plot(rec(1:essup));
set(ax,'xlim',[1 50]);
essup = essup*2;
end
subplot(311)
title(['Approximation signals, obtained from a single coefficient at
levels 1 to 3'])
```

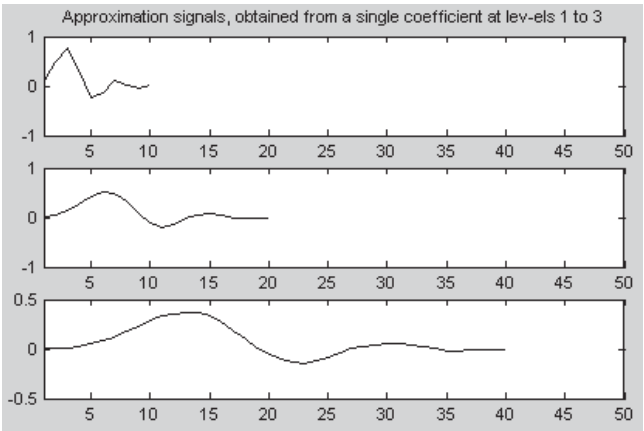


Рис. 3.3.8. Восстановление сигнала из единственного коэффициента

Совершенно аналогично можно вычислить детализирующие сигналы, получаемые из того же единичного коэффициента (считая его детализирующим) на уровни от 1 до 3 вверх.

8. Функция ndwt – недецимированное 1D-вейвлет-преобразование. Функция делает разложение фильтрами вейвлета без последующей децимации.

Синтаксис:

WT = ndwt(X,N,'wname')
WT = ndwt(X,N,'wname','mode','ExtM')
WT = ndwt(X,N,WF,...)

Выходной параметр WT представляет собой струкутру с шестью полями:

- rowvect – логическое значение true, если X – вектор;
- level – глубина разложения N;
- mode – имя метода расширения для вейвлет-разложения;
- filters – структура с четырьмя полями: LoD, HiD, LoR, HiR, это фильтры, используемые в разложении;
- dec – массив ячеек размера 1×N, содержащих коэффициенты разложения. dec{1} содержат коэффициенты аппроксимации, а ячейки dec{j} j = 2, ..., N + 1 содержат детализирующие коэффициенты уровня N + 1 – j;
- longs – вектор, содержащий длины компонент, [length (cA_N), length (cD_N), ..., length (cD_1), length (s)]. Надо помнить, что это недецимированные коэффициенты.

Пример 10. Создадим сигнал длины 1000 и сделаем недецимированное вейвлет-разложение до уровня 4 вейвлетом Добеши db2. Определим длины коэффициентов разложения.

```
t = linspace(0,1,1000);
x = 4*sin(4*pi*t);
x = x - sign(t - .3) - sign(.72 - t);
W = ndwt(x,4,'db2','mode','per');
>> W.longs

ans =
      1012        1012        1009        1006        1003        1000
```

9. Функция indwt – обратное недецимированное 1D-вейвлет-преобразование. Производит вейвлет-восстановление из структуры многоуровневого недецимированного разложения W.

Синтаксис:

X = indwt(W)
X= indwt(W,'a',0)
X= indwt(W,'ca',0)
C = indwt(W,TYPE,n)
C = indwt(W,TYPE)

Первые три функции X = indwt(W), X= indwt(W,'a',0) или X= indwt(W,'ca',0) производят полное восстановление сигнала по структуре разложения W.

Функция C = indwt(W,TYPE,n) вычисляет компоненту, восстановленную по одной ветви коэффициентов с уровня n ≤ N вейвлет-разложения уровеня N. Эта ветвь коэффициентов определяется параметром TYPE. Допустимые значения параметра TYPE следующие:

- символ 'a' (или 'l' или 'A' или 'L') определяет низкочастотную компоненту (аппроксимацию уровня n);
- символ 'd' (или 'h' или 'D' или 'H') определяет высокочастотную компоненту (детали уровня n);
- для извлечения коэффициентов к этим символам надо добавить префикс 'c' или 'C'.

C = indwt(W,TYPE) эквивалентно C = indwt(W,TYPE,N), где N – уровень разложения.

10. Функция wmaxlev – максимальный уровень вейвлет-разложения.

Синтаксис:

L = wmaxlev(S,'wname')

Описание

Функция L = wmaxlev (S, 'wname') позволяет wmaxlev определить максимальный разумный уровень разложения L. Функция применима также для изображений или для пакетного разложения. Обычные значения L = 5 для одномерного случая, L = 3 для двумерного случая.

Дискретное вейвлет-преобразование стационарного сигнала

Сигнал *s* рассматривается как случайный процесс с дискретным временем *t*. Тогда сигнал *s*(*t*) с конечной дисперсией называется стационарным (в широком смысле), если его математическое ожидание постоянно, M[*s*(*t*)] = m_s, а ковариационная функция зависит только от разности аргументов $\tau = t_2 - t_1$. Для исследования таких сигналов используются функции:

- **swt** – дискретное вейвлет-преобразование стационарного сигнала;
- **iswt** – обратное вейвлет-преобразование стационарного сигнала.

Особенности swt:

1. Делается многоуровневое разложение, в отличие от одноуровневого преобразования dwt.

2. В отличие от обычного алгоритма вейвлет-преобразования dwt и wavedec, в данном случае после применения фильтров декомпозиции Lo_D и Hi_D децимация не производится. Поэтому коэффициенты cA1 и cD1 имеют ту же длину, что и исходный сигнал s. Далее к cA1 применяется аналогичная процедура, но с использованием модифицированных фильтров F и G, которые получены соответственно из Lo_D и Hi_D обратной децимацией.
3. Длина сигнала length(s) должна быть кратной 2N, где N – уровень разложения. Результатом вычислений является матрица SWC, содержащая векторы коэффициентов в качестве строк: для $1 \le i \le N$ строка SWC (i,:) содержит детализирующие коэффициенты уровня i и SWC(N+1,:) содержит аппроксимирующие коэффициенты уровня N.

1. Функция swt – дискретное многоуровневое вейвлет-разложение стационарного сигнала.

Синтаксис функции swt:

```
SWC = swt(s,N,'wname')
SWC = swt(s,N,Lo_D,Hi_D)
[SWA,SWD] = swt(s,N,'wname')
[SWA,SWD] = swt(s,N,Lo_D,Hi_D)
```

Входящие параметры:

- N – уровень разложения;
- 'wname' – имя ортогонального вейвлета;
- Lo_D и Hi_D – его фильтры декомпозиции нижних и верхних частот.

Выходные данные:

- SWC – матрица векторов-строк коэффициентов разложения всех уровней от 1 до N. Для $1 \le i \le N$ строки SWC(i,:) содержат коэффициенты деталей уровня i, а SWC(N+1,:) содержит аппроксимирующие коэффициенты уровня N.
- SWA – матрица аппроксимирующих коэффициентов всех уровней от 1 до N. Эти коэффициенты являются строками матрицы в соответствии со своими номерами: SWA(i,:) = cAi.
- SWD – матрица, состоящая из строк детализирующих коэффициентов всех уровней: SWD(i,:) = cDi , $1 \le i \le N$.

2. Функция iswt – дискретное многоуровневое обратное вейвлет-преобразование стационарного сигнала.

Синтаксис функции iswt:

```
X = iswt(SWC,'wname')
X= iswt(SWA,SWD,'wname')
```

```
X= iswt(SWA(end,:),SWD,'wname')
X = iswt(SWC,Lo_R,Hi_R)
X= iswt(SWA,SWD,Lo_R,Hi_R)
X = iswt(SWA(end,:),SWD,Lo_R,Hi_R)
```

Входящие параметры:

- SWC – матрица векторов-строк коэффициентов разложения всех уровней;
- SWA – матрица векторов-строк коэффициентов аппроксимации всех уровней;
- SWD – матрица векторов-строк коэффициентов детализации всех уровней;
- 'wname' – имя ортогонального вейвлета;
- Lo_D и Hi_D – его фильтры декомпозиции нижних и верхних частот.

Выходные данные: восстановленный сигнал X.

Моделирование дробного броуновского движения

Броуновское движение (винеровский процесс) есть однородный гауссовский процесс $X(t)$ с независимыми приращениями. Это одна из математических моделей хаотического перемещения взвешенных в жидкости или газе мелких частиц. С математической точки зрения броуновское движение определяется как непрерывный по времени и по значениям случайный процесс $X(t)$, обладающий свойствами:

$$X(0) = 0,$$
$$M[X(t) - X(s)] = 0,$$
$$D[X(t) - X(s)] = t - s, s \le t,$$

где M – математическое ожидание и D – дисперсия.

Броуновское движение обладает следующими характерными свойствами. Броуновская траектория нигде не дифференцируема. При выходе из какой-либо точки x эта траектория за сколь угодно малое время δ с вероятностью 1 бесконечное число раз пересекает уровень x (возвращаясь в исходную точку). Рассматриваемая на фиксированном отрезке $[0, t]$, эта траектория имеет тенденцию достигать экстремальных значений вблизи концевых точек.

Броуновское движение является непрерывным аналогом случайного блуждания частицы, которая в дискретные моменты времени $t = k\Delta t$ в результате случайного воздействия каждый раз независимо от предшествующих обстоятельств смещается на величину $\Delta X(t)$, такую что $M[\Delta X(t)] = 0$ и $D[\Delta X(t)] = \Delta t$. Точнее, если $\Delta t = 1/n$, то

$$X(t) = \sum_{k=0}^{m-1} \Delta X(k/n) + (nt - m)\Delta X(m/n),\ 0 \le t \le 1,$$

где $m = [nt]$ – целая часть, будет некоторым приближением броуновского движения.

Дробное броуновское движение с показателем Херста $0 < H < 1$ есть непрерывный по времени и по значениям гауссов процесс $X(t)$ с нулевым средним значением, такой что $X(0) = 0$ и

$$D[X(t) - X(s)] = v|t - s|^{2H},$$

где v есть положительная константа. Он является обобщением обычного броуновского движения, соответствующего показателю $H = 0.5$, производная которого есть белый шум. Дробное броуновское движение является нестационарным процессом.

Для моделирования дробного броуновского движения при помощи вейвлетов предназначены две функции:

- **wfbm** – синтез дробного броуновского движения;
- **wfbmesti** – оценка параметра дробного броуновского движения.

1. Функция wfbm – синтез дробного броуновского движения.

Синтаксис:

```
FBM = wfbm(H,L)
FBM = wfbm(H,L,'plot')
FBM = wfbm(H,L,NS,W) = FBM = wfbm(H,L,W,NS)
wfbm(H,L,'plot',NS)
wfbm(H,L,'plot',W)
wfbm(H,L,'plot',NS,W) = wfbm(H,L,'plot',W,NS)
```

Описание.

Функция FBM = wfbm(H,L) возвращает сигнал дробного броуновского движения FBM с параметром Херста H (0 < H < 1) длины L, следуя алгоритму Абри–Селана [AS] (Abry P., Sellan F.).

Функция FBM = wfbm(H,L,'plot') генерирует и строит график сигнала FBM.

Функция FBM = wfbm(H,L,NS,W) или FBM = wfbm(H,L,W,NS) возвращает FBM, используя NS шагов реконструкции и достаточно регулярный ортогональный вейвлет W.

Функции wfbm(H,L,'plot',NS), или wfbm(H,L,'plot',W), или wfbm(H,L,'plot',NS,W), или wfbm(H,L,'plot',W,NS) генерируют и строят график сигнала FBM.

Функция wfbm(H,L) эквивалентна WFBM(H,L,6,'db10').

Функция wfbm(H,L,NS) эквивалентна WFBM(H,L,NS,'db10').

Функция wfbm(H,L,W) эквивалентна WFBM(H,L,W,6).

Пример 11. Продемонстрируем на примере, что для значения $H > 0.5$ движение будет более регулярным, а при $H < 0.5$ – более хаотичным. Создадим дробное броуновское движение для $H = 0.3$ и для $H = 0.7$.

```
   % Инициализация генератора randn
randn('state',1)
   % установка параметра H и длины сигнала
H = 0.3; lg = 1000;
   % Создание и график сигнала для H = 0.3
fBm03 = wfbm(H,lg,'plot');
```

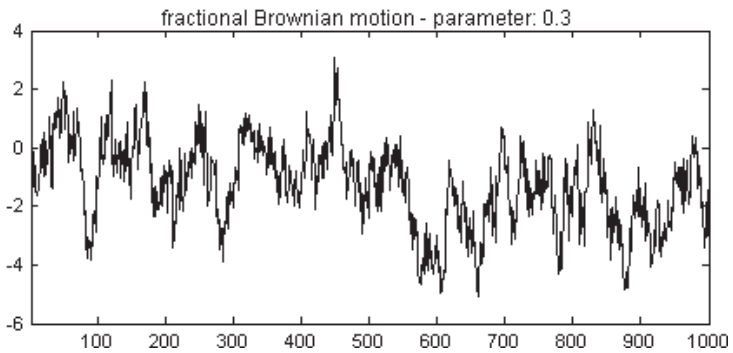


Рис. 3.3.9. Дробное броуновское движение для показателя Херста $H = 0.3$

```
   % Инициализация генератора randn и установка нового H
randn('state',1); H = 0.7;
   % Создание и график сигнала для H = 0.7
fBm07 = wfbm(H,lg,'plot');
```

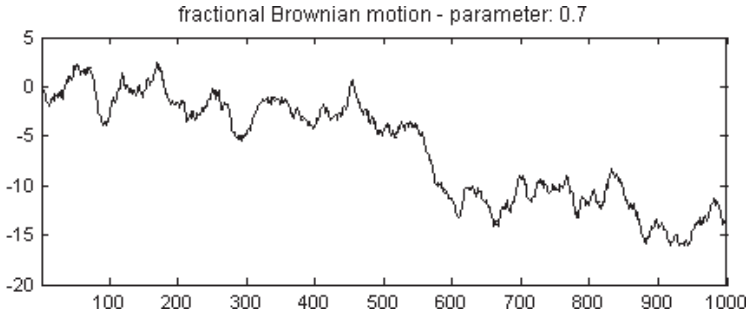


Рис. 3.3.10. Дробное броуновское движение для показателя Херста $H = 0.7$

Алгоритм основан на построении коэффициентов аппроксимации дробным процессом ARIMA. Процесс *ARIMA*(0,*s*,0) можно получить из белого гауссова шума с нулевым средним сверткой с фильтром бесконечной импульсной характеристики $\alpha_k^{(s)}$, для которого *z*-преобразование записывается в виде $A(z) = (1 - z^{-1})^{-s}$. Поэтому процесс ARIMA b_h получается из следующего приближения: $b_h = \gamma^* \alpha^{(1)} * t\alpha^{(d)}$, где $t\alpha^{(d)}$ – усеченный фильтр заранее выбранного порядка *d*. Подробнее об этом см. в [AS] (статья доступна через Интернет).

Коэффициенты деталей берутся при помощи генератора randn нормально распределенных случайных чисел с нулевым средним и дисперсией 1. Затем применяется многократное вейвлет-восстановление. См. описание функции wfbm в документации MATLAB и книгу [M].

2. Функция wfbmesti – оценка параметра дробного броуновского движения. Производит оценку индекса фрактальности Херста *H* сигнала как дробного броуновского движения.

Синтаксис:

```
HEST = wfbmesti(X)
```

Характеристики сигнала и его приближения

Это следующие функции:

- **wenergy** – энергия 1D-вейвлет или вейвлет-пакетных коэффициентов;
- **wvarchg** – поиск точек изменения дисперсии сигнала;
- **measerr** – оценки качества приближения.

1. Функция wenergy – энергия 1D-вейвлет- или вейвлет-пакетных коэффициентов.

Синтаксис:

```
[Ea,Ed]= wenergy(C,L)
E = wenergy(T)
```

Описание.

Для одномерного вейвлет-разложения [C, L] (см. wavedec) функция [Ea,Ed]= wenergy(C,L) возвращает значение Ea, которое есть процент энергии коэффициентов аппроксимации, и Ed – вектор, содержащий проценты энергии коэффициентов деталей.

Для пакетного дерева T (см. wptree, wpdec, wpdec2) функция E = wenergy (T) возвращает вектор E, который содержит проценты энергии коэффициентов в терминальных (конечных) узлах дерева T.

2. Функция wvarchg – поиск точек изменения дисперсии сигнала.
Синтаксис:

[PTS_OPT,KOPT,T_EST] = wvarchg(Y,K,D)

Описание.

Функция позволяет выделить в сигнале некоторые блоки по признаку дисперсии. Используя вейвлет-коэффициенты, функция вычисляет примерные точки перехода дисперсии сигнала Y для j точек перехода, j = 0, 1, 2..., K. Целое число D является минимальной задержкой между двумя точками перехода. Числа K и D должны быть целыми и такими, что 1< K << length(Y) и 1 ≤ D << length(Y). Сигнал Y должен быть с нулевым средним.

Целое число KOPT – это предложенное число точек перехода (0 ≤ KOPT ≤ K). Вектор PTS_OPT содержит соответствующие точки перехода.

Матрица T_EST имеет следующую структуру. Для 1 ≤ k ≤ K матрица T_EST (k + 1, 1:k) еще содержит k моментов точек изменения дисперсии и затем если KOPT > 0, PTS_OPT = T_EST (KOPT+1,1:KOPT), иначе PTS_OPT = [].

wvarchg(Y,K) эквивалентно wvarchg(Y,K,10),
wvarchg(Y) эквивалентно wvarchg(Y,6,10).

3. Функция measerr – оценки качества приближения.
Синтаксис:

[PSNR,MSE,MAXERR,L2RAT]= measerr(X,XAPP)
[...] = measerr(...,BPS)

Описание.

Функция [PSNR,MSE,MAXERR,L2RAT]= measerr(X,XAPP) возвращает:

- PSNR – пиковое отношение сигнал–шум;
- MSE – среднеквадратичную ошибку;
- MAXERR – максимальную квадратичную ошибку и
- L2RAT – отношение квадратичных норм для входного сигнала или изображения X и его приближения – XAPP.

3.3.2. Анализ изображений

В данном разделе рассмотрим вейвлет-анализ двумерных сигналов, называмых обычно изображениями. Для их анализа используются следующие группы функций.

Функции разложения и восстановления:

- **dwt2** – одноуровневое дискретное двумерное вейвлет-преобразование;
- **idwt2** – одноуровневое обратное двумерное вейвлет-преобразование;
- **wavedec2** – многоуровневое двумерное вейвлет-разложение;
- **waverec2** – многоуровневое двумерное вейвлет-восстановление;
- **appcoef2** – нахождение коэффициентов аппроксимации;
- **detcoef2** – нахождение детализирующих коэффициентов;
- **wrcoef2** – 2D-восстановление сигнала по одной ветви коэффициентов;
- **upcoef2** – прямое восстановление из 2D-вейвлет-коэффициентов;
- **upwlev2** – одноуровневая реконструкция 2D-вейвлет-разложения;
- **ndwt2** – недецимированное 2D-вейвлет-преобразование;
- **indwt2** – обратное недецимированное 2D-вейвлет-преобразование;
- **wenergy2** – энергия для 2D-вейвлет-разложения.

Стационарное 2D-вейвлет-преобразование:

- **swt2** – дискретное стационарное 2D-вейвлет-преобразование;
- **iswt2** – обратное стационарное 2D-вейвлет-преобразование.

Наложение изображений:

- **wfusimg** – слияние двух изображений;
- **wfusmat** – слияние двух матриц или массивов.

Методы лифтинга для изображений: lwt2, ilwt2, iswt2 и lwtcoef2 – выполняют то же саммое, что и обычные функции, но при помощи подправленного лифтингом вейвлета. Для двумерного вейвлет-анализа сигнала создан графический интерфейс пользователя wavemenu и можно также использовать общие функции:

- **wcodemat** – расширенное псевдоцветное матричное масштабирование;
- **dwtmode** – мода расширения дискретного вейвлет-преобразования;
- **measerr** – оценки качества приближения;
- **wkeep** – извлечение части вектора или матрицы.

Изображения в MATLAB

Изображение в MatLab может быть задано несколькими способами.

Бинарное (черно-белое) изображение задается матрицей, элементы которой принимают два значения: 0 и 1.

Полутоновое (grayscale) изображение задается матрицей, элементы которой принимают действительные значения от 0 до 1 и характеризуют яркость соответствующего пикселя дисплея.

Палитровое изображение задается двумя матрицами X и map. Цветовая карта map состоит из трех столбцов, задающих интенсивности красного (R), зеленого (G) и синего (B) цветов. Значение элемента матрицы X есть номер строки в матрице map, в которой указаны интенсивности цветов пикселя, соответствующего элементу матрицы X.

Полноцветное (RGB) изображение представляет собой трехмерный массив типа размера $n \times m \times 3$. Это три $(n \times m)$-матрицы, содержащие интенсивности красного (R), зеленого (G) и синего (B) цветов каждого пикселя.

Цветовая карта **map** обычно сопутствует палитровому изображению. Цветовую карту можно задать командой colormap. При этом можно задать дополнительно оттенки и количество цветов (размер). Наример, map=colormap, map=colormap(hsv(256)), map= colormap(jet), map = pink(255), map = jet(1000). При загрузке тестовых сигналов MATLAB, например

```
load woman;
```

загружается матрица изображения X порядка 256 и одновременно матрица map из трех столбцов размера 255. При этом значения элементов матрицы X есть целые числа – номера строк в матрице map, в которых указаны интенсивности цветов, соответствующие элементам матрицы X. Подробности см. в Help MatLab.

Функция imshow – вывод на экран изображения. Варианты использования:

- imshow(I) выводит на экран полутоновое изображение I;
- imshow(I,[low high]) выводит на экран полутоновое изображение I с заданным диапазоном [low high]. Значения low (и любые меньшие, чем low) отображаются как черные; значения high (и любые значения, большие high) отображаются как белые. Промежуточные значения отображаются как промежуточные оттенки серого, используя число по умолчанию уровней яркости. Если использовать пустую матрицу ([]) для [low high], то imshow использует [min(I(:)) max(I(:))], то есть минимальное значение в I отображается как черное, а максимальное значение – как белое;
- imshow(RGB) выводит на экран полноцветное изображение RGB;
- imshow(BW) выводит на экран бинарное (черно-белое) изображение BW;

- imshow(X,map) выводит на экран палитровое изображение X с цветовой картой map. У цветовой матрицы map может быть любое число рядов, но должно быть точно 3 столбца. Каждая строка ряда интерпретируется как цвет, с первым элементом, определяющим интенсивность красного цвета, вторым – зеленого, третьим – синего цвета. Цветовая интенсивность может быть определена на интервале от 0.0 до 1.0;
- imshow(filename) выводит на экран изображение из файла.

Функция image – визуализация матрицы и вывод на экран изображения. Функция **image** может быть применена к любому сигналу для его визуализации. Каждому элементу матрицы MATLAB ставит в соответствие квадратик в матрице, закрашенный определенным образом.

Синтаксис:

image(C)
image(x,y,C)
image(x,y,C,'PropertyName',PropertyValue,...)

Описание.

Функция image(C) показывает матрицу как изображение. Каждый элемент определяет цвет соответствующего квадратика.

Функция image(x,y,C), где x и y – двухэлементные векторы, определяет диапазон значений по осям x и y, но отображается то же самое (полностью) изображение, что и image(C). Изменение отметок по осям иногда бывает необходимо.

Тестовые изображения в MatLab представляют палитровые изображения, которые задаются двумя матрицами X и map, где map – цветовая карта, а значения элементов матрицы X есть номера строк в матрице map, в которых указаны интенсивности цветов, соответствующие элементам матрицы X. Поэтому если применить просто команду image(X), то мы получим искаженное изображение (номера строк в матрице map). Чтобы получить реальное изображение, необходимо еще применить команду colormap(map):

```
image(X);
colormap(map);
```

Функция wcodemat – псевдоцветное матричное масштабирование. Для того чтобы получить на экране более качественное изображение, можно сначала применить к матрице X псевдоцветное матричное масштабирование командой wcodemat, для указания количества градаций цветов (размер цветовой карты).

Синтаксис:

Y = wcodemat(X)
Y = wcodemat(X,NBCODES)
Y = wcodemat(X,NBCODES,OPT)
Y = wcodemat(X,NBCODES,OPT,ABSOL)

Описание.

Функция Y = wcodemat(X) перемасштабирует матрицу X в целые числа в диапазоне [1,16].

Функция Y = wcodemat(X,NBCODES) перемасштабирует матрицу X в целые числа в диапазоне [1,NBCODES].

Функция Y = wcodemat(X,NBCODES,OPT) перемасштабирует матрицу X в целые числа в диапазоне [1,NBCODES] вдоль направления, указанного параметром OPT. Допустимые значения OPT: 'column' (или 'c'), 'row' (или 'r') и 'mat' (или 'm') – X масштабируется глобально (по умолчанию).

Функция Y = wcodemat(X,NBCODES,OPT,ABSOL) перемасштабирует матрицу X в целые числа на основе абсолютных значений, если ABSOL=1 (по умолчанию) и с учетом знаков – если ABSOL=0.

Например, Y=wcodemat(X). Тогда image(Y) – это изображение с использованием 16 градаций синего цвета. Хорошие результаты дает использование дополнительных оттенков, например командой colormap(pink(n)) перед применением image.

Пример 12. Загружаем тестовый сигнал «woman». При этом в рабочее пространство загружаются матрица изображения X размера 256 на 256 и цветовая карта map из трех одинаковых столбцов, то есть размера 255 на 3. Сначала выводим на экран изображение без масштабирования в цветовой карте map, а затем масштабируем по строкам в диапазоне от 1 до 50 и выводим на экран (рис. 3.3.11). Изображение справа получилось темнее, поскольку диапазон стал меньше: от 1 до 50, а для левого – от 1 до 250.

```
load woman; % Загрузка тестового сигнала: матрица X и матрица map
NBCOL = size(map,1);
subplot(1,2,1);
image(X);
title('Unscaled Image');
   % Масштабирование по строкам с числом градаций от 1 до 50
subplot(1,2,2);
image(wcodemat(X,50,'r'));
colormap(map);
title('Scaled Image');
```

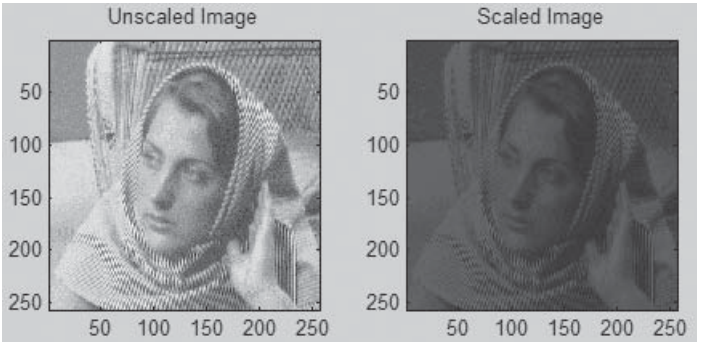


Рис. 3.3.11. Масштабирование изображения

Функция imread – чтение изображения из файла.
Синтаксис:

A=imread(filename, fmt)
[X,map]=imread(filename, fmt)

Описание. Входные аргументы: filename, fmt – это файл, из которого считывается изображение и его формат. Поддерживаются практически все форматы fmt: bmp, jpg, pcx, tif, hdf, png, xwd (см. полный список в Help для imread).

Если исходное изображение полутоновое, то A – матрица, если полноцветное, то A – трехмерный массив *n*-на-*m*-на-3.

Второй вариант функции [X,map]=imread(filename, fmt) создает палитровое изображение X с картой цветов map. Заметим, что в случае чтения из файла, например wpeppers.jpg, цветовая карта отсутствует. Карта цветов map сопутствует тестовым изображениям.

Функция imwrite – запись изображения в файл.
Синтаксис:

imwrite(A,filename,fmt)
imwrite(X,map,filename,fmt)

Допускаются те же форматы: bmp, jpg, pcx, tif, hdf, png, xwd и т. д. Если A – матрица, то изображение полутоновое, если A – трехмерный массив, то изображение полноцветное. Второй вариант функции применяется к палитровому изображению X с картой цветов map.

Преобразование форматов:

- [X,map]=gray2ind(I,n) – преобразует полутоновое изображение в палитровое. Палитра map содержит n градаций серого цвета;

- I= ind2gray(X,map) – преобразует палитровое изображение в полутоновое;
- RGB= ind2rgb(X,map) – преобразует палитровое изображение в полноцветное;
- I= mat2gray(A) – преобразует матрицу A в полутоновое изображение;
- I= rgb2gray(RGB) – преобразует полноцветное изображение в полутоновое;
- [X,map]= rgb2ind(RGB,tol) – преобразует полноцветное изображение в палитровое, используя палитру всех цветов исходного изображения с шагом tol;
- [X,map]= rgb2ind(RGB,n) – преобразует полноцветное изображение в палитровое, используя n цветов;
- X= rgb2ind(RGB,map) – преобразует полноцветное изображение в палитровое с использованием заданной цветовой карты.

Более подробную информацию можно найти в Help MATLAB\Graphics\Images\Image File Operations.

Вейвлет-преобразования двумерных сигналов

Двумерные сигналы $s(x, y)$ будем называть изображениями. Алгоритм разложения изображений аналогичен тому, что применяется в одномерном случае. Пусть $\varphi(x)$ – масштабирующая вейвлет-функция и $\psi(x)$ – вейвлет. Как известно, они порождают базисные функции $\varphi_{j,n}(x)$ и $\psi_{j,k}(x)$. Двумерный сигнал $s(x, y)$ раскладывается по базисным в $L^2(\mathbf{R}^2)$ функциям $\varphi_{j,n}(x)\varphi_{j,m}(y)$, $\varphi_{j,n}(x)\psi_{j,m}(y)$, $\psi_{j,n}(x)\varphi_{j,m}(y)$ и $\psi_{j,n}(x)\ \psi_{j,m}(y)$. Соответствующие коэффициенты принято называть следующим образом.

Аппроксимирующие коэффициенты cA_j получаются как коэффициенты разложения по вейвлет-базису $\varphi_{j,n}(x)\varphi_{j,m}(y)$.

Горизонтальные детализирующие коэффициенты cH_j получаются как коэффициенты разложения по вейвлет-базису $\varphi_{j,n}(x)\psi_{j,m}(y)$.

Вертикальные детализирующие коэффициенты cV_j получаются как коэффициенты разложения по вейвлет-базису $\psi_{j,n}(x)\varphi_{j,m}(y)$.

Диагональные детализирующие коэффициенты cD_j получаются как коэффициенты разложения по вейвлет-базису $\psi_{j,n}(x)\psi_{j,m}(y)$.

Практически входной сигнал задается матрицей *s*. При разложении этого сигнала получаются указанные выше четыре типа коэффициентов. Например, горизонтальные детализирующие коэффициенты cH_1 получаются сверткой строк матрицы *s* с низкочастотным фильтром вейвлета и децимацией и затем сверткой столбцов полу-

ченной матрицы с высокочастотным фильтром и децимацией. Схему разложения можно изобразить в виде

$C_0 \rightarrow (cA_1, cH_1, cV_1, cD_1) \rightarrow (cA_2, cH_2, cV_2, cD_2, cH_1, cV_1, cD_1) \rightarrow ...$

Поскольку массив начальных коэффициентов двумерный, то более естественно схему разложения сигнала C_0 изобразить в (принятом в MatLab) виде (рис. 3.3.12).

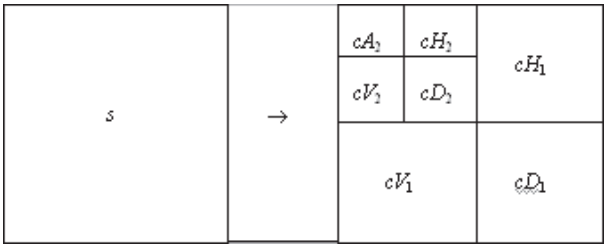


Рис. 3.3.12. Схема разложения двумерного сигнала

Заметим, что эта схема отражает изменение размеров массивов коэффициентов разложения. А именно если сигнал C_0 имел размерность $n \times n$, то коэффициенты первого уровня разложения cA_1, cH_1, cD_1, cV_1 будут иметь размерность $n/2 \times n/2$.

Функции разложения и восстановления

Это следующие функции:

- **dwt2** – одноуровневое дискретное двумерное вейвлет-преобразование;
- **idwt2** – одноуровневое обратное двумерное вейвлет-преобразование;
- **wavedec2** – многоуровневое двумерное вейвлет-разложение;
- **waverec2** – многоуровневое двумерное вейвлет-восстановление;
- **appcoef2** – нахождение коэффициентов аппроксимации;
- **detcoef2** – нахождение детализирующих коэффициентов;
- **wrcoef2** – 2D-восстановление сигнала по одной ветви коэффициентов;
- **upwlev2** – одноуровневая реконструкция 2D-вейвлет-разложения;
- **upcoef2** – прямое восстановление из 2D-вейвлет-коэффициентов;

- **ndwt2** – недецимированное 2D-вейвлет-преобразование;
- **indwt2** – обратное недецимированное 2D-вейвлет-преобразование;
- **wenergy2** – энергия для 2D-вейвлет-разложения.

Рассмотрим их подробнее.

1. Функция dwt2 – дискретное двумерное вейвлет-преобразование. Для входной матрицы X функция dwt2 вычисляет аппроксимирующие коэффициенты *cA* и детализирующие коэффициенты первого уровня трех типов:

- горизонтальные *cH*;
- вертикальные *cV* и
- диагональные *cD*,

используя для этого тензорное произведение $w(x)w(y)$ вейвлета $w(x)$. Все полученные коэффициенты образуют матрицы размера в два раза меньше, чем исходная матрица X.

Синтаксис:

```
[cA,cH,cV,cD] = dwt2(X,'wname')
[cA,cH,cV,cD] = dwt2(X,Lo_D,Hi_D)
[cA,cH,cV,cD] = dwt2(...,'mode',MODE).
```

Здесь: 'wname' – имя вейвлета, Lo_D и Hi_D – фильтры разложения. Можно установить моду расширения вейвлет-разложения параметром MODE. Например, [cA,cH,cV,cD] = dwt2(x,'db1','mode','sym'). Если мода расширения DWT установлена 'per' (периодичность), то

size(cA) = size(cH) = size(cV) = size(cD) = ceil(sx/2),

в остальных случаях

size(cA) = size(cH) = size(cV) = size(cD) = floor((sx+lf-1)/2),

где sx = size(X) и lf – длина фильтров. Напомним, что ceil(y) – это ближайшее к y целое число справа, а floor(y) – это целая часть числа.

Пример 13. Загружаем тестовый сигнал detail (магический квадрат Дюрера). При этом X содержит загруженное изображение, а map – цветовую карту. Делаем 1-уровневое разложение. В заключение вызываем на экран изображения полученных разложений.

```
load detail; nbc = size(map,1);
[cA1,cH1,cV1,cD1] = dwt2(X,'db1');
colormap(pink(nbc));
cod_X = wcodemat(X,nbc); cod_cA1 = wcodemat(cA1,nbc);
cod_cH1 = wcodemat(cH1,nbc); cod_cV1 = wcodemat(cV1,nbc);
cod_cD1 = wcodemat(cD1,nbc); dec2d = [cod_cA1,cod_cH1;cod_cV1,cod_cD1];
I= mat2gray(dec2d); imshow(I); %image(dec2d)
```

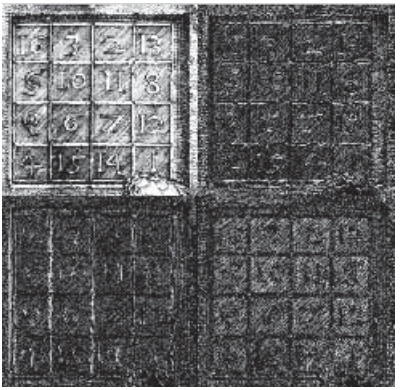


Рис. 3.3.13. Разложение
магического квадрата Дюрера

На рисунке хорошо видно, что горизонтальные коэффициенты cH1 (правый верхний угол) больше отражают горизонтальные элементы рисунка, а вертикальные коэффициенты cV1 (левый нижний угол) больше отражают вертикальные элементы рисунка. В правом нижнем углу преобладают диагональные линии, соответствующие диагональным коэффициентам cD1.

2. Функция idwt2 – одноуровневое обратное двумерное преобразование. Это обратная функция к dwt2. По известным аппроксимирующим коэффициентам *cA* и детализирующим коэффициентам трех типов *cH, cV, cD* команда idwt2 вычисляет значения сигнала *X,* используя либо вейвлет, либо его фильтры реконструкции. Алгоритм восстановления – обратный к разложению.

Синтаксис:

```
X = idwt2(cA,cH,cV,cD,'wname')
X = idwt2(cA,cH,cV,cD,Lo_R,Hi_R)
X = idwt2(cA,cH,cV,cD,'wname',S)
X = idwt2(cA,cH,cV,cD,Lo_R,Hi_R,S)
X = idwt2(...,'mode',MODE)
X = idwt2(cA,[],[],[],...)
X = idwt2([],cH,[],[],...)
```

Входные параметры:

- cA,cH,cV,cD – коэффициенты разложения;
- 'wname' – имя вейвлета;
- Lo_R – низкочастотный фильтр восстановления;

- Hi_R – высокочастотный фильтр восстановления;
- S – размер центральной части массива, которую нужно восстановить командой idwt2;
- MODE – определяет метод расширения при вейвлет-преобразовании.

Функция X = idwt2(cA,[],[],[],...) осуществляет одноуровневое восстановление коэффициентов вектора s только по аппроксимирующим коэффициентам cA.

Функция X = idwt2([],cH,[],[],...) осуществляет одноуровневое восстановление коэффициентов вектора s по детализирующим коэффициентам cH. То же самое относится к X = idwt2([],[],cV,[],...) и X = idwt2([],[],[],cD,...).

3. wavedec2 – многоуровневое двумерное вейвлет-разложение.

Синтаксис:

```
[C,S] = wavedec2(X,N,'wname')
[C,S] = wavedec2(X,N,Lo_D,Hi_D)
```

Функция производит вейвлет-разложение матрицы X до уровня N, используя вейвлет 'wname' или фильтры Lo_D,Hi_D. На выходе получаются вектор разложения C и соответствующая матрица учета длин S.

Вектор коэффициентов C устроен следующим образом:

```
C = [ A(N), H(N), V(N), D(N), H(N-1), V(N-1), D(N-1), ...,
H(1), V(1), D(1) ],
```

где A, H, V, D – строки векторов, такие что:

- A – коэффициенты аппроксимации;
- H – горизонтальные детализирующие коэффициенты;
- V– вертикальные детализирующие коэффициенты;
- D – диагональные детализирующие коэффициенты.

Каждый вектор есть вектор столбцеобразного хранения матрицы.

Матрица учета размерностей S порядка (N + 2)×2 имеет следующее строение:

- два числа первой строки S(1,:) – это размер [p, q] матрицы p на q аппроксимирующих коэффициентов уровня N;
- следующие строки S(i,:) – это размеры матриц детализирующих коэффициентов уровней N – i + 2 для i = 2, ... N + 1;
- последняя строка S(N+2,:) = size(X) – размер [n, m] матрицы исходного сигнала.

Алгоритм разложения изображений аналогичен тому, что применяется в одномерном случае. Особенностью двумерного случая яв-

ляется то, что при разложении аппроксимирующих коэффициентов уровня *j* получаются четыре компоненты: аппроксимация на уровне *j* + 1 и детали в трех ориентациях (горизонтальной, вертикальной и диагональной).

Пример 14. Загружаем изображение X. Производим разложение до уровня 2, используя вейвлет db1. Изучаем строение векторов разложения.

```
load woman;
[c,s] = wavedec2(X,2,'db1');
sizex = size(X)
sizex =
    256  256
sizec = size(c)
sizec =
    1  65536
s =
    64  64
    64  64
    128  128
    256  256
```

4. Функция waverec2 – многоуровневое вейвлет-восстановление.

Синтаксис:

X = waverec2(C,S,'wname')
X = waverec2(C,S,Lo_R,Hi_R).

Обратная функция к wavedec2. Восстанавливает матрицу X, используя структуру вейвлет-разложения [C,S] (см. wavedec2). Отметим, что X = waverec2(C,S,'wname') эквивалентно X = appcoef2(C,S,'wname',0).

5. Функция appcoef2 – нахождение коэффициентов аппроксимации. Команда appcoef2 вычисляет аппроксимирующие коэффициенты сигнала задаваемого уровня n, используя полученные ранее векторы многоуровневого разложения [C,S].

Синтаксис:

A = appcoef2(C,S,'wname',n),
A = appcoef2(C,S,'wname'),
A = appcoef2(C,S,Lo_R,Hi_R),
A = appcoef2(C,S,Lo_R,Hi_R,n).

Здесь [C,S] – вейвлет-разложение, найденное функцией wavedec2, n – уровень, для которого нужно найти аппроксимирующие коэф-

фициенты разложения, 'wname' – имя вейвлета. Если n не указано, то берется последний уровень N разложения сигнала. Исходное разложение [C,S] может быть более глубоким, чем n. Отметим, что X = appcoef2(C,S,'wname',0) эквивалентно X = waverec2(C,S,'wname').

6. Функция detcoef2 – нахождение детализирующих коэффициентов. Команда detcoef2 вычисляет детализирующие коэффициенты сигнала задаваемого уровня n, используя полученные ранее векторы многоуровневого разложения [C,S].

Синтаксис:

D = detcoef2(O,C,S,n).

Функция извлекает из вейвлет-разложения [C,S] (см. wavedec2) горизонтальные, вертикальные и диагональные детализирующие коэффициенты уровня N в зависимости от значения параметра O = 'h', или 'v', или 'd', соответственно.

Функция [H,V,D] = detcoef2('all',C,S,n) извлекает все коэффициенты уровня n – горизонтальные, вертикальные и диагональные.

7. wrcoef2 – восстановление двумерного сигнала по одной ветви вейвлет-коэффициентов. Если известно вейвлет-разложение [C,S] некоторого уровня разложения N0, то команда wrcoef2 реконструирует матрицу X по одной ветви вейвлет-коэффициентов на уровне разложения N ≤ N0.

Синтаксис:

X = wrcoef2('type',C,S,'wname',N),
X = wrcoef2('type',C,S,Lo_R,Hi_R,N),
X = wrcoef2('type',C,S,'wname'),
X = wrcoef2('type',C,S,Lo_R,Hi_R).

Здесь 'type' определяет тип ветви:

- 'type' = 'a' – восстановление по аппроксимирующим коэффициентам;
- 'type' = 'h' или 'v', или 'd' – восстановление по детализирующим коэффициентам, соответственно горизонтальным, вертикальным или диагональным.

Если N не указано, то имеется в виду максимальный уровень N0.

8. Функция upcoef2 – прямое восстановление из 2D-вейвлет-коэффициентов. Функция upcoef восстанавливает матрицу Y из коэффициентов X заданного уровня N, применяя N раз процедуру обратного вейвлет-преобразования. При этом если X считается детализирующими коэффициентами (O = 'd'), то аппроксимирующие коэффициенты берутся нулевыми, и наоборот.

Синтаксис:

```
Y = upcoef2(O,X,'wname',N,S),   Y = upcoef2(O,X,Lo_R,Hi_R,N,S),
Y = upcoef2(O,X,'wname',N),     Y = upcoef2(O,X,Lo_R,Hi_R,N),
Y = upcoef2(O,X,'wname'),       Y = upcoef2(O,X,Lo_R,Hi_R).
```

Функция Y = upcoef2(O,X,'wname',N,S) вычисляет N-шаговую реконструкцию матрицы X и берет центральную часть размера S. 'wname' – имя вейвлета. Если O = 'a', то реконструируются аппроксимирующие коэффициенты X. Если O = 'h', 'v' или 'd', то восстанавливаются детализирующие коэффициенты X. Вместо вейвлета можно использовать фильтры реконструкции Lo_R и Hi_R.

Y = upcoef2(O,X,'wname') эквивалентно
Y = upcoef2(O,X,'wname',1).

9. Функция upwlev2 – одноуровневая реконструкция 2D-вейвлет-разложения. Если известны векторы вейвлет-разложения [C,S] (см. wavedec2) некоторого уровня разложения N0, то команда upwlev2 находит новое вейвлет-разложение [NC,NS] уровня разложения N0 – 1, то есть на один уровень меньше. Кроме того, извлекаются также последние аппроксимирующие коэффициенты cA, то есть уровня N0.

Синтаксис:

```
[NC,NS,cA] = upwlev2(C,S,'wname'),
[NC,NS,cA] = upwlev2(C,S,Lo_R,Hi_R)
```

Вместо вейвлета 'wname' можно использовать фильтры реконструкции Lo_R и Hi_R.

10. Функция ndwt2 – недецимированное 2D-вейвлет-преобразование. Функция делает разложение фильтрами вейвлета без последующей децимации.

Синтаксис:

```
WT = ndwt2(X,N,'wname')
WT = ndwt2(X,N,'wname','mode','ExtM')
WT = ndwt2(X,N,WF,...)
```

Выходной параметр WT представляет собой структуру с шестью полями:

- sizeINI – размер двумерного массива X;
- level – глубина разложения N;
- mode – имя метода расширения для вейвлет-разложения;
- filters – структура с четырьмя полями: LoD, HiD, LoR, HiR – это фильтры, используемые в разложении;

- dec – массив ячеек размера 1×(3*N + 1), содержащих коэффициенты разложения. dec{1} содержат коэффициенты аппроксимации, а ячейки dec{j} j = 2,..., 3*N + 1 содержат детализирующие коэффициенты от уровня N до 1;
- sizes – массив размера (N + 1)×2, содержащий размеры компонент. Надо помнить, что это недецимированные коэффициенты.

11. Функция indwt2 – обратное недецимированное 2D-вейвлет-преобразование. Производит вейвлет-восстановление из структуры многоуровневого недецимированного разложения W.

Синтаксис:

```
X = indwt2(W)
X= indwt2(W,'a',0)
X= indwt2(W,'ca',0)
C = indwt2(W,TYPE,n)
C = indwt2(W,TYPE)
```

Первые три функции X = indwt(W), X= indwt(W,'a',0) или X= indwt(W,'ca',0) производят полное восстановление сигнала по структуре разложения W.

Функция C = indwt(W,TYPE,n) вычисляет компоненту, восстановленную по одной ветви коэффициентов с уровня n ≤ N вейвлет-разложения уровня N. Эта ветвь коэффициентов определяется параметром TYPE. Допустимые значения параметра TYPE следующие:

- группа символов 'xy', где 'x' и 'y' могут быть из множества {'a','d','l','h'} или {'A','D','L','H'}), где 'A' или 'L' определяют низкочастотную компоненту и 'D' или 'H' – высокочастотную;
- символ 'd' (или 'h' или 'D' или 'H'), определяет сумму всех компонент, отличных от высокочастотной (все детали уровня n).

C = indwt2(W,TYPE) эквивалентно C = indwt2(W,TYPE,N), где N – уровень разложения.

12. Функция wenergy2 – энергия 2D-вейвлет- или вейвлет-пакетных коэффициентов.

Синтаксис:

```
[Ea,Eh,Ev,Ed]= wenergy2(C,S)
[Ea,EDetail] = wenergy2(C,S)
```

Описание.

Для 2D-вейвлет-разложения [C,S] (см. wavedec2) функция [Ea,Eh,Ev,Ed] = wenergy2(C,S) возвращает значение Ea, которое есть процент энергии коэффициентов аппроксимации, и векторы Eh, Ev,

Ed, содержащие проценты энергии коэффициентов деталей. EDetail – это сумма всех векторов Eh, Ev и Ed.

Стационарное 2D-вейвлет-преобразование

1. Функция swt2 – дискретное стационарное 2D-вейвлет-преобразование. В этом случае децимация не производится и коэффициенты разложения имеют те же размеры, что и исходный сигнал.

Синтаксис:

SWC = swt2(X,N,'wname'), SWC = swt2(X,N,Lo_D,Hi_D),
[A,H,V,D] = swt2(X,N,'wname'),
[A,H,V,D] = swt2(X,N,Lo_D,Hi_D)

Функция swt2 производит разложение матрицы X до уровня N > 0, используя ортогональный вейвлет 'wname' или его фильтры разложения. Размеры size(X,1) и size(X,2) матрицы X должны быть кратными 2^N, где N – уровень разложения.

Выходные трехмерные массивы [A,H,V,D] имеют следующую структуру. Для 1 ≤ i ≤ N выходная матрица A(:,:,i) содержит аппроксимирующие коэффициенты уровня i, выходные матрицы H(:,:,i), V(:,:,i) и D(:,:,i) содержат детализирующие коэффициенты (горизонтальные, вертикальные и диагональные) уровня i.

SWC = [H(:,:,1:N) ; V(:,:,1:N) ; D(:,:,1:N) ; A(:,:,N)].

2. Функция iswt2 – обратное стационарное 2D-вейвлет-преобразование.

Синтаксис:

X = iswt2(SWC,'*wname*'), X = iswt2(SWC,Lo_R,Hi_R),
X = iswt2(A,H,V,D,'*wname*'), X = iswt2(A,H,V,D,Lo_R,Hi_R).

Функция iswt2 производит стационарное двумерное вейвлет-восстановление, используя ортогональный вейвлет 'wname', или его фильтры реконструкции на основе известной структуры разложения SWC или [A,H,V,D] (см. swt2).

Слияние (наложение) изображений

Слияние (наложение) изображений и массивов можно сделать при помощи приложения **Image Fusion** главного меню пакета Wavelet Toolbox, а также из командной строки MATLAB при помощи функций:

- **wfusimg** – слияние двух изображений;
- **wfusmat** – слияние двух матриц или массивов.

Идея *слияния изображений* заключается в следующем. После вейвлет-разложения изображений делается выбор коэффициентов для нового изображения из коэффициентов аппроксимации и детализации исходных двух изображений. Затем по выбранным коэффициентам восстанавливается изображение обычным методом вейвлет-восстановления.

Возможно несколько *методов выбора* коэффициентов для нового изображения из двух, соответствующих двум оригинальным изображениям:

- 'max' – выбор максимального из двух коэффициентов;
- 'min' – выбор минимального из двух коэффициентов;
- 'mean' – выбор среднего значения из двух коэффициентов;
- 'img1' – выбор коэффициента первого изображения;
- 'img2' – выбор коэффициента второго изображения;
- 'rand' – случайный выбор из двух коэффициентов.

Кроме того, имеется еще несколько параметрических методов выбора. Поскольку коэффициенты вейвлет-разложений изображений составлены из матриц, поясним эти методы на матрицах. Пусть A(i, j) и B(i, j) представляют, соответственно, коэффициенты вейвлет-разложений первого и второго изображений и C(i, j) – коэффициенты вейвлет-разложения для объединенного изображения. Тогда доступные параметрические, зависящие от параметра $0 \le t \le 1$, методы выбора коэффициентов следующие:

- 'linear' – линейный, тогда C = A* t + B*(1 – t), $0 \le t \le 1$;
- 'UD_fusion' – слияние сверху вниз с параметром $t \ge 0$. Для определения этого метода задается линейно распределенный массив

  x = linspace(0,1,size(A,1));

  определяется степенная функция $P = x^t$:

  P = x.^t;

  Тогда каждая строка матрицы C вычисляется по формуле

  C(i,:) = A(i,:)*(1-P(i)) + B(i,:)*P(i);

  Этот метод обеспечивает плавное перетекание сверху вниз первого изображения во второе (в соответствии с функцией x^t).
- 'DU_fusion' – слияние снизу вверх;
- 'RL_fusion' – слияние справа налево;
- 'UserDEF' – слияние слева направо;

- 'UserDEF' – определенный пользователем метод, тогда вместо параметра задается функция пользователя Func.Name, такая что C = Func.Name(A, B).

1. Функция wfusimg – слияние двух изображений.

Синтаксис

```
XFUS = wfusimg(X1,X2,WNAME,LEVEL,
AFUSMETH,DFUSMETH)
[XFUS,TXFUS,TX1,TX2] =
wfusimg(X1,X2,WNAME,LEVEL,AFUSMETH,DFUSMETH)
wfusimg(X1,X2,WNAME,LEVEL,AFUSMETH,DFUSMETH,
FLAGPLOT)
```

Описание.

Принцип слияния изображений с использованием вейвлетов заключается в слиянии коэффициентов аппроксимации и детализации вейвлет-разложения двух оригинальных изображений. После вейвлет-разложения изображений выбор коэффициентов для нового изображения делается из коэффициентов аппроксимации и детализации оригинальных изображений. Возможен выбор максимального, минимального, среднего, элемента первого изображения, элемента второго изображения, или случайным выбором элемента. Функция

```
XFUS = wfusimg(X1,X2,WNAME,LEVEL,
AFUSMETH,DFUSMETH)
```

возвращает изображение XFUS, полученное слиянием двух оригинальных изображений X1 и X2. Каждый метод слияния, заданный параметрами AFUSMETH и DFUSMETH, определенным способом объединяет коэффициенты разложений X1 и X2 на уровне LEVEL и использует вейвлет WNAME.

Матрицы X1 и X2 должны иметь один и тот же размер и должны быть связаны с изображениями общей цветовой картой colormap. Параметры

```
AFUSMETH и DFUSMETH
```

определяют метод слияния для приближений и деталей соответственно. Функция

```
[XFUS,TXFUS,TX1,TX2]=
wfusimg(X1,X2,WNAME,LEVEL,AFUSMETH,DFUSMETH)
```

возвращает, в дополнение к матрице XFUS, три объекта класса WDECTREE, ассоциированные с XFUS, X1 и X2 соответственно. Функция

wfusimg(X1,X2,WNAME,LEVEL,AFUSMETH,DFUSMETH, FLAGPLOT)

строит также графики объектов TXFUS, TX1 и TX2.

Fusmeth обозначает AFUSMETH или DFUSMETH. Доступные методы слияния:

- простой – может быть 'max', 'min', 'mean', 'img1', 'img2' или 'rand', когда слияние коэффициентов аппроксимаций и деталей, полученных из X1 и X2, делается выбором максимального, минимального, среднего из соответствующих элементов двух изображений, элемента первого изображения, элемента второго изображения или случайным выбором элемента;
- зависящий от параметра – в следующей форме:

  Fusmeth = struct('name',nameMETH,'param',paramMETH)

  где nameMETH может быть
  - 'linear';
  - 'UD_fusion' – слияние сверху вниз;
  - 'DU_fusion' – слияние снизу вверх;
  - 'RL_fusion' – слияние справа налево;
  - 'UserDEF' – пользовательское слияние.

Описание этих методов дано ниже для слияния матриц.

2. Функция wfusmat – слияние двух матриц или массивов.

Синтаксис

C = wfusmat(A,B,METHOD), [C,D] = wfusmat(A,B,METHOD)

Описание.

Функция C=wfusmat(A,B,METHOD) возвращает объединенную матрицу C, полученную *слиянием матриц* A и B с использованием метода слияния, определенного параметром METHOD. Матрицы A и B должны иметь одинаковый размер. Матрица вывода C имеет тот же самый размер, что A и B. Булева матрица D указывает, откуда выбран элемент. Параметр METHOD может принимать те же значения. При этом в функции

METHOD = struct('name',nameMETH,'param',paramMETH)

параметр nameMETH может быть

- 'linear' – тогда C = A*paramMETH + B*(1-paramMETH), где $0 \le$ paramMETH ≤ 1;
- 'UD_fusion' – слияние сверху вниз с параметром paramMETH ≥ 0.

```
x = linspace(0,1,size(A,1));
P = x.^paramMETH;
```

Тогда каждая строка C вычисляется по формуле

```
C(i,:) = A(i,:)*(1-P(i)) + B(i,:)*P(i);
```

В этом случае C(1,:) = A(1,:) и C(end,:) = A(end,:). Этот метод обеспечивает плавное перетекание сверху вниз первого массива во второй (в соответствии с функцией $p = x^t$);

- 'DU_fusion' – слияние снизу вверх;
- 'RL_fusion' – слияние справа налево;
- 'UserDEF' – слияние слева направо;
- 'UserDEF' – определенный пользователем метод, тогда paramMETH есть строка 'userFUNCTION', содержащая имя функции, такой что C = userFUNCTION (A, B).

Рассмотрим два простых примера слияния, предложенные в документации MATLAB. Первый пример объединяет два различных изображения, что приводит к новому изображению, а второй восстанавливает изображение из двух нечетких версий оригинального изображения.

Пример 15. Слияние двух различных изображений. Загружаем два оригинальных изображения: маска и бюст (файлы mask.mat и bust.mat находятся в каталоге C:\Program Files\MATLAB\R2006a\toolbox\wavelet\wavedemo\). Выполняем слияние двух изображений двумя различными методами слияния из вейвлет-разложения уровня 5, используя вейвлет db2. Сначала делается слияние выбором среднего значения и для коэффициентов аппроксимации, и для деталей. Во втором случае – слияние выбором максимальных значений для коэффициентов аппроксимации и минимальных – для деталей.

```
load mask; X1 = X;
load bust; X2 = X;
XFUSmean = wfusimg(X1,X2,'db2',5,'mean','mean');
XFUSmaxmin = wfusimg(X1,X2,'db2',5,'max','min');
% Графики оригинальных и синтезированных изображений
colormap(map);
subplot(221), image(X1), axis square, title('Mask')
subplot(222), image(X2), axis square, title('Bust')
subplot(223), image(XFUSmean), axis square,
title('Synthesized image, mean-mean')
subplot(224), image(XFUSmaxmin), axis square,
title('Synthesized image, max-min')
```

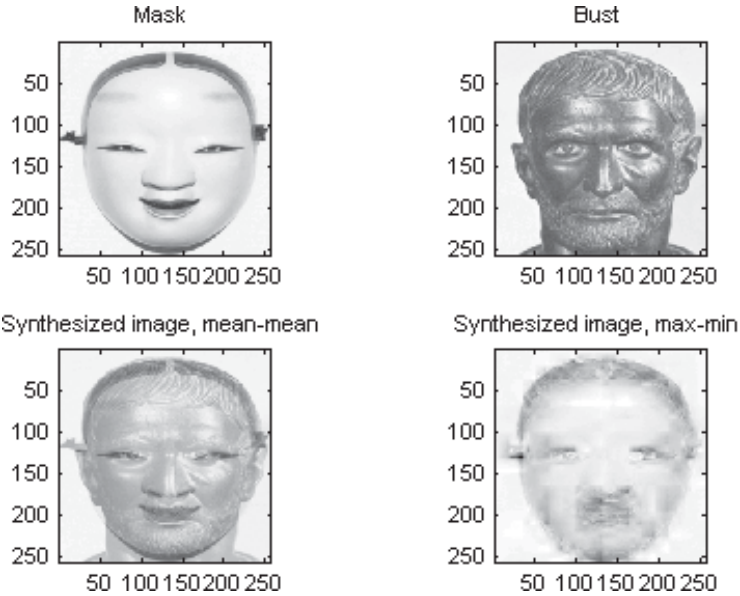


Рис. 3.3.14. Оригинальные
и синтезированные изображения

Пример 16. Восстановление изображения из двух нечетких изображений. Загружаем две оригинальные нечеткие фотографии. Файлы cathe_1.mat и cathe_2.mat находятся в каталоге C:\ProgramFiles\MATLAB\R2006a\toolbox\wavelet\wavedemo. Выполняем слияние двух изображений из вейвлет-разложения уровня 5, используя sym4, выбором максимумов абсолютных значений коэффициентов и аппроксимации, и детализации.

```
load cathe_1; X1 = X;
load cathe_2; X2 = X;
XFUS = wfusimg(X1,X2,'sym4',5,'max','max');
   % Графики оригинальных и синтезированного изображения
colormap(map);
subplot(221), image(X1), axis square, title('Catherine 1')
subplot(222), image(X2), axis square, title('Catherine 2')
subplot(223), image(XFUS), axis square, title('Synthesized image')
```

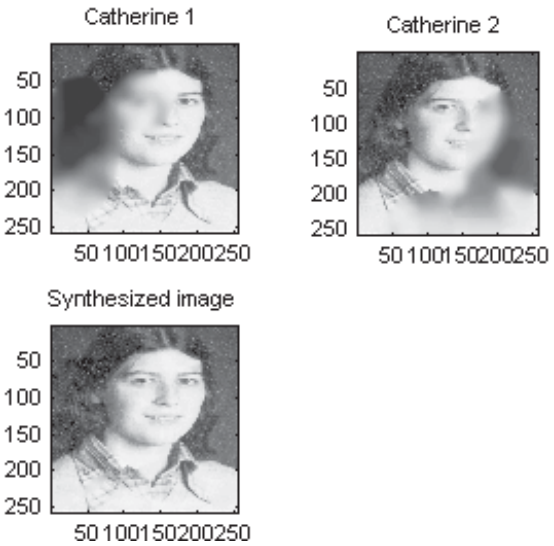


Рис. 3.3.15. Оригинальные
и восстановленное изображения

3.3.3. Трехмерный анализ

Пакет расширения Wavelet Toolbox имеет ряд функций, специально созданных для анализа трехмерных сигналов. Их можно использовать как из командной строки, так и через графический интерфейс главного вейвлет-меню wavemenu (этому будет посвящен параграф в следующей главе). В этом разделе мы рассмотрим основные функции вейвлет-анализа трехмерных сигналов, которыми можно пользоваться из командной строки:

- **dwt3** – одноуровневое дискретное 3D-вейвлет-преобразование;
- **idwt3** – одноуровневое обратное дискретное 3D-вейвлет-преобразование;
- **wavedec3** – многоуровневое 3D-вейвлет-разложение;
- **waverec3** – многоуровневое 3D-вейвлет-восстановление;
- **dwtmode** – мода расширения сигнала (см. раздел 3.3.1).

Трехмерный аналоговый сигнал задается функцией $s(x, y, z)$ трех переменных. Алгоритм разложения сигнала $s(x, y, z)$ аналогичен

тому, что применяется в двумерном случае. Пусть $\varphi(x)$ – масштабирующая (одномерная) вейвлет-функция и $\psi(x)$ – вейвлет. Как известно, они порождают базисные функции $\varphi_{j,n}(x)$ и $\psi_{j,k}(x)$. Трехмерный сигнал $s(x, y, z)$ раскладывается по базисным в $L^2(\mathbf{R}^3)$ функциям:

$$\varphi_{j,n}(x)\varphi_{j,m}(y)\varphi_{j,k}(z),$$
$$\varphi_{j,n}(x)\varphi_{j,m}(y)\psi_{j,k}(z),\quad \varphi_{j,n}(x)\psi_{j,m}(y)\varphi_{j,k}(z),\quad \psi_{j,n}(x)\varphi_{j,m}(y)\varphi_{j,k}(z),$$
$$\varphi_{j,n}(x)\psi_{j,m}(y)\psi_{j,k}(z),\quad \psi_{j,n}(x)\varphi_{j,m}(y)\varphi_{j,k}(z),\quad \psi_{j,n}(x)\psi_{j,m}(y)\varphi_{j,k}(z),$$
$$\psi_{j,n}(x)\psi_{j,m}(y)\psi_{j,k}(z).$$

Получается 8 групп коэффициентов. Они организованы в трехмерный 2×2×2 массив ячеек. Ячейка dec{i,j,k}, $i, j, k = 1, 2$ содержит коэффициенты, которые получены при помощи низкочастотного фильтра, соответствующего функции $\varphi(x)$, если i, j или $k = 1$, и получены при помощи высокочастотного фильтра, соответствующего функции $\psi(x)$, если i, j или $k = 2$. Например, ячейка dec{1,1,1} содержит аппроксимирующие коэффициенты, а ячейка dec{1,1,2} содержит детализирующие коэффициенты, соответствующие вейвлетам $\varphi_{j,n}(x)\varphi_{j,m}(y)\psi_{j,k}(z)$. Группы коэффициентов разложения в MATLAB обозначаются также символами: AAA, AAD, ADA, ADD, DAA, AAD, DDA, DDD, смысл которых понятен. Напомним, что все размеры массивов коэффициентов разложения уменьшаются вдвое по сравнению с размерами исходного массива.

При изучении данного раздела очень полезно рассмотреть демонстрационный пример Wavelet Analysis for 3D Data, представленный в Help\Wavelet Toolbox\Wavelet Toolbox Examples (функция wavelet3ddemo.m), и примеры трехмерного вейвлет-анализа в главном вейвлет-меню wavemenu. Рассмотрим функции трехмерного анализа.

1. Функция dwt3 – одноуровневое дискретное 3D-вейвлет-преобразование.

Синтаксис.

```
WT = dwt3(X,'wname')
WT = dwt3(X,'wname','mode','ExtM')
WT = dwt3(X,W,...)
WT = dwt3(X,WF,...)
```

Входные агрументы.

- X – анализируемый трехмерный массив;
- 'wname' – имя анализирующего вейвлета;
- 'mode','ExtM' – использование метода 'ExtM' расширения массива (методы раширения приведены в табл. 3.3.1).

Выходной параметр WT есть структура с четырьмя полями:

- sizeINI – размеры трехмерного массива X;
- mode – имя метода расширения для вейвлет-разложения;
- filters – структура с четырьмя полями: LoD, HiD, LoR, HiR – это фильтры, используемые в разложении;
- dec – трехмерный массив ячеек 2×2×2, содержащих коэффициенты разложения.

Функция WT = dwt3(X,W,...) использует для разложения три вейвлета – по одному для каждого направления. W = {'wname1', 'wname2','wname3'} или W есть структура с тремя полями 'w1', 'w2', 'w3', содержащая строки с именами вейвлетов.

Функция WT = dwt3(X,WF,...) использует для разложения четыре фильтра, два для разложения и два для восстановления, или 3×4 фильтра (одна четверка для направления). WF есть либо массив ячеек (1×4) или (3×4): {LoD,HiD,LoR,HiR}, либо структура с 4 полями 'LoD','HiD','LoR','HiR'.

Примеры корректой записи команд:

```
[LoD,HiD,LoR,HiR] = wfilters('db2');
wt = dwt3(X,{LoD,HiD,LoR,HiR})

% Разложение X с использованием различных вейвлетов, по одному
% на каждое направление (db1, db2 и снова db1).
WS = struct('w1','db1','w2','db2','w3','db1');
wt = dwt3(X,WS,'mode','per')
wt =
    sizeINI: [4 4 4]
    filters: [1x1 struct]
       mode: 'per'
        dec: {2x2x2 cell}
WF = wt.filters;
% Разложение X с использованием фильтров из структуры WF и
% использование симметричного метода расширения.
wtBIS = dwt3(X,WF,'mode','sym')
```

2. Функция idwt3 – одноуровневое обратное дискретное 3D-вейвлет-преобразование, использующее данные 3D-вейвлет-разложения из структуры WT.

Синтаксис.

X = idwt3(WT)
C = idwt3(WT,TYPE)

Первая функция X = idwt3(WT) производит полное восстановление сигнала, а вторая функция C = idwt3(WT,TYPE) вычисляет компоненту, восстановленную по одной ветви коэффициентов трехмерного вейвлет-разложения. Этот набор коэффициентов определяется параметром TYPE. Допустимые значения параметра TYPE следующие:

- группа из трех символов 'xyz', по одному на направление, где 'x', 'y' и 'z' выбираются во множестве {'a','d','l','h'} или из заглавных символов {'A','D','L','H'}, где 'A' (или 'L') определяют низкочастотный фильтр, и 'D' (или 'H') определяют высочастотный фильтр;
- символ 'd' (или 'h' или 'D' или 'H'), который определяет сумму всех компонент, отличающихся от низкочастотной компоненты.

Примеры корректной записи команды:

```
   % Вычисление аппроксимации, то есть низкочастотной компоненты
A  = idwt3(wt,'aaa');
   % Вычисление суммы всех компонент, отличных от низкочастотной
D  = idwt3(wt,'d');
   % Восстанавление компоненты, низкочастотной в направлениях  X и Z и
   % высокочастоной в направлении Y.
ADA  = idwt3(wt,'ada');
```

3. Функция wavedec3 – многоуровневое 3D-вейвлет-разложение. Производит вейвлет-разложение матрицы трехмерного сигнала X до уровня N, используя вейвлет 'wname' или фильтры {LoD,HiD,LoR,HiR}, а также с использованием моды расширения.

Синтаксис.

WDEC = wavedec3(X,N,'wname')
WDEC = wavedec3(X,N,'wname','mode','ExtM')
WDEC = wavedec3(X,N,{LoD,HiD,LoR,HiR})

Входные агрументы.

- X – анализируемый трехмерный массив;
- N – уровень разложения;

- 'wname' – имя анализирующего вейвлета;
- 'mode','ExtM' – использование метода 'ExtM' расширения массива (методы раширения приведены в табл. 3.3.1).

Выходной параметр WDEC есть структура с шестью полями:

- sizeINI – размеры трехмерного массива X;
- level – глубина разложения N;
- mode – имя метода расширения для вейвлет-разложения;
- filters – структура с четырьмя полями: LoD, HiD, LoR, HiR – это фильтры, используемые в разложении;
- dec – массив ячеек размера M×1, содержащих коэффициенты разложения. M = 7* WDEC. level + 1. dec{1} содержат коэффициенты низкочастотной части разложения уровня N (коэффициенты аппроксимации). Если k = 0, 7, 14, ..., 7*(WDEC.level – 1), то ячейки dec{k + 2}, ..., dec{k + 8} содержат 3D-вейвлет-коэффициенты детализации одного уровня разложения, начиная с самого высокого. Например, если уровень разложения N = WDEC.level = 3, то ячейки dec{2}, ..., dec{8} содержат вейвлет-коэффициенты уровня 3 (k = 0), ячейки dec{9}, ..., dec{15} содержат вейвлет-коэффициенты уровня 2 (k = 7), и ячейки dec{16}, ..., dec{22} содержат вейвлет-коэффициенты уровня 1 (k = 7*(WDEC.level – 1)). На каждом уровне вейвлет-коэффициенты в dec{k + 2}, ..., dec{k + 8} расположены в следующем порядке: 'HLL', 'LHL', 'HHL', 'LLH', 'HLH', 'LHH', 'HHH';
- sizes – последовательные размеры компонентов разложения.

4. Функция waverec3 – многоуровневое 3D-вейвлет-восстановление. Производит вейвлет-восстановление из структуры многоуровневого разложения WDEC.

Синтаксис.

```
X = waverec3(WDEC)
C = waverec3(WDEC,TYPE,n)
X = waverec3(WDEC,'a',0)
X= waverec3(WDEC,'ca',0)
C = waverec3(WDEC,TYPE)
C= waverec3(WDEC,TYPE,N)
```

Первая функция X = waverec3(WDEC) производит полное восстановление сигнала по структуре WDEC.

Вторая функция C = waverec3(WDEC,TYPE,n) вычисляет компоненту, восстановленную по одной ветви коэффициентов с уровня

n ≤ N трехмерного вейвлет-разложения (N – уровень разложения). Этот набор коэффициентов определяется параметром TYPE. Допустимые значения параметра TYPE следующие:

- группа из трех символов 'xyz', по одному на направление, где 'x', 'y' и 'z' выбираются во множестве {'a','d','l','h'} или из заглавных символов {'A','D','L','H'}, где 'A' (или 'L') определяют низкочастотный фильтр и 'D' (или 'H') определяют высочастотный фильтр;
- символ 'd' (или 'h', или 'D', или 'H'), который определяет сумму всех компонент, отличающихся от низкочастотной компоненты;
- символ 'a' (или 'l', или 'A', или 'L') определяет низкочастотную компоненту (аппроксимацию уровня n).

X = waverec3(WDEC,'a',0) или X= waverec3(WDEC,'ca',0) эквивалентно X = waverec3(WDEC).

C = waverec3(WDEC,TYPE) эквивалентно C = waverec3(WDEC, TYPE,N), где N – уровень разложения.

3.3.4. Анализ мультисигналов

Мультисигналом будем называть набор из нескольких сигналов одинаковой длины. Например, это могут быть строки матрицы, представляющей изображение, числовые данные нескольких каналов кардиосигнала.

Пакет расширения Wavelet Toolbox имеет ряд функций (см. раздел 3.10.5), специально созданных для анализа мультисигнала. Их можно использовать как из командной строки, так и через графический интерфейс главного вейвлет-меню (wavemenu). В следующей главе мы рассмотрим анализ мультисигнала при помощи графического интерфейса (wavemenu), а здесь обсудим основные функции вейвлет-анализа многомерных сигналов, которыми можно пользоваться из командной строки. Это следующие функции:

- **mdwtdec** – вейвлет-разложение мультисигнала;
- **mdwtrec** – вейвлет-восстановление мультисигнала и извлечение коэффициентов аппроксимации и деталей;
- **chgwdeccfs** – изменение 1D-коэффициентов разложения мультисигнала;
- **wdcenergy** – распределение энергии 1D-разложения мультисигнала;
- **mswcmp** – 1D-вейвлет-сжатие мультисигнала;

- **mswcmptp** – вычисление порогов и энергии 1D-сжатия мультисигнала;
- **mswden** – 1D-вейвлет-удаление шума мультисигнала;
- **mswthresh** – выполнение 1D-пороговой обработки мультисигнала;
- **mswcmpscr** – оценка 1D-вейвлет-сжатия мультисигнала;
- **mdwtcluster** – кластеризация 1D-мультисигнала.

Для вейвлет-анализа мультисигнала можно использовать также главное вейвлет-меню wavemenu и функции:

- **wkeep** – извлечение части вектора или матрицы;
- **wmspca** – многомасштабный анализ главных компонент;
- **dwtmode** – мода расширения дискретного вейвлет-преобразования.

1. Функция mdwtdec – вейвлет-разложение мультисигнала.

Синтаксис:

```
DEC = mdwtdec(DIRDEC,X,LEV,WNAME)
DEC = mdwtdec(DIRDEC,X,LEV,LoD,HiD,LoR,HiR)
DEC = mdwtdec(...,'mode',EXTMODE)
```

Функция DEC = mdwtdec(DIRDEC,X,LEV,WNAME) возвращает вейвлет-разложение до уровня LEV каждой строки (если DIRDEC = = 'r') или каждого столбца (если DIRDEC= 'c') матрицы X, используя вейвлет WNAME.

Выходной параметр DEC есть структура с полями:

- 'dirDec' – индикатор направления: 'r' (row) или 'c' (column);
- 'level' – уровень разложения;
- 'wname' – имя вейвлета;
- 'dwtFilters' – структура с 4 полями: LoD, HiD, LoR и HiR;
- 'dwtEXTM' – мода расширения;
- 'dwtShift' – параметр отклонения DWT, (0 или 1);
- 'dataSize' – размеры X;
- 'ca' – коэффициенты аппроксимации уровня LEV;
- 'cd' – массив ячеек детализирующих коэффициентов от уровня 1 до уровня LEV.

Коэффициенты cA и cD{k} (для k= 1 до LEV) – это матрицы, которые хранятся и сохраняются в строках, если DIRDEC = 'r', или в столбцах, если DIRDEC = 'c'.

Пример 17. Загрузим тестовый сигнал MATLAB thinker.mat. Он содержит единственную переменную X. Это матрица double размера 192 на 96, представляющая мультисигнал – серию из 192 сигналов-

строк. Выполним вейвлет-разложение сигналов-строк до уровня 2, используя вейвлет db2.

```
load thinker
dec = mdwtdec('r',X,2,'db2');
Функция mdwtdec создает структуру разложения dec следующего вида:
dec =
        dirDec: 'r'
         level: 2
         wname: 'db2'
    dwtFilters: [1x1 struct]
       dwtEXTM: 'sym'
      dwtShift: 0
      dataSize: [192 96]
            ca: [192x26 double]
            cd: {[192x49 double]  [192x26 double]}
```

2. Функция mdwtrec – вейвлет-восстановление мультисигнала и извлечение коэффициентов аппроксимации и деталей.

Синтаксис:

X = mdwtrec(DEC)
X = mdwtrec(DEC,IDXSIG)
Y = mdwtrec(DEC,TYPE,LEV)
A = mdwtrec(DEC,'a') эквивалентно A = mdwtrec(DEC,'a',LEVDEC)
D = mdwtrec(DEC,'d')
CA = mdwtrec(DEC,'ca') эквивалентно
CA= mdwtrec(DEC,'ca',LEVDEC)
CD = mdwtrec(DEC,'cd',MODE)
CFS = mdwtrec(DEC,'cfs',MODE)
Y = mdwtrec(...,IDXSIG)

Функция X = mdwtrec(DEC) возвращает матрицу оригинального сигнала, исходя из структуры вейвлет-разложения DEC.

Параметр IDXSIG указывает на восстановление элементов мультисигнала, индексы которых заданы вектором IDXSIG.

Функция Y = mdwtrec(DEC,TYPE,LEV) извлекает, или реконструирует, детали или коэффициенты аппроксимации указанного уровня LEV в зависимости от значения параметра TYPE. Если параметр TYPE есть:

- 'cd' или 'ca', то извлекаются коэффициенты уровня LEV;
- 'd' или 'a', то реконструируются коэффициенты уровня LEV;
- 'a' или 'ca', то LEV должен быть таким, что $0 \le LEV \le LEVDEC$;
- 'd' или 'cd', то LEV должен быть таким, что $1 \le LEV \le LEVDEC$.

Функция D = mdwtrec(DEC,'d') возвращает матрицу, содержащую сумму всех деталей, так что X= A + D.

Функция CD = mdwtrec(DEC,'cd',MODE) возвращает матрицу, содержащую все коэффициенты деталей.

Функция CFS = mdwtrec(DEC,'cfs',MODE) возвращает матрицу, содержащую все коэффициенты.

Функция Y = mdwtrec(...,IDXSIG) извлекает или реконструирует коэффициенты аппроксимации или деталей элементов мультисигнала, индексы которых заданы вектором IDXSIG.

Пример 18. Выполним вейвлет-реконструкцию сигналов из структуры dec примера 1 и создадим графики некоторых из новых сигналов из нового мультисигнала Xbis:

```
Xbis = mdwtrec(dec);
figure;
plot(Xbis(1:5,:)','r');   hold on
plot(Xbis(21:25,:)','b'); plot(Xbis(31:35,:)','g')
grid; set(gca,'Xlim',[1,96])
```

Еще один пример. Установим нулевыми вейвлет-коэффициенты на уровне 1 и 2 для сигналов 31–35, выполним вейвлет-реконструкцию сигнала 31 и сравним на графике некоторые из старых и новых сигналов (рис. 3.3.16).

```
decTER = chgwdeccfs(dec,'cd',0,1:2,31:35);
Y = mdwtrec(decTER,'a',0,31);
figure;
plot(X([1 31],:)','r','linewidth',1);   hold on
plot([Xbis(1,:) ; Y]','b','linewidth',2);
grid; set(gca,'Xlim',[1,96])
```

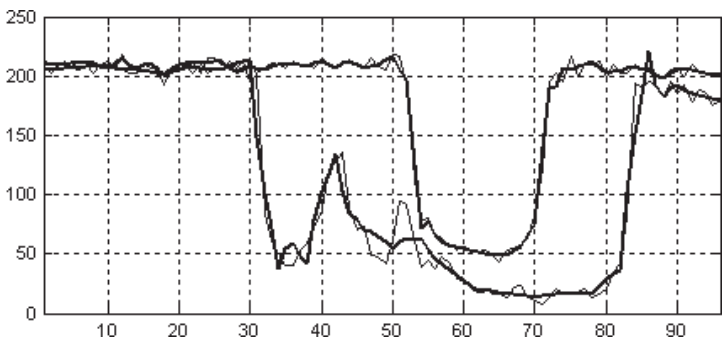


Рис. 3.3.16. Графики двух сигналов, исходных и после реконструкции (п/ж)

3. Функция chgwdeccfs – изменение 1D-коэффициентов разложения мультисигнала.

Синтаксис:

DEC = chgwdeccfs(DEC,'ca',COEFS)
DEC = chgwdeccfs(DEC,'cd',COEFS,LEV)
DEC = chgwdeccfs(DEC,'all',CA,CD)
DEC= chgwdeccfs(DEC,'all',V)
DEC = chgwdeccfs(...,IDXSIG

Функция DEC = chgwdeccfs(DEC,'ca',COEFS) заменяет коэффициенты аппроксимации на уровне DEC.level на те, которые содержатся в матрице COEFS. Если COEFS есть одно значение V, то все коэффициенты заменяются на V.

Функция DEC = chgwdeccfs(DEC,'cd',COEFS,LEV) заменяет коэффициенты деталей на уровне LEV на те, которые содержатся в матрице COEFS. Если COEFS есть одно значение V, тогда LEV может быть вектором уровней, и тогда все коэффициенты этих уровней заменяются на V. LEV должен быть таким: $1 \le LEV \le DEC.level$.

Функция DEC = chgwdeccfs(DEC,'all',CA,CD) заменяет все коэффициенты аппроксимации и деталей. CA должен быть матрицей, а CD – массивом ячеек длины DEC.level. Если COEFS (или CA, или CD) есть одно значение, то на него заменяются все коэффициенты. Для числа V функция DEC= chgwdeccfs(DEC,'all',V) заменяет все коэффициенты на V.

Функция DEC = chgwdeccfs(...,IDXSIG) заменяет только коэффициенты элементов мультисигнала, которые указаны в векторе индексов IDXSIG.

Пример 19. В качестве примера применения в полученной в предыдущем примере 1 структуре dec для каждого сигнала из мультисигнала изменим вейвлет-коэффициенты, устанавливая нулевыми все коэффициенты деталей уровня 1:

```
decBIS = chgwdeccfs(dec,'cd',0,1);
```

В результате создается новая структура разложения decBIS.

4. Функция wdecenergy. Вычисление энергии (L^2-нормы) сигналов и энергии элементов 1D-декомпозиции мультисигнала.

Синтаксис:

E,PEC,PECFS] = wdecenergy(DEC)
[E,PEC,PECFS,IDXSORT,LONGS] = wdecenergy(DEC,'sort')
[E,PEC,PECFS] = wdecenergy(DEC,OPTSORT,IDXSIG)

[E,PEC,PECFS,IDXSORT,LONGS] =
wdecenergy(DEC,OPTSORT,IDXSIG)

Функция [E,PEC,PECFS] = wdecenergy(DEC) вычисляет вектор E, который содержит энергию (L2-норму) каждого разложенного сигнала из мультисигнала, матрицу PEC, которая содержит процент энергии для каждой вейвлет-компоненты (аппроксимация и детали) каждого сигнала, и матрицу PECFS, которая содержит процент энергии каждого коэффициента. При этом:

- E(i) – энергия (L2-норма) *i*-го сигнала;
- PEC(i,1) – процент энергии для аппроксимации уровня MAXLEV = DEC.level для *i*-го сигнала;
- PEC(i,j), j=2,...,MAXLEV+1 – процент энергии для деталей уровня (MAXLEV+1-j) для *i*-го сигнала;
- PECFS(i,j) – процент энергии j-го коэффициента *i*-го сигнала.

Функция [E,PEC,PECFS,IDXSORT,LONGS] = wdecenergy (DEC,'sort') возвращает PECFS отсортированнные (по строкам) в порядке возрастания и вектор индексов IDXSORT. Замена 'sort' на 'ascend' дает тот же самый результат. Замена 'sort' на 'descend' возвращает PECFS, отсортированные в порядке убывания. LONGS есть вектор, содержащий длины каждого семейства коэффициентов.

Функция[E,PEC,PECFS]=wdecenergy(DEC,OPTSORT,IDXSIG) возвращает значения, относящиеся к сигналам, указанным в векторе индексов IDXSIG.

Пример 20. Вычислим энергию сигналов и проценты от энергии для вейвлет-коэффициентов из предыдущих примеров.

```
[E,PEC,PECFS] = wdecenergy(dec);
Энергия сигналов 1 и 31:
Ener_1_31 = E([1 31])
Ener_1_31 =
  1.0e+006 *
    3.7534
    2.2411
```

Вычислим процент от энергии для вейвлет-коэффициентов сигналов 1 и 31.

```
PEC_1_31 = PEC([1 31],:)
PEC_1_31 =
   99.7760    0.1718    0.0522
   99.3850    0.2926    0.3225
```

Первый столбец показывает процент от энергии всего сигнала для коэффициентов аппроксимации на уровне 2. Столбцы 2 и 3 показывают проценты от энергии для коэффициентов деталей на уровне 2 и 1 соответственно.

5. Функция mswcmp – вейвлет-сжатие 1D-мультисигнала. Вычисляет пороги и, опционально, выполняет вейвлет-сжатие 1D-мультисигнала.

Синтаксис:

```
[XC,DECCMP,THRESH] = mswcmp('cmp',DEC,METH)
[XC,DECCMP,THRESH] = mswcmp('cmp',DEC,METH,PARAM)
[XC,THRESH] = mswcmp('cmpsig',...)
[DECCMP,THRESH] = mswcmp('cmpdec',...)
```

Функция возвращает сжатую (что отмечается параметром 'cmp') версию XC оригинального мультисигнала X, который имеет структуру разложения DEC. Вывод XC получается пороговой обработкой вейвлет-коэффициентов, DECCMP является структурой вейвлет-разложения для XC (см. mdwtdec), и THRESH является матрицей пороговых значений. Параметр METH является именем метода сжатия, PARAM является соответствующим параметром, если он требуется. Например, если методами являются 'L2_perf' – отношение энергии или 'N0_perf' – процент нулевых коэффициентов, то PARAM есть вещественное число от 0 до 100. Подробнее об этой функции см. в разделе 3.6 – сжатие сигналов.

Например, сожмем сигналы из мультисигнала X, устанавливая процент нулей для вейвлет-коэффициентов около 95%.

```
[XC,decCMP,THRESH] = mswcmp('cmp',dec,'N0_perf',95);
[Ecmp,PECcmp,PECFScmp] = wdecenergy(decCMP);
```

Сигналы XC – сжатые версии оригинальных сигналов в направлении строк. Сожмем сигналы XC в направлении столбцов:

```
XX = mswcmp('cmpsig','c',XC,'db2',2,'N0_perf',60);
```

Отметим, что вместо 'cmp' вводится опция 'cmpsig', тогда используется для ввода сигнал и выводом будет сжатый сигнал, а не структура разложения. Можно также использовать 'cmpdec' или 'thr'. В случае 'thr' возвращаются вычисленные пороги, но сжатие не выполняется.

6. Функция mswcmptp. Вычисление порогов 1D-сжатия мультисигнала и энергии сжатых сигналов мультисигнала.

Синтаксис:

[THR_VAL,L2_Perf,N0_Perf] = mswcmptp(DEC,METH)
[THR_VAL,L2_Perf,N0_Perf]= mswcmptp(DEC,METH,PARAM)

Функции вычисляют векторы THR_VAL, L2_Perf и N0_Perf, полученные после сжатия, используя метод METH и, если требуется, параметр PARAM (см. mswcmp для информации относительно METH и PARAM, раздел 3.6).

Выходные параметры:

- THR_VAL(i) – порог, применяемый к вейвлет-коэффициентам. Для метода, зависящего от уровня, THR_VAL(i,j) – это порог, применяемый к вейвлет-коэффициентам деталей уровня j.
- L2_Perf(i) есть процент энергии (L2-нормы), сохранившейся после сжатия.
- N0_Perf(i) есть процент нулей, полученных после сжатия.

Пример 21. Вычислим пороги, проценты сохраняемой энергии и проценты нулевых коэффициентов в соответствии с методом с L2_perf, сохраняющим, по крайней мере, 95% энергии.

```
[THR_VAL,L2_Perf,N0_Perf] = mswcmptp(dec,'L2_perf',95);
idxSIG = [1,31];
Thr   = THR_VAL(idxSIG)
Thr =
  256.1914
  158.6085
L2per = L2_Perf(idxSIG)
L2per =
   96.5488
   94.7197
N0per = N0_Perf(idxSIG)
N0per =
   79.2079
   86.1386
```

7. Функция mswden – вейвлет-удаление шума 1D-мультисигнала. Вычисляет пороги и, опционально, выполняет удаление шума 1D-мультисигнала.

Синтаксис:

[XD,DECDEN,THRESH] = mswden('den', DEC,METH,PARAM)
[XD,THRESH] = mswden('densig',...)
[DECDEN,THRESH]= mswden('dendec',...)
THRESH = mswden('thr',...)

Функция [XD,DECDEN,THRESH] = mswden('den',...) находит очищенную от шума версию XD оригинального мультисигнала X по структуре вейвлет-разложения DEC. Сигнал на выходе XD получается пороговой обработкой вейвлет-коэффициентов, DECDEN – это вейвлет-разложение, ассоциированное с XD (см. mdwtdec), и THRESH – матрица значений порогов. Входной параметр METH есть метод удаления шума, и PARAM – параметры, если они необходимы.

Пример 22. Удалим шумы сигналов, используя универсальный порог:

```
[XD,decDEN,THRESH] = mswden('den',dec,'sqtwolog','sln');
figure;
plot(X([1 31],:)','r','linewidth',2);   hold on
plot(XD([1 31],:)','b','linewidth',2);
grid; set(gca,'Xlim',[1,96])
```

В результате получается XD – очищенные от шума версии оригинальных сигналов по строкам. Очистим от шума сигналы XD по столбцам

```
XX = mswden('densig','c',XD,'db2',2,'sqtwolog','sln');
```

8. Функция mswthresh – выполнение 1D-пороговой обработки мультисигнала. Вычисляет пороги и, опционально, выполняет удаление шума 1D-мультисигнала.

Синтаксис:

Y = mswthresh(X,SORH,T)
Y = mswthresh(X,SORH,T,'c')
Y = mswthresh(X,'s',T)
Y = mswthresh(X,'h',T)

Функция Y = mswthresh(X,SORH,T) возвращает мягкую (если SORH='s') или жесткую (если SORH='h') T-пороговую обработку входной матрицы X. T может быть одним значением, матрицей того же размера, что и X, или вектором. В последнем случае трешолдинг выполняется по строкам, и LT= length(T) должна быть такой, что size(X,1) ≤ LT. Параметр 'c' указывает на трешолдинг по столбцам, и тогда size(X,2) ≤ LT.

9. Функция mswcmpscr – оценка 1D-вейвлет-сжатия мультисигнала.

Синтаксис:

[THR,L2SCR,NOSCR,IDXSORT] = mswcmpscr(DEC)

Функция вычисляет четыре матрицы: пороги THR, оценка сжатия L2SCR и NOSCR и индексы IDXSORT. Разложение DEC соответствует матрице вейвлет-коэффициентов CFS:

CFS = [cd{DEC.level}, ..., cd{1}] или
CFS = [ca, cd{DEC.level}, ..., cd{1}]

Для каждой строки IDXSORT содержит порядок коэффициентов.

10. Функция mdwtcluster – кластеризация 1D-мультисигнала. Выделяет группу сигналов в мультисигнале по заданным свойствам.

Синтаксис:

S = mdwtcluster(X)
S = mdwtcluster(X,'PropName1',PropVal1,'PropName2',PropVal2,...)

Функция S = mdwtcluster(X) строит кластеры из иерархического дерева группы. Входная матрица X раскладывается по строкам с использованием функции DWT с вейвлетом Хаара и максимально допустимым уровнем. Подробнее об этой функции см. в Help MATLAB.

3.4. Вейвлет-пакеты

При обычном ортогональном вейвлет-разложении аппроксимирующие коэффициенты раскладываются на аппроксимирующие и детализирующие коэффициенты более низкого уровня, а затем процедура применяется к вновь полученным аппроксимирующим коэффициентам. Детализирующие коэффициенты далее не анализируются.

Идея пакетного вейвлет-анализа состоит в том, чтобы раскладывать также и детализирующие коэффициенты, используя тот же самый метод разложения (см. dwt и wavedec). Это дает более богатый анализ: двоичное дерево T коэффициентов разложения, начинающееся с исходного сигнала $s(0,0)$:

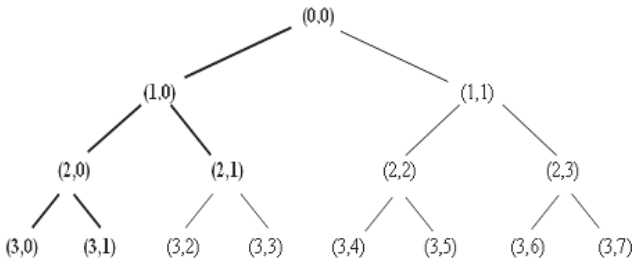


Рис. 3.4.1. Схема пакетного разложения

При разложении данных некоторого узла ниже слева находятся аппроксимирующие коэффициенты, а справа – детализирующие коэффициенты.

Пакетные вейвлет-функции $w_n(x)$ могут быть определены рекурсивно. Пусть $\varphi(x)$ и $\psi(x)$ – ортогональные масштабирующая функция и вейвлет с фильтрами h_n и g_n соответственно. Полагаем $w_0(x) = \varphi(x)$ и $w_1(x) = \psi(x)$ и определим для любого натурального n функции

$$w_{2n}(x) = \sqrt{2}\sum_{k\in\mathbf{Z}} h_k w_n(2x-k), \qquad w_{2n+1}(x) = \sqrt{2}\sum_{k\in\mathbf{Z}} g_k w_n(2x-k).$$

Обозначим символом Ω_n пространство, порожденное целочисленными сдвигами функций $w_n(x)$. Пространства Ω_0 и Ω_1 соответствуют введенным ранее пространствам V_0 и V_1. Известно, что функции $w_n(x-k)$ для любых целых k и неотрицательных целых n образуют базис пространства $L^2(\mathbf{R})$. Тогда

$$L^2(\mathbf{R}) = \Omega_0 + \Omega_1 + \Omega_2 + \ldots$$

есть разложение пространства $L^2(\mathbf{R})$ в ортогональную прямую сумму замкнутых подпространств. Разложение функции $f(x)$ по базису $w_n(x-k)$ называется вейвлет-пакетным разложением.

Основные функции пакетного вейвлет-анализа:

- **wpfun** – вейвлет-пакетные функции;
- **wpdec** – пакетное вейвлет-разложение одномерного сигнала;
- **wprec** – пакетное 1D-вейвлет-восстановление;
- **plot** – изображение активного дерева пакетного вейвлет-разложения;
- **cfs2wpt** – построение вейвлет-пакетного дерева из коэффициентов;
- **wp2wtree** – извлечение вейвлет-дерева из вейвлет-пакетного дерева;
- **wpcoef** – коэффициенты пакетного вейвлет-преобразования;
- **wprcoef** – восстановление по коэффициентам пакетного преобразования;
- **wpsplt** – ращепление узла вейвлет-пакетного дерева;
- **wpcutree** – обрезание пакетного дерева;
- **wpjoin** – отрезание отдельных ветвей дерева;
- **allnodes** – нумерация узлов дерева;
- **tnodes** – определение терминальных узлов;
- **depo2ind** – преобразование бинарной нумерации узлов в одинарную;

- **ind2depo** – преобразование нумерации узлов по порядку в бинарную;
- **otnodes** – порядок терминальных узлов вейвлет-пакетного дерева;
- **wpviewcf** – цветное изображение коэффициентов терминальных узлов;
- **write (read)** – запись (считывание) значений в терминальные узлы дерева;
- **wentropy** – вычисление энтропии сигнала;
- **entrupd** – замена функции энтропии в пакетном анализе;
- **besttree** – наилучшее дерево пакетного вейвлет-разложения;
- **bestlevt** – дерево пакетного вейвлет-разложения наилучшего уровня;
- **wpdec2** – пакетное 2D-вейвлет-разложение;
- **wprec2** – пакетное 2D-вейвлет-восстановление;
- **wpspectrum** – вейвлет-пакетный спектр.

Рассмотрим их подробнее.

1. Функция wpfun – вейвлет-пакетные функции. Находит пакетные функции $w_n(x)$ для заданного n.

Синтаксис:

```
[WPWS,X] = wpfun('wname',NUM,PREC)
[WPWS,X] = wpfun('wname',NUM) эквивалентно
[WPWS,X] = wpfun('wname',NUM,7)
```

Функция WPWS,X] = wpfun('wname',NUM,PREC) вычисляет вейвлет-пакетные функции на основе вейвлета 'wname' на сетке X, шаг которой определяется целым положительным числом PREC, $\Delta t < 2^{-PREC}$. NUM определяет количество вейвлет-функций $w_n(x)$. Выходная матрица WPWS содержит функции $w_n(x)$ с индексами от 0 до NUM, хранящиеся по строкам как [W0; W1; ...; WNUM].

2. Функция wpdec – пакетное вейвлет-разложение одномерного сигнала. Находит полное дерево T пакетного разложения до уровня N.

Синтаксис:

```
T = wpdec(X,N,'wname',E,P),
T = wpdec(X,N,'wname') эквивалентно
T = wpdec(X,N,'wname','shannon').
```

Здесь:

- X – исходный сигнал;
- N – уровень разложения;

- E – тип функции «энтропии», позволяющей оценить информативность набора коэффициентов в узлах (см. wentropy). Параметр E может принимать строковые значения: 'shannon', 'threshold', 'norm', 'log energy', 'sure', 'user';
- P – дополнительный параметр.

В случае 'shannon' или 'log energy' параметр P не используется, если 'threshold' или 'sure' – тогда P есть пороговое значение, $P \geq 0$, в случае 'norm' – P есть степень нормы ($P \geq 1$), если 'user' – тогда P определен как имя функции, заданной пользователем.

T = wpdec(X,N,'wname') эквивалентно
T = wpdec(X,N,'wname','shannon').

3. Функция wprec – пакетное вейвлет-восстановление. Это обратная функция к wpdec. Она восстанавливает вектор X по вейвлет-пакетному дереву коэффициентов T и записывается в виде

X = wprec(T)

4. Функция plot – изображение активного дерева пакетного вейвлет-разложения.

Синтаксис:

plot(T)

Команда plot(T) вызывает графическое окно, состоящее из двух частей. Слева – изображение дерева, а справа – график коэффициентов в узлах дерева, который появляется, если соответствующий узел слева активировать мышкой.

Пример 1. Разложим сигнал x на глубину 3 с помощью пакета вейвлета Добеши db1, используя функцию энтропии 'shannon', и построим дерево этого разложения. Справа – график коэффициентов в узле (3,1) дерева.

```
load noisdopp; x = noisdopp; %загружаем тестовый сигнал noisdopp
wpt = wpdec(x,3,'db1','shannon');
plot(wpt)
```

5. Функция cfs2wpt – построение вейвлет-пакетного дерева из коэффициентов.

Синтаксис:

T = cfs2wpt(' wname ',size,tnodes,order,Cfs);

Входные параметры:

- size – размеры сигнала или изображения;
- tnodes – номера выбранных узлов;

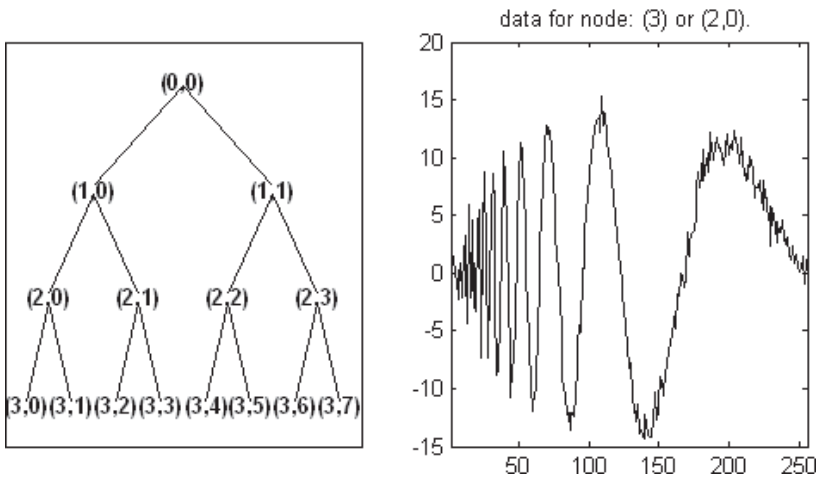


Рис. 3.4.2. Дерево разложения и данные в узле (2,0)

- order = 2 для 1D и =4 для 2D;
- Cfs – коэффициенты, используемые для реконструкции сигнала или изображения.

6. Функция wp2wtree – извлечение вейвлет-дерева из вейвлет-пакетного дерева.

Синтаксис:

T = wp2wtree(T)

7. Функция wpcoef – коэффициенты пакетного вейвлет-преобразования. Эта функция вычисляет коэффициенты пакетного вейвлет-преобразования в узле N дерева. Если указанного узла не существует, то X = [ ].

Синтаксис:

X = wpcoef(T,N),
X = wpcoef(T) эквивалентно X = wpcoef(T,0).

N может быть целым числом либо парой N = [i j] в зависимости от способа нумерации.

8. Функция wprcoef – восстановление по коэффициентам пакетного преобразования. Эта функция восстанавливает сигнал по коэффициентам пакетного вейвлет-преобразования в узле N дерева. При этом считается, что коэффициенты в остальных узлах этого уровня равны нулю.

Синтаксис:

X = wprcoef(T,N),
X = wprcoef(T) эквивалентно X = wprcoef(T,0).

Пример 2. Делаем пакетное разложение сигнала x, выбираем коэффициенты в узле N=(2,1), по этим коэффициентам (и только по ним) восстанавливаем сигнал, обозначим его x21, делаем пакетное разложение сигнала x21, снова выбираем коэффициенты в узле N=(2,1) дерева разложения сигнала x21 и сравниваем их на графике с первоначальными.

```
load noisdopp; x = noisdopp; %загружаем тестовый сигнал noisdopp
t = wpdec(x,3,'db1','shannon');
cfs = wpcoef(t,[2 1]);
x21 = wprcoef(t,[2 1]);
tr21 = wpdec(x21,3,'db1','shannon');
cfs21 = wpcoef(tr21,[2 1]);
figure(1); subplot(411);  plot(x); title('Original signal');
subplot(412); plot(cfs); title('Packet (2,1) coefficients');
subplot(413); plot(x21); title('Reconstructed packet (2,1)');
subplot(414); plot(cfs21); title('Packet x21 (2,1) coefficients');
plot(tr21);
```

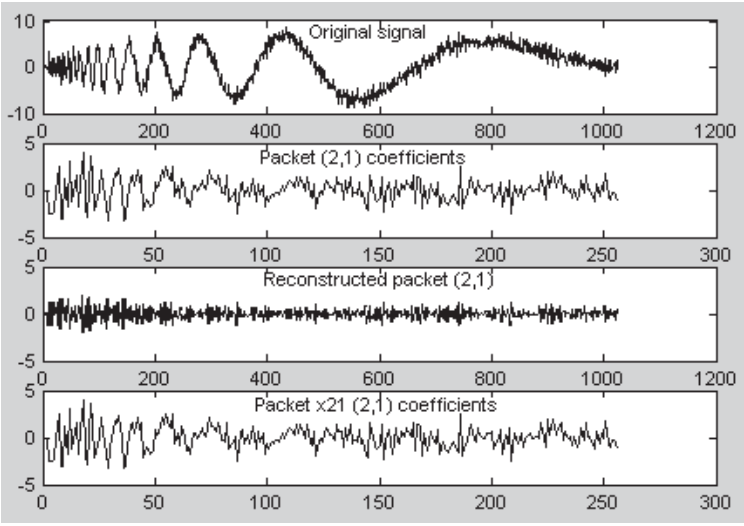


Рис. 3.4.3. Восстановление части сигнала по данным в узле (2,1)

9. Функция wpsplt – ращепление узла вейвлет-пакетного дерева. Это функция одно- и двумерного вейвлет-анализа. Она добавляет к уже найденному дереву T вейвлет-разложение коэффициентов узла N.

Синтаксис:

T1 = wpsplt(T,N),
[T1,cA,cD] = wpsplt(T,N),
[T1,cA,cH,cV,cD] = wpsplt(T,N).

Результатом операции является модифицированное дерево T1 с дополнительно разложенным узлом N. Коэффициенты cA – аппроксимирующие и cD – детализирующие для узла N.

В двумерном случае функция применяется в виде

[T1,cA,cH,cV,cD] = wpsplt(T,N).

Пример 3.

```
load noisdopp; x = noisdopp; %загружаем тестовый сигнал noisdopp
wpt = wpdec(x,2,'db1');
   %plot(wpt)
wpt1 = wpsplt(wpt,[2 0]); %Разложим пакет в узле N=(2,0), можно
wpsplt(wpt,3)
plot(wpt1)
```

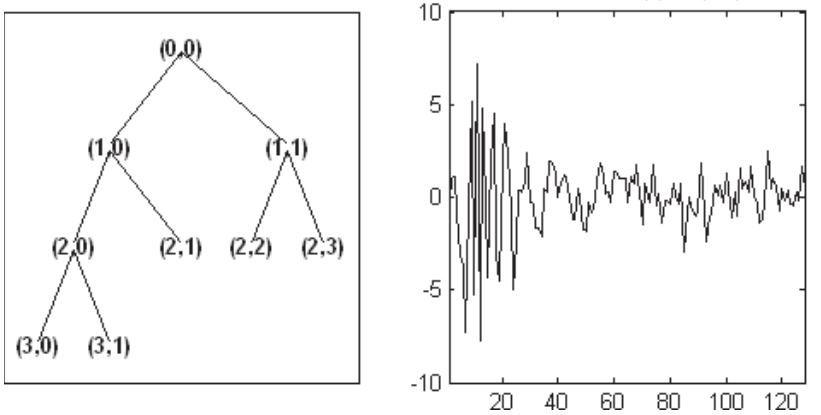


Рис. 3.4.4. Разложение данных в узле (2.0)
и коэффициенты в узле (3,1)

10. Функция wpcutree – обрезание пакетного дерева.
Синтаксис:

T1 = wpcutree(T,L),
[T1,RN] = wpcutree(T,L).

Новое дерево T1 получается обрезанием всех ветвей дерева T на уровне L. Команда [T1,RN] = wpcutree(T,L) дополнительно определяет вектор RN, содержащий индексы узлов перестроенного дерева.

11. Функция wpjoin – отрезание отдельных ветвей дерева. Функция отрезает ветвь дерева T в узле N.

Синтаксис:

T1 = wpjoin(T,N), [T1, RN] = wpjoin(T,N),
T1 = wpjoin(T), [T1, RN] = wpjoin(T).

Функция [T1, RN] = wpjoin(T,N) дополнительно устанавливает индексы RN узлов нового дерева. T1 = wpjoin(T) эквивалентно T1 = wpjoin(T,0), то есть обрезает все, кроме вершины.

12. Функция allnodes – нумерация узлов дерева. Нумерация возможна двумя способами: номера по порядку или по глубине и положению на данном уровне. Узлы нумеруются слева направо и сверху вниз. Начальный индекс (исходного сигнала) есть 0. Дерево не обязательно полное.

Синтаксис:

N= allnodes(T),
N= allnodes(T,'deppos'),

Первая функция N = allnodes(T) дает номера по порядку всех узлов дерева T в виде столбца N. Вторая функция N = allnodes(T,'deppos') нумерует по глубине и положению на данном уровне всех узлов в виде матрицы N из двух столбцов. Индексы узлов – это строки (i,j), первый индекс i – номер уровня, а второй j – положение узла на i-м уровне, причем крайний левый узел на данном уровне имеет нулевой номер j = 0 (см. приведенный выше рисунок).

13. Функция tnodes – определение терминальных узлов дерева.
Синтаксис:

N = tnodes(T),
N = tnodes(T,'deppos').
[N,K] = tnodes(T),
[N,K] = tnodes(T,'deppos').

В случае N = tnodes(T) N – это столбец номеров терминальных, то есть конечных, узлов дерева T. Узлы перечисляются слева направо и сверху вниз. Индекс корня – 0.

В случае N = tnodes(T,'deppos') N – это матрица из двух столбцов, у которой элемент первого столбца i – номер уровня конечного узла, а элемент j второго столбца – это положение конечного узла на i-м уровне, то есть N(n,1) – глубина n-го терминального узла. N(n,2) – позиция n-го терминального узла.

В случае [N,K] = tnodes(T) или [N,K] = tnodes(T,'deppos') дается еще одна нумерация K терминальных узлов. Она начинается с единицы и учитывает положение узлов в дереве T.

14. Функция depo2ind – преобразование бинарной нумерации узлов в одинарную. Функция N = depo2ind(ORD,[D P]) вычисляет индексы N узлов, у которых глубина и положение на уровне заданы в матрице [D P], где P – номер уровня, а P –положение узла на уровне P.

15. Функция ind2depo – преобразование нумерации узлов по порядку в бинарную (уровень, положение).

Синтаксис:

```
[depth,pos] = ind2depo(ord,N);
```

Здесь N – вектор-столбец индексов узлов, которые необходимо преобразовать.

16. Функция otnodes – порядок терминальных узлов вейвлет-пакетного дерева.

Синтаксис:

```
[Tn_Pal,Tn_Seq] = otnodes(WPT)
[Tn_Pal,Tn_Seq,I,J] = otnodes(WPT)
[DP_Pal,DP_Seq] = otnodes(WPT,'dp')
```

Функция [Tn_Pal,Tn_Seq] = otnodes(WPT) возвращает индексы терминальных узлов вейвлет-пакетного дерева WPT. Tn_Pal – в естественном порядке и Tn_Seq – в частотном порядке (по возрастанию частоты). Индексы I, J отражают перестановки чисел (1, 2, ..., N), когда Tn_Seq получается из Tn_Pal, и наоборот: Tn_Seq = Tn_Pal(I) и Tn_Pal = Tn_Seq(J).

Функция [DP_Pal,DP_Seq] = otnodes(WPT,'dp') выполняет то же самое, но в бинарном формате нумерации.

17. Функция wpviewcf – цветное изображение коэффициентов терминальных узлов. Производится в виде раскрашенной разными

способами матрицы, строками которой являются коэффициенты терминальных узлов, занумерованных по порядку целыми числами. Строки коэффициентов в матрице могут быть расставлены в соответствии с методом раскраски.

Синтаксис:

wpviewcf(T,CMODE)
wpviewcf(T,CMODE,NBCOL)

18. Функция write (read) – запись (считывание) значений в терминальные узлы дерева. Кроме записи новых значений в конечные узлы дерева, функция write вычисляет новое дерево T1 с восстановленными коэффициентами во всех узлах с учетом внесенных изменений.

Синтаксис:

T1 = write(T,'cfs',NODE,COEFS),
T1 = write(T,'cfs',N1,CFS1,'cfs',N2,CFS2, ...)

В первом случае записываются коэффициенты COEFS в терминальный (конечный) узел NODE дерева. А во втором – записываются коэффициенты CFS1, CFS2, ... в терминальные (конечные) узлы N1, N2, ... дерева.

Новое дерево T1 будет иметь восстановленные коэффициенты во всех узлах.

Предостережение. Записываемые коэффициенты должны иметь соответствующий размер. Для правильного определения размера можно использовать функцию read с условием 'sizes':

S = read(T,'sizes',NODE) или S = read(T,'sizes',[N1;N2; ...]),

которая определяет размеры массивов в узлах. Функция read позволяет также считывать значения коэффициентов в терминальных узлах дерева. Для этого применяется параметр 'cfs':

cfs = read(T,'cfs',NODE), или эквивалентно cfs =
read(T,'data',NODE).

См. справку по функции read для других возможностей read.

Пример 4. Отрезаем ветви в узлах с номерами 4 и 5, а затем в эти узлы записываем новые данные.

```
load noisdopp; x = noisdopp;
t = wpdec(x,3,'db3');
t = wpjoin(t,[4;5]);
plot(t);
```

```
sNod = read(t,'sizes',[4,5,7]);
cfs4 = zeros(sNod(1,:));
cfs5 = zeros(sNod(2,:));
cfs7 = zeros(sNod(3,:));
t1 = write(t,'cfs',4,cfs4,'cfs',5,cfs5,'cfs',7,cfs7); % Запись нулевых
значений в узлы.
plot(t1)
```

19. Функция wentropy – вычисление энтропии сигнала. Используется при выборе оптимального пакетного вейвлет-разложения.

Полное дерево пакетного вейвлет-разложения содержит много коэффициентов. Изучение всех полученных коэффициентов пакета затруднительно ввиду их большого числа. Кроме того, некоторые из коэффициентов могут быть малоинформативны. Поэтому важно получить не все дерево, а некоторое поддерево оптимальной величины в смысле числа коэффициентов и их информативности. Представляет интерес нахождение оптимального разложения относительно удобного критерия. Как известно, *энтропия* – общая концепция оценки информативности во многих областях, главным образом в обработке сигналов. Поэтому для оценки дерева разложения естественно использовать классические критерии на основе энтропии, которые обладают свойством аддитивности (объединения) по отношению к сигналам и дают информационные оценки сигнала.

Рассмотрим четыре различных критерия энтропии. В следующих выражениях s является сигналом, а s_i являются значениями сигнала s. Во всех формулах используется сумма, поскольку она обладает свойством аддитивности по отношению к объединению массивов, задающих сигналы.

1. Энтропия *Шеннона* ('shannon')
   $$E1(s) = -\sum_i s_i^2 \log(s_i^2),$$
   при следующем соглашении: 0*log(0) = 0.
2. *Норма* пространства l^p, $p \geq 1$ ('norm')
   $$E2(s) = \sum_i |s_i|^p = \|s\|_p^p .$$
3. Энтропия *логарифм энергии* ('log energy')
   $$E3(s) = \sum_i \log(s_i^2),$$
   при следующем соглашении: log(0) = 0.
4. *Пороговая* энтропия ('threshold') $E4(s_i) = 1$, если $|s_i| > \varepsilon$ и 0 – в противном случае, тогда $E4(s)$ есть число элементов сигнала, величина которых больше порогового значения ε.

5. Энтропия «SURE» *E5*(*s*). Дополнительная информация имеется в Help Matlab: Wavelet Toolbox\Using the Wavelet Toolbox\Advanced Concepts\Choosing the Optimal Decomposition.

Синтаксис функции wentropy:

E = wentropy(X,T,P),
E = wentropy(X,T).

Функция E = wentropy(X,T,P) вычисляет энтропию E вектора или матрицы X, T указывает тип энтропии: 'shannon', 'threshold', 'norm', 'log energy', 'sure', 'user'. P – дополнительный параметр:

- 'shannon' или 'log energy' – P не используется;
- 'threshold' or 'sure' – тогда P – пороговое значение, P > 0;
- 'norm' – тогда P – степень (P ≥ 1);
- 'user' – тогда P определен как имя функции, определенной пользователем.

E = wentropy(X,T) эквивалентно E = wentropy(X,T,0).

Пример 5. Для случайного сигнала x = randn(1,200).

```
x = randn(1,200); e = wentropy(x,'shannon')
e =
    -142.7607
e = wentropy(x,'threshold',0.2)
e =
    162
e = wentropy(x,'norm',1.1)
e =
     160.1583
```

20. Функция entrupd – замена функции энтропии в пакетном анализе.

Синтаксис:

T = entrupd(T,E),
T = entrupd(T,E,P)

Возвращает то же самое дерево T, но с новой функцией энтропии E и, возможно, с параметром P.

Пример 6.

```
load noisdopp; x = noisdopp;
t = wpdec(x,2,'db1','shannon');
nodes = allnodes(t);
ent = read(t,'ent',nodes)
ent =
```

```
    1.0e+04 *
    -5.8615 -6.8204 -0.0350 -7.7901 -0.0497 -0.0205 -0.0138
t = entrupd(t,'threshold',0.5);
nent = read(t,'ent')
nent =
    937 488 320 241 175 170 163
```

21. Функция besttree – наилучшее дерево пакетного вейвлет-разложения. Данная команда находит наилучшее дерево пакетного вейвлет-разложения и наилучшую функцию энтропии, если получено первоначальное дерево T командой wpdec. Энтропия указана в функции wpdec.

Алгоритм. Коэффициенты вейвлет-разложения являются проекциями сигнала на базисные функции $\varphi_{j,n}(x)$ и $\psi_{j,n}(x)$. Энтропия характеризует усредненность (размазанность) сигнала по базисным направлениям, она мала, когда большая часть сигнала сосредоточена на малом числе базисных направлений. Именно в этом случае коэффициентов становится меньше, и информация о сигнале может быть сжата. Рассмотрим одномерный случай. Начиная с корневого узла, лучшее дерево находится по следующей схеме. Узел N разбивается на два узла N1 и N2, если и только если сумма энтропии N1 и N2 меньше, чем энтропия N. Это локальный критерий, основанный лишь на информации узла N.

Синтаксис функции besttree:

T1 = besttree(T),
[T1,E] = besttree(T),
[T1,E,N] = besttree(T)

Функция T1 = besttree(T) вычисляет наилучшее дерево T1, соответствующее наилучшему значению энтропии.

Функция [T1,E] = besttree(T) вычисляет наилучшее дерево T1 и, дополнительно, наилучшие значения энтропии E. Это вектор, у которого E(j) – это оптимальные значения энтропии в узле с номером j – 1 (см. allnodes) оптимального дерева T1.

Функция [T1,E,N] = besttree(T) вычисляет наилучшее дерево T1, наилучшие значения энтропии E и, дополнительно, вектор N, содержащий номера соединенных (неразложенных) узлов в исходном дереве T.

Пример 7. Наилучшее дерево.

```
load noisdopp; x = noisdopp;
wpt = wpdec(x,3,'db1');
[bt,e,n] = besttree(wpt);
plot(bt)
```

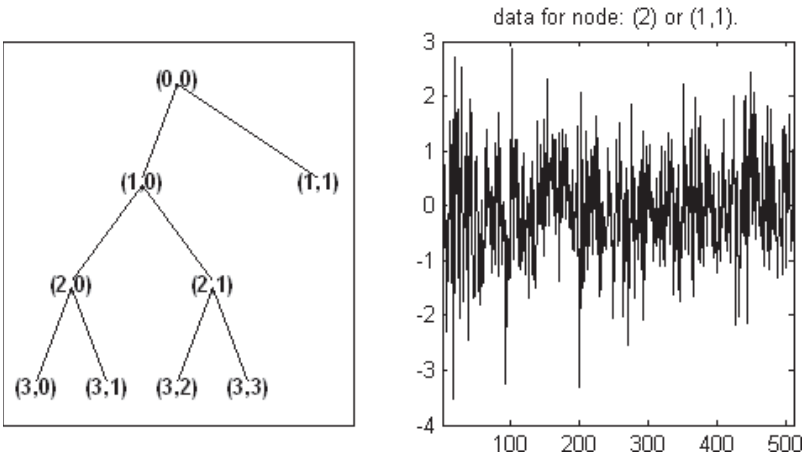


Рис. 3.4.5. Наилучшее дерево

В этом случае соединенный узел имеет номер 2, индекс (1,1). Справа изображен сигнал в этом неразложенном узле.

22. Функция bestlevt – дерево пакетного вейвлет-разложения наилучшего уровня. Данная команда находит оптимальное полное поддерево T1 первоначального дерева T относительно критерия энтропии. Единственное отличие алгоритма от besttree заключается в том, что оптимальное дерево ищется среди полных поддеревьев начального дерева вместо всех поддеревьев. Найденное оптимальное дерево будет полным, но может иметь меньшую глубину, чем начальное.

Синтаксис:

T1 = bestlevt(T),
[T1,E] = bestlevt(T).

Во втором случае дополнительно вычисляются лучшие значение энтропии E в узлах дерева T1.

Пример 8.

```
load noisdopp; x = noisdopp;
wpt = wpdec(x,3,'db1');
wpt = wpsplt(wpt,[3 0]);
plot(wpt)
blt = bestlevt(wpt);
plot(blt)
```

23. Функция wpdec2 – пакетное 2D-вейвлет-разложение.
Синтаксис:

T = wpdec2(X,N,'wname',E,P),
T = wpdec2(X,N,'wname').

Функция wpdec2 находит пакетное дерево T, соответствующее пакетному разложению матрицы X до уровня N, используя вейвлет 'wname'.

E – строка, указывающая тип энтропии (см wentropy): E = 'shannon', 'threshold', 'norm', 'log energy', 'sure', 'user'.

P – дополнительный параметр. Если E = 'shannon' или 'log energy', то P не используется, если E = 'threshold' или 'sure', то P – порог, если E = 'norm', то P – степень.

T = wpdec2(X,N,'wname') эквивалентно
T = wpdec2(X,N,'wname','shannon').

24. Функция wprec2 – пакетное 2D-вейвлет-восстановление. Обратная функция к wpdec2. Находит восстановленную матрицу X = wprec2(T), соответствующую вейвлет-пакетному дереву T.

25. Функция wpspectrum – вейвлет-пакетный спектр.
Синтаксис:

[SPEC,TIMES,FREQ] = wpspectrum(WPT,Fs)
[...] = wpspectrum(WPT,Fs,'plot')
[...,TNFO] = wpspectrum(...)

Входные параметры:

- WPT – вейвлет-пакетное дерево;
- Fs – частота выборки в герцах. По умолчанию 1;
- plot – для изображения пакетного спектра.

Выходные параметры:

- SPEC – вейвлет-пакетный спектр, это матрица размера 2^J на N, где J есть уровень разложения и N – длина начального узла 0 в пакетном дереве. Интервал частоты между строками SPEC есть $Fs/2^J + 1$;
- TIMES – вектор времени в секундах, длины начального узла 0 в пакетном дереве. Интервал между элементами есть 1/Fs;
- FREQ – вектор частот длины 2^J, где J есть уровень разложения. Интервал частоты в FREQ есть $Fs/2^J + 1$;
- TNFO – терминальные узлы. Вектор номеров терминальных узлов, упорядоченных в частотном порядке.

3.5. Удаление шума и сжатие сигнала

Вейвлет-анализ эффективно используется для удаления шума в сигнале. В простейшей модели предполагается, что зашумленный сигнал имеет вид $s(n) = f(n) + \sigma e(n)$, где $f(n)$ – полезный сигнал, σ – уровень шума и $e(n)$ – гауссов белый шум, то есть стационарная случайная последовательность с нулевым математическим ожиданием, абсолютно некоррелированная и с дисперсией, равной единице. Название «белый шум» отражает тот факт, что такой сигнал $e(n)$ имеет постоянный спектр на всех частотах по аналогии с белым светом, имеющим в видимой части равномерный сплошной спектр. Цель состоит в том, чтобы подавить шумовую часть $e(n)$ сигнала и восстановить f.

Вейвлет-анализ также эффективно используется для сжатия сигналов путем многоуровневого вейвлет-разложения и удаления части коэффициентов аппроксимации. Процедура сжатия вполне аналогична удалению шума в сигнале при использовании жесткой пороговой обработки коэффициентов.

Пороговая обработка вейвлет-коэффициентов для удаления шума

При вейвлет-анализе сигнал раскладывается на аппроксимирующие коэффициенты, которые представляют сглаженный сигнал, и детализирующие коэффициенты, описывающие колебания. Следовательно, шумовая компонента больше отражается в детализирующих коэффициентах cD_j. Поэтому при удалении шума обрабатывают обычно детализирующие коэффициенты. Второе предположение заключается в том, что шумовая компонента представляет собой сигнал, меньший по модулю, чем основной. Поэтому простейший способ удаления шума состоит в том, чтобы сделать нулевыми значения коэффициентов, меньшие некоторого порогового значения. Эта процедура называется *пороговой обработкой* (трешолдинг) коэффициентов. Широкое распространение получили такие методы пороговой обработки, как *жесткий трешолдинг* и *мягкий трешолдинг*.

При жесткой пороговой обработке сохраняются неизменными все коэффициенты, большие или равные по абсолютной величине порога τ, а меньшие коэффициенты обращаются в нуль. При мягкой пороговой обработке наряду с обращением в нуль коэффициентов, по модулю меньших, чем τ, происходит уменьшение по модулю остальных коэффициентов на величину τ.

При решении задачи шумоподавления необходимо: оценить спектральный состав шумовой компоненты, выбрать тип пороговой обработки (трешолдинга) и критерий расчета самого порога.

От выбора порогового уровня фона (оценки дисперсии шума) зависит качество шумоподавления сигнала, оцениваемое в виде отношения сигнал/шум. Задание малых значений порога сохраняет фон в коэффициентах детализации и поэтому приводит лишь к незначительному увеличению отношения сигнал/шум. При больших значениях порога можно потерять коэффициенты, которые несут существенную информацию. Поиск оптимального значения τ_0 означает отыскание такого порога, который при наименьшем изменении восстановленного сигнала обеспечивает наибольшее значение отношения сигнал/шум.

Критерий Штейна несмещенной оценки риска (*Stein's unbiased risk estimation*) позволяет определить оптимальное значение порога τ_0, соответствующее уровню j разложения сигнала, как аргумент некоторой функции риска $R_j(\tau)$, при котором данная функция принимает минимальные значения.

Качество шумоподавления сигнала (отношение сигнал/шум) зависит также от способа применения трешолдинга. Используются следующие способы пороговой обработки:

- общий трешолдинг, осуществляемый с использованием фиксированного значения порога τ – значения, единого для всех уровней разложения и коэффициентов детализации сигнала;
- многоуровневый трешолдинг, осуществляемый с использованием порога τ, значения которого изменяются от уровня к уровню;
- локальный трешолдинг, подразумевающий использование порога τ, переменного не только по уровню разложения, но также зависящего от позиции коэффициентов детализации на данном уровне.

Таким образом, процедура удаления шума сигнала и компрессии изображения состоит из трех шагов:

1. Декомпозиция. Выбирается вейвлет и уровень разложения N. Вычисляется вейвлет-разложение исходного сигнала до уровня N.
2. Пороговая обработка детализирующих коэффициентов. Для каждого уровня от 1 до N выбирается порог и применяется мягкая (а в случае изображений – жесткая) пороговая обработка детализирующих коэффициентов.

3. Реконструкция. Производится вейвлет-реконструкция, основанная на первоначальных аппроксимирующих коэффициентах уровня N и модифицированных детализирующих коэффициентах уровней от 1 до N.

Подробности и примеры можно найти в Help Wavelet Toolbox в разделе «Denoising».

3.5.1. Функции MATLAB для удаления шума и сжатия

Основные функции очистки сигналов от шума и для сжатия:

- **thselect** – выбор порога для удаления шума 1D;
- **wthresh** – мягкая или жесткая пороговая обработка (1D, 2D);
- **wthcoef** – пороговая обработка вейвлет-коэффициентов;
- **wthcoef2** – двумерная пороговая обработка вейвлет-коэффициентов;
- **ddencmp** – параметры по умолчанию для удаления шума (1D, 2D);
- **wden** – автоматическое удаление шума 1D;
- **wdencmp** – удаление шума и сжатие при помощи вейвлетов (1D, 2D);
- **wcompress** – подлинное сжатие с использованием вейвлетов;
- **wbmpen** – штрафной порог для 1D- или 2D-удаления шума;
- **wdcbm** – пороги для 1D-сигналов по стратегии Birge-Massart;
- **wdcbm2** – пороги для 2D-сигналов по стратегии Birge-Massart;
- **cmddenoise** – удаление шума, зависящее от выбора интервала;
- **wthrmngr** – управление установкой порогов;
- **wnoisest** – оценка шума 1D-вейвлет-коэффициентов;
- **wnoise** – тестовые стигналы.

Удаление шума с вейвлет-пакетным разложением:

- **wpbmpen** – штрафной порог удаления шума вейвлет-пакета;
- **wpthcoef** – пороговая обработка пакетных коэффициентов (1D, 2D).

Многовариантное удаление шума:

- **wmulden** – многовариантное удаление шума;
- **wmspca** – многомасштабный анализ главных компонент.

Для вейвлет-анализа мультисигнала можно использовать также главное вейвлет-меню wavemenu, которое дает более наглядный и эффективный инструментарий для удаления шума, а также рассмотренные ранее функции удаления шума мультисигнала (раздел 3.4.5):

1. Функция thselect – выбор порога для удаления шума 1D.
Синтаксис:

THR = thselect(X,TPTR).

Находит значение порога для сигнала X, используя правило, определенное строкой TPTR. Этот параметр может принимать следующие значения:

- TPTR = 'rigrsure' – адаптивный выбор порога с использованием принципа Штейна несмещенной оценки риска;
- TPTR = 'heursure' – эвристический вариант первого выбора;
- TPTR = 'sqtwolog' – порог sqrt(2*log(length(X)));
- TPTR = 'minimaxi' – минимаксный порог.

Правила выбора порога базируются на основной модели $y = f(t) + e$, где e является белым шумом N(0, 1). Если шум немасштабированный или не белый, можно его обработать, используя перемасштабирование выходного порога THR (см. параметр SCAL в описании команды wden).

Пример 1. Создаем гауссов белый шум. Находим порог для каждого правила выбора.

```
init = 2055415866; randn('state',init); x = randn(1,1000);
thr = thselect(x,'rigrsure')
thr =
    1.8065
thr = thselect(x,'sqtwolog')
thr =
    3.7169
thr = thselect(x,'minimaxi')
thr =
    2.2163
```

2. Функция wthresh – пороговая обработка (1D, 2D).
Синтаксис:

Y = wthresh(X,SORH,T)
Y = wthresh(X,'s',T)
Y = wthresh(X,'h',T)

Функция выполняет мягкую (если SORH = 's') или жесткую (если SORH = 'h') пороговую обработку вектора или матрицы X. T – значение порога.

3. Функции wthcoef и wthcoef2 – одномерная и двумерная пороговая обработка. Рассмотрим вторую функцию как более общую.

Если известна структура вейвлет-разложения [C,S] (см. wavedec2) некоторого уровня разложения N0, то функция wthcoef2 позволяет провести пороговую обработку коэффициентов разложения и получить модифицированное разложение [NC,S].

Синтаксис:

```
NC= wthcoef2('type',C,S,N,T,SORH)
NC= wthcoef2('type',C,S,N)
NC = wthcoef2('a',C,S)
NC = wthcoef2('t',C,S,N,T,SORH)
```

Функция NC = wthcoef2('type',C,S,N,T,SORH) для 'type' = 'h', 'v' или 'd' находит соответственно горизонтальные, вертикальные или диагональные коэффициенты, полученные из структуры вейвлет-разложения [C,S] либо мягкой (SORH = 's'), либо жесткой (SORH = = 'h') пороговой обработкой, определенной векторами N и T. N содержит уровни деталей для обработки, а T – соответствующие пороги.

Функция NC = wthcoef2('type',C,S,N) для 'type' = 'h', 'v' или 'd' находит соответственно горизонтальные, вертикальные или диагональные коэффициенты, полученные из вейвлет-разложения [C,S], полагая равными нулю все детализирующие коэффициенты уровня N.

Функция NC = wthcoef2('a',C,S) находит коэффициенты, полученные обнулением коэффициентов аппроксимации.

Функция NC = wthcoef2('t',C,S,N,T,SORH) находит детализирующие коэффициенты, полученные из структуры вейвлет-разложения [C,S] либо мягкой (SORH = 's'), либо жесткой (SORH = 'h') пороговой обработкой, определенной векторами N и T. Вектор N содержит уровни деталей, которые подвергаются пороговой обработке, а T – соответствующие пороги. Вектор N должен быть таким, что $1 \le N(i) \le$ $\le$ size(S,1)-2.

[NC,S] есть структура модифицированного вейвлет-разложения.

4. Функция ddencmp – вычисление параметров по умолчанию для удаления шума или сжатия (1D, 2D). Функция находит значения параметров по умолчанию для всех общих процедур, связанных с удалением шума или сжатием одно- или двумерных сигналов, используя вейвлеты или вейвлет-пакеты. Параметры по умолчанию для удаления шума:

- мягкий пороговый метод и сохранение коэффициентов аппроксимации;

- thr = sqrt(2*log(n))*s, где s есть оценка уровня шума, n = = prod(size(x)).

Параметры по умолчанию для компрессии:

- жесткий пороговый метод и сохранение коэффициентов аппроксимации;
- thr = median(abs(cD1)), если отлично от нуля, либо
- thr = 0.05 * max(abs(cD_1)).

Параметры по умолчанию при удалении шума для вейвлет-пакетов (четыре параметра вывода):

- мягкий пороговый метод и сохранение коэффициентов аппроксимации;
- thr = sqrt(2*log(n*log(n)/log(2))), n = prod(size(x)), уровень шума предполагается равным 1;
- энтропия есть критерий 'sure'.

Синтаксис функции wdencmp:

[THR,SORH,KEEPAPP,CRIT] = ddencmp(IN1,IN2,X),
[THR,SORH,KEEPAPP] = ddencmp(IN1,'wv',X),
[THR,SORH,KEEPAPP,CRIT] = ddencmp(IN1,'wp',X).

В первом случае функция находит значения по умолчанию для удаления шума или сжатия для входного вектора или матрицы X.

Исходные параметры:

- X – одно- или двумерный сигнал (вектор или матрица);
- IN1 есть 'den' для удаления шума или 'cmp' для компрессии;
- IN2 есть 'wv' для вейвлета или 'wp' для вейвлет-пакета.

Выходные параметры:

- THR – значение порога;
- SORH – мягкий или жесткий пороговый метод;
- KEEPAPP = 1, то коэффициенты аппроксимации не подвергаются пороговой обработке, иначе это возможно;
- CRIT – критерий энтропии, используется только для вейвлет-пакетов.

Для вейвлетов три параметра вывода: THR, SORH, KEEPAPP. Для вейвлет-пакетов – четыре: THR, SORH, KEEPAPP, CRIT.

Пример 2. Создаем гауссов одномерный шум. Находим значения по умолчанию для вейвлетов ('wv',) три параметра вывода. Эти значения могут использоваться для wdencmp с вейвлетом 'wv'.

Удаление шума.

```
init = 2055415866; randn('seed',init);  x = randn(1,1000);
     [thr,sorh,keepapp] = ddencmp('den','wv',x)
```

```
thr =    3.8593
sorh =     s
keepapp =     1
```

Компрессия.

```
[thr,sorh,keepapp] = ddencmp('cmp','wv',x);
```

Удаление шума для вейвлет-пакетов.

```
[thr,sorh,keepapp,crit] = ddencmp('den','wp',x);
```

5. Функция wden – автоматическое удаление шума (1D). Функция wden автоматически удаляет шум одномерного сигнала X, используя вейвлеты.

Синтаксис:

[XD,CXD,LXD] = wden(X,TPTR,SORH,SCAL,N,'wname'),
[XD,CXD,LXD] = wden(C,L,TPTR,SORH,SCAL,N,'wname').

Первая версия функции вычисляет очищенную от шума версию XD входного сигнала X, используя пороговую обработку вейвлет-коэффициентов. Дополнительно находит вейвлет-коэффициенты [CXD,LXD] разложения очищенного сигнала XD.

Исходные параметры:

- X – сигнал;
- TPTR – строка, определяющая выбор порога:
  - rigrsure' – адаптивный выбор порога с использованием принципа Штейна несмещенной оценки риска (SURE);
  - heursure' – эвристический вариант первого выбора;
  - qtwolog' – универсальный порог sqrt(2*log(length(X)));
  - 'minimaxi' – минимаксный порог;
- SORH ='s' или 'h' – выбор мягкого или жесткого порога;
- SCAL – строка, которая определяет мультипликативное пороговое перемасштабирование. Если шум вне пределов [0, 1] или не белый, то порог должен быть перемасштабирован с использованием оценки уровня шума. Возможны следующие варианты:
  - 'one' – без перемасштабирования, когда используется базовый метод;
  - 'sln' – для перемасштабирования порога с использованием оценки уровня шума на базе коэффициентов первого уровня;
  - 'mln' – для перемасштабирования порога с использованием оценки уровня шума, зависящей от уровня разложения;

- N – уровень вейвлет-разложения и 'wname' – имя ортогонального вейвлета.

Во втором случае функция wden производит то же самое, но использует уже полученное вейвлет-разложение [C,L] исходного сигнала до уровня N и ортогональный вейвлет 'wname'.

Пример 3. Создаем зашумленный сигнал функцией wnoise. Для удаления шума используем мягкий эвристический метод SURE. Разложение x берем до уровня 5 по вейвлету sym8.

```
snr = 3; init = 2055615866;
[xref,x] = wnoise(3,11,snr,init);
xd = wden(x,'heursure','s','one',5,'sym8');
subplot(311), plot(xref), axis([1 2048 -10 10]); title('Original
signal');
subplot(312), plot(x), axis([1 2048 -10 10]); title('Noisy signal');
subplot(313), plot(xd), axis([1 2048 -10 10]); title('De-noised signal -
heuristic SURE');
```

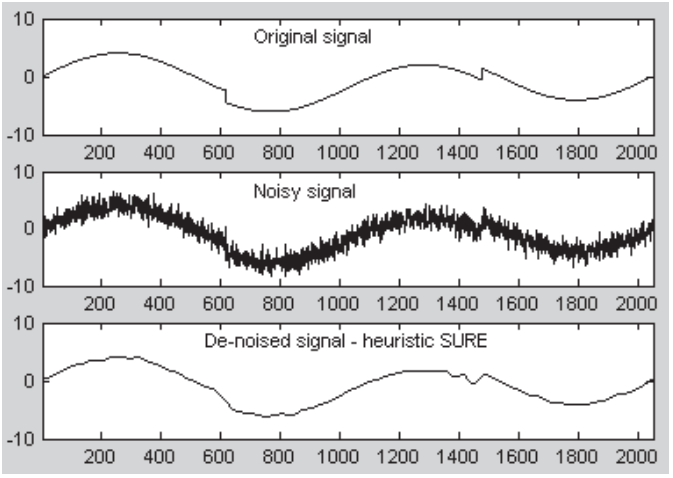


Рис. 3.5.1. Зашумление и очистка

6. Функция wdencmp – удаление шума и сжатие при помощи вейвлетов (1D, 2D).

Синтаксис:

[XC,CXC,LXC, PR0, PRL2] =
= wdencmp('gbl',X,'wname',N,THR,SORH,KEEPAPP),

[XC,CXC,LXC, PR0, PRL2] =
= wdencmp('lvd',X,'wname',N,THR,SORH),
[XC,CXC,LXC, PR0, PRL2] =
= wdencmp('lvd',C,L,'wname',N,THR,SORH).

Описание. Случай параметра 'gbl'. Функция производит:

- XC – очищенную от шума версию входного сигнала X (1D или 2D);
- [CXC,LXC] – структуру вейвлет-разложения сигнала XC;
- PR0 – оценку числа обнуленных коэффициентов сигнала X в процентах;
- PRL2 – отношение L2-норм сжатого и первоначального сигналов в процентах. PRL2 = 100 * (norm(CXC)/ norm(C))2 , где [C,L] – вейвлет-разложение X. Если X – одномерный сигнал и 'wname' – ортогональный вейвлет, то PRL2 = 100 * $\|XC\|^2/\|X\|^2$.

Функция использует:

- X – сигнал, или изображение;
- wname' – имя вейвлета и N – уровень вейвлет-разложения;
- THR – значение порога;
- SORH = 's' или 'h' – выбор мягкого или жесткого порогового метода;
- если KEEPAPP = 1, то коэффициенты аппроксимации не подвергаются пороговой обработке, иначе это возможно.

Функция wdencmp('gbl',C,L,'wname',N,THR,SORH,KEEPAPP) выполняет то же самое, используя вейвлет-разложение [C,L] исходного сигнала до уровня N.

В одномерном случае и опции 'lvd'. Функция выполняет то же самое, но используя зависимые от уровня пороги, указанные в векторе THR (THR должен иметь длину N). Кроме того, коэффициенты аппроксимации сохраняются.

Для двумерного случая и опции 'lvd'. В этом случае THR должен быть матрицей 3×N, содержащей зависимые от уровня пороги в трех направлениях: по горизонтали, по диагонали и по вертикали.

Отметим, что сравнительно с wden (автоматическое удаление шума) wdencmp дает больше возможностей и можно осуществить свою стратегию удаления шума.

Пример 4. Загружаем изображение woman = (X, map). Делаем вейвлет-разложение x при n = 5; w = 'sym2'. Используем wdencmp для компрессии изображения.

```
load woman; x=X(100:200,100:200);
nbc = size(map,1); n = 5; w = 'sym2';
[c,l] = wavedec2(x,n,w); thr=20;
[xd,cxd,lxd,perf0,perfl2]=wdencmp('gbl',c,l,w,n,thr,'h',1);
colormap(pink(nbc));
subplot(221), image(wcodemat(x,nbc)), title('Original image')
subplot(222), image(wcodemat(xd,nbc)), title('De-noised image')
xlab1 = ['2-norm rec.: ',num2str(perfl2)];
xlab2 = [' %  -- zero cfs: ',num2str(perf0), ' %'];
xlabel([xlab1 xlab2]);
```

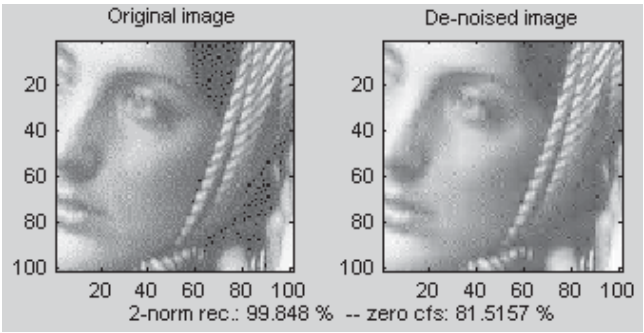


Рис. 3.5.2. Удаление шума в изображении

Пример 5. Кроме того, первая опция 'lvd' позволяет указать порог в зависимости от уровня и ориентации. Аппроксимирующие коэффициенты сохраняются. Для сигнала x из примера 2, используя разложение до второго уровня, полагаем:

```
thr_h = [17 18];    % Горизонтальные пороги по двум уровням;
thr_d = [19 20];    % Диагональные пороги по двум уровням;
thr_v = [21 22];    % Вертикальные пороги по двум уровням;
thr = [thr_h ; thr_d ; thr_v].
[xd,cxd,lxd,perf0,perfl2] = wdencmp('lvd',x,'sym8',2,thr,'h').
```

7. Функция wcompress – полное сжатие с использованием вейвлетов. В этой функции наряду с вейвлет-разложением и пороговой обработкой коэффициентов разложения используется дополнительная последовательно, несколько раз, применяемая процедура квантования и кодирования. Полная каскадная последовательность сжатия включает фазы вейвлет-обработки, квантования, кодирования и раскодирования. Квантование сигнала – это его разбиение на определенные квадраты и выбор в каждом квадрате одного усредненного

значения сигнала. Основной параметр процедуры полного сжатия – число циклов (по умолчанию 10). Увеличение этого числа приводит к лучшему восстановлению, но худшей степени сжатия.

Синтаксис:

```
wcompress('c',X,SAV_FILENAME,COMP_METHOD)
wcompress(...,'ParName1',ParVal1,'ParName2',ParVal2,...)
[CR,BPP] = wcompress('c',...)
XC = wcompress('u',SAV_FILENAME)
XC = wcompress('u',SAV_FILENAME,'plot')
XC = wcompress('u',SAV_FILENAME,'step')
```

Рассмотрим первую функцию wcompress('c',X,SAV_FILENAME, COMP_METHOD).

Входные параметры:

- 'c' – символ, обозначающий дополнительную компрессию, если 'u' – то без дополнительных шагов кодирования квантованием, только вейвлет-обработка коэффициентов и восстановление;
- X – входной сигнал;
- SAV_FILENAME – имя файла сжатого изображения для его сохранения;
- COMP_METHOD – метод сжатия,он может быть двух типов:
  - метод прогрессивных значений коэффициентов (PCSM), может принимать значения: 'ezw', 'spiht', 'stw', 'wdr', 'aswdr' и 'spiht_3d';
  - метод трешолдинга коэффициентов: 'lvl_mmc', 'gbl_mmc_f' и 'gbl_mmc_h'.

Во втором варианте этой функции: [CR,BPP] = wcompress('c',...) – сохраняются еще и две характеристики сжатия сигнала:

- CR – коэффициент сжатия. Он означает, что сжатый образ хранится с использованием только CR% от размера хранения начального образа;
- BPP – коэффициент Bit на Pixel – это число битов, необходимых для хранения одного пикселя изображения. Для серых (grayscale) исходных изображений BPP равно 8, а для полноцветных начальное значение BPP равно 24 (RGB – это три матрицы).

Функция XC = wcompress('u',SAV_FILENAME) обрабатывает файл SAV_FILENAME без дополнительного квантования и кодирования и возвращает изображение XC.

Подробнее об этой функции см. Help Wavelet Toolbox.

8. Функция wbmpen – штрафной порог для 1D- или 2D-удаления шума.

Синтаксис:

THR = wbmpen(C,L,SIGMA,ALPHA)

Функция находит глобальный порог THR для удаления шума. Параметры:

- [C,L] – структура вейвлет-разложения анализируемого сигнала или изображения;
- SIGMA есть стандартное отклонение гауссова белого шума в анализируемом сигнале или изображении (см. wnoisest);
- ALPHA является параметром настройки для штрафного метода. Он должен быть вещественным числом, большим 1. Обычно ALPHA = 2.

THR минимизирует критерий штрафа, заданный формулой: crit(t) = –sum(c(k)^2,k<=t) + 2*SIGMA^2*t*(ALPHA + log(n/t)), где c(k) – вейвлет-коэффициенты в порядке убывания их абсолютных значений и n – число коэффициентов. Если t* – точка минимума, то THR = |c(t*)|.

Пример 6. Удаление шума в изображении. Загружаем тестовое изображение tire (колесо автомобиля), накладываем шум типа «соль и перец» и делаем его разложение до уровня 3 по вейвлету coif2. Применяем wbmpen для выбора глобального порога для образа. Применяем wdencmp для очистки изображения от шума, используя найденное значение порога с мягким пороговым методом и сохраняя аппроксимацию.

```
load tire; nbc = size(map,1); wname = 'coif2'; lev = 3;
Y = imnoise(X, 'salt & pepper',0.1);
[c,s] = wavedec2(Y,lev,wname);
det1 = detcoef2('compact',c,s,1);
sigma = median(abs(det1))/0.6745;
alpha = 5;
thr = wbmpen(c,3,sigma,alpha)
keepapp = 1;
xd = wdencmp('gbl',c,s,wname,lev,thr,'s',keepapp);
colormap(gray(nbc));
subplot(331), image(wcodemat(X,nbc)); title('Original image')
subplot(332), image(wcodemat(Y,nbc)); title('Noised image')
subplot(333), image(wcodemat(xd,nbc)); title('De-noised image')
```

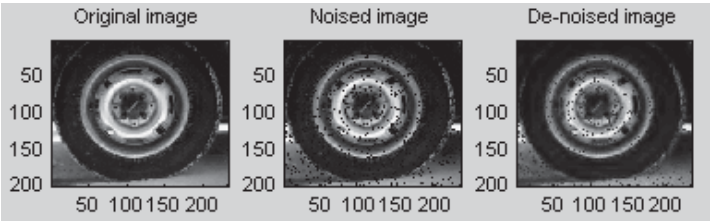


Рис. 3.5.3. Удаление шума в изображении

9. Функции wdcbm и wdcbm2 – пороги для 1D- и 2D-сигналов, использующие стратегию Birge-Massart. Находят зависящие от уровня пороги и число сохраненных коэффициентов при многоуровневом разложении глубины N.

Синтаксис:

[THR,NKEEP] = wdcbm(C,L,ALPHA), [THR,NKEEP] =
= wdcbm(C,L,ALPHA,M).
[THR,NKEEP] = wdcbm2(C,S,ALPHA), [THR,NKEEP] =
= wdcbm2(C,S,ALPHA,M)

Функции находят пороги THR – вектор длины N и число сохраненных коэффициентов NKEEP – тоже вектор длины N. Уровень N и числа M, ALPHA определяют стратегию:

- на уровне N + 1 (и более) все сохраняется;
- для уровня i от 1 до N – ni самых больших коэффициентов сохраняется, где $ni = M(N + 2 - i)^{ALPHA}$.

Обычные значения ALPHA = 1.5 – для сжатия и ALPHA = 3 – для удаления шума.

10. Функция cmddenoise – удаление шума, зависящее от выбора интервала. Функция sigden = cmddenoise(sig,wname,level) возвращает очищенный от шума сигнал sigden, полученный интервально зависимым удалением шума–сигнала sig, используя ортогональный или биортогональный вейвлет wname и уровень разложения level. Функция cmddenoise разбивает сигнал на интервалы, основываясь на точках перехода дисперсии в коэффициентах деталей первого уровня и устанавливая пороги для каждого интервала отдельно. Положение и число точек перехода дисперсии находятся автоматически. Подробнее об этой функции см. Help для cmddenoise и приведенные там примеры для каждого варианта использования cmddenoise. Графический интерфейс wavemenu дает дополнительные возможности выбора интервалов для удаления шума.

11. Функция wnoisest – оценка шума 1D-вейвлет-коэффициентов. Вычисляет оценки стандартного отклонения детализирующих вейвлет-коэффициентов для уровней, указанных вектором S.

Синтаксис:

STDC = wnoisest(C,L,S)
STDC= wnoisest(C)

Здесь [C,L] является структурой вейвлет-разложения входного сигнала.

Если C есть одномерный массив ячеек, то STDC = wnoisest (C) возвращает вектор, такой что STDC(k) является оценкой стандартного отклонения C{k}.

Если C – числовой массив, то STDC = wnoisest (C) возвращает вектор, такой что STDC(k) является оценкой стандартного отклонения k-й строки C(k,:).

При оценке используется медиана абсолютного отклонения, деленная на 0.6745, что хорошо подходит для гауссова белого шума в одномерной модели (см. thselect).

Пример 7. Генерируем гауссов белый шум, делаем вейвлет-разложение полученного сигнала и находим стандартное отклонение на уровнях 1 и 2. Поскольку x – гауссов белый шум с отклонением единица, то и оценки должны быть близки к 1.

```
init = 2055415866;  randn('seed',init);  x = randn(1,1000);
[c,l] = wavedec(x,2,'db3');
wnoisest(c,l,1:2)
ans = 1.0111, 1.0763
```

12. Функция wnoise – тестовые стигналы. Позволяет получить тестовый сигнал, определенный параметром FUN на сетке из 2^N точек промежутка [0, 1].

Синтаксис:

X = wnoise(FUN,N)
[X,XN] = wnoise(FUN,N,SQRT_SNR)
[X,XN] = wnoise(FUN,N,SQRT_SNR,INIT)

Параметр FUN может принимать значения:

- FUN = 1 или 'blocks';
- FUN = 2 или 'bumps';
- FUN = 3 или 'heavy sine';
- FUN = 4 или 'doppler';
- FUN = 5 или 'quadchirp';
- FUN = 6 'mishmash'.

Выходной параметр XN содержит тот же самый тестовый вектор, испорченный аддитивным гауссовым белым шумом N(0,1). При этом XN имеет отношение сигнал–шум, равное SNR = (SQRT_SNR)^2.

Удаление шума с вейвлет-пакетным разложением

Можно использовать функции пакетного вейвлет-анализа (раздел 3.4) и две специальные функции для удаления шума:

- **wpbmpen** – штрафной порог удаления шума вейвлет-пакета;
- **wpthcoef** – пороговую обработку пакетных коэффициентов (1D, 2D).

13. Функция wpbmpen – штрафной порог удаления шума вейвлет-пакета.

Синтаксис:

THR = wpbmpen(T,SIGMA,ALPHA),
THR = wpbmpen(T,SIGMA,ALPHA,ARG).

Функция находит глобальный порог THR для удаления шума, используя штрафной метод Birge-Massart. Параметры:

- T есть вейвлет-пакетное дерево сигнала или изображения, которые подвергаются компрессии либо удалению шума;
- SIGMA является стандартным отклонением гауссового белого шума в модели удаления шума;
- ALPHA есть параметр в штрафной функции. Он должен быть действительным числом, большим, чем 1. Обычно ALPHA = 2;
- THR минимизирует следующий штрафной критерий. Пусть сначала t* – минимум функции crit(t) = –sum(c(k)^2,k<=t) + + 2*SIGMA^2*t*(ALPHA + log(n/t)), где c(k) – это коэффициенты вейвлет-пакета, упорядоченные по убыванию их модулей, и n – число коэффициентов. Тогда THR = $|c(t^*)|$.

Функция wpbmpen(T,SIGMA,ALPHA,ARG) вычисляет глобальный порог и дополнительно изображает три кривые:

- 2*SIGMA^2*t*(ALPHA + log(n/t));
- sum(c(k)^2,k<=t);
- crit(t).

14. Функция wpthcoef – пороговая обработка пакетных вейвлет-коэффициентов (1D, 2D). Функция wpthcoef есть утилита для удаления шума или компрессии. Она находит новое пакетное дерево NT, которое получается из пакетного дерева T пороговой обработкой вейвлет-коэффициентов.

Синтаксис:

NT = wpthcoef(T,KEEPAPP,SORH,THR),

где:

- NT – новое пакетное дерево;
- T – исходное пакетное дерево;
- если KEEPAPP = 1, то коэффициенты аппроксимации не обрабатываются, в противном случае они также обрабатываются;
- если SORH = 's', то используется мягкая пороговая обработка, если SORH = 'h', то жесткая пороговая обработка;
- THR – значение порога.

3.5.2. Многовариантное удаление шума

Пакет Wavelet Toolbox предлагает методы удаления шума мультисигнала, основанные на одновременном использовании одномерного вейвлет-анализа и анализа главных компонент (PCA).

Дискретный мультисигнал X можно рассматривать как последовательность точек $\{X_k\}$ в n-мерном пространстве $\mathbf{R}^n$, или как набор из n одномерных сигналов, составленных из координат X_{ik} последовательности $\{X_k\}$. Поэтому в этом разделе будем называть его многомерным сигналом. Как показано в параграфе 3.3.4, для удаления шума многомерного сигнала можно использовать методы одномерного вейвлет-анализа, примененные отдельно для каждой i-й компоненты $\{X_{ik}\}$ многомерного сигнала. В то же время имеются специальные методы анализа многомерных сигналов. Достаточно распространенным является анализ главных компонент (Principal Components Analysis, PCA) многомерного временного ряда [Ра], [ДЖ].

Пакет Wavelet Toolbox предлагает методы удаления шума многомерного сигнала, основанные на одновременном использовании одномерного вейвлет-анализа и анализа главных компонент (PCA). Напомним сначала кратко теоретические основы PCA.

Анализ главных компонент

Рассмотрим вещественный конечный n-мерный сигнал $X = \{X_k\}$, $k = 1, 2, ..., N$, составленный из n координатных сигналов $\{X_{ik}\}$. Образуем матрицу $X = \{X_{ik}\}$ из n строк и N столбцов (каждая строка представляет отдельный одномерный сигнал).

Основная идея метода главных компонент заключается в том, чтобы из матрицы X размера n на N получить какие-нибудь ее алгебраические характеристики, которые потом можно будет использовать для анализа сигнала. Предлагается следующий подход. Умножим мат-

рицу X на транспонированную, получим квадратную симметричную матрицу $\Sigma = X \cdot X^T$ порядка n. Она имеет вещественные собственные числа $\lambda_1 \geq \lambda_2 \geq ... \geq \lambda_n \geq 0$ и ортонормированную систему собственных векторов $\{v_1, v_2, ..., v_n\}$ (адаптированный базис). Тогда симметричная матрица Σ раскладывается в виде произведения $\Sigma = V \Lambda V^T$ диагональной матрицы Λ собственных чисел и ортогональной матрицы V собственных векторов-столбцов $v_1, v_2, ..., v_n$:

$$\Lambda = \begin{pmatrix} \lambda_1 & 0 & \cdots & 0 \\ 0 & \lambda_2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \lambda_n \end{pmatrix}, \quad V = \begin{pmatrix} V_{11} & V_{21} & \cdots & V_{n1} \\ V_{12} & V_{22} & \cdots & V_{n2} \\ \vdots & \vdots & \ddots & \vdots \\ V_{1n} & V_{2n} & \cdots & V_{nn} \end{pmatrix}.$$

Это равносильно диагонализации ковариационной матрицы Σ за счет перехода к другому базису $v_i = \sum_{j=1}^{n} V_{ij} e_j$, $e_j = \sum_{s=1}^{n} V_{sj} v_s$.

Как уже отмечалось, многомерный дискретный сигнал X можно рассматривать как последовательность точек $\{X_k\}$ в n-мерном пространстве $\mathbf{R}^n$ или как набор из n одномерных сигналов, составленных из координат X_{ik} последовательности $\{X_k\}$ в стандартном базисе $\{e_1, e_2, ..., e_n\}$ пространства $\mathbf{R}^n$. Перейдем теперь к новому базису $\{v_1, v_2, ..., v_n\}$. Ясно, что координаты точек $\{X_k\}$ изменятся: $X_k = (Y_{1k}, ..., Y_{nk})^T$. Найдем эти координаты:

$$X_k = \sum_{j=1}^{n} X_{jk} e_j = \sum_{j,s=1}^{n} X_{jk} V_{sj} v_s = \sum_{j,s=1}^{n} V_{sj} X_{jk} v_s = \sum_{s=1}^{n} Y_{sk} v_s,$$

где $Y_{sk} = \sum_{j=1}^{n} V_{sj} X_{jk}$.

Таким образом, новые компоненты $\{Y_{sk}\}$ сигнала X в адаптированном базисе находятся по формуле $Y_{sk} = \sum_{j=1}^{n} V_{sj} X_{jk}$, то есть представляют собой матрицу вида $Y = V \cdot X$. Эти новые компоненты $\{Y_{sk}\}$ сигнала X называют главными компонентами. Все они соответствуют собственным числам $\lambda_1 \geq \lambda_2 \geq ... \geq \lambda_n \geq 0$. Считается, что чем меньше собственное число λ_s, тем меньшее значение в сигнале X имеет соответствующая компоенента $Y_s = \{Y_{sk}\}$. При малых λ_s ее можно не учитывать, что приводит к выделению в сигнале главных компонент. Этот метод позволяет получить: n числовых характеристик, это собственные числа $\lambda_1 \geq \lambda_2 \geq ... \geq \lambda_n \geq 0$ ковариационной матрицы Σ; адаптированный базис собственных векторов $\{v_1, v_2, ..., v_n\}$ и главные компоненты сигнала. После удаления «неглавных компонент», которые оценены как

несущественные (шумовые), остаются только главные компоненты $(Y_1, ..., Y_i, 0, ..., 0)^T$, которые образуют новую матрицу $\tilde{Y}$. Из этой матрицы $\tilde{Y}$ и из матрицы V восстанавливается новая матрица сигнала по формуле $\tilde{X} = V^T\tilde{Y}$. Этот сигнал можно считать как многомерный сигнал, очищенный от шумовых компонент методом PCA.

Изложим метод главных компонент более аккуратно. Пусть $X = \{X_k\}$ – конечный n-мерный сигнал, $k = 1, 2, ..., N$, составленный из n координатных сигналов $\{X_{ik}\}$. Образуем матрицу $X = \{X_{ik}\}$ из n строк и N столбцов (каждая строка представляет отдельный одномерный сигнал). Вычислим средние арифметические и стандартные отклонения по строкам матрицы X:

$$\overline{X}_i = \frac{1}{N}\sum_{k=1}^{N} X_{ik}, \quad S_i = \sqrt{\frac{1}{N}\sum_{k=1}^{N}(X_{ik} - \overline{X}_i)^2}.$$

Обозначим через X^* матрицу, полученную из X центрированием по строкам и нормированием по стандартным отклонениям S_i, $X^*_{ik} = (X_{ik} - \overline{X}_i)/S_i$. Образуем выборочную ковариационную матрицу Σ размера n на n как произведение X^* и транспонированной:

$$\Sigma = \frac{1}{N}X^*(X^*)^T, \quad \Sigma_{ki} = \frac{1}{N}\sum_{j=1}^{N}\frac{1}{S_k S_i}(X_{kj} - \overline{X}_k)(X_{ij} - \overline{X}_i).$$

Найдем собственные числа $\lambda_1 \geq \lambda_2 \geq ... \geq \lambda_n \geq 0$ и собственные векторы симметричной матрицы Σ. Другими словами, найдем ее разложение в виде произведения $\Sigma = V\Lambda V^T$ диагональной матрицы Λ собственных чисел и ортогональной матрицы V собственных векторов-столбцов:

$$\Lambda = \begin{pmatrix} \lambda_1 & 0 & \cdots & 0 \\ 0 & \lambda_2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \lambda_n \end{pmatrix}, \quad V = \begin{pmatrix} V_{11} & V_{21} & \cdots & V_{n1} \\ V_{12} & V_{22} & \cdots & V_{n2} \\ \vdots & \vdots & \ddots & \vdots \\ V_{1n} & V_{2n} & \cdots & V_{nn} \end{pmatrix}.$$

Это равносильно диагонализации ковариационной матрицы Σ за счет перехода к другому базису.

Главными компонентами $Y_1, ..., Y_n$ многомерного сигнала X называются строки следующей матрицы $Y = V \cdot X^*$. В этом случае собственные числа имеют смысл выборочных дисперсий главных компонент $Y_1, ..., Y_n$. Отметим, что сумма собственных чисел равна n, поэтому величина $(\lambda_i/n)100\%$ имеет смысл доли дисперсии в процентах, использующейся для оценки доли информации, которую несет главная компонента Y_i.

Практически это используется так, что если доля $(\lambda_i/n)100\%$ главной компоненты оценивается как малая, то считается, что соответствующая главная компонента Y_i несет мало информации и может быть удалена как шумовая компонента. После удаления компонент сигнала, которые оценены как несущественные, остаются только главные компоненты (Y_1, ..., Y_i, 0, ..., 0), которые образуют новую матрицу $\tilde{Y}$. Из этой матрицы $\tilde{Y}$ и из матрицы V восстанавливается новая матрица сигнала по формуле $\tilde{X}^* = V^T\tilde{Y}$. Затем производятся децентрирование и денормализация матрицы $\tilde{Y}^*$. Полученная в итоге матрица $\tilde{X}$ представляет многомерный сигнал, очищенный от шумовых компонент методом PCA.

Функции многовариантного удаления шума

Для совместного использования методов вейвлет-анализа и анализа главных компонент, то есть *многовариантного удаления шума*, пакет Wavelet Toolbox предлагает графический интерфейс пользователя wavemenu и две функции:

- **wmulden** – многовариантное удаление шума;
- **wmspca** – многомасштабный анализ главных компонент.

15. Функция wmulden – многовариантное удаление шума.
Синтаксис:

```
[X_DEN,NPC,NESTCOV,DEC_DEN,PCA_Params,DEN_Params]= ...
wmulden(X,LEVEL,WNAME,NPC_APP,NPC_FIN,TPTR,SORH)
[...] = wmulden(X,LEVEL,WNAME,'mode',EXTMODE,NPC_APP,...)
[...] = wmulden(DEC,NPC_APP)
[DEC,PCA_Params] =
= wmulden('estimate',DEC,NPC_APP,NPC_FIN)
[X_DEN,NPC,DEC_DEN,PCA_Params] =
= wmulden('execute',DEC,PCA_Params)
[...]= wmulden('execute',DEC,PCA_Params,TPTR,SORH)
```

Рассмотрим первую функцию

```
[X_DEN,NPC,NESTCOV,DEC_DEN,PCA_Params,
DEN_Params]= ... wmulden(X,LEVEL,WNAME,NPC_APP,
NPC_FIN,TPTR,SORH).
```

Она возвращает очищенную от шума версию X_DEN из входной матрицы X. Стратегия комбинирует одномерное вейвлет-удаление шума компонент сигнала в базисе, где ковариационная матрица Σ

является диагональной с нецентрированным анализом главных компонент (PCA) на вейвлет-аппроксимациях или PCA на финальном этапе (после восстановления). Входная матрица X содержит P сигналов-столбцов длины N, где N > P.

Входные параметры:

- X,LEVEL,WNAME – не требуют пояснений;
- NPC_APP, NPC_FIN – определяют способ выбора главных компонент для вейвлет-аппроксимаций на уровне LEVEL и для заключительного PCA после вейвлет-реконструкции соответственно. Если NPC_APP (или NPC_FIN) является целым числом, то это число сохраненных главных компонент. Если NPC_APP или NPC_FIN = 'kais' (или 'heur'), то выбор числа сохраненных главных компонент производится автоматически по правилу Кайзера (или по эвристическому):
  - правило Кайзера оставляет компоненты, которые соответствуют собственным числам, большим среднего из всех собственных чисел;
  - эвристическое правило оставляет компоненты, которые соответствуют собственным числам, большим, чем 0.05 от суммы всех собственных чисел;
- TPTR – строка, определяющая выбор порога. Может принимать значения: 'rigsure', 'heursure', 'sqtwolog' (по умолчанию), 'minimaxi', 'penalhi', 'penalme', 'penallo';
- SORH – тип пороговой обработки. Если SORH = 's' (по умолчанию), то используется мягкая, а если 'h' – то жесткая пороговая обработка.

Выходные параметры:

- X_DEN – очищенная от шума матрица сигнала X;
- NPC – вектор чисел сохраненных главных компонент для аппроксимаций и для заключительного PCA;
- NESTCOV – ковариационная матрица Σ оценки шума;
- DEC_DEN – вейвлет-разложение X_DEN;
- PCA_Params – структура, такая что:
  - PCA_Params.NEST = {pc_NEST,var_NEST,NESTCOV};
  - PCA_Params.APP = {pc_APP,var_APP,npc_APP};
  - PCA_Params.FIN = {pc_FIN,var_FIN,npc_FIN};
- DEN_Params – структура, такая что:
  - DEN_Params.thrVAL – это вектор длины LEVEL, который содержит значения порогов для каждого уровня;

– DEN_Params.thrMETH – это строка, содержащая имя метода удаления шума (TPTR), например 'sqtwolog';
– DEN_Params.thrTYPE – это символ типа трешолдинга (SORH), то есть 's' или 'h'.

Остальные варианты использования wmulden понятны с уточнениями из Help для wmulden.

Пример 8. Загрузим многомерный тестовый сигнал ex4mwden.mat, выполняя следующие команды в строке MATLAB:

```
load ex4mwden
whos
```

Вторая команда показывает, что в рабочее пространство MATLAB загружаются следующие данные:

```
Name     Size     Bytes    Class

covar    4x4      128      double
x        1024x4   32768    double
x_orig   1024x4   32768    double
```

Итак, мы будем использовать массивы: x_orig – набор из 4 оригинальных сигналов, covar – ковариационная матрица Σ оригинальных сигналов x_orig и набор из 4 зашумленных сигналов x. Исследуемые сигналы x – это зашумленные версии простых комбинаций двух оригинальных сигналов. Первый есть «Blocks», который является достаточно нерегулярным, и второй – «HeavySine», который является регулярным всюду, за исключением двух точек. Другие два сигнала – сумма и разность двух первых оригинальных сигналов. Многомерный гауссов белый шум добавлен к результирующим четырем сигналам для получения исследуемых сигналов в x.

Отображаем (на экран) оригинальные и зашумленные сигналы (рис. 3.5.4):

```
kp = 0;
for i = 1:4
    subplot(4,2,kp+1), plot(x_orig(:,i));
    title(['Original signal ',num2str(i)])
    subplot(4,2,kp+2), plot(x(:,i));
    title(['Observed signal ',num2str(i)])
    kp = kp + 2;
end
```

Приступим к удалению шума простой многомерной пороговой обработкой с использованием анализа главных компонент (PCA). Сна-

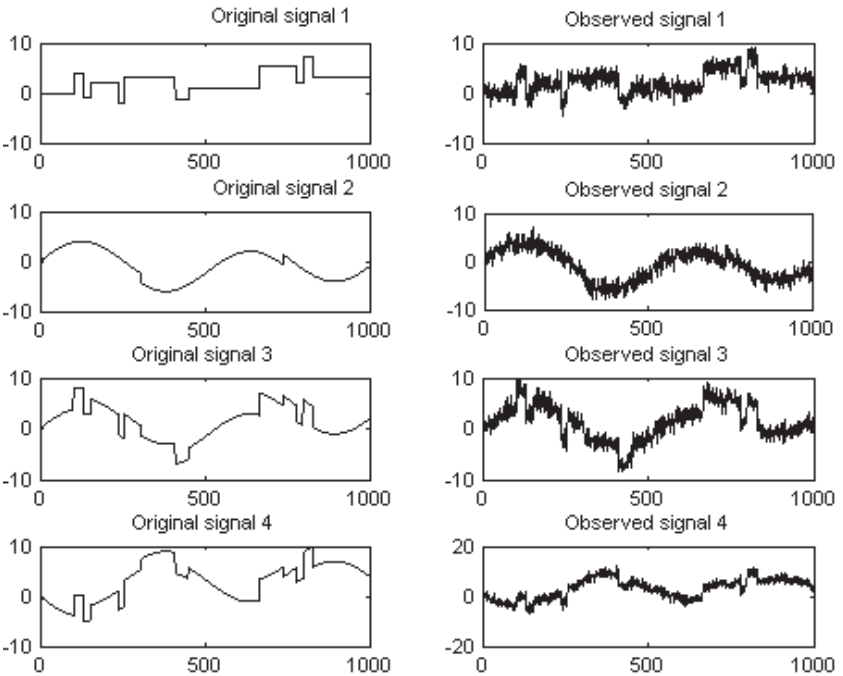


Рис. 3.5.4. Графики оригинальных и зашумленных сигналов

чала выполняем одномерное удаление шума. Установим параметры удаления шума:

```
level = 5;
wname = 'sym4';
tptr  = 'sqtwolog';
sorh  = 's';
```

Также установим параметры PCA, сохраняя все 4 главные компоненты:

```
npc_app = 4;
npc_fin = 4;
```

Наконец, выполняем многомерное удаление шума командой:

```
x_den_1 = wmulden(x, level, wname, npc_app, npc_fin, tptr, sorh);
```

В данном случае были оставлены все главные компоненты. Поэтому анализ главных компонент по существу не применялся. Но результаты достаточно хорошие (рис. 3.5.5, левый столбец). Результаты

можно улучшить за счет выбора главных компонент мультисигнала. Функция предусматривает автоматический выбор числа сохраняемых главных компонент по правилу Кайзера (который сохраняет компоненты, соответствующие собственным значениям, превышающим среднее значение из всех собственных значений). Для этого нужно положить

```
npc_app = 'kais';
npc_fin = 'kais';
```

и выполнить многомерное удаление шума снова:

```
[x_den_2,npc,nestcov]=wmulden(x,level,wname,npc_app,npc_fin, tptr, sorh);
```

В последней команде второй параметр вывода npc дает числа сохраненных главных компонент для PCA для приближений и конечного PCA. Отобразим число сохраненных главных компонент:

```
npc
npc =
    2     2
```

Метод сохраняет по две главные компоненты. Третий параметр вывода nestco содержит оценочную шумовую ковариационную матрицу.

Отобразим графики очищенных сигналов в первом и во втором случаях:

```
kp = 0;
for i = 1:4
    subplot(4,2,kp+1), plot(x_den_1(:,i));
    title(['De-noised signal ',num2str(i)])
    subplot(4,2,kp+2), plot(x_den_2(:,i));
    title(['De-noised signal, kais ',num2str(i)])
    kp = kp + 2;
end
```

Результат получается лучше (рис. 3.5.5, второй столбец), особенно для второго оригинального сигнала.

Многомасшабный анализ главных компонет

В предыдущем разделе анализ главных компонент применялся к многомерному сигналу *X* совместно с вейвлет-удалением шума. Однако при вейвлет-разложении сигнала *X* получаются новые многомерные сигналы, состоящие из матриц вейвлет-коэффициентов аппроксимации и детализации сигнала. Естественно применить анализ

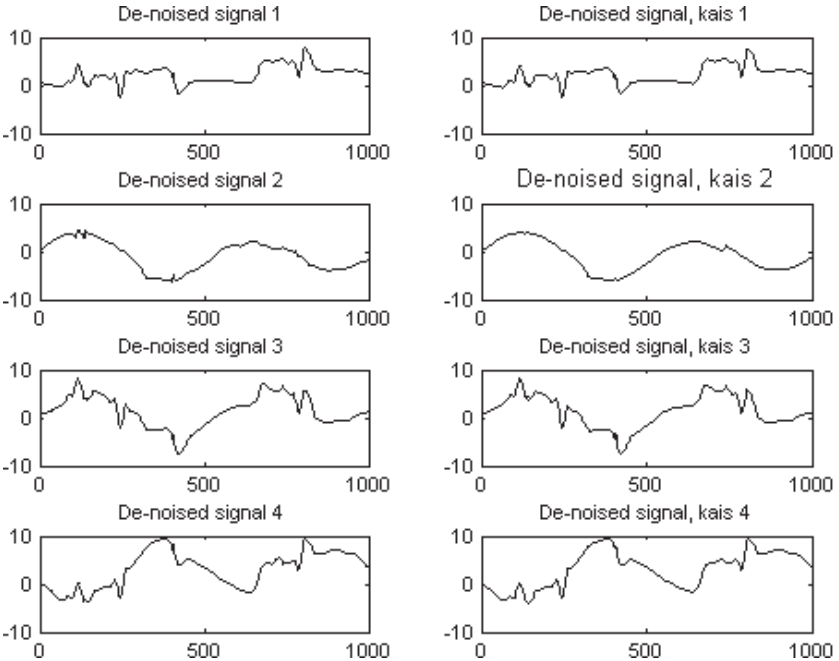


Рис. 3.5.5. Результаты двух вариантов удаления шума

главных компонент и для обработки этих новых многомерных сигналов с последующим вейвлет-восстановлением. Тогда сигнал упрощается за счет выбора главных компонент вейвлет-коэффициентов.

В этом разделе покажем особенности многомасштабного вейвлет-анализа многомерного сигнала с применением анализа *главных компонент* не только к самому сигналу, но и к вейвлет-коэффициентам и вейвлет-компонентам многомерного сигнала *X*. Для проведения такого анализа пакет Wavelet Toolbox содержит функцию **wmspca**. Эта функция может использоваться и в командной строке, и при помощи графического интерфейса пользователя wavemenu. Рассмотрим кратко работу функции wmspca. Подробности см. в документации Wavelet Toolbox.

16. Функция wmspca – многомасшабный анализ главных компонент.

Синтаксис:

```
[X_SIM,QUAL,NPC,DEC_SIM,PCA_Params] =
= wmspca(X,LEVEL,WNAME,NPC)
```

```
[...] = wmspca(X,LEVEL,WNAME,'mode',EXTMODE,NPC)
[...] = wmspca(DEC,NPC)
```

Рассмотрим первую функцию

```
[X_SIM,QUAL,NPC,DEC_SIM,PCA_Params] =
= wmspca(X,LEVEL,WNAME,NPC)
```

Она возвращает упрощенную версию X_SIM из входной матрицы мультисигнала X, полученную на основе многомасшабного анализа главных компонент.

Входные параметры:

- X, LEVEL,WNAME – не требуют пояснений. Входная матрица X содержит P сигналов-столбцов длины N (N > P);
- NPC – указывает число сохраняемых главных компонент. Он может быть одним числом – для всех уровней, или вектором – для указания числа главных компонент для коэффициентов каждого уровня разложения. Различные варианты параметра NPC:
  - если NPC является вектором, тогда он должен иметь длину LEVEL+2. Тогда для 1 <= d <= LEVEL, NPC(d) является числом сохраненных нецентрированных главных компонент детализирующих коэффициентов на уровне d. NPC(LEVEL+1) является числом сохраненных нецентрированных главных компонент для коэффициентов аппроксимации уровня LEVEL. NPC (LEVEL+2) является числом сохраненных главных компонент для финального PCA после вейвлет-реконструкции;
  - если NPC = 'kais' (соответственно, 'heur'), то число сохраненных основных компонентов отбирается автоматически, по правилу Кайзера (или эвристическому правилу). Правило Кайзера сохраняет компоненты, соответствующие собственным значениям, которые больше среднего из всех собственных значений. Эвристическое правило сохраняет компоненты, соответствующие собственным значениям, которые больше, чем 0.05 от суммы всех собственных значений;
  - если NPC = 'nodet', то все коэффициенты деталей «уничтожаются», а все приближения сохраняются.

Выходные параметры:

- X_SIM – упрощенная версия матрицы X;
- QUAL – вектор длины P, показывающий качество реконструкций столбцов через средние квадратичные отклонения в процентах;

- NPC – вектор чисел сохраненных главных компонент;
- DEC_SIM – структура вейвлет-разложения сигнала X_SIM;
- PCA_Params – массив структуры длины LEVEL+2, такой что:
  - PCA_Params (d).pc – матрица P на P главных компонент. Столбцы сохранены в порядке убывания дисперсий;
  - PCA_Params (d).variances – вектор дисперсий главных компонент;
  - PCA_Params(d).npc = NPC.

Остальные варианты использования wmspca понятны с уточнениями из Help для wmspca.

Пример 9. Загрузим те же 4 сигнала и установим параметры:

```
load ex4mwden
level = 5;
wname = 'sym4';
npc = 'kais';
```

Сделаем первый раз многомасшабный PCA.

```
[x_sim, qual, npc] = wmspca(x,level,wname,npc);
```

Можно отобразить графики зашумленных и упрощенных сигналов.

```
kp = 0;
for i = 1:4
    subplot(4,2,kp+1), plot(x (:,i));
    title(['Original signal ',num2str(i)])
    subplot(4,2,kp+2), plot(x_sim(:,i));
    title(['Simplified signal ',num2str(i)])
    kp = kp + 2;
end
```

Результаты удовлетворительны (рис. 3.5.6, левый столбец). Рассмотрим вектор npc числа оставленных главных компонент

```
npc
npc =
     1     1     1     1     1     2     2
```

Результаты могут быть улучшены подавлением шума, уничтожением полностью деталей на уровнях 1–3. Для этого достаточно взять нулевые значения числа главных компонент:

```
npc(1:3) = zeros(1,3);
```

Сделаем многомасшабный PCA с новыми параметрами.

```
[x_sim_1, qual_1, npc_1] = wmspca(x,level,wname,npc);
```

Отобразим графики упрощенных сигналов в первом и во втором случаях (рис. 3.5.6):

```
kp = 0;
for i = 1:4
    subplot(4,2,kp+1), plot(x_sim(:,i));
    title(['Simplified signal ',num2str(i)])
    subplot(4,2,kp+2), plot(x_sim_1(:,i));
    title(['Simplified signal ',num2str(i)])
    kp = kp + 2;
end
```

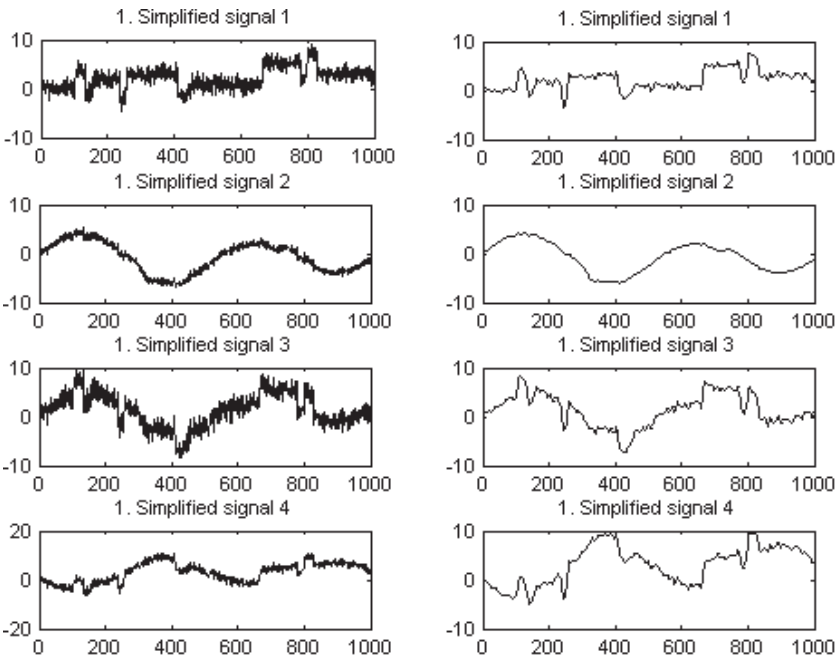


Рис. 3.5.6. Результаты двух вариантов применения PCA

3.5.3. Сжатие сигнала

Вейвлет-анализ эффективно используется для сжатия сигналов путем многоуровневого вейвлет-разложения и удаления части коэффициентов аппроксимации. Процедура сжатия вполне аналогична удалению шума в сигнале при использовании жесткой пороговой обработки коэффициентов.

Основные функции для сжатия сигналов:

- **wmpalg** – процесс подгонки (matching pursuit);
- **wmpdictionary** – словарь для процесса подгонки;
- **wcompress** – подлинное сжатие с использованием вейвлетов.

Более наглядный и эффективный инструментарий для сжатия сигнала дает главное меню wavemenu вейвлет-анализа.

Процесс подгонки (matching pursuit)

Две функции **wmpalg** и **wmpdictionary** позволяют для сигнала подобрать базис, в котором он разлагается с наименьшим числом коэффициентов, становится разреженным.

Иногда, для более точного представления сигнала $f(x)$, бывает удобно расширить базис пространства сигналов $L^2(\mathbf{R})$, то есть добавить еще некоторые элементы. Полученная система $\{\varphi_k(x)\}$ будет полной, но избыточной. Возможно, это будет фрейм. В данном разделе такую систему функций принято называть словарем, а отдельные элементы $\varphi_k(x)$ называются атомами словаря. Предполагается, что все они имеют единичную норму. Процедура подгонки заключается в определенной серии операций по нахождению главных составляющих сигнала. Она производится в несколько этапов:

1. Находим атом словаря $\varphi_p(x)$, для которого скалярное произведение с $f(x)$ будет максимальным по модулю.
2. Вычисляем скалярное произведение $(f,\varphi_p)_{L^2}$ и находим остаток $R^1(f) = f - (f,\varphi_p)_{L^2}\varphi_p$.
3. Применяем шаги 1 и 2 для получаемых остатков. На m-м шаге получаем остаток $R^{m+1}(f) = R^m(f) - (R^m(f),\varphi_k)_{L^2}\varphi_k$.
4. Заканчиваем процедуру подгонки, когда остаток $R^{m+1}(f)$ будет мал в определенном смысле.

Подбирая такое разложение сигнала, мы делаем сигнал более разреженным, то есть представленным меньшим числом коэффициентов.

Реально сигнал задается выборкой значений $Y = \{y_k\}$, $k = 1, ..., N$. Тогда мы в качестве пространства сигналов должны рассматривать $\mathbf{R}^N$. Возьмем словарь Φ в этом пространстве, состоящий из $M > N$ атомов. Они образуют полную, но избыточную систему нормальных векторов (столбцов) пространства $\mathbf{R}^N$. Будем рассматривать словарь Φ как матрицу M-на-N. Тогда скалярные произведения сигнала Y с атомами словаря находятся как произведение $\Phi^T \cdot Y$ транспонированной матрицы Φ^T на вектор-столбец Y. Пусть максимум модуля до-

стигается для p-го атома (столбца) φ_p. Тогда $R^1(Y) = Y - (\varphi_p, Y)\varphi_p$, где (φ_p, Y) – обычное скалярное произведение в $\mathbf{R}^N$.

На втором этапе, если мы выбрали еще один атом φ_q для остатка $R^1(Y)$, мы проектируем остаток $R^1(Y)$ на пространство, порожденное векторами φ_p и φ_q, и из остатка $R^1(Y)$ вычитаем эту проекцию: $R^2(Y) = R^1(Y) - P(R^1(Y))$. Остаток $R^2(Y)$ получается ортогональным атомам-векторам φ_p и φ_q. Образуем матрицу Ψ из векторов-столбцов атомов φ_p и φ_q. Тогда оператор проектирования P находится по общей формуле $P = \Psi(\Psi^T\Psi)^{-1}\Psi^T$. Дальнейшие шаги по нахождению $R^{m+1}(Y)$ совершенно понятны.

Подробнее об этом см. http://en.wikipedia.org/wiki/Matching_pursuit и имеющиеся там ссылки.

По умолчанию используется словарь, состоящий из следующих групп (подсловарей):

- **sym4-lev5** – комплект векторов, значений функций вейвлет-базиса симлетов порядка 4 до уровня масштабирования 5;
- **wp sym4-lev5** – комплект векторов, значений функций пакетного вейвлет-базиса симлетов порядка 4 до уровня масштабирования 5;
- **dct** – унитарное дискретное косинусное преобразование массива длины N. Ортонормированный базис состоит из элементов:

$$\varphi_k(n) = \begin{cases} \dfrac{1}{\sqrt{N}} & k = 0 \\ \sqrt{\dfrac{2}{N}} \cos\dfrac{\pi(2n+1)k}{2N} & k = 1,2,\ldots,N-1 \end{cases};$$

- **sin** – подсловарь синусов. Он состоит из следующих столбцов: $\varphi_k(t) = \sin(2\pi kt)$, $k = 1, 2, ..., [N/2]$, $0 \le t \le 1$, где t есть линейно распределенный N-точечный вектор;
- **cos** – подсловарь косинусов. Он состоит из следующих столбцов: $\varphi_k(t) = \cos(2\pi kt)$, $k = 1, 2, ..., [N/2]$, $0 \le t \le 1$, где t есть линейно распределенный N-точечный вектор;
- **poly** – полиномиальный подсловарь. Он состоит из следующих столбцов: $p_n(t) = t^{n-1}$, $n = 1, 2, ..., 20$, $0 \le t \le 1$, где t есть линейно распределенный N-точечный вектор;
- **RnIdent** – подсловарь смещенных символов Кронекера. Он состоит из следующих столбцов: $\varphi_k(n) = \delta(n - k)$, $k = 0, 1, ..., N$. Это просто стандартный базис пространства $\mathbf{R}^N$.

Рассмотрим функции, выполняющие процедуру подгонки.

1. Функция wmpdictionary – создание словаря для процесса подгонки.

Синтаксис:

```
MPDICT = wmpdictionary(N)
[MPDICT,NBVECT]= wmpdictionary(N)
[MPDICT,NBVECT]=wmpdictionary(N,Name,Value)
[MPDICT,NBVECT,LST]= wmpdictionary(N,Name,Value)
[MPDICT,NBVECT,LST,LONGS]= wmpdictionary(N,Name,Value)
```

Описание:

Функция MPDICT = wmpdictionary(N) создает словарь MPDICT в виде матрицы N на P для подсловарей по умолчанию {{'sym4',5},{'wpsym4',5},'dct','sin'}. Отметим, что число столбцов P зависит от N.

Функция [MPDICT,NBVECT] = wmpdictionary(N) возвращает еще столбец NBVECT, который содержит числа векторов в каждом подсловаре. Порядок элементов в NBVECT соответствует порядку подсловарей. Сумма элементов в NBVECT равна числу столбцов словаря MPDICT.

Функция [MPDICT,NBVECT] = wmpdictionary(N,Name,Value) создает словарь MPDICT, используя дополнительные параметры, указанные парой аргументов Name,Value.

Функция [MPDICT,NBVECT,LST] = wmpdictionary(N,Name, Value) возвращает еще массив ячеек LST с описаниями подсловарей.

Функция [MPDICT,NBVECT,LST,LONGS] = wmpdictionary (N,Name,Value) возвращает еще массив ячеек LONGS, содержащий числа векторов в каждом подсловаре. Массив LONGS полезен только в случае вейвлетных подсловарей. В этом случае элементы из LONGS дают числа масштабирующих функций на самом грубом уровне и количество вейвлет-функций по уровням.

Возможные значения пары аргументов Name,Value:

- 'addbeg' – подсловарь, вставляемый вперед. Это матрица N на M, где N – длина входного сигнала. При этом не проверяется, что эти M векторов-столбцов образуют базис. Если параметр **lstcpt** создания словаря не использовался, то подсловарь вставляется перед словарями по умолчанию {{'sym4',5},{'wpsym4',5}, 'dct','sin'}. Столбцы векторов в добавляемом словаре не обязаны иметь единичную норму;
- 'addend' – подсловарь, добавляемый в конец. Это матрица N на M, где N – длина входного сигнала. При этом не проверяется, что эти M векторов-столбцов образуют базис. Если параметр

lstcpt создания словаря не использовался, то подсловарь вставляется после словарей по умолчанию. Столбцы векторов в добавляемом словаре не обязаны иметь единичную норму;

- 'lstcpt' – массив ячеек для создания словаря из предлагаемых MATLAB функций. Кроме уже упоминаемых ранее значений по умолчанию {{'sym4',5}, {'wpsym4',5}, 'dct', 'sin'}, можно использовать имена подсловарей 'cos', 'RnIdent', 'poly'. Кроме того, можно использовать короткие имена подходящих ортогональных или биортогональных вейвлет-семейств, например {'sym4', 5}, или вейвлет-пакетов, например {'wpsym4',5}. Если не указать уровень или моду расширения, то будет считаться по умолчанию уровень 5, мода 'per'.

Примеры использования функции:

```
dict = {'db2','poly'}; %Определение словаря как массива ячеек
mpdict = wmpdictionary(1000,'LstCpt',dict); % Создание словаря

[mpdict,~,~,longs] = wmpdictionary(100,'lstcpt',{{'haar',2}});

load cuspamax; %Загрузка сигнала
dict = {{'sym4',2},'sym4','dct'}; %Определение словаря
mpdict = wmpdictionary(length(cuspamax),'LstCpt',dict); %Создание словаря
```

2. Функция wmpalg – процесс подгонки. Позволяет для сигнала Y подобрать базис, в котором он разлагается с наименьшим числом коэффициентов, становится разреженным.

Синтаксис:

YFIT = wmpalg(MPALG,Y,MPDICT)
[YFIT,R] = wmpalg(...)
[YFIT,R,COEFF] = wmpalg(...)
[YFIT,R,COEFF,IOPT] = wmpalg(...)
[YFIT,R,COEFF,IOPT,QUAL] = wmpalg(...)
[YFIT,R,COEFF,IOPT,QUAL,X] = wmpalg(...)
[YFIT,R,COEFF,IOPT,QUAL,X] = wmpalg(..., Name, Value)

Описание:

Функция YFIT = wmpalg(MPALG,Y,MPDICT) возвращает адаптивную жадную аппроксимацию входного сигнала Y в словаре MPDICT. Для этого используется алгоритм процесса подгонки, определенный параметром MPALG. Словарь состоит из полной, но избыточной системы векторов и строится обычно функцией wmpdictionary.

Жадные аппроксимации – это способ нелинейного приближения функций с помощью некоторой фиксированной полной системы функций (в частности, переполненной системы), при котором выбирается конечное число элементов приближения с наибольшими, в том или ином смысле, коэффициентами. Жадные аппроксимации активно изучались с середины 80-х годов XX века. Основную роль в становлении теории жадных аппроксимаций сыграли Дж. Фридман, В. Стузл, С. Маллат, Ж. Жанг, П. Хьюбер, Л. Джоунс, А. Бэррон, Р. ДеВор, В. Н. Темляков, Д. Донохо и др. Свое применение жадные аппроксимации нашли в теории обработки сигналов, передачи и хранения информации.

Функция [YFIT,R] = wmpalg(...) возвращает еще остаток R процедуры подгонки как разность между векторами Y и YFIT.

Функция [YFIT,R,COEFF] = wmpalg(...) возвращает еще коэффициенты разложения COEFF. Число этих коэффициентов зависит от числа итераций.

Функция [YFIT,R,COEFF,IOPT] = wmpalg(...) возвращает еще столбец индексов IOPT сохраняемых атомов.

Функция [YFIT,R,COEFF,IOPT,QUAL] = wmpalg(...) возвращает еще показатель QUAL качества приближения на каждом этапе итераций. QUAL – это отношение l2 квадрата нормы вектора COEFF коэффициентов разложения к l2 квадрату нормы входного сигнала.

В последней функции получается еще нормализованный словарь X. Он получается из столбцов исходного словаря MPDICT их нормировкой по l2-норме.

Методы MPALG алгоритма подгонки могут быть следующими:

- 'BMP' – базисный метод процесса подгонки;
- 'OMP' – ортогональный процесс подгонки;
- 'WMP' – слабоортогональный процесс подгонки.

Пример использования функции:

```
load cuspamax;
mpdict = wmpdictionary(length(cuspamax),'LstCpt',{{'wpsym4',2},'dct'});
[yfit,r,coeff,iopt,qual] = wmpalg('OMP',cuspamax,mpdict);
```

Полное сжатие

Используемая функция:

- **wcompress** – полное сжатие с использованием вейвлетов.

Этот тип сжатия сигнала использует вейвлет-разложение и пороговую обработку коэффициентов совместно с процедурой квантования и кодирования. Полная каскадная последовательность сжатия

включает фазы вейвлет-обработки, квантования, кодирования и раскодирования. На рис. 3.5.7 выделены обычное вейвлет-сжатие (левая часть) и полное сжатие с использованием дополнительных процедур. Отметим, что формат JPEG2000 также использует вейвлет-разложение совместно с процедурой квантования и кодирования коэффициентов. Напомним, что квантование (или округление) производится по следующей формуле:

$$q = sign(y)\left[\frac{|y|}{\Delta}\right]\Delta,$$

где y – исходное значение коэффициента, $sign(y)$ – знак коэффициента, Δ – значение шага округления и $[x]$ – целая часть числа. Мертвая зона квантователя – это интервал диапазоном 2Δ около нуля, она дает большее количество нулей на выходе. Идеи кодирования несравнимо сложнее, см. об этом на сайте http://rain.ifmo.ru/cat/view.php/.

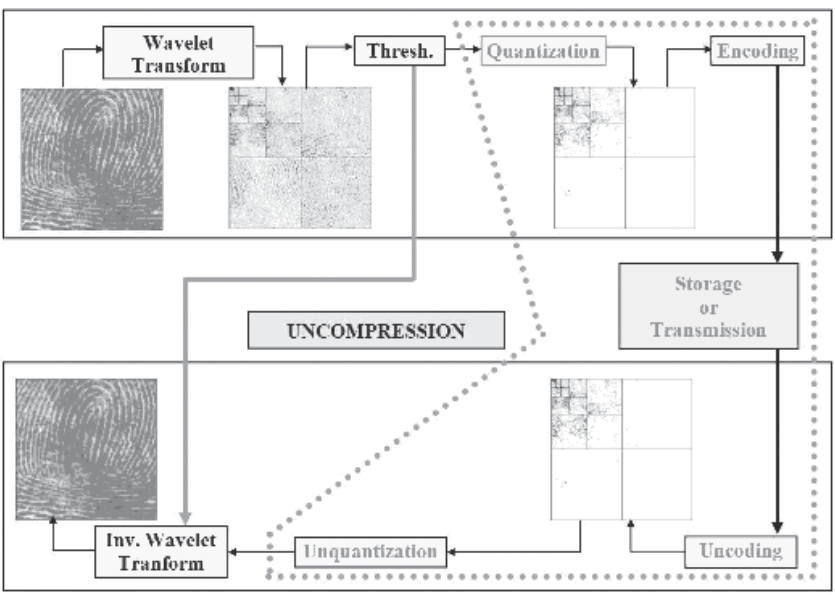


Рис. 3.5.7. Схема подлинного сжатия
(рисунок из документации MATLAB)

Методы сжатия разбиваются на две группы. Первую группу представляют методы пороговой обработки коэффициентов (СТМ-методы). В MATLAB предложены три метода, которые произво-

дятся каскадно, простыми шагами с выполнением пороговой обработки (глобальной или уровневой) и кодирования квантованием. Используется либо фиксированный, либо метод Хаффмана в зависимости от выбора метода. О кодировании Хаффмана см., напр., http://ru.wikipedia.org/wiki/Код Хаффмана и приведенные там источники.

Методы пороговой обработки коэффициентов (СТМ-методы):

- 'gbl_mmc_f' – глобальная пороговая обработка коэффициентов и фиксированное кодирование;
- 'gbl_mmc_h' – глобальная пороговая обработка коэффициентов и фиксированное кодирование Хаффмана;
- 'lvl_mmc' – субполосная пороговая обработка коэффициентов и фиксированное кодирование Хаффмана.

Доступны более сложные методы, которые комбинируют вейвлет-разложение и квантование. Это основной принцип прогрессивных методов. Прогрессивность позволяет во время декодирования получить изображение, разрешение которого постепенно увеличивается. Кроме того, возможно получить ряд степеней компрессии, основанных на длине сохраненного кода. Это сжатие обычно вызывает потери информации, но данный вид алгоритмов включает также сжатие без потерь. Такие методы основаны на нескольких идеях. Во-первых, можно использовать вейвлет-разложение, для того чтобы добиться разреженности (большого количества нулевых коэффициентов), и классические методы кодирования. Идея, имеющая решающее значение для использования вейвлетов в сжатии изображений, состоит в том, чтобы использовать фундаментально древовидную структуру вейвлет-разложения. Определенные коды, разработанные с 1993 до 2000 год, используют эту идею, в частности алгоритм кодирования EZW вложенного нуль-дерева, введенный Шапиро (Shapiro J. M. (1993). Embedded image coding using zerotrees of wavelet coefficients IEEETrans. – Signal Proc. – Vol. 41. – No. 12. – P. 3445–3462). Алгоритм EZW комбинирует пошаговую пороговую обработку и последовательное квантование, сосредоточиваясь на более эффективном способе кодирования коэффициентов изображения, чтобы минимизировать отношение компрессии. Две разновидности SPIHT и STW являются усовершенствованными версиями первоначального алгоритма EZW.

Полный обзор этих прогрессивных методов можно найти в работе: Walker J. S. Wavelet-Based Image Compression. – University of Wisconsin, Eau Claire, Wisconsin, USA, 1999.

Прогрессивные методы кодирования:

- 'ezw' – метод Шапиро вложенного нуль-дерева (Embedded Zero tree Wavelet);
- 'spiht', 'spiht_3d' – метод разбиения множества в иерархических деревьях (Set Partitioning In Hierarchical Trees);
- 'stw' – метод пространственно ориентированного вейвлет дерева (Spatial-orientation Tree Wavelet);
- 'wdr' – метод редукции вейвлет разности (Wavelet Difference Reduction);
- 'aswdr' – метод редукции адаптивно сканируемой вейвлет разности (Adaptively Scanned Wavelet Difference Reduction).

3. Функция wcompress – полное сжатие с использованием вейвлетов. В этой функции наряду с вейвлет-разложением и пороговой обработкой коэффициентов разложения используется дополнительная, последовательно, несколько раз, применяемая процедура квантования и кодирования. Полная каскадная последовательность сжатия включает фазы вейвлет-обработки, квантования, кодирования и раскодирования.

Синтаксис:

```
wcompress('c',X,SAV_FILENAME,COMP_METHOD)
wcompress(...,'ParName1',ParVal1,'ParName2',ParVal2,...)
[CR,BPP] = wcompress('c',...)
XC = wcompress('u',SAV_FILENAME)
XC = wcompress('u',SAV_FILENAME,'plot')
XC = wcompress('u',SAV_FILENAME,'step')
```

Рассмотрим первую функцию wcompress('c',X,SAV_FILENAME, COMP_METHOD).

Входные параметры:

- 'c' – символ, обозначающий дополнительную компрессию, если 'u' – то без дополнительных шагов кодирования и квантования, только вейвлет-обработка коэффициентов и восстановление;
- X – входной сигнал;
- SAV_FILENAME – имя файла сжатого изображения для его сохранения;
- COMP_METHOD – метод сжатия, он может быть двух типов:
  - метод прогрессивных значений коэффициентов (PCSM), может принимать значения: 'ezw', 'spiht', 'stw', 'wdr', 'aswdr' и 'spiht_3d';
  - метод трешолдинга коэффициентов: 'lvl_mmc', 'gbl_mmc_f' и 'gbl_mmc_h'.

Во втором варианте этой функции: [CR,BPP] = wcompress('c',...) – сохраняются еще и две характеристики сжатия сигнала:

- CR – коэффициент сжатия. Он означает, что сжатый образ хранится, используя только CR% от размера хранения начального образа;
- BPP – коэффициент Bit-Per-Pixel – это число битов, необходимых для хранения одного пикселя изображения. Для серых (grayscale) исходных изображений BPP равно 8, а для полноцветных начальное значение BPP равно 24 (RGB – это три матрицы).

Функция XC = wcompress('u',SAV_FILENAME) обрабатывает файл SAV_FILENAME без дополнительного квантования и кодирования и возвращает изображение XC.

Подробнее об этой функции см. Help Wavelet Toolbox.

3.6. Тестовые сигналы в MATLAB

В MATLAB имеется большой набор тестовых демонстрационных одномерных сигналов. Кроме того, имеется функция wnoise, генерирующая серию тестовых сигналов.

3.6.1. Одномерные тестовые сигналы

Тестовые демонстрационные сигналы являются файлами типа *.mat и загружаются командой load, после которой можно указать диапазон этого массива. Например, vonkoch.mat – фрактальный сигнал длины 8192, его фрагмент длины 510 загружается командой:

```
load vonkoch;
s= vonkoch(1:510);
plot(s);
```

Приведем имена нескольких наиболее популярных сигналов (графики и имена всех тестовых сигналов Wavelet Toolbox можно найти на сайте matlab.ru):

- **vonkoch** – фрактальная кривая Коха;
- **wcantor** – канторова лестница;
- **sinfract** – фрактал с синусоидальной огибающей;
- **leleccum** – демонстрационный сигнал с шумом;
- **nearbrk** – демонстрационный сигнал с шумом;
- **sumsin** – демонстрационный сигнал на основе синуса;
- **noisdopp** – допплерова кривая, напоминает x*sin(1/x) с шумом;
- **ndoppr1** – допплерова кривая с шумом;

- **noisbloc** – зашумленные прямоугольные импульсы;
- **whitnois** – белый шум;
- **wstep** – ступенчатый сигнал;
- **brkintri** – треугольник;
- **noispol** – полиномиальный сигнал;
- **wnoislop** – демонстрационный сигнал с шумом;
- **wntrsin** – суперпозиция зашумленной треугольной функции и синусоиды;
- **freqbrk** – время-частотное распределение двух гармоник;
- **noissin** – композиция синусоиды и аддитивного шума;
- **qdchirp** – осциллирующий сигнал (чирп);
- **sumlichr** – комбинация линейно развернутых синусоид;
- **sumsin** – сумма синусов.

3.6.2. Изображения

Тестовые изображения в MatLab являются файлами формата *.mat. Они представляют палитровые изображения, которые задаются двумя матрицами X и map, где map – цветовая карта, а значения элементов матрицы X есть номера строк в матрице map, в которых указаны интенсивности цветов пикселя, соответствующие элементам матрицы X. Получить изображение на экране можно командой imshow(X,map) либо image(X), предварительно применив к X команду wcodemat для псевдоцветного матричного масштабирования. Загрузка тестового изображения производится командой load, например load woman. В результате загружаются (в рабочее пространство MATLAB) массив X и цветовая карта map.

Укажем наиболее популярные двумерные тестовые изображения:

- **detail** – магический квадрат Дюрера;
- **woman** – девушка в платке (фрагмент);
- **detfingr** – отпечаток пальца;
- **tire** – колесо автомобиля;
- **chess** – шахматы;
- **wbarb** – девушка в платке;
- **facets** – многогранник;
- **geometry** – геометрическое изображение;
- **sinsin** – сигнал типа f(x,y) = sin(ax)sin(ay);
- **tartan** – клеенка;
- **mandel** – множество Мандельброта;
- **julia** – множество Жюлиа;
- **wifs** – лист папоротника;

- **noiswom** – девушка в платке (фрагмент с помехами);
- **noissi2d** – сигнал sin(ax)sin(ay) с помехами;
- **mandril** – обезьяна;
- **belmont1** – фотография 1;
- **belmont2** – фотография 2.

3.6.3. Генерирование сигналов

Рассмотрим функции, которые позволяют создать тестовый сигнал:

- **randn** – нормально распределенные случайные числа и массивы;
- **rand** – псевдослучайные значения из открытого интервала (0, 1);
- **wnoise** – создание тестовых сигналов;
- **imnoise** – наложение шума на изображение.

1. Функция randn – нормально распределенные случайные числа и массивы. Функция randn генерирует массив нормально распределенных случайных чисел с нулевым средним, дисперсией $\varsigma^2 = 1$ и стандартным отклонением $\varsigma = 1$. На самом деле генерируются не случайные, а так называемые псевдослучайные числа – они выглядят случайно, но вычисляются по вполне конкретной формуле. Узнать состояние генератора случайных чисел и изменить его параметры можно при помощи функции **rng**.

Синтаксис:

```
r = randn(n)
r = randn(m,n), r = randn([m,n])
r = randn(m,n,p,...), r = randn([m,n,p,...])
r = randn
r = randn(size(A))
```

Функция Y = randn(n) генерирует матрицу n на n случайных элементов. Другие варианты:

- r = rand(m,n) или r = rand([m,n]) возвращает матрицу m на n;
- r = rand(m,n,p,...) или r = rand([m,n,p,...]) возвращает массив размера m на n на p на ...;
- r = rand возвращает случайное число;
- r = rand(size(A)) возвращает массив того же размера, что и A.

Пример 1.

```
R = randn(3,4).  %Случайная матрица размера 3x4.
```

Пример 2. Формула x = .6 + sqrt(0.1) * randn(5) создает случайный массив 5×5, у которого среднее значение равно 0.6, дисперсия $\varsigma^2 = 0.1$, а стандартное отклонение ς = sqrt(0.1).

Пример 3.

```
y = sin(x)+sqrt(0.1) * randn(size(x)); % зашумленный синус.
```

2. Функция rand – псевдослучайные значения из открытого интервала (0, 1). Функция генерирует случайные числа из интервала (0, 1) стандартного равномерного распределения. На самом деле генерируются не случайные, а так называемые псевдослучайные числа – они выглядят случайно, но вычисляются по вполне конкретной формуле. Узнать состояние генератора случайных чисел и изменить его параметры можно при помощи функции **rng**.

Синтаксис:

r = rand(n)
r = rand(m,n), r = rand([m,n])
r = rand(m,n,p,...), r = rand([m,n,p,...])
r = rand
r = rand(size(A))

Функция r = rand(n) возвращает матрицу n на n, содержащую псевдослучайные значения из открытого интервала (0, 1) стандартного равномерного распределения. Другие варианты такие же, что для команды randn.

3. Функция wnoise – создание тестовых сигналов. Функция генерирует тестовые сигналы, в том числе и с шумом.

Синтаксис:

X = wnoise(FUN,N),
[X,XN] = wnoise(FUN,N,SQRT_SNR).

Функция X = wnoise(FUN,N) вычисляет «чистые» значения тестового сигнала, определенного параметром FUN (возможные значения FUN указаны ниже), на сетке из 2^N значений промежутка [0, 1].

Функция [X,XN] = wnoise(FUN,N,SQRT_SNR) вычисляет тестовый вектор X, как и выше, перемасштабированный так, чтобы стандартное отклонение std(X) сигнала X принимало бы значение, заданное числом SQRT_SNR.

Вектор XN получается из вектора X добавлением гауссова белого шума N(0,1). Тогда XN имеет отношение сигнала к шуму SNR = = $(SQRT\_SNR)^2$.

Функция [X,XN] = wnoise(FUN,N,SQRT_SNR,INIT) генерирует векторы X и XN, как и выше, но начальное значение генератора установлено значением INIT.

Следующие шесть функций введены Donoho и Johnstone. Их графики приведены ниже.

FUN = 1 или 'blocks', FUN = 2 или 'bumps',
FUN = 3 или 'heavy sine', FUN = 4 или 'doppler',
FUN = 5 или 'quadchirp', FUN = 6 или 'mishmash'.

Пример 4. Построение графиков тестовых сигналов.

```
ind = linspace(0,1,2^10);
for i = 1:6  x = wnoise(i,10);
    subplot(6,1,i), plot(ind,x)
end
```

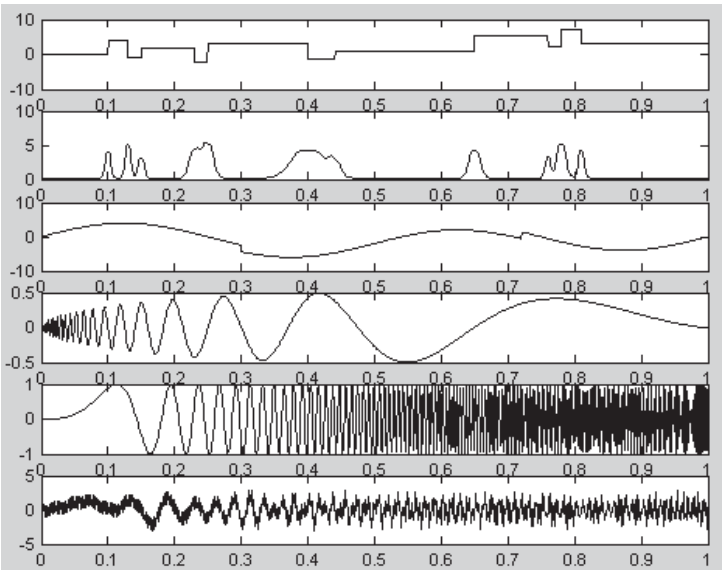


Рис. 3.6.1. Тестовые сигналы Donoho и Johnstone

3. Функция imnoise – наложение шума на изображение. Функция добавляет шум к изображению. Исходное изображение I может быть не только полутоновым, но и цветным. Входное изображение I должно быть класса uint8, uint16 или double. Выходное изображение J получается того же класса.

Синтаксис:

J = imnoise(I,type)

J = imnoise(I,type,parameters)
J = imnoise(I,'gaussian',m,v)
J = imnoise(I,'localvar',V)
J = imnoise(I,'localvar',image_intensity,var)
J = imnoise(I,'poisson')
J = imnoise(I,'salt & pepper',d)
J = imnoise(I,'speckle',v)

Тип добавляемого шума (type) может быть следующий:

- 'gaussian' – гауссов белый шум;
- 'localvar' – гауссов белый шум с нулевым средним и зависящей от интенсивности дисперсией;
- 'poisson' – пуассонов шум;
- 'salt & pepper' – соль и перец;
- 'speckle' – веснушки, мультипликативный шум.

Если функция применяется в виде J = imnoise(I,type,parameters), то задаются дополнительные параметры. Если параметры опущены, то по умолчанию.

В случае J = imnoise(I,'gaussian',m,v) добавляется гауссов белый шум с математическим ожиданием m и дисперсией v. По умолчанию m = 0, v = 0.01.

В случае J = imnoise(I,'localvar',V) добавляется гауссов белый шум с локальной дисперсией V. При этом V имеет тот же размер, что и I.

В случае J = imnoise(I,'localvar',image_intensity,var) добавляется гауссов белый шум с нулевым средним с локальной дисперсией var, зависящей от интенсивности I.

В случае imnoise(I,'poisson') производится шум Пуассона из данных, а не добавляется искусственный шум к данным.

В случае J = imnoise(I,'salt & pepper',d) добавляется «соль и перец», d – плотность шума. По умолчанию d = 0.05.

В случае J = imnoise(I,'speckle',v) добавляется мультипликативный шум согласно уравнению J = I + n*I, где n – равномерно распределенный случайный шум со средним значением m = 0 и отклонением v. По умолчанию v = 0.04.

Пример 5. Загружаем изображение eight.tif (из MATLAB) и добавляем «соль и перец» с плотностью d = 0.02.

```
I = imread('eight.tif');
J = imnoise(I,'salt & pepper',0.02);
subplot(221), imshow(I); title('Original image')
subplot(222), imshow(J); title('Noised image')
```

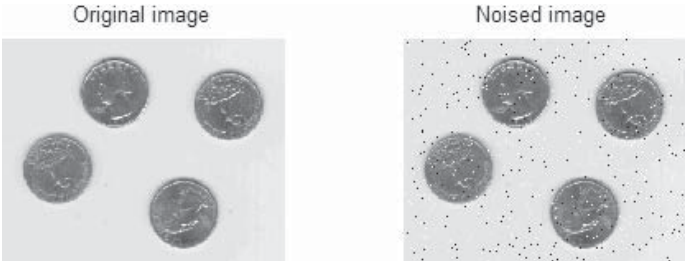


Рис. 3.6.2. Добавление шума в тестовый сигнал

3.7. Вейвлет-анализ кардиосигнала

В данном параграфе мы используем вейвлет-функции MATLAB для анализа кардиосигнала.

Имеется обширная литература, посвященная методам анализа зубцов кардиосигнала, длительности комплексов зубцов, вариабельности длин различных интервалов кардиосигнала. Традиционно используются статистические методы. Сильным инструментом для изучения кардиосигнала являются также разложение в ряд Фурье и анализ спектра. Вейвлет-анализ позволяет получить новые методы для изучения кардиосигналов. В работах [IM], [Is] и [MF] продемонстрированы методы непрерывного вейвлет-преобразования для изучения кардиосигналов. Числовые характеристики высокочастотных компонент кардиосигнала получены в работах [По3], [По4], [По5], [ППС], [СЦ1] и [СЦ2]. Из последних работ в этом направлении отметим [К].

В предыдущих разделах книги показано, что методы вейвлет-анализа могут успешно использоваться для сглаживания и удаления шума кардиосигнала. Кардиосигнал, очищенный от шума, выглядит более наглядно, и его объем составляет от 10% до 5% от исходного сигнала, что в большой степени решает проблему хранения кардиозаписей. В работе [ИЧЩ] вейвлет-анализ использовался для определения характерных всплесков кардиосигнала. В последних выпусках MATLAB Wavelet Toolbox также разработаны новые вейвлет-методы (New Wavelet for CWT), которые могут использоваться для нахождения характерных волн кардиосигнала.

Считается, что верхняя граничная частота нормального (без нагрузки) кардиосигнала, заметно влияющая на ее форму, не превышает 100 Гц. Поэтому при контурном анализе ЭКГ частоты выше 100 Гц практически не учитываются. Преимущество вейвлет-анализа

заключается еще и в том, что имеется возможность для изучения высокочастотных компонент кардиосигнала. При исследовании кардиосигнала традиционно анализируются форма зубцов, длительность комплексов зубцов, вариабельность длин различных интервалов кардиосигнала – это все низкочастотные характеристики. Более того, для сглаживания кардиограммы высокочастотные составляющие обычно удаляют при помощи различных фильтров. Ясно, что при этом теряется часть информации, зарегистрированная кардиографом. Природа возникновения высоких частот кардиосигнала до конца не выяснена. Некоторое влияние оказывает фоновый (сетевой) шум и электронный шум усилителей электрокардиографа. Причиной высокочастотных шумов могут быть электрическая активность тканей, через которые проводится импульс (например, скелетные мышцы), сопротивление тканей, особенно кожи, а также сопротивление на входе усилителя. Мы полагаем, что физиологические высокочастотные шумы являются в большой степени следствием электрической активности сердца, поскольку они регистрируются датчиками, расположенными вблизи сердца. В современных технических средствах при соответствующем заземлении аппаратные шумы практически незначительны, по сравнению с физиологическими шумами. Для регистрации высоких частот в настоящее время используются электрокардиографы высокого разрешения с частотой дискретизации 5, 10 и 20 кГц. Эффективное выделение высокочастотных компонент возможно с использованием вейвлет-преобразования сигнала. В настоящее время имеются работы, в которых высокие частоты кардиосигнала анализируются при помощи непрерывного вейвлет-преобразования. Так, например, в работах [IM], [Is] и [MF] рассматривались оцифрованные кардиосигналы с частотой дискретизации 5, 10, и 20 кГц и изучались свойства кардиосигнала на частотах от 25 Гц до 400 Гц.

В данном разделе мы рассмотрим применения вейвлет-анализа в среде MATLAB для сглаживания, сжатия и изучения высокочастотных компонент кардиосигнала. Для анализа используются оцифрованные кардиосигналы, снятые на кардиографе высокого разрешения (1028 отсчетов в секунду) «Кардиотехника-4000, EcgShell». Кардиосигнал регистрируется по 8 общепринятым каналам: L – левая рука (+) и правая рука (–), F – левая нога (+) и правая рука (–) и шесть грудных отведений, обозначаемых $C1$–$C6$.

Из данных 8 каналов L, F, $C1$, $C2$, $C3$, $C4$, $C5$, $C6$ кардиографа образуются 12 так называемых стандартных отведений: I, II, III, aVR, aVL, aVF, V1, V2, V3, V4, V5, V6 – по формулам:

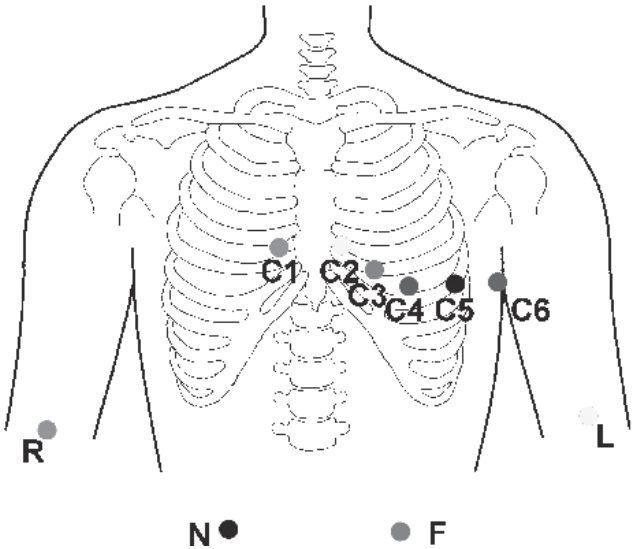


Рис. 3.7.1. Восемь каналов регистрации ЭКГ

I = L, II = F, III = $F - L$,
aVR = $-(L + F)/2$, aVL = $L - F/2$, aVF = $F - L/2$,
Vi = Ci $- (L + F)/3$, $i = 1, 2, ..., 6$.

Обоснование определения таких отведений дается в любом учебнике по кардиологии, см., напр., [В], [ПК].

3.7.1. Многоуровневый анализ кардиосигнала

Рассмотрим для примера кардиосигнал (второе грудное отведение V2), записанный на кардиографе высокого разрешения с частотой дискретизации 1024 отсчета в секунду. Для вейвлет-разложения применим вейвлет Добеши db4, имеющий носитель на промежутке [0, 7].

Загружаем кардиосигнал (автора данной книги) sm.txt. Его длина составляет более 70 000 отсчетов. Выбираем фрагмент этого сигнала длиной 4096. Проведем разложение сигнала до уровня N = 3 и построим графики сигнала и вейвлет-коэффициентов (рис. 3.7.2 и 3.7.3).

```
load sm.txt;
SV2=sm(601:4696); %Фрагмент сигнала
w='db4'; Fr = centfrq(w) % выбор вейвлета, определение его центральной
частоты
[c,l] = wavedec(SV2,3,w); %Разложение до уровня 3
```

```
%cA3=appcoef(c,l,w,3); %Выбор аппрксимирующих коэффициентов
[cD1,cD2,cD3]=detcoef(c,l,[1,2,3]); %Выбор детализирующих коэффициентов
figure(1)
plot(SV2);title('ECG V2')
figure(2)
subplot 311; plot(cD3);title('Detail coefs. lev. 3')
subplot 312; plot(cD2);title('Detail coefs. lev. 2')
subplot 313; plot(cD1);title('Detail coefs. lev. 1')
```

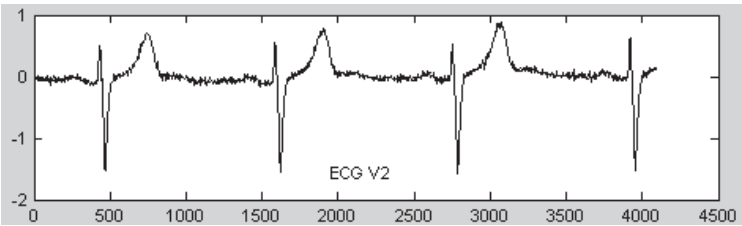


Рис. 3.7.2. График кардиосигнала sm

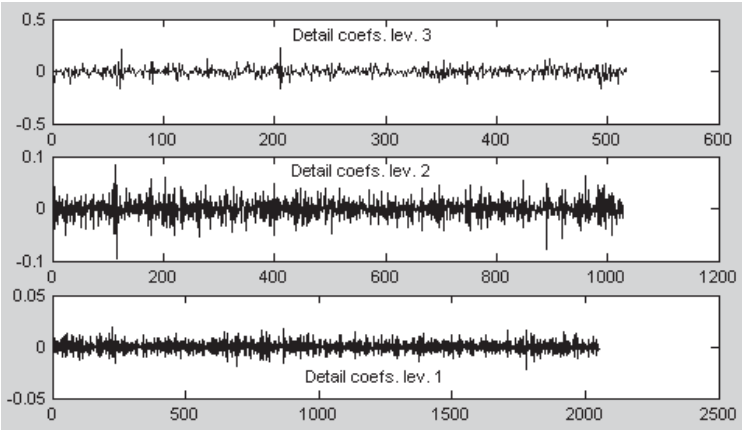


Рис. 3.7.3. Графики детализирующих коэффициентов sm

Вейвлет $\psi(t)$ Добеши db4 имеет носитель на промежутке [0, 7] и центральную частоту Fr = 0.7143 Гц. Поскольку $\Delta t = 1/1024$, то носитель вейвлета $\tilde{\psi}(t) = \psi(t/\Delta t)$ находится на промежутке $[0, 7\Delta t]$ и центральная частота вейвлета $\tilde{\psi}(t)$, используемого для первого уровня разложения, равна Fr1 = 0.7143*1024 = 734.30 Гц. Для второго уровня разрешения частота вейвлета будет в два раза меньше, Fr2 = 367.15 Гц,

а для третьего уровня разрешения Fr3 = 183.57 Гц. Вейвлет-коэффициенты cD1, cD2, cD3 отражают характеристики кардиосигнала на указанных частотах.

Больший интерес представляют компоненты сигнала, которые соответствуют найденным вейвлет-коэффициентам cD1, cD2, cD3. Это будут компоненты сигнала на частотах 183.57 Гц, 367.15 Гц и 734.30 Гц. Отнесение данных компонент к указанным частотам достаточно условно и не соответствует аналогичному понятию для разложения Фурье. В частности, частота 734.30 Гц, очевидно, выше частоты Найквиста, равной 512 Гц. Однако в дальнейшем мы увидим, что спектр частот компонент сигнала достаточно хорошо локализован (рис. 3.7.7). Для нахождения компонент сигнала сделаем прямое восстановление отдельно по каждому набору детализирующих коэффициентов (рис. 3.7.4).

```
figure(3)
ScD1=upcoef('d',cD1,w,1);
ScD2=upcoef('d',cD2,w,2);
ScD3=upcoef('d',cD3,w,3);
subplot 311; plot(ScD3);title('ECG Frec. 183 Hz')
subplot 312; plot(ScD2);title('ECG Frec. 367 Hz')
subplot 313; plot(ScD1);title('ECG Frec. 734 Hz')
```

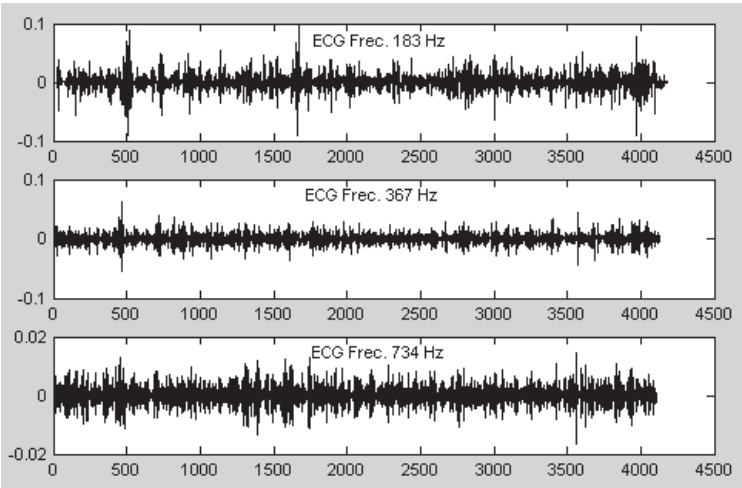


Рис. 3.7.4. Восстановленный сигнал
по детализирующим коэффициентам

Для сравнения сделаем такой же анализ кардиосигнала (второе грудное отведение V2) пациента, перенесшего месяц назад инфаркт миокарда и находящегося на лечении в санатории (рис. 3.7.5 и 3.7.6). Обе кардиограммы сняты на одном и том же кардиографе в одно и то же время, сначала – уже рассмотренный пациент (обозначим его символом Sm), затем сразу – другой пациент (обозначим его символом Th).

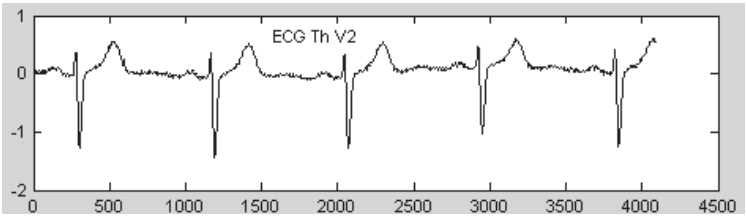


Рис. 3.7.5. График кардиосигнала отведения V2 пациента Th

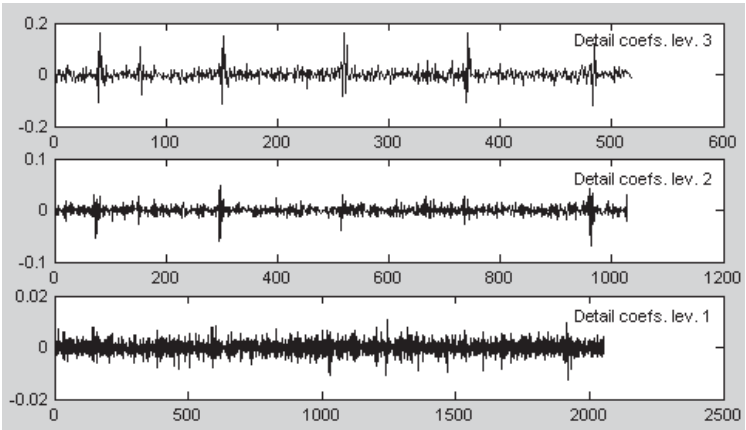


Рис. 3.7.6. Графики детализирующих коэффициентов пациента Th

Имеется визуальная разница графиков детализирующих коэффициентов двух пациентов. Однако хотелось бы выразить эту разницу в числах. Для этого можно использовать, по крайней мере, четыре способа:

1. Статистические характеристики.
2. Энтропия.
3. Энергетический спектр.
4. Стохастические характеристики.

1. Статистические характеристики

Для использования статистического метода мы считаем, что кардиосигнал является случайной величиной X, представленной в виде конечной выборки $x_1, ..., x_N$. Другие свойства: частота сердечных сокращений, длительности комплексов QRS, или длительности интервалов между зубцами, – также можно считать случайными величинами, функциями от X. Для анализа этих величин можно использовать оценки:

- выборочное среднее $m_X = \sum_{k=1}^{N} x_k / N$;
- выборочная (несмещенная) дисперсия $D_X = \sum_{k=1}^{N} (x_k - m_X)^2 / (N-1)$;
- выборочное стандартное отклонение $s_X = \sqrt{D_X}$;
- автокорреляционная функция $K_X(n) = \sum_{k=1}^{N-n} x_k x_{k+n} / (N-n+1)$.

Сравнивая данные характеристики для разных кардиосигналов или других величин, делают вывод о наличии патологических явлений.

Рассмотрим статистические характеристики как вейвлет-коэффициентов, так и компонентов сигнала, восстановленных по этим коэффициентам. Очевидно (и это подтверждается вычислениями), что среднее значение детализирующих коэффициентов и компонент равно нулю. Найдем среднее квадратичное отклонение (команда std MATLAB) полученных выше вейвлет-коэффициентов cD1, cD2, cD3 и компонент сигнала ScD1, ScD2, ScD3 пациентов Sm и Th для сравнения.

```
Sd1=std(cD1) Sd2=std(cD2) Sd3=std(cD3)
Sd1S=std(ScD1) Sd2S=std(ScD2) Sd3S=std(ScD3)
```

Результаты вычислений приведены в таблице.

Таблица 3.7.1. Статистические характеристики вейвлет-коэффициентов

Пациенты	Вейвлет-коэффициенты			Компоненты сигнала		
	cD1	cD2	cD3	ScD1	ScD1	ScD1
Пациент Sm	0.0047	0.0186	0.0448	0.0033	0.0093	0.0168
Пациент Th	0.0025	0.0092	0.0281	0.0018	0.0046	0.0099
Отношение Sm/Th	1.88	2.02	1.59	1.83	2.02	1.60

Мы видим, что стандартные отклонения вейвлет-коэффициентов двух различных пациентов различаются почти в два раза. Поэтому они имеют диагностическую значимость.

2. Энтропия

Можно вычислить также энтропию вейвлет-коэффициентов в каждом узле разложения и для каждого отведения. Есть несколько концепций энтропии. Например, энтропия *Шеннона* ('shannon') нормированного сигнала $s = \{s_i\}$ определяется формулой:

$$E1(s) = -\sum_i s_i^2 \log(s_i^2).$$

Она характеризует разброс значений коэффициентов. Энтропия Шеннона принимает максимальное значение, когда все значения s_i одинаковы, и минимальное значение – когда имеется всего один ненулевой элемент сигнала $s = \{s_i\}$.

Напомним другие критерии энтропии в MATLAB:

1. *Норма* пространства l^p, $p \geq 1$ ('norm')

   $$E2(s) = \sum_i |s_i|^p = \|s\|_p^p .$$

2. Энтропия *логарифм энергии* ('log energy')

   $$E3(s) = \sum_i \log(s_i^2),$$

   при следующем соглашении: $\log(0) = 0$.
3. *Пороговая* энтропия ('threshold') $E4(s_i) = 1$ если $|s_i| > \varepsilon$ и 0 – в противнм случае, тогда $E4(s)$ есть число элементов сигнала, величина которых больше порогового значения ε.

Описание функции wentropy вычисления энтропии сигнала в MATLAB см. в разделе 3.4.

Покажем использование функции wentropy на примере. Возьмем кардиосигналы двух пациентов, условно обозначенных Sm и Th, сделаем пакетное преобразование до 3-го уровня с вейвлетом db4 и вычислим значения энтропии Шеннона во всех терминальных узлах дерева, кроме первого узла (3, 0), который содержит аппроксимацию сигнала.

```
load SmCh4.txt; SV2=SmCh4(601:4696);
X=SV2; w='db4'; N=3; P='shannon';
T = wpdec(X,N,w);
for k=1:(2^N-1);
    XI=wpcoef(T,[N,k]);
    E(k)=wenropy(XI,P);
end
E
```

Результаты представлены в следующей табл. 3.7.2.

Таблица 3.7.2. Значения энтропии пакетных вейвлет-коэффициентов

Пациенты	Значения энтропии в терминальных узлах						
Индексы узлов	(3,1)	(3,2)	(3,3)	(3,4)	(3,5)	(3,6)	(3,7)
Пациент Sm	5.3781	0.5496	1.9093	0.0084	0.0446	0.2365	0.1373
Пациент Th	2.2253	0.1811	0.5132	0.0048	0.0261	0.0731	0.0400
Отношение Sm/Th	2.4	3.0	3.7	1.7	1.7	3.2	3.4

Разница значений энтропии пакетных вейвлет-коэффициентов пациентов Sm и Th еще более значительна, чем для статистических характеристик в табл. 3.7.1. Поэтому она также может иметь диагностическую значимость.

При желании можно использовать другой вейвлет, другой тип энтропии или рассматривать пакетные разложения другого уровня N.

3. Энергетический спектр

Второй естественный способ получения количественных характеристик сигналов cD1, cD2, cD3 и ScD1, ScD2, ScD3 состоит в том, чтобы исследовать *энергетический спектр* их преобразований Фурье. Напомним, что дискретное преобразование Фурье сигнала $\{x_n\}$ длины N производится по формуле

$$y_k = \sum_{n=0}^{N-1} x_n \, e^{-i\frac{2\pi}{N}k\,n}, \quad k = 0, ..., N-1.$$

Полученный сигнал $\{y_k\}$ отражает частотные свойства массива $\{x_n\}$, поэтому его иногда называют спектром сигнала $\{x_n\}$. При этом если числа $\{x_n\}$ вещественные, то массив $\{y_k\}$ имеет симметрию относительно середины $N/2$ (рис. 3.7.7):

$$y_{(N/2)+1} = \overline{y_{(N/2)-1}}, \quad y_{(N/2)+2} = \overline{y_{(N/2)-2}}, \ldots, \quad y_N = \overline{y_0} = \sum_{n=0}^{N-1} x_n.$$

Первые значения $y_0, y_1, ..., y_{N/2}$ содержат полную информацию относительно всего массива $\{y_k\}$. Поскольку значения $\{y_k\}$ могут быть комплексными, представляет интерес так называемый спектр мощности частот, вычисляемый по формуле

$$P_k = \frac{|y_k|^2}{N}, \quad k = 0, 1, \ldots, N-1.$$

При спектральном анализе всего кардиосигнала результаты получаются не вполне наглядными, поскольку мощность низкочастотных элементов кардиосигнала несравнимо больше мощности высокочастотных компонент. Представляет интерес провести спектральный

анализ высокочастотных компонент отдельно. При этом правильнее использовать не сами вейвлет-коэффициенты, а компоненты сигналов (то есть восстановленные по одной ветви коэффициентов). Например, вычисляем спектр компонент сигнала ScD1, ScD2, ScD3 пациентов Sm и Th для сравнения (рис. 3.7.7 и 3.7.8).

```
N1=length(ScD1) % находим длину сигнала
YS1=fft(ScD1);   % выполняем преобразование Фурье
PS1=YS1.*conj(YS1)/N1;   вычисляем энергетический спектр
M1=length(TScD1)
YT1=fft(TScD1);
PT1=YT1.*conj(YT1)/M1;
figure(1)
subplot 211; plot(PS1);title('ScD1 for Sm')
subplot 212; plot(PT1);title('ScD1 for Th')
```

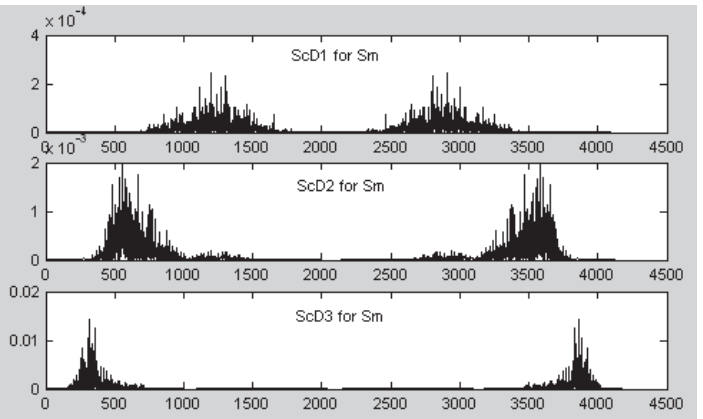


Рис. 3.7.7. Энергетический спектр компонент сигнала пациента Sm

Как и ожидалось, энергетические спектры компонент сигналов на разных частотах достаточно хорошо локализованы. Видна существенная разница спектров двух пациентов, особенно по самой высокочастотной компоненте, соответствующей первому уровню разложения – ScD1 и TcD1.

4. Стохастические характеристики

Как мы видели на рис. 3.7.3, 3.7.4, 3.7.6, поведение вейвлет-коэффициентов и компонент сигналов является достаточно сложным,

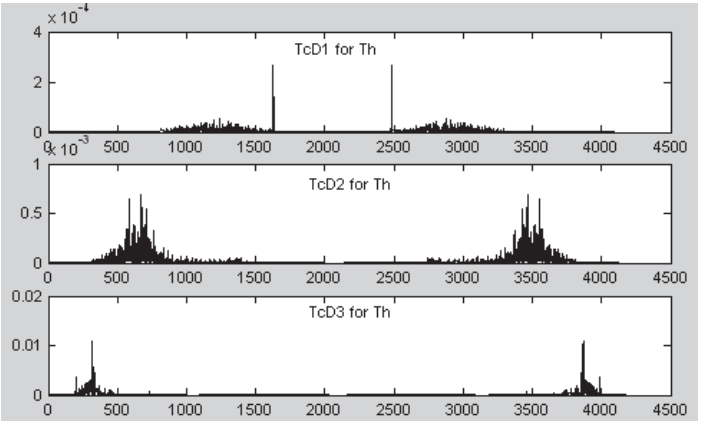


Рис. 3.7.8. Энергетический спектр компонент сигнала пациента Th

имеет характер хаотичности. Для анализа таких сигналов в настоящее время разработаны математические методы. Во-первых, можно оценить степень хаотичности параметром Херста. Далее по имеющимся одномерным данным можно построить динамическую систему в многомерном фазовом пространстве, для которой наблюдаемая переменная будет одной из координат, а траектории системы лежат на некотором множестве, имеющем фрактальную структуру и дробную размерность. Поэтому для оценки хаотичности вейвлет-коэффициентов можно использовать такие характеристики, как:

- показатель Херста;
- размерность фазового пространства;
- фрактальная размерность;
- корреляционная размерность.

Показатель Херста (индекс фрактальности) H достаточно известен (см. также раздел 3.3.1) в анализе временных рядов. Он характеризует степень хаотичности значений ряда. Коэффициент Херста характеризует вероятность того, что два соседних отсчета могут быть одинаковыми. Известно также, что показатель Херста представляет собой меру персистентности, то есть склонности процесса к трендам (в отличие от обычного броуновского движения). Значение $H > 0.5$ говорит о том, что направленная в определенную сторону динамика процесса в прошлом, вероятнее всего, повлечет продолжение движения в том же направлении. Если $H < 0.5$, то прогнозируется, что

процесс изменит направленность, случай $H = 0.5$ означает неопределенность.

Вычисление показателя Херста для сигнала $\{x_n\}$ обычно производится на основе так называемого *RS-анализа*. В MATLAB имеется функция wfbmesti, которая производит оценку индекса фрактальности Херста H сигнала. В работах [ДК1] и [ДК2] предложены методы оценки параметра Херста для временных рядов с трендом методом вейвлет-преобразования.

Напомним классический RS-метод для нахождения показателя Херста. Пусть дан сигнал $\{x_n\}$ длины N. Тогда коэффициент Херста H находится из соотношения

$$R/S = (N/2)^H,$$

S – стандартное отклонение:

$$S = \sqrt{\frac{1}{N}\sum_{k=1}^{N}(x_k - m_X)^2}\text{, с выборочным средним } m_X = \sum_{k=1}^{N} x_k / N,$$

а R – так называемый размах, накопленное отклонение от среднего:

$$R = \max_{1\le n\le N}\sum_{k=1}^{n}(x_k - m_X) - \min_{1\le n\le N}\sum_{k=1}^{n}(x_k - m_X).$$

Для бесконечных сигналов определение выборочного показателя Херста $H_N(n)$ по выборке длины N на шаге n можно найти, например, в [ККМОП].

Размерность фазового пространства есть минимальная размерность пространства, в которое можно вложить наш ряд значений в виде динамической системы. Напомним кратко основные конструкции, связанные с этим понятием. Рассмотрим динамическую систему $\dot{X} = F(X)$ в многомерном пространстве. Например, систему уравнений Лоренца:

$$\begin{cases} x' &= \sigma(y - x) \\ y' &= -y + (r - z)x \\ z' &= -bz + xy \end{cases}$$

которая определяет аттрактор Лоренца (рис. 3.7.9) в трехмерном пространстве. Выберем значения параметров: $\sigma = 10$, $r = 24$ и $b = 2.5$, промежуток времени [0, 30], начальные условия $x(0) = 5$, $y(0) = 7$, $z(0) = 9$. Траектория решения $X(t) = (x(t), y(t), z(t))$ имеет вид (рис. 3.7.9):

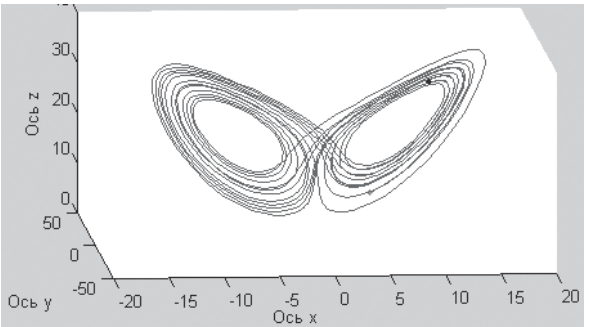


Рис. 3.7.9. Странный аттрактор Лоренца

Мы видим, что траектории системы притягиваются к некоторому достаточно сложному множеству в трехмерном пространстве. Такое множество принято называть аттрактором.

Напомним, что *аттрактор* – это компактное подмножество пространства динамической системы, такое что все траектории, начинающиеся из некоторой его окрестности, стремятся к нему при времени, стремящемся к бесконечности. Это притягивающее множество неустойчивых траекторий в пространстве динамической системы. Аттрактором может являться притягивающая неподвижная точка, периодическая траектория, или некоторая ограниченная область с неустойчивыми траекториями внутри. *Странный аттрактор*, в отличие от аттрактора, не является многообразием (то есть не является кривой или поверхностью); его геометрическое устройство очень сложно, а его структура фрактальна. Поэтому он и получил название «странный» [Д. Рюэль (D.Ruelle), Ф. Такенс (F. Takens)].

Обратимся снова к системе Лоренца. Если мы построим график одной координаты решения этой системы, например $x(t)$, то получим кривую, которая меняется довольно хаотично (рис. 3.7.10).

Возникает естественный вопрос: если у нас имеется временной ряд $\{x_n\}$, значения которого меняются хаотично, можно ли для него найти динамическую систему в многомерном пространстве, для которой наш ряд значений будет выборкой значениями некоторой координаты? Оказывается, это возможно! По имеющимся одномерным данным можно построить динамическую систему в многомерном пространстве, для которой наблюдаемая переменная будет одной из координат. В 1981 г. Такенсом была доказана теорема [Tak], утверж-

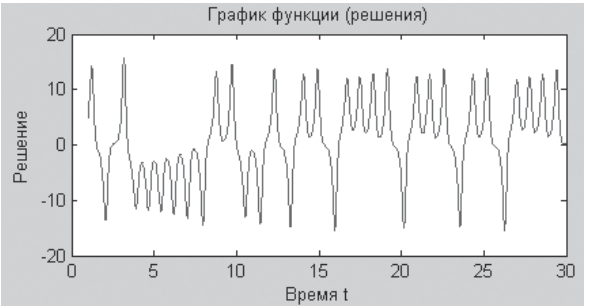


Рис. 3.7.10. График одной координаты решения системы Лоренца

дающая, что если траектория $X(t)$ некоторой n-мерной динамической системы $\dot{X} = F(X)$ лежит на аттракторе A, принадлежащем гладкому d-мерному многообразию, тогда по одномерной проекции $x(t)$ этой траектории можно получить m-мерную реконструкцию $Z(t)$ динамической системы в пространстве $\mathbf{R}^m$ при $m > 2d$ с некоторым аттрактором A_R. Реконструкция имеет вид

$$Z(t) = \Lambda_m(x(t)) = \{x(t), x(t+\tau), \ldots, x(t+(m-1)\tau)\} \in \mathrm{R}^m,$$

для почти любого $\tau > 0$. Такое число m и называется размерностью фазового пространства.

Таким образом, методом, предложеным Такенсом [Tak], по имеющимся одномерным данным $\{x_k\}$ можно построить динамическую систему в многомерном фазовом пространстве, для которой наблюдаемая переменная будет одной из координат. Поведение решения динамической системы является достаточно сложным и может восприниматься как хаотическое. Для характеристики таких движений применяется понятие «динамический хаос». В работе Рюэля и Такенса [RT], опубликованной в 1971 году, был введен новый математический образ динамического хаоса – странный аттрактор. Слово «странный» подчеркивает два свойства аттрактора. Это, во-первых, необычность его геометрической структуры. Размерность странного аттрактора является дробной (фрактальной). Во-вторых, странный аттрактор – это притягивающая область для траекторий. При этом все траектории внутри странного аттрактора динамически неустойчивы, что выражается в сильной (экспоненциальной) расходимости близких в начальный момент траекторий.

Для временного ряда $\{x_k\}$, $k = 0, 1, ..., N - 1$, в m-мерном фазовом пространстве строится последовательность точек вида

$$X_k = (x_k, x_{k+\tau}, \cdots, x_{k+(m-1)\tau}), \quad k = 0, \cdots, s-1, \quad s = N - (m-1)\tau.$$

Эта последовательность представляет дискретизацию динамической системы в $\mathbf{R}^m$. Для оценки характеристик реального исследуемого аттрактора можно вычислять характеристики восстановленного аттрактора на основе точек $X_k \in \mathbf{R}^m$.

Выбор временной задержки τ. Для малых шагов по времени Δt значения x_k и x_{k+1} будут близкими, поэтому большое значение приобретает правильный выбор временной задержки τ. Необходимо стремиться выбрать τ так, чтобы корреляция между x_k и $x_{k+\tau}$ была по возможности минимальной. Традиционный способ выбора временной задержки состоит в вычислении автокорреляционной функции временного ряда:

$$B(\tau) = \frac{1}{s}\sum_{k=0}^{s-1}(x_k - \overline{x})(x_{k+\tau} - \overline{x}), \quad s = N - \tau,$$

где $\overline{x}$ – среднее значение. Задержка τ выбирается равной времени первого пересечения нуля автокорреляционной функции.

После восстановления странного аттрактора можно определить еще две характеристики аттрактора временного ряда $\{x_n\}$.

Фрактальная размерность D_F некоторого множества (аттрактора) в n-мерном пространстве определяется как предел

$$D_F = \overline{\lim}_{\varepsilon \to 0} \frac{\ln M(\varepsilon)}{\ln(1/\varepsilon)},$$

где $M(\varepsilon)$ есть минимальное число n-мерных кубиков с ребром ε, необходимых для покрытия данного множества [Фе]. Для нетривиальных множеств размерность D_F может оказаться дробной.

Корреляционная размерность D_C некоторого множества (аттрактора) определяется формулой [GP1], [GP2]

$$D_C = \lim_{\varepsilon \to 0}\left(\frac{1}{\ln \varepsilon}\ln\left(\sum_{i=1}^{M(\varepsilon)} p_i^2\right)\right),$$

где $M(\varepsilon)$ есть минимальное число n-мерных кубиков с ребром ε, необходимых для покрытия данного множества, и где p_i^2 – вероятность того, что пара точек аттрактора принадлежит i-му кубику. Корреляционную размерность можно представить в виде:

$$D_C = \lim_{\varepsilon \to 0}(\ln C(\varepsilon)/\ln \varepsilon),$$

$$C(\varepsilon) = \lim_{m\to\infty} \frac{1}{m^2} \sum_{i,j=1}^{m} \theta(\varepsilon - \| X_i - X_j \|),$$

где $C(\varepsilon)$ – корреляционный интеграл, $\theta(\alpha) = \begin{cases} 1, \alpha \geq 0, \\ 0, \alpha < 0 \end{cases}$ – функция Хевисайда, X_i – точки в фазовом пространстве. Таким образом, размерность D_C определяется значением корреляционного интеграла $C(\varepsilon)$, характеризующим относительное число пар точек X_i, X_j, удаленных на расстояние $r_{ij} = \| X_i - X_j \| \leq \varepsilon$. Если выполняется условие $C(\varepsilon) \sim \varepsilon^D$, то D считают корреляционной размерностью аттрактора.

Для вычисления всех этих характеристик разработаны методики, известные в литературе по динамическому хаосу. Можно использовать программу «Fractan». Она создана В. В. Сычевым и имеется на сайте http://www.impb.ru/files.php в свободном доступе, см. также его диссертацию [С].

Возвращаясь к нашему случаю, отметим, что стохастические характеристики кардиосигналов наших пациентов Sm и Th, вычисленные при помощи программы «Fractan», существенно различны. Поэтому они также представляют диагностическую ценность. В частности, для компонент кардиосигналов на частоте Fr2 = 367.15 Гц (второй уровень разложения) имеем следующие данные (табл. 3.7.3):

Таблица 3.7.3. Стохастические характеристики компонент кардиосигнала

Пациенты	Показатель Херста	Фрактальная размерность	Корреляционная размерность	Размерность фазового пространства
Пациент Sm	0.2280	1.7669	3.829	32
Пациент Th	0.3103	1.68972	6.423	17

Непрерывный вейвлет-анализ кордиосигнала

Загрузим кардиосигнал (отведение V2) из файла SmCh4.txt, выберем фрагмент сигнала длиной 4096 отсчетов и сделаем его непрерывное вейвлет-разложение. Если мы интересуемся не только высокочастотными составляющими, но и хотим отразить и нижние частоты, то масштабный коэффициент нужно взять достаточно большим. Пределы изменения масштабного коэффициента зависят от выбора вейвлета и от того, какие частоты мы хотим отразить в результате разложения. Например, центральная частота вейвлета «мексиканская шляпа» равна 0,25 Гц. Поскольку в нашем случае $\Delta t = 1/1024$, то центральная частота вейвлета «мексиканская шляпа», используемо-

го для первого уровня разложения (когда масштабный коэффициент $a = 1$), равна Fr = 0.25*1024 = 256 Гц. Для значения $a = 100$ имеем частоту 256/100 = 2,56 Гц. Значение $a = 200$ обеспечивает частоту просмотра 1,28 Гц. Таким образом, если a в пределах от 1 до 200, то в вейвлет-коэффициентах будут отображены свойства кардиограммы в пределах частот от 256 Гц до 1,28 Гц. Для того чтобы отразить более высокие частоты, нужно выбрать значения масштабного коэффициента a, меньшие единицы, либо выбрать другой вейвлет с большей центральной частотой. Укажем центральные частоты некоторых часто используемых вейвлетов: Fr(db2) = 0.667 Гц, Fr(db4) = 0.7143 Гц, Fr(morl) = = 0.8125 Гц, Fr(haar) = 0.9961 Гц.

Сделаем непрерывное вейвлет-преобразование с использованием вейвлета Добеши db4, выбрав значение масштаба a в пределах от 1 до 600 с шагом 4, сначала для пациента Sm, а затем – для Th. Это значит, что будут отображены частоты от 734.3 Гц до 1,22 Гц (рис. 3.7.11, 3.7.12).

```
load SmCh4.txt;
SV2=SmCh4(601:4696);
w='db4'; Fr = centfrq(w)
subplot 211; plot(SV2); %График сигнала
subplot 212; Sc = cwt(SV2,1:4:600,w,'plot'); %разложение на частотах от
734.3 Гц до 1,22 Гц
```

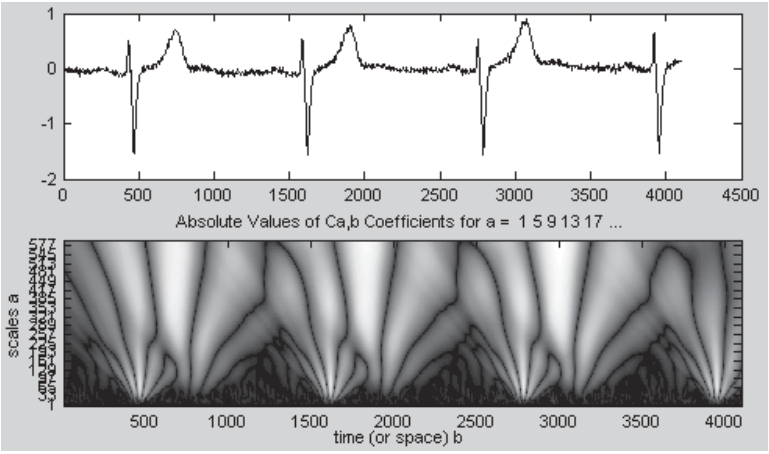


Рис. 3.7.11. График кардиосигнала и спектрограмма пациента Sm

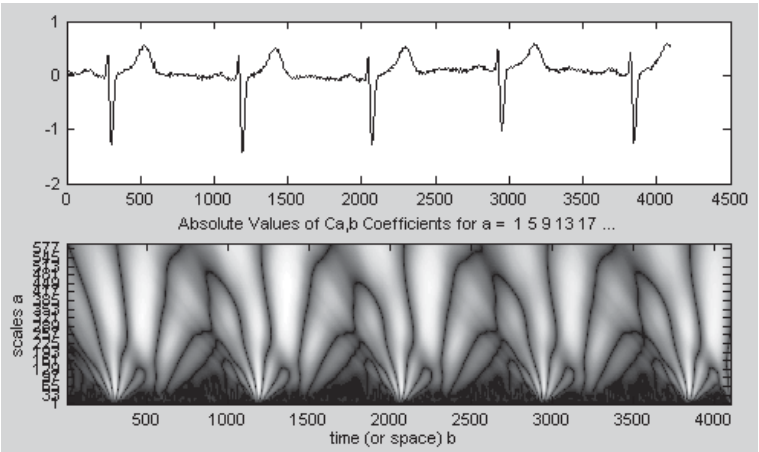


Рис. 3.7.12. График кардиосигнала и спектрограмма пациента Th

На спектрограммах хорошо виден комплекс QRS на высоких частотах, пик T проявляется на более низких частотах. Хорошо отображен пик P, а также виден пик U, едва заметный на кардиограмме. Видны также другие элементы на высоких частотах, которые на кардиограмме незаметны. Отчетливо видна разница двумерных изображений кардиограмм (рис. 3.7.11 и 3.7.12). Для того чтобы придать смысл полученным изображениям, нужен анализ серии кардиосигналов. Сейчас пока можно отметить только то, что спектрограмма кардиосигнала нагляднее и полнее отражает сигнал.

Для получения более крупного изображения нужно применить команду Sc = cwt(SV2,1:4:600,w,'plot') построения спектрограммы в отдельном окне.

Для большей наглядности можно построить график полученной матрицы коэффициентов (рис. 3.7.13). С этой целью выберем фрагмент матрицы, соответствующий одному полному циклу кардиосигнала. Рассмотрим отсчеты кардиосигнала от 1 до 1200 и масштаб от 1 до 300, что соответствует частотам от 734.3 Гц до 2,44 Гц. Полученный график можно увеличивать и «вращать», для того чтобы лучше увидеть разные элементы.

```
for k=1:1200;
    for l=1:75;
        Sd(l,k)=Sc(l,k);
    end
```

```
end
surfl(Sd)
colormap(gray);
shading interp;
```

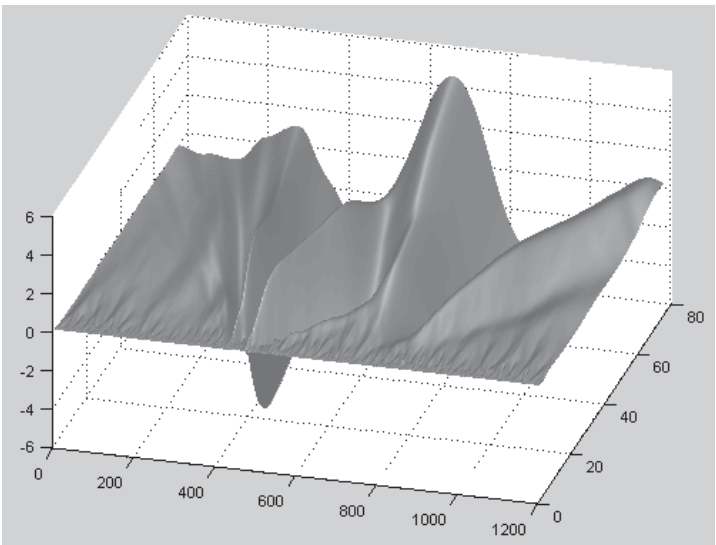


Рис. 3.7.13. График вейвлет-коэффициентов пациента Sm

Использование логарифмического масштаба. Мы применили непрерывное вейвлет-преобразование вейвлетом db4 с центральной частотой F_w = 0.7143 Гц. Масштабный коэффициент менялся равномерно от 1 до 600 с шагом 4. Соответствующие частоты изменялись от F_0 = 0.7143*1024 = 734.3 Гц до $F_0/600$ = 1,22 Гц. При этом частота при переходе с одной строки коэффициентов на другую менялась очень неравномерно. Первая, самая нижняя строка коэффициентов соответствует масштабному коэффициенту a = 1 и частоте F_0 = 0.7143*1024 = 731.4 Гц, следующая строка коэффициентов соответствует масштабному коэффициенту a = 2 и частоте F_1 = 731.4/2 = = 365.7 Гц, следующая за ней строка коэффициентов соответствует масштабному коэффициенту a = 3 и частоте F_1 = 731.4/3 = 243.8 Гц и т. д. Мы видим, что высокие частоты на спектрограмме отображены лишь несколькими линиями и, следовательно, незаметны. Большим значениям масштабного коэффициента соответствуют низкие часто-

ты, и изменение низких частот при переходе от одной строки коэффициентов к другой – незначительное. Поэтому при равномерном изменении масштабного коэффициента мы не увидим графического изображения высокочастотных компонент кардиосигнала, зато низкие частоты будут отображены хорошо. Эта неравномерность может быть исправлена применением логарифмической шкалы, когда вместо значений $a = k$, $k = 1, ..., N$, берутся степени a^k, $k = 0, 1, ..., N$, некоторого числа a. Теперь отношение частот двух соседних строк коэффициентов будет постоянно и равно a.

Пусть F_0 – частота первой строки коэффициентов (соответствует масштабному коэффициенту $a = 1$) и ΔF – желаемый (начальный) шаг изменения частоты. Тогда вторая строка коэффициентов будет иметь частоту $F_1 = F_0 - \Delta F$. Найдем значение масштабного параметра a, соответствующее этой частоте. $F_1 = F_0/a$. Из равенства $F_0/a = F_0 - \Delta F$ получаем: $a = F_0/F_1 = F_0/(F_0 - \Delta F)$. Теперь в качестве набора масштабных коэффициентов берем степени a^k, $k = 0, 1, ..., N$, этого числа a.

Процедура использования логарифмического масштаба легко реализуется в MATLAB. Вейвлет Добеши db4 имеет достаточно высокую центральную частоту 0.7143, поэтому в следующем примере используем гауссов вейвлет gaus3 с меньшей центральной частотой, равной 0.4, тогда $F_0 = 409.6$ Гц, и если взять начальное приращение частоты $\Delta F = 10$, то $a = 1.025$.

```
w='gaus3'; F0 = centfrq(w)*1024
DF=10;
N=270;
a=F0/(F0-DF)
for k=1:N; A(k)=a^(k-1); end
load SmCh4.txt;
SV2=SmCh4(601:4696);
Sc = cwt(SV2,A,w,'plot'); %разложение, с частотой 410 Гц с шагом 10
```

Мы видим (рис. 3.7.14), что высокочастотные компоненты кардиосигнала на спектрограмме фактически не видны. Только со значения $a = 4$ (100 Гц) мы можем заметить достаточно четкие проявления высокочастотного комплекса QRS. Это показывает, что высокочастотные компоненты кардиосигнала имеют шумовой характер и для их исследования лучше использовать статистические и стохастические методы.

Поиск характерных фрагментов кардиосигнала

Иногда бывает так, что анализируемый сигнал содержит достаточно характерные всплески, например комплекс QRS кардиосиг-

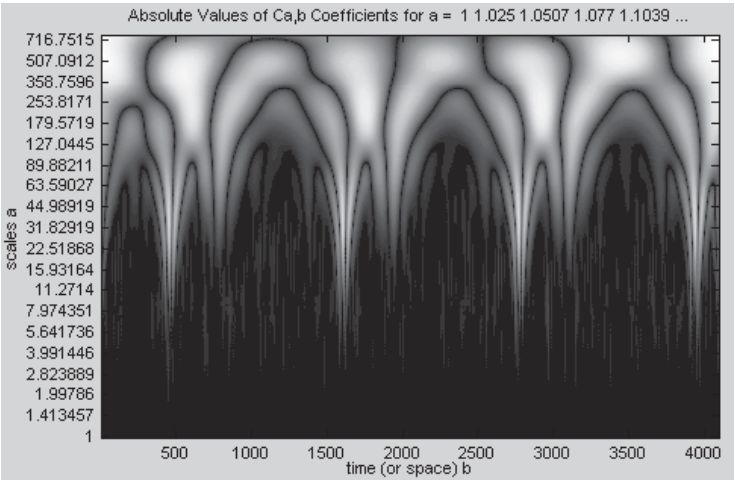


Рис. 3.7.14. Спектрограмма
в логарифмическом масштабе

нала, или другой всплеск, характеризующий сердечную патологию. Для обнаружения таких характерных колебаний при помощи непрерывного вейвлет-преобразования желательно использовать вейвлет, который имеет вид этого характерного всплеска. Однако имеются некоторые проблемы. Вейвлет для непрерывного вейвлет-преобразования должен иметь нулевой интеграл, а характерная волна, которую мы хотим обнаружить, может иметь ненулевой интеграл. Для решения этой проблемы Wavelet Toolbox предлагает функцию pat2cwav, которая находит достаточно регулярное приближение с нулевым интегралом к указанной волне методом наименьших квадратов. Описание функции приведено в разделах 3.1.3. и 4.9. Функция

```
[PSI,XVAL,NC] = pat2cwav(YPAT,METHOD,POLDEGREE,
REGULARITY)
```

вычисляет допустимый вейвлет, нормы равной 1, для CWT (заданный массивами XVAL и PSI), приближенный к образцу, определенному вектором YPAT. Значения аргумента x для образца устанавливаются как равномерно распределенные на отрезке [0, 1]. Константа NC является такой, что NC*PSI аппроксимирует YPAT на интервале [0, 1] по методу наименьших квадратов, используя:

- полином степени POLDEGREE, когда METHOD равен 'polynomial';

- проекцию на пространство функций, ортогональных к константам, когда METHOD равен 'othconst'.

Параметр REGULARITY определяет граничные условия в точках 0 и 1. Допустимые значения 'continuous', 'differentiable' и 'none'.

Рассмотрим на примере использование этой функции для построения нового вейвлета на основе волны комплекса QRS.

Пример. Рассмотрим задачу выделения типичного для кардиосигнала комплекса QRS (типы комплексов QRS зависят от выбора отведения и от состояния пациента). При вейвлет-анализе кардиосигнала естественно использовать вейвлет, который является хорошим приближением к выбранному типу комплекса QRS. Сначала загрузим фрагмент кардиосигнала (отведение 4) и выберем комплекс QRS. Влияние шумовой компоненты незначительно, удаление шума не применяем (хотя для следующей за QRS волны T потребовалось бы удаление шума).

```
% Программа для просмотра ЭКГ и выбора фрагмента
load SmCh4.txt
%График
Sm=SmCh4(100:6000);
figure(1)
plot(Sm); axis([0 length(Sm) min(Sm) max(Sm)]);
S1=Sm(501:1700); %Выборка фрагментов
S=Sm(901:1040); %Уточнение
N1=length(S1); % Длина данных
N=length(S);
%Графики фрагментов
figure(2)
subplot 121; plot(S1); axis([0 N1 min(S1) max(S1)]);
subplot 122; plot(S); axis([0 N min(S) max(S)]);
save SmQRS.txt -ascii S %Сохранение выбранного комплекса
```

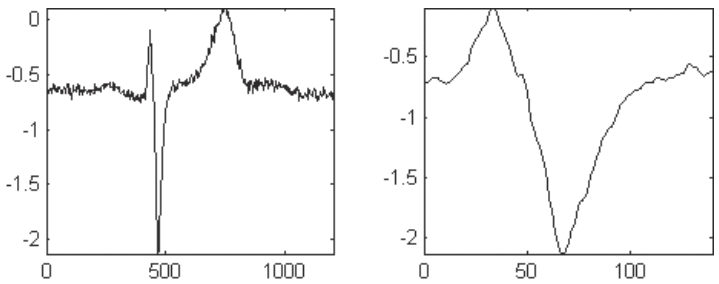


Рис. 3.7.15. Комплекс QRS кардиосигнала

Выбранный комплекс QRS рассматриваем в качестве образца. Загружаем этот образец и применяем команду pat2cwav для нахождения адаптированного вейвлета. В конце работы сохраняем результаты в mat-файле WavSmQRS.mat.

```
load SmQRS.txt; % Загрузка образца
Y=SmQRS;
   % Построение вейвлетов по образцу
[psi_p,xval_p,nc_p] = pat2cwav(Y, 'polynomial',10, 'continuous') ;
[psi_o,xval_o,nc_o] = pat2cwav(Y, 'orthconst',2, 'continuous') ;
   % Просмотр результатов
subplot 121; plot(xval_p,Y,xval_p,nc_p*psi_p,'.')
   title('polynomial')
subplot 122; plot(xval_o,Y,xval_o,nc_o*psi_o,'.')
   title('orthconst')
   % Сохранение результатов
savefile = 'WavSmQRS.mat';
save(savefile, 'psi_p','xval_p','nc_p','psi_o','xval_o','nc_o')
```

Полученные вейвлеты вместе с оригинальными образцами изображены на рис. 3.7.16. Рисунок справа показывает, что метод 'othconst' при значении параметров poldegree=2 и regularity = 'continuous' гораздо точнее отражает особенности образца.

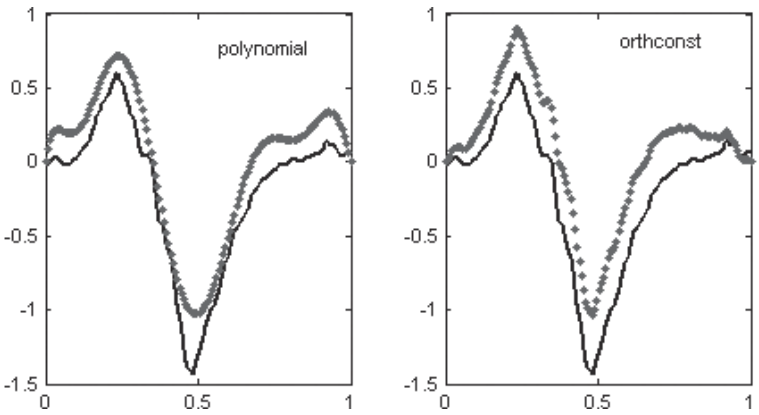


Рис. 3.7.16. Вейвлет, адаптированный к комплексу QRS кардиосигнала

Теперь используем полученный новый вейвлет для непрерывного вейвлет-разложения. Новые два адаптированных вейвлета сохранены нами в файле WavSmQRS.mat. Загружаем исследуемый сигнал,

загружаем данные файла WavSmQRS.mat и для разложения используем вейвлет psi_o. Результаты приведены на рис. 3.7.17 (сравните с рис. 3.7.11 и 3.7.12):

```
load SmCh4.txt;
load WavSmQRS.mat;
SV2=SmCh4(601:4696);
subplot 211; plot(SV2); %График сигнала
subplot 212; Sc = cwt(SV2,1:4:600,psi_o,'plot'); %
```

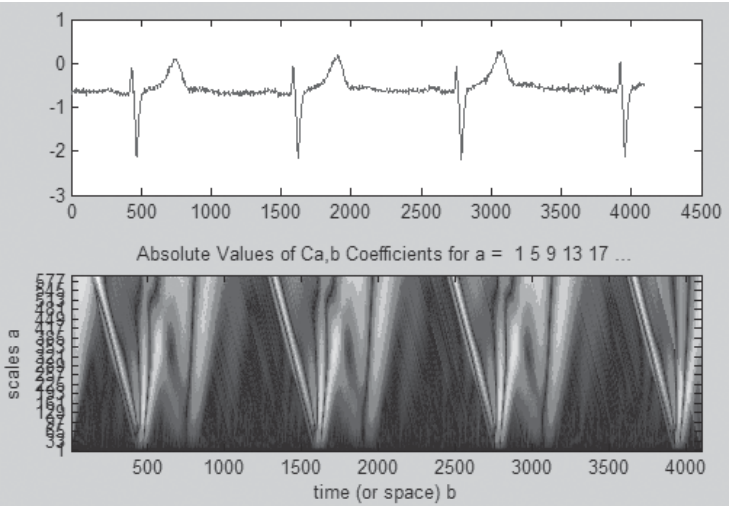


Рис. 3.7.17. Вейвлет-анализ с адаптированным вейвлетом

Предложенная схема построения адаптированного вейвлета может также применяться в диагностических целях для автоматического обнаружения характерных колебаний кардиосигнала методами вейвлет-анализа cwt.

3.7.3. Удаление шума, компрессия и сглаживание кардиосигнала

В случае кардиографов высокого разрешения кардиосигнал может иметь очень большой объем. Поэтому задача сжатия кардиосигнала для его хранения является достаточно актуальной. Сжатие сигнала можно провести за счет удаления шума. Шумом принято считать высокочастотные компоненты кардиосигнала. Удаление шума при-

водит к сжатию и сглаживанию кардиосигнала. Считается, что верхняя граничная частота нормального (без нагрузки) кардиосигнала, заметно влияющая на ее форму, не превышает 100 Гц (что подтверждает рис. 3.10.13). Поэтому компоненты сигнала частоты выше 100 Гц можно удалить без существенного изменения формы сигнала. Таким образом, мы получаем один из вариантов сжатия и сглаживания.

Предположим, что мы используем вейвлет Добеши db4. Он имеет центральную частоту Fr = 0.7143 Гц. Поскольку $\Delta t = 1/1024$, то центральная частота вейвлета db4, используемого для первого уровня разложения, равна Fr1 = 0.7143*1024 = 734.30 Гц, далее для второго уровня Fr2 = 367.15 Гц, для третьего уровня Fr3 = 183.57 Гц, и для четвертого – Fr3 = 91.8 Гц. Поэтому для удаления шума можно сделать разложение до четвертого уровня и для восстановления взять нулевыми все детализирующие коэффициенты cD1, cD2, cD3 и cD4. В этом случае мы получаем сжатие сигнала в $2^4 = 16$ раз.

```
load SmCh4.txt; SV2=SmCh4(601:4696);
w='db4'; [c,l] = wavedec(SV2,4,w);
cA4=appcoef(c,l,w,4); ScA4=upcoef('a',cA4,w,4);
subplot 211; plot(SV2);title('ECG V2')
subplot 212; plot(ScA4);title('Approx. at level 4')
```

Однако в результате получается сигнал немного длиннее и слегка искаженный на концах. Поэтому нужно либо его обрезать на концах, либо воспользоваться функцией wdencmp, которая дает очень неплохие результаты. Отметим некоторые дополнительные возможности Wavelets Toolbox для обработки кардиограммы.

1. Определение порога для удаления шума кардиосигнала. Функция THR = thselect(X,TPTR) находит значение порога для сигнала, используя правило, определенное строкой TPTR. Этот параметр может принимать следующие значения: 'rigrsure' – адаптивный выбор порога с использованием принципа Штейна несмещенной оценки риска, 'heursure' – эвристический вариант первого выбора, 'sqtwolog' – порог sqrt(2*log(length(X))) и 'minimaxi' – минимаксный порог.

Определим пороги для каждого правила выбора для кардиосигнала SV2 пациента Sm.

```
thr = thselect(SV2,'rigrsure')
thr =
    2.1746
thr = thselect(SV2,'sqtwolog')
thr =
    4.0787
```

```
thr = thselect(SV2,'minimaxi')
thr =
    2.5884
```

2. Оценка шума вейвлет-коэффициентов. Используем функцию STDC = wnoisest(C,L,S), которая находит оценки стандартного отклонения вейвлет-коэффициентов, полученных при дискретном разложении. При оценке используется медиана абсолютного отклонения, деленная на 0.6745, что хорошо подходит для гауссова белого шума в одномерной модели (см. thselect).

```
load SmCh4.txt;
SV2=SmCh4(601:4696); %Фрагмент сигнала
w='db4'; Fr = centfrq(w) % выбор вейвлета, определение его центральной
частоты
[c,l] = wavedec(SV2,4,w); %Разложение до уровня 3
STDC = wnoisest(c,l,[1 2 3 4])
STDC =
    0.0042    0.0165    0.0351    0.0648
```

3. Удаление шума и сжатие при помощи вейвлетов. Функция [XC, CXC, LXC, PR0, PRL2] = wdencmp('gbl',X,'wname', N, THR, SORH, KEEPAPP) производит в случае глобального порога gbl:

- очищенную от шума версию входного сигнала;
- структуру вейвлет-разложения очищенного сигнала;
- оценку числа обнуленных коэффициентов сигнала в процентах;
- отношение L^2-норм сжатого и первоначального сигналов в процентах.

Функция использует уровень вейвлет-разложения N, порог THR, метод пороговой обработки SORH='s' или 'h' и параметр KEEPAPP = 1 – тогда коэффициенты аппроксимации не подвергаются пороговой обработке.

Используем wdencmp для компрессии кардиосигнала при фиксированном пороге. Загружаем кардиосигнал и применяем команду wdencmp.

```
load SmCh4.txt; SV2=SmCh4(601:4696); thr=4.1;
[Sd,cxd,lxd,perf0,perfl2] =wdencmp('gbl',SV2,'db4',4,thr,'h',1);
subplot(211), plot(SV2); title('Original signal');
subplot(212), plot(Sd); title('Compressed signal');
% Plot original and compressed signals.
xlab1 = ['2-norm rec.: ',num2str(perfl2)];
xlab2 = [' %  -- zero cfs: ',num2str(perf0), ' %'];
xlabel([xlab1 xlab2]);
```

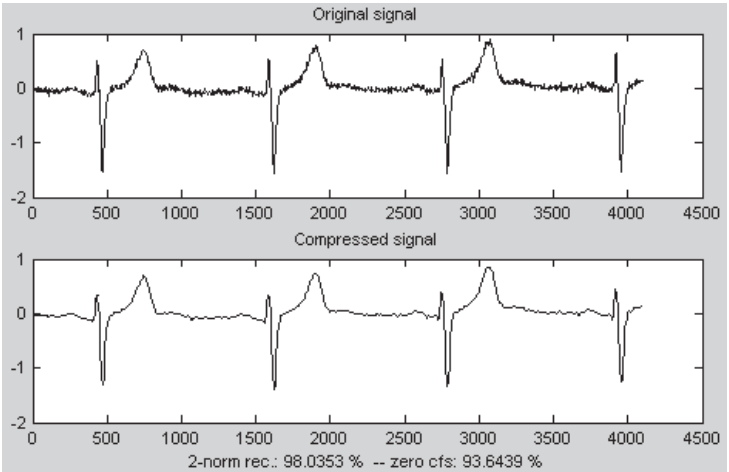


Рис. 3.7.18. Графики кардиосигнала и его сжатия

В результате получен сигнал, сжатый более чем в 10 раз. При этом в сжатом сигнале зубцы кардиосигнала видны более явно (рис. 3.7.18), хорошо отражена их форма. Можно также получить всю удаленную шумовую компоненту в виде разности исходного сигнала и очищенного. Шумовая компонента может быть использована для получения статистических и стохастических характеристик высокочастотной части сигнала.

3.7.4. Использование пакетных разложений

Для получения новых числовых характеристик кардиосигнала можно также использовать пакетные вейвлет-коэффициенты и соответствующие компоненты сигналов. Команда T = wpdec(X,N,'wname') производит пакетное разложение, а команда X = wprcoef(T,N) производит прямое восстановление только по одному набору коэффициентов в узле N. Для полученной таким образом компоненты сигнала мы можем получить энтропию, статистические и стохастические характеристики. Однако характерной чертой пакетного разложения является наличие нескольких ярко выраженных основных частот (рис. 3.7.21). Данный раздел посвящен исследованию именно частотных характеристик высокочастотных компонент. Отметим еще раз, что рассматриваются пакетные компоненты сигнала, а не коэффициенты разложения.

Для анализа брались оцифрованные кардиосигналы, снятые на кардиографе высокого разрешения (1028 отсчетов в секунду) «Кардиотехника-4000, EcgShell». Кардиосигнал регистрируется по 8 общепринятым каналам: L – левая рука (+) и правая рука (–), F – левая нога (+) и правая рука (–) и шесть грудных отведений, обозначаемых $C1$–$C6$ (рис. 3.7.1). Из данных 8 каналов L, F, $C1$, $C2$, $C3$, $C4$, $C5$, $C6$ кардиографа образуются 12 так называемых стандартных отведений I, II, III, aVR, aVL, aVF, V1,V2,V3,V4,V5,V6 по формулам:

$$\mathrm{I} = L, \quad \mathrm{II} = F, \quad \mathrm{III} = F - L,$$
$$\mathrm{aVR} = -(L + F)/2, \quad \mathrm{aVL} = L - F/2, \quad \mathrm{aVF} = F - L/2,$$
$$\mathrm{Vi} = C\mathrm{i} - (L + F)/3, \quad i = 1, 2, ..., 6.$$

Обоснование определения таких отведений дается в любом учебнике по кардиологии, см., напр., [В], [ПК].

Для выделения высокочастотных компонент сигнала используется пакетное вейвлет-разложение сигнала до четвертого уровня вида, указанного на рис. 3.7.19. В узле (0.0) находится исследуемый сигнал s, который далее раскладывается дискретным вейвлет-преобразованием на сглаженную компоненту (1.0) и высокочастотную часть (1.1) по формулам

$$a_{1,k} = \sum_n h_n^* s_{2k-n}, \quad d_{1,k} = \sum_n g_n^* s_{2k-n},$$

где $\{h_n\}$ – масштабирующий фильтр вейвлета, $h_n^* = \overline{h}_{-n}$, $g_n^* = \overline{g}_{-n}$ и $g_n = (-1)^n \overline{h}_{-1-n}$. Восстановление сигнала производится по формуле

$$s_n = \sum_{k \in Z} (h_{n-2k} a_{1,k} + g_{n-2k} d_{1,k}).$$

При этом если в последней формуле коэффициенты $\{d_{1,k}\}$ считать нулевыми, то восстановленный сигнал является сглаженной компонентой сигнала, если коэффициенты $\{a_{1,k}\}$ считать нулевыми, то восстановленный сигнал является высокочастотной компонентой сигнала.

Процедуру разложения можно применить к полученным коэффициентам $\{a_{1,k}\}$ и $\{d_{1,k}\}$. В результате получается дерево пакетного разложения, часть которого изображена на рис. 3.7.19. В качестве высокочастотных компонент сигнала мы берем компоненты, соответствующие коэффициентам в узлах (2,2), (2,3), (3,2), (3,3), (4.1), (4,2) и (4,3).

Причина такого выбора заключается в том, что функция MATLAB, besttree – наилучшее дерево пакетного вейвлет-разложения, примененная к кардиосигналу, дает именно такое дерево разложения.

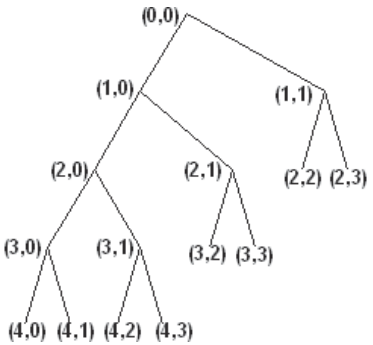


Рис. 3.7.19. Дерево разложения сигнала

Высокочастотные компоненты имеют близкое к нулю математическое ожидание. Для каждой компоненты сигнала вычисляется стандартное отклонение. При нулевом среднем стандартное отклонение может быть использовано в качестве меры интенсивности компоненты.

В качестве примера рассмотрим стандартные отклонения высокочастотных компонент в различных каналах и для различных пациентов (Sm2 и Ple). Диаграммы на рис. 3.7.20 показывают, что стандартные отклонения различных пациентов существенно отличаются (каналы занумерованы числами 1, 2, ..., 8).

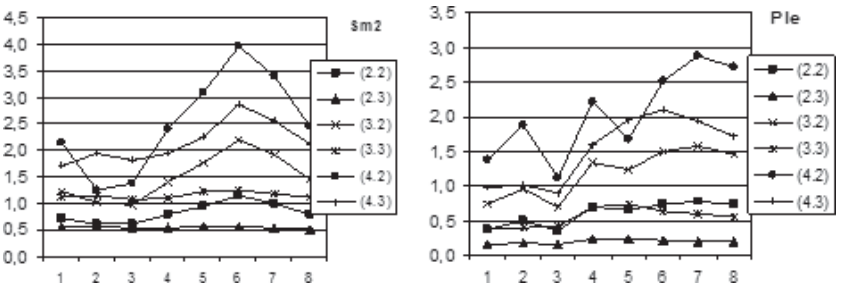


Рис. 3.7.20. Диаграммы интенсивностей высокочастотных компонент

Для анализа частотных характеристик компонент сигнала можно применить дискретное преобразование Фурье с дальнейшим анализом спектра. Оказывается, что вейвлет-компоненты сигнала доста-

точно хорошо локализованы по частоте, они концентрируются в основном в диапазонах частот, указанных в табл. 3.7.3. На следующем рис. 3.7.21 изображены частотные спектры высокочастотных компонент в нескольких первых узлах при разложении с использованием биортогонального вейвлета bior1.5 с основной частотой 0,7781 Гц.

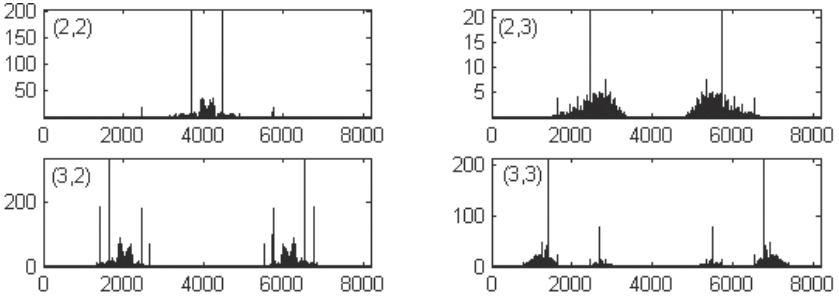


Рис. 3.7.21. Частотные спектры компонент сигнала

Хорошо видно, что имеются несколько выделяющихся (основных) частот. К основным частотам мы отнесем те, у которых амплитуда составляет более 1/3 от максимальной амплитуды. Количество основных частот, их значения и амплитуды зависят от узла разложения, отведения, вейвлета и от пациента. Данные зависимости позволяют надеяться, что основные частоты не являются аппаратными шумами, а характеризуют работу сердца. Наиболее часто встречаются значения (в герцах) основных частот, указанные в табл. 3.7.4.

Таблица 3.7.4. Основные электрические частоты

Узлы	(2,2)	(2,3)	(3,2)	(3,3)	(4,1)	(4,2)	(4,3)
Диапазон частот	400–500	250–350	200–300	100–150	40–70	70–140	60–125
Первая частота	464.0 514.0	307.0 299.8	207.1 257.0	157.1 149.9	50.0 48.6	114.3 100.0 128.5	78.6 92.9
Вторая частота	464.06 449.7	207.1 357.0 349.8	307.0 178.5 199.9	178.5 185.7	78.6	142.8 92.9 100.0	50.0 178.5 92.9

Статистические и частотные характеристики мы вычисляем для компонентов сигнала (то есть восстановленных по одному набору коэффициентов). Для пакетного вейвлет-разложения в MATLAB есть функция вычисления еще одной числовой характеристики – значе-

ния энтропии коэффициентов в узлах дерева разложения. Она также представляет интерес. Покажем это на примере. Возьмем кардиосигналы двух пациентов, условно обозначенных Sm и Th, сделаем пакетное преобразование до 3-го уровня с вейвлетом db4 и вычислим значения энтропии Шеннона во всех терминальных узлах дерева, кроме первого узла (3,0), который содержит аппроксимацию сигнала.

```
load SmCh4.txt; SV2=SmCh4(601:4696);
X=SV2; w='db4'; N=3; P='shannon';
T = wpdec(X,N,w);
for k=1:(2^N-1);
    XI=wpcoef(T,[N,k]);
    E(k)=wenropy(XI,P);
end
E
```

Результаты представлены в табл. 3.7.5.

Таблица 3.7.5. Значения энтропии пакетных вейвлет-коэффициентов

Пациенты	Значения энтропии в терминальных узлах						
Индексы узлов	(3,1)	(3,2)	(3,3)	(3,4)	(3,5)	(3,6)	(3,7)
Пациент Sm	5.3781	0.5496	1.9093	0.0084	0.0446	0.2365	0.1373
Пациент Th	2.2253	0.1811	0.5132	0.0048	0.0261	0.0731	0.0400
Отношение Sm/Th	2.4	3.0	3.7	1.7	1.7	3.2	3.4

Разница значений энтропии пакетных вейвлет-коэффициентов пациентов Sm и Th еще более значительна, чем для статистических характеристик в табл. 3.7.1. Поэтому она также может иметь диагностическую значимость.

При желании можно использовать другой вейвлет, другой тип энтропии или рассматривать пакетные разложения другого уровня N.

Существенная разница числовых характеристик высокочастотных компонент кардиосигнала пациентов Sm и Th наблюдается во всех 12 стандартных отведениях. Здесь в качестве примера приведен анализ данных одного из самых информативных отведений – V2. Полученные результаты показывают, что числовые характеристики высокочастотных компонент кардиосигналов у здорового человека существенно больше по абсолютной величине и более разнообразны по своей структуре.

Таким образом, для высокочастотных компонент кардиосигнала представляют интерес следующие числовые характеристики:

- интенсивности (стандартные отклонения);

- энтропия;
- значения основных частот;
- число основных частот;
- амплитуды основных частот высокочастотных компонент.

В работах [По3], [По4], [По5], [ППС], [СЦ1] и [СЦ2] проведено достаточно полное исследование данных числовых характеристик на основе 76 кардиосигналов больных и здоровых пациентов. Изучены также следующие вопросы:

- зависимость числовых характеристик высокочастотных компонент кардиосигнала от выбора вейвлета;
- устойчивость числовых характеристик высокочастотных компонент кардиосигнала для одного пациента при различных регистрациях кардиосигнала и
- зависимость числовых характеристик высокочастотных компонент кардиосигнала от пациента при одном вейвлете.

Как уже отмечалось, вопрос о диагностической значимости числовых характеристик высокочастотных компонент остается открытым. Пока, основываясь на ряде наблюдений, можно отметить, что эти числовые характеристики существенно зависят от пациента. Поэтому они (может быть, опосредованно) характеризуют работу сердца. Ввиду больших различий значений зависимости от аппаратных шумов не выявлено. По результатам проведенных наблюдений можно отметить высокую степень устойчивости значений стандартных отклонений, энтропии, основных частот и их количества в серии снятых кардиосигналов одного и того же пациента. Диаграммы интенсивностей высокочастотных компонент пациентов, перенесших инфаркт миокарда достаточно давно, мало отличаются от диаграмм интенсивностей здоровых пациентов. У пациентов с патологическими изменениями не более месяца назад диаграммы интенсивностей достаточно существенно отличаются от диаграмм пациентов здоровых пациентов.

Для анализа использовалась следующая программа на MATLAB.

Программа анализа кардиосигнала

Разработана для вычисления статистических характеристик компонент сигнала, частотного спектра и выделения основных частот. Она написана на m-языке MATLAB в виде m-файла. Обсудим основные блоки программы, полный листинг приведен в конце раздела. Она начинается с выбора вейвлета для разложения и глубины разложения *N*.

```
w='bior1.5';     % выбор вейвлета
Fw = centfrq(w)  % частота вейвлета
N=4;             % выбор уровня% разложения%
```

Затем загружается оцифрованный кардиосигнал, представленный в виде форматированного текстового файла из 8 столбцов. В результате загрузки получается матрица из 8 строк, причем строки соответствуют каналам *L*, *F*, *C*1, *C*2, *C*3, *C*4, *C*5 и *C*6.

```
v=fopen('D:\Med_Issled\Data\Dat_Stud\Vorob_4.txt','rt');
S8=fscanf(v,'%g',[8 inf]);       %Считывание данных из файла *.txt
S8n=S8/(10^6);                   % Нормировка
L=S8n(1,:) ;F=S8n(2,:);
C1=S8n(3,:);C2=S8n(4,:);C3=S8n(5,:);C4=S8n(6,:);C5=S8n(7,:);C6=S8n(8,:);
```

Далее из полученных каналов образуем 12 стандартных отведений по формулам:

$$\mathrm{I} = L, \quad \mathrm{II} = F, \quad \mathrm{III} = F - L,$$
$$\mathrm{aVR} = -(L + F)/2, \quad \mathrm{aVL} = L - F/2, \quad \mathrm{aVF} = F - L/2,$$
$$\mathrm{Vi} = C\mathrm{i} - (L + F)/3, \quad i = 1, 2, ..., 6.$$

Делается пакетное вейвлет-разложение до 4-го уровня каждой строки-отведения – это цикл по k = 1:12. Коэффициенты пакетного разложения выбираем только те, которые соответствуют узлам (2,2), (2,3), (3,2) (3,3), (4.1), (4,2) и (4,3). Эти узлы занумерованы в дереве пакетного разложения соответственно числами 5, 6, 9, 10, 17, 18, и 16 – это цикл по p = 5, 6, 9, 10, 17, 18, 16.

Находим энтропию Шеннона коэффициентов разложения. Напомним, что энтропия *Шеннона* ('shannon') сигнала *s* определяется формулой

$$E1(s) = -\sum_i s_i^2 \log(s_i^2).$$

Затем мы находим высокочастотные компоненты сигнала, восстанавливая сигнал только по одной ветви вейвлет-коэффициентов. Находим стандартные отклонения полученных компонент.

```
  Tk=wpdec(St(k,:),N,w);    % Разложение в пакет к-й строки-отведения%
for p=[5,6,9,10,17,18,16];  % цикл по номерам терминальных узлов
     Ko=wpcoef(Tk,p);
     E='shannon';
     En(k,p)=wentropy(Ko,E);%энтропия коэффициентов
  %Восстановление компоненты по набору коэффициентов
KoSi=wprcoef(Tk,p);
       Sd(k,p)=std(KoSi);   %Средние квадратичные отклонения%
```

Далее делаются дискретное преобразование Фурье каждой компоненты сигнала и образование спектра мощности частот по формуле

$$PKo(l) = \frac{|F(l)|^2}{Q}, \quad l = 1, 2, \ldots, Q,$$

где Q – длина массива F.

```
    Q=sz(2);
  FK=fft(KoSi'); %Преобразование Фурье компоненты сигнала
  F=FK';
  PKo=F.*conj(F)/Q;
```

Теперь делается выборка основных частот. Для этого массив `PKo` обрезается наполовину (поскольку он симметричен относительно середины). Полученный массив P сортируется по убыванию мощности частот. Так как функция `sort` MATLAB сортирует данные по возрастанию, то применяем ее к массиву –P. Выбираются первая и вторая частоты как первый и второй элементы вектора после сортировки. Номер I основной частоты в исходном спектре используется для определения частоты в герцах по формуле

```
Fr1(k,p) = I(1)/(Q/1028).
Q=sz(2);
FK=fft(KoSi'); %Преобразование Фурье компоненты сигнала
F=FK';
PKo=F.*conj(F)/Q;
  for l=1:(Q/2); P(l)=PKo(l); end
     [B,I]=sort(-P);             % сортировка
        Fr1(k,p)=I(1)/(Q/1028); % Первая основная частота
        Fr2(k,p)=I(2)/(Q/1028); % Вторая частота
        PFr1(k,p)=-B(1)*(10^6); % Значение основной частоты
        PFr2(k,p)=-B(2)*(10^6);
```

Находим количество основных частот, то есть таких частот, амплитуда которых больше или равна одной трети максимальной амплитуды первой основной частоты.

```
  Ma=-B(1)/3;
     for s=1:(Q/4);
        if -B(s)>=Ma; NN(k,p)=s;
           end
     end
```

В конце программы из всех полученных данных формируется выходной массив, который экспортируется в рабочий лист Excel при

помощи приложения Excel Link MATLAB. Диаграммы полученных данных строятся в Excel.

Листинг программы

```
   %Программа для пакетного разложения кардиосигнала по 12 каналам
   %I, II, III, aVR,aVL, aVF, V1,V2,V3,V4,V5,V6 и анализа компонент
w='bior1.5';     % выбор вейвлета
Fw = centfrq(w)  % частота вейвлета
N=4;             %выбор уровня% разложения%
   %Пересчет 8 каналов I, II, C1,C2,C3,C4,C5,C6 --
   %в станд 12 каналов %I, II, III, aVR, aVL, aVF, V1,V2,V3,V4,V5,V6
   %Открытие файла *.txt с данными
v=fopen('D:\Med_Issled\Data\Dat_Stud\Vorob_4.txt','rt');
S8=fscanf(v,'%g',[8 inf]);       %Считывание данных из файла *.txt
S8n=S8/(10^6);                   % Нормировка
L=S8n(1,:) ;F=S8n(2,:);
C1=S8n(3,:);C2=S8n(4,:);C3=S8n(5,:);C4=S8n(6,:);C5=S8n(7,:);C6=S8n(8,:);
St(1,:)=L;
St(2,:)=F;
St(3,:)=F-L;
St(4,:)=-(L+F)/2;
St(5,:)=L-F/2;
St(6,:)=F-L/2;
St(7,:)=C1-(L+F)/3;
St(8,:)=C2-(L+F)/3;
St(9,:)=C3-(L+F)/3;
St(10,:)=C4-(L+F)/3;
St(11,:)=C5-(L+F)/3;
St(12,:)=C6-(L+F)/3;
sz=size(St); %Размер данных
for k=1:12
   Tk=wpdec(St(k,:),N,w);  % Разложение в пакет к-й строки-отведения%
for p=[5,6,9,10,17,18,16]; % цикл по номерам терминальных узлов
           Ko=wpcoef(Tk,p);
           E='shannon';
           En(k,p)=wentropy(Ko,E);       %энтропия коэффициентов
   %Восстановление компоненты по набору коэффициентов
  KoSi=wprcoef(Tk,p);
       Sd(k,p)=std(KoSi);  %Средние квадратичные отклонения%
   Q=sz(2);
   FK=fft(KoSi');          %Преобразование Фурье компоненты сигнала
   F=FK';
       PKo=F.*conj(F)/Q;
       for l=1:(Q/2); P(l)=PKo(l); end
       [B,I]=sort(-P);           % сортировка
```

```
       Fr1(k,p)=I(1)/(Q/1028);    % Первая основная частота
       Fr2(k,p)=I(2)/(Q/1028);    % Вторая частота
       PFr1(k,p)=-B(1)*(10^6);    % Значение основной частоты
       PFr2(k,p)=-B(2)*(10^6);
             Ma=-B(1)/3;
             for s=1:(Q/4);
                 if -B(s)>=Ma; NN(k,p)=s;
             end
             end
         Ko=0; KoSi=0; B=0; I=0; FK=0;
F=0; PKo=0; P=0; Ma=0;    %Обнуление после цикла
end                       %конец цикла по p
Tk=0;                     %NoSi=0; %DSt;
end                       %конец цикла по к
   % Формирование массивов для вывода
   % Стандартные отклонения
Std=[Sd(:,5), Sd(:,6), Sd(:,9), Sd(:,10), Sd(:,17), Sd(:,18), Sd(:,16)];
   % Энтропия коэффициентов
Ent=[En(:,5),En(:,6),En(:,9),En(:,10),En(:,17),En(:,18),En(:,16)];
   % Основные частоты, их значения и количество
Fre=[Fr1(:,5),Fr2(:,5),PFr1(:,5),PFr2(:,5),NN(:,5), …
       Fr1(:,6),Fr2(:,6),PFr1(:,6),PFr2(:,6),NN(:,6), …
       Fr1(:,9),Fr2(:,9),PFr1(:,9),PFr2(:,9),NN(:,9), …
       Fr1(:,10),Fr2(:,10),PFr1(:,10),PFr2(:,10),NN(:,10), …
       Fr1(:,17),Fr2(:,17),PFr1(:,17),PFr2(:,17),NN(:,17), …
       Fr1(:,18),Fr2(:,18),PFr1(:,18),PFr2(:,18),NN(:,18), …
       Fr1(:,16),Fr2(:,16),PFr1(:,16),PFr2(:,16),NN(:,16)];
Freq=Fre';
D=[Std,Ent]*(10^4);
   % Выходной массив
Dat=[D';Freq];
```

Выводы. Вейвлет-анализ позволяет получить принципиально новые характеристики кардиосигнала. Вейвлет-спектрограмма кардиосигнала визуально более информативна, чем кардиограмма. Вейвлет-анализ позволяет произвести эффективное сжатие и сглаживание кардиосигнала.

Рассмотренные здесь методы исследования высокочастотных компонент кардиосигнала и многие другие возможности доступны не только в командном режиме работы MATLAB, но и в большом комплексе графических сред, объединенных в приложение Wavelet Toolbox MATLAB под названием «Главное меню». Возможности главного меню будут рассмотрены в следующей главе.

Главное меню пакета Wavelet Toolbox



Главное волновое меню представляет собой большой пакет графических сред, удобных для использования широкого комплекса команд практически по всем направлениям вейвлет-анализа сигналов. Главное меню называется также графическим интерфейсом пользователя – Graphical User Interface (GUI).

Главное меню вызывается из MatLab командой **wavemenu**. После этого открывается окно, на основном поле которого расположены группы кнопок, показанные на рис. 4.0.1.

Пользователь может выбрать любую из предложенных возможностей. Каждая кнопка вызывает графическую оболочку, специально разработанную для определенного раздела вейвлет-анализа сигналов и изображений и для изучения вейвлетов.

В этой главе рассмотрим все девять групп графических оболочек главного меню. Начнем изучение с раздела **Display**.

4.1. Просмотр вейвлетов

Раздел **Display** предназначен для знакомства и получения информации о вейвлетах, используемых в MATLAB. Он состоит из двух частей:

- **Wavelet Display** – просмотр вейвлетов;
- **Wavelet Packet Display** – просмотр пакетных вейвлетов.

4.1.1. Просмотр вейвлетов

Для запуска **Wavelet Display** нужно вызвать главное волновое меню командой wavemenu и нажать кнопку **Wavelet Display**. Открывается окно, которое содержит обычное меню вверху, а также специальные кнопки меню справа и внизу.



Рис. 4.0.1. Окно главного волнового меню

Правое меню. Вверху справа (**Wavelet**) есть возможность выбрать имя или тип вейвлета и его порядок. Далее можно установить уровень разрешения **Refinement** – это число N, такое что значения вейвлетов вычисляются с шагом $1/2^N$. По умолчанию N = 8. Затем нажать кнопку **Display**, тогда в графическом окне появятся графики вейвлет-функций $\varphi(x)$ и $\psi(x)$ в пределах своего носителя. Ниже изображаются значения фильтров разложения и реконструкции.

Напомним, что фильтры $\{h_n\}_{n\in\mathbf{Z}}$ и $\{g_n\}_{n\in\mathbf{Z}}$ вейвлетов $\varphi(x)$ и $\psi(x)$ – это коэффициенты масштабирующих уравнений $\varphi(x) = \sqrt{2}\sum_n h_n \varphi(2x-n)$

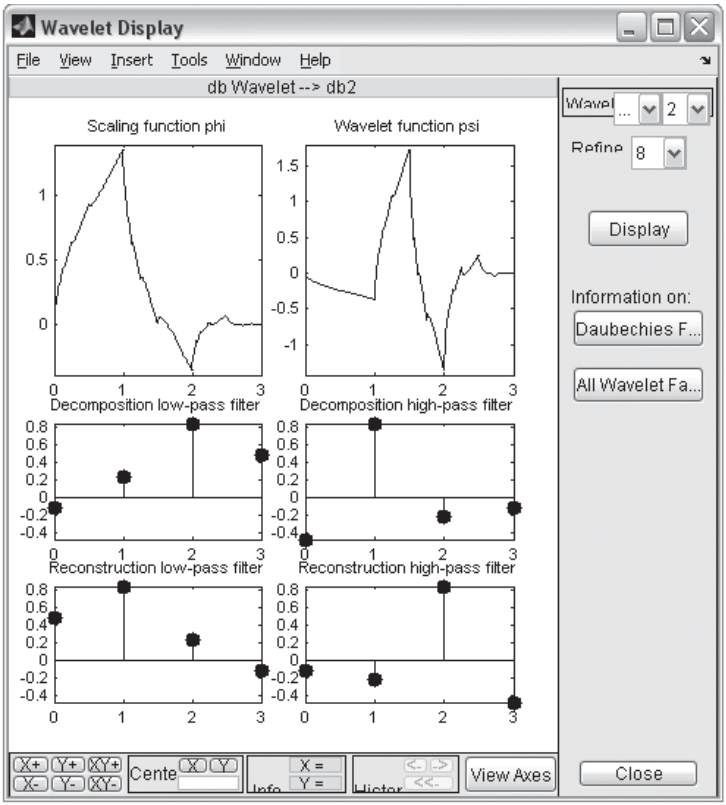


Рис. 4.1.1. Основное окно раздела Wavelet Display

и $\psi(x) = \sqrt{2}\sum_n g_n \varphi(2x-n)$, где $g_n = (-1)^n \overline{h}_{N-n}$. Фильтр $\{h_n\}$ называется низкочастотным, а фильтр $\{g_n\}$ – высокочастотным. При этом для восстановления используются фильтры $\{h_n\}$ и $\{g_n\}$ вейвлетов $\varphi(x)$ и $\psi(x)$, а для вейвлет-разложения – сопряженные (транспонированные) фильтры $h_n^* = \overline{h_{-n}}$ и $g_n^* = \overline{g_{-n}}$. Все это хорошо видно при изображении фильтров в графическом окне. В приведенном выше примере имеем в левом нижнем окне фильтр $\{h_n\}$ $\varphi(x)$-функции Добеши 'db2':

```
W = dbwavf('db2'); H=sqrt(2)*W
H =  0.4830   0.8365   0.2241   -0.1294   % фильтр Lo_R
```

Внизу справа – фильтр $g_n = (-1)^n h_{3-n}$ вейвлета Добеши $\psi(x)$:

```
G = -0.1294   -0.2241    0.8365   -0.4830   % фильтр Hi_R
```

Отметим, что функция dbwavf в MATLAB вычисляет масштабирующий фильтр W, то есть коэффициенты w_n уравнения $\varphi(x) = 2\sum_n w_n \varphi(2x - n)$, где φ – масштабирующая функция вейвлета. Тогда $w_n = 1/\sqrt{2}\, h_n$.

Справа имеются еще две кнопки, при нажатии которых открывается дополнительное окно с информацией о выбранном вейвлете либо о всех вейвлетах MATLAB (**All Wavelet Families**).

Нижнее меню. Предназначено для более детального рассмотрения графиков.

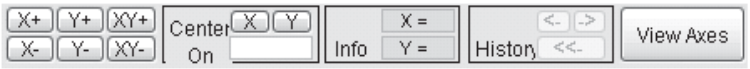


Рис. 4.1.2. Кнопки нижнего меню

Во-первых, кнопка **Wiev Axes** запускает дополнительное окно, кнопки которого позволяют открыть соответствующий график «во все окно». Вернуться к исходному окну (рис. 4.1.1) можно, «отжав» кнопку на панели **Wiev Axes** либо выключив ее полностью.

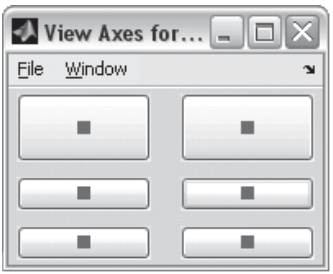


Рис. 4.1.3. Окно Wiev Axes

Левой кнопкой мыши можно выбрать фрагмент графика и кнопками **X+**, **X-**, **Y+**, **Y-**, **XY+**, **XY-** увеличить или уменьшить его масштаб по координатным осям X и Y отдельно либо одновременно. Кнопка **History** позволяет вернуться обратно в процессе изменения масштабов. При нажатии правой кнопкой мыши на точку графика в поле **Info** указываются координаты выбранной точки. Кнопка **Center On** позволяет центрировать график, если задать координаты центра. Отметим, что кнопки нижнего меню работают как в режиме основного окна (рис. 4.1.1), так и в режиме **Wiev Axes**.

4.1.2. Просмотр пакетных вейвлетов

Раздел **Wavelet Packet Display** главного меню посвящен вейвлет-пакетным функциям. Он работает совершенно аналогично **Wavelet Display**. Напомним понятие пакетных вейвлетов. Пусть $\varphi(x)$ – ортогональная масштабитующая функция и $\psi(x)$ – соответствующий вейвлет. Они определяют следующее ортогональное разложение:

$$L^2(\mathbf{R}) = V_0 \oplus W_0 \oplus W_1 \oplus ... \oplus W_j \oplus ... ,$$

где V_0 – пространство, порожденное функциями $\varphi(x - n)$, $n \in \mathbf{Z}$ и W_j – пространства, порожденные функциями $\{\psi_{jn}(x) = \sqrt{2^j}\psi(2^j x - n)$, $j, n \in \mathbf{Z}$. Эти пространства имеют в определенном смысле разный «размер». Например, два первых пространства, V_0 и W_0, имеют одинаковый «размер» – они порождены целочисленными сдвигами функций $\varphi(x)$, $\psi(x)$, то есть функциями $\varphi(x - n)$, $\psi(x - k)$, а третье пространство W_1 «в два раза больше», оно порождено полуцелыми сдвигами функции $\psi(2x)$, то есть функциями $\psi(2x - k)$. Вейвлет-пакетные функции возникли из желания исправить это неравенство. С этой целью пространство W_1 раскладывается на два ортогональных подпространства, каждое из которых порождается сдвигами одной функции:

$$W_1 = W_{10} \oplus W_{11},$$

где W_{10} и W_{11} порождаются целочисленными сдвигами, соответственно, функций:

$$w_2(x) = \sqrt{2}\sum_{n\in\mathbf{Z}} h_n \psi(2x - n) \quad \text{и} \quad w_3(x) = \sqrt{2}\sum_{n\in\mathbf{Z}} g_n \psi(2x - n).$$

Продолжая эту процедуру, мы получаем разложение пространства $L^2(\mathbf{R})$ в прямую сумму ортогональных замкнутых подпространств

$$L^2(\mathbf{R}) = \Omega_0 + \Omega_1 + \Omega_2 + ...,$$

где $\Omega_0 = V_0$ и $\Omega_1 = W_0$ и каждое следующее пространство Ω_n порождено целочисленными сдвигами одной функции $w_n(x)$. При этом функции $w_n(x)$ определяются рекурсивно. Полагаем $w_0(x) = \varphi(x)$ и $w_1(x) = \psi(x)$ и определим *вейвлет-пакетные* функции для любого натурального n следующим образом:

$$w_{2n}(x) = \sqrt{2}\sum_{k\in\mathbf{Z}} h_k w_n(2x - k), \quad w_{2n+1}(x) = \sqrt{2}\sum_{n\in\mathbf{Z}} g_n w_n(2x - n).$$

Подробнее об этом см. в разделе 2.4.5.

Wavelet Packet Display главного меню позволяет просмотреть общую информацию о вейвлет-пакетных функциях и построить их графики (рис. 4.1.4).

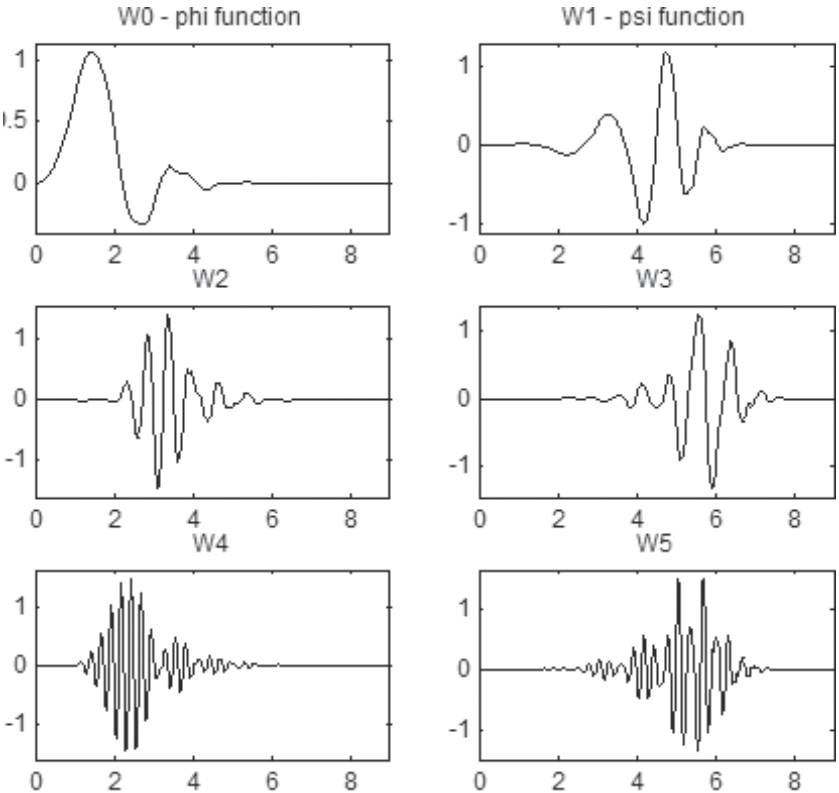


Рис. 4.1.4. Вейвлет-пакетные функции для db5

4.2. Продолжение сигналов и изображений (Extension)

Как известно, применение фильтров к сигналам конечной длины может привести к тому, что граничные значения выходного сигнала могут быть не определены или искажены. Проблема может быть решена продолжением сигнала несколькими способами: периодическим продолжением, симметричным отражением относительно конца, постоянным продолжением вправо или влево и т. д.

Для продолжения сигнала *X* или изображения из командной строки служит команда **wextend**. Она имеет следующий синтаксис:

Y = wextend(TYPE,MODE,X,L,LOC)
Y = wextend(TYPE,MODE,X,L)

Описание параметров.

Параметр TYPE может принимать значения:

- 1, '1', '1d' или '1D' – для одномерного сигнала;
- 2, '2', '2d' или '2D' – для двумерного сигнала;
- 'ar' or 'addrow' – добавление строки;
- 'ac' or 'addcol' – добавление столбца.

Параметр MODE может принимать значения:

- 'sym' или 'symh' – симметричное расширение относительно точки, расположенной на расстоянии $^1/_2$ от концевой точки, в этом случае концевая точка «повторяется дважды» (default mode);
- 'symw' – симметричное расширение относительно граничной точки;
- 'asym' или 'asymh' – симметричное продолжение относительно точки, расположенной на расстоянии $^1/_2$ от концевой точки ($t = n + {}^1/_2$) с отражением по оси значений Ox относительно значения $x = 0$, в этом случае концевая точка «повторяется», а значения меняют знак;
- 'zpd' – дополнение нулями;
- 'spd' или 'sp1' – гладкое продолжение, то есть линейное продолжение в направлении первой производной;
- 'sp0' – непрерывное продолжение (постоянное продолжение первого значения – влево и последнего значения – направо);
- 'ppd' – периодическое продолжение значений;
- 'per' – отличается от 'ppd', только если длина сигнала нечетна. Тогда wextend добавляет дополнительное значение, равное последнему значению справа, и выполняет расширение, используя режим 'ppd'.

Параметры L и Loc могут быть следующими:

- Для TYPE = {1, '1', '1d' or '1D'}:
  - LOC = 'l' (или 'u') для расширения влево (или вверх);
  - LOC = 'r' (или 'd') для расширения вправо (или вниз);
  - LOC = 'b' для расширения в обе стороны;
  - LOC = 'n' нулевое расширение;
  - по умолчанию LOC = 'b';
  - L есть длина расширения.

- Для TYPE = {'ar', 'addrow'} и {'ac', 'addcol'}:
  - LOC есть местоположение 1D-расширения;
  - по умолчанию LOC = 'b';
  - L есть число строк (столбцов) для добавления.
- Для TYPE = {2, '2', '2d' или '2D'}:
  - LOC = [LOCROW,LOCCOL], где LOCROW и LOCCOL – местоположение 1D-расширения или 'n';
  - по умолчанию LOC = 'bb';
  - L = [LROW,LCOL], где LROW есть число строк для добавления и LCOL есть число столбцов для добавления.

Продолжение сигнала или изображения удобно сделать в разделе **Extension** главного меню пакета Wavelet Toolbox. Для запуска этого приложения нужно вызвать главное меню командой wavemenu и нажать кнопку **Signal extension**. На правой части окна будет отображено следующее:

- длина сигнала (**Length**);
- следующая степень двойки (Next Power of 2);
- предыдущая степень двойки (Previous Power of 2);
- требуемая длина сигнала (Desired Length), необходимо выбрать;
- направление продолжения (Direction to extend), необходимо выбрать;
- мода продолжения (**Extension Mode**), необходимо выбрать.

Раздел **Signal extension** позволяет выбрать следующие способы *продолжения сигнала* (изображения):

- **Symmetric (Whole-Point)** – симметричное продолжение относительно концевой точки;
- **Symmetric (Half-Point)** – симметричное продолжение относительно точки, расположенной на расстоянии $^1/_2$ от концевой точки, в этом случае концевая точка «повторяется»;
- **Antisymmetric (Whole-Point)** – симметричное продолжение относительно концевой точки ($t = n$) с отражением по оси значений Ox относительно значения на конце $x = x_n$;
- **Antisymmetric (Half-Point)** – симметричное продолжение относительно точки, расположенной на расстоянии $^1/_2$ от концевой точки ($t = n + {}^1/_2$) с отражением по оси значений Ox относительно значения $x = 0$, в этом случае концевая точка «повторяется», а значения меняют знак;
- **Periodic** – периодическое продолжение;
- **Zero Padding** – продолжение нулями;

- **Continuous** – непрерывное продолжение (постоянное продолжение первого значения влево и последнего значения – направо);
- **Smooth** – гладкое продолжение, линейное продолжение в направлении первой производной;
- **For SWT** – для SWT. Как известно, стационарное вейвлет-преобразование глубины разложения L требует, чтобы длина сигнала делилась бы на 2^L. Поэтому режим продолжения For SWT делает периодическое продолжение до ближайшего большего числа, которое делится на 2^L.

После выбора параметров продолжения необходимо нажать кнопку **Extend**. Если требуемая длина меньше, чем длина сигнала, то используются методы усечения и кнопка **Truncate**.

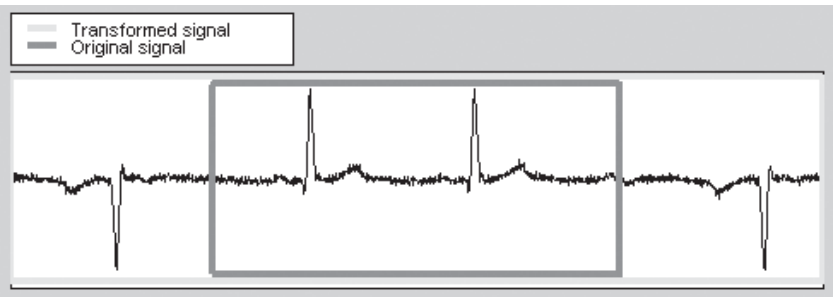


Рис. 4.2.1. Антисимметричное (Whole-Point) продолжение

В появившемся окне (рис. 4.2.1) исходный сигнал изображен в красной рамке, а продолженный – в желтой. Продолженный сигнал можно сохранить на диске.

В случае изображений параметры продолжения нужно указывать отдельно для горизонтали и для вертикали (рис. 4.2.2).

4.3. Одномерный вейвлет-анализ

Раздел одномерного вейвлет-анализа представлен пятью программами:

- **Wavelet 1D** – для вейвлет-разложения и анализа дискретного сигнала;
- **Wavelet Packet 1D** – для пакетного вейвлет-разложения и анализа;

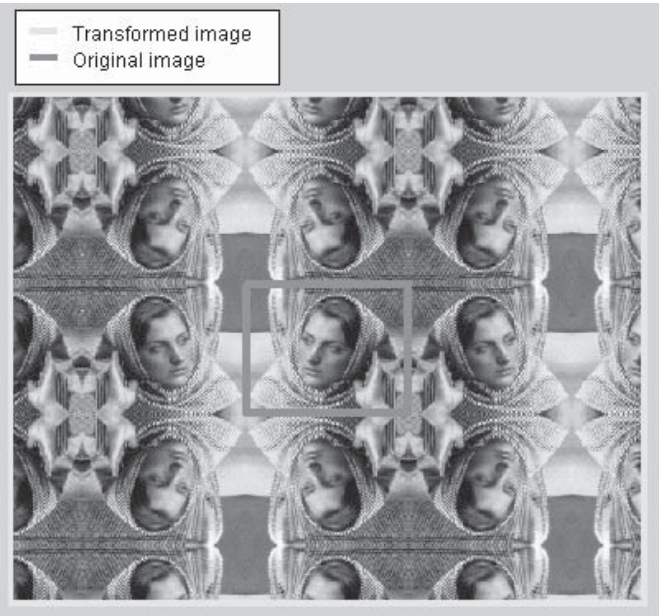


Рис. 4.2.2. Симметричное продолжение изображения

- **Continuous Wavelet 1D** – для непрерывного вейвлет-анализа;
- **Complex Continuous Wavelet 1D** – для комплексного непрерывного вейвлет-анализа;
- **Continuous Wavelet 1D (Usimg FFT)** – для непрерывного вейвлет-анализа на основе дискретного преобразования Фурье.

Рассмотрим все эти возможности главного меню wavemenu.

4.3.1. Одномерный вейвлет-анализ (Wavelet 1D)

Раздел **Wavelet 1D** главного меню предназначен для вейвлет-разложения и анализа дискретного сигнала. Для запуска **Wavelet 1D** нужно вызвать главное меню командой **wavemenu** и нажать кнопку **Wavelet 1D**. Запускается графическая среда, удобная для работы большого комплекса команд по анализу сигналов. Окно графического интерфейса содержит обычное меню вверху, специальные кнопки меню справа и внизу, а также основное поле для визуализации результатов анализа.

Верхнее меню. Оно содержит возможности, обычные для Windows-приложений. Используя меню **File**, вы сможете загрузить дан-

ные, которые предполагаются к анализу. Значительное удобство заключается в том, что можно использовать демонстрационные сигналы. Их можно найти в меню **File/Example Analysis**. Хотя в этих примерах уже проведен вейвлет-анализ, выбранный демонстрационный сигнал можно проанализировать заново. После обработки можно сохранить коэффициенты, синтезированный сигнал и разложение.

Кнопки нижнего меню работают точно так же, как и в рассмотренном выше разделе **Wavelet Display**. Они предназначены для более детального рассмотрения графиков. Кнопка **Wiev Axes** запускает дополнительное окно, кнопки которого позволяют открыть соответствующий график «во все окно». При этом левой кнопкой мыши можно выбрать фрагмент графика и кнопками **X+**, **X–**, **Y+**, **Y–**, **XY+**, **XY–** увеличить либо уменьшить его масштаб по координатным осям X и Y отдельно или одновременно.

Правое меню активируется после того, как будет загружен сигнал. Меню предназначено для задания вейвлета, при помощи которого будет анализироваться сигнал, уровня разложения, задания видов отображения на экране вейвлет-коэффициентов и указания целей анализа: получение статистических характеристик, гистограмм, сжатия сигнала или удаления шума.

Рассмотрим работу данного комплекса на примере анализа электрокардиограммы.

Вейвлет-разложение

Загрузим сигнал ЭКГ, содержащий два комплекса QRS (файл ECG.mat). Появится график сигнала, и справа в верхнем поле будет указана длина выборки. Далее выбираем вейвлет (**Wavelet**), его порядок и уровень (**Level**) вейвлет-разложения (по умолчанию 5). После этого нужно нажать на кнопку **Analyse**, для того чтобы произвести вейвлет-разложение. Результаты разложения будут изображены на основном поле графического окна.

На следующем рисунке приведены результаты разложения с использованием вейвлета sym3 до третьего уровня. Приведены график исходного сигнала **s** и графики компонент сигнала, восстановленные только по одной ветви аппроксимирующих $\mathbf{a}_3$ или детализирующих коэффициентов $\mathbf{d}_1$, $\mathbf{d}_2$, $\mathbf{d}_3$. Это не графики вейвлет-коэфициентов!

Выбор различных видов разложения (Display mode)

В данном меню (справа в центре) имеется возможность выбора следующих видов вейвлет-разложения:

- **Full Decomposition Mode** – режим по умолчанию, показывает графики аппроксимации самого верхнего уровня разложения и детализирующих компонент сигнала (рис. 4.3.1);

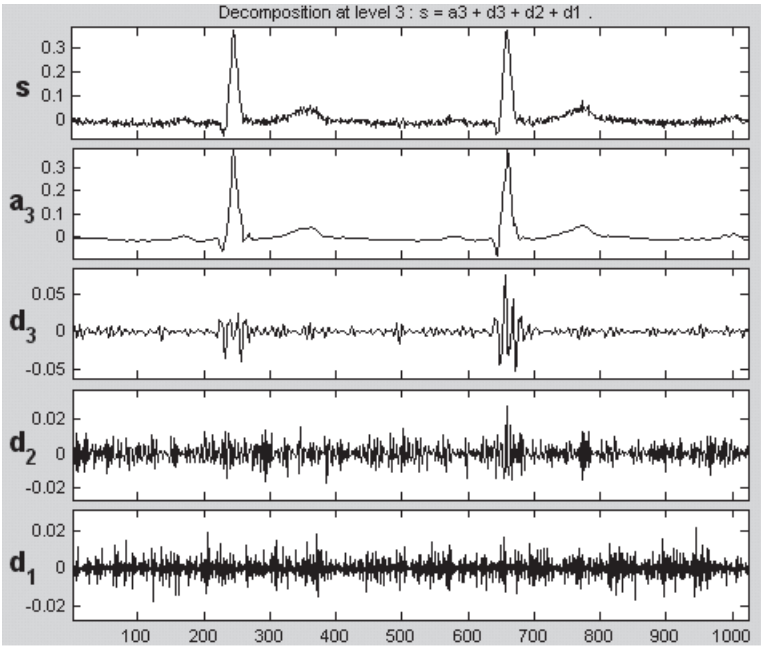


Рис. 4.3.1. Дискретное вейвлет-разложение до 3-го уровня

- **Separate Mode** – показывает детали и все аппроксимации в отдельных столбцах;
- **Superimpose Mode** – показывает детали графиками в одном окне разными цветами. Для всех аппроксимаций аналогичные графики;
- **Tree Mode** – показывает дерево вейвлет-разложения сигнала, график сигнала и одну дополнительную компоненту по выбору. Выбрать компоненту сигнала можно мышкой, щелкнув по обозначению компоненты на дереве;
- **Show and Scroll Mode (Stem Cfs)** – отображает три окна. Первое показывает первоначальный сигнал, наложенный на выбранное приближение. Второе окно показывает выбранные детали. Третье окно показывает вейвлет-коэффициенты.

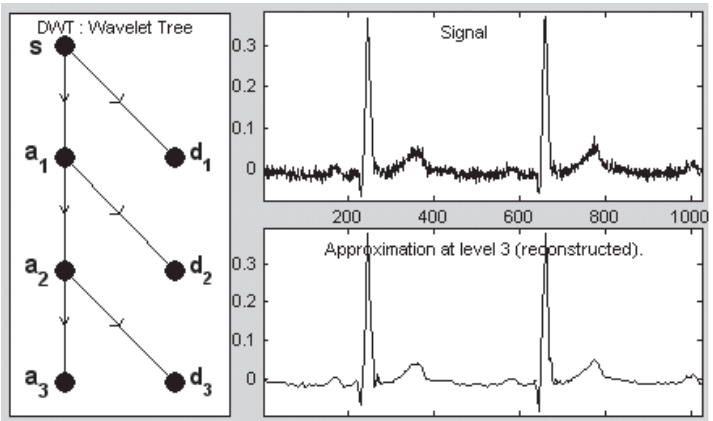


Рис. 4.3.2. Дерево дискретного вейвлет-разложения

Статистические характеристики коэффициентов разложения

В комплексе программ **Wavelet 1-D** предусмотрена возможность вычисления статистических характеристик компонент разложения. Для этого необходимо нажать кнопку **Statistic** справа. Появится новое окно **Wavelet 1-D Statistics**, содержащее статистические данные об исходном сигнале. Напомним основные понятия теории вероятностей, которые отражаются в этом окне.

Случайной величиной называют такую величину, которая в результате испытаний принимает то или иное значение, которое зависит от случайных обстоятельств и заранее не может быть предсказано. *Функцией распределения* случайной величины X называется функция $F(x)$, равная вероятности того, что случайная величина X примет значение, меньшее, чем x: $F(x) = P\{X < x\}$, $x \in R$. Из определения сразу следует, что $F(x)$ – возрастающая функция, такая что $F(-\infty) = 0$ и $F(+\infty) = 1$. Вероятность попадания случайной величины X в интервал $[x_1, x_2)$ равна разности значений функции $F(x)$, $P\{x_1 \le x < x_2\} = F(x_2) - F(x_1)$.

Плотность распределения вероятности случайной величины X есть производная функции распределения, $p(x) = F'(x)$. Вероятность попадания случайной величины X в промежуток $[x_1, x_2)$ равна интегралу $p(x)$ по этому промежутку

$$P\{x_1 \le x < x_2\} = \int_{x_1}^{x_2} p(x)dx.$$

В частности, можно считать, что вероятность попадания случайной величины X в малый промежуток длины Δx равна $\Delta P = p(x)\Delta x$, где x – некоторая точка выбранного промежутка.

Числовые характеристики случайной величины *X*:

- *математическое ожидание* $M[x] = \int_{-\infty}^{\infty} xp(x)dx$;
- *дисперсия* – это математическое ожидание квадрата отклонения случайной величины X от математического ожидания $M[x]$

$$D[x] = M[(X - M[x])^2] = \int_{-\infty}^{\infty} (x - M[x])^2 p(x)dx;$$

- *среднее квадратичное отклонение* $\sigma_x = \sqrt{D[x]}$;
- *медиана* – это такое значение x_0, которое делит пополам площадь под графиком плотности распределения $y = p(x)$. Другими словами, медиана x_0 является корнем уравнения $F(x) = {}^1/_2$;
- *мода* случайной величины – это значение x_m, при котором плотность распределения $p(x)$ имеет максимальное значение.

Предположим, что на основании некоторых дополнительных данных о характере случайной величины X можно сделать обоснованное заключение о виде закона распределения или плотности вероятности случайной величины X. Однако параметры этого распределения неизвестны и их нужно оценить по результатам наблюдений, представленных в виде конечной выборки $x_1, ..., x_n$. В этом случае используют выборочные точечные оценки числовых характеристик случайной величины. Рассмотрим наиболее распространенные оценки случайной величины.

- *Выборочное среднее*

$$\bar{m}_X = \frac{1}{n}\sum_{k=1}^{n} x_k.$$

- *Выборочная дисперсия*

$$\bar{D}_X = \frac{1}{n}\sum_{k=1}^{n} (x_k - \bar{m}_X)^2 = \frac{1}{n}\sum_{k=1}^{n} x_k^2 - (\bar{m}_X)^2.$$

- *Выборочное стандартное отклонение* $\bar{\sigma}_X = \sqrt{\bar{D}_X}$.
- *Несмещенная выборочная дисперсия*

$$s^2 = \frac{1}{n-1}\sum_{k=1}^{n} (x_k - \bar{m}_X)^2.$$

Тогда s – *несмещенное стандартное отклонение*. Заметим, что оценка $\bar{m}_X$ является несмещенной, а оценка $\bar{D}_X$ – смещенная:

$$M[\overline{D}_X] = \frac{n-1}{n} D_X,$$

где D_X – дисперсия (невыборочная) случайной величины X.

Оценка закона распределения и плотности распределения случайной величины. Чтобы получить представление о законе распределения случайной величины на основе наблюдений ее значений, поступают следующим образом. Область экспериментальных значений случайной величины разбивается на m одинаковых интервалов длиной Δx, и вычисляется относительная плотность точек в каждом интервале $p_k = n_k/(n\Delta x)$ – отношение относительных частот попадания в этот интервал к длине интервала. Эта величина является оценкой плотности распределения. Действительно, вероятность попадания в k-й интервал равна относительной частоте попадания в этот интервал $\Delta P = p_k \Delta x = n_k/n$. Найденные таким образом значения p_k изображают в виде *гистограммы* – над каждым k-м интервалом строится прямоугольник с высотой p_k. Выбор количества интервалов зависит от величины выборки n. Одна из рекомендуемых формул для числа m следующая: $m \approx 5\ln n$, в Matlab берется $m = n/4$. По виду гистограммы делают предположение о типе функции распределения случайной величины. Сопоставляя гистограмму с графиками различных теоретических распределений и подбирая наиболее близкое, делают вывод о характере распределения неизвестной случайной величины.

При статистической обработке вейвлет-коэффициентов считается, что набор коэффициентов разложения представляет конечную выборку значений случайной величины. Вид гистограммы отражает плотность распределения, а вид комулятивной гистограммы отражает вид функции распределения.

Для вызова окна, предназначенного для изучения статистик, необходимо нажать кнопку **Statistic** справа. Появится новое окно **Wavelet 1D Statistics**, содержащее статистические данные об исходном сигнале (по умолчанию). Статистические характеристики можно получить для исходного сигнала, синтезированного сигнала или коэффициентов разложения. Для этого нужно выбрать справа компоненту, статистику которой необходимо получить. Кнопка **Approximation** – для выбора коэффициентов аппроксимации, и кнопка **Detail** – для выбора детализирующих коэффициентов. В случае выбора коэффициентов разложения необходимо указать их уровень (**Level**). Затем нужно нажать кнопку **Show Statistics**.

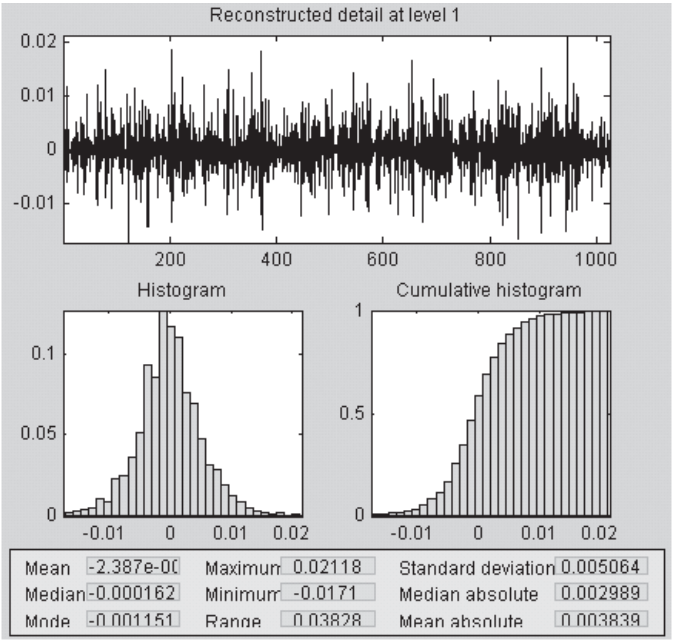


Рис. 4.3.3. Статистические характеристики компонент сигнала

На основном поле появившегося окна будут изображены график выбранной компоненты и гистограммы относительных частот. Для построения гистограммы область значений сигнала разбивается, по умолчанию, на 30 равных частей. Число промежутков разбиения можно задать самостоятельно в поле **Number of Bins**. Внизу приведены числовые характеристики компонент сигнала, такие как среднее, мода, медиана, максимальное и минимальное значения, диапазон, среднеквадратичное отклонение.

Отметим, что все эти характеристики в случае вейвлет-коэффициентов кардиосигнала имеют диагностическую значимость, поскольку они характеризуют высокочастотные флуктуации, обычно незаметные на кардиограмме.

***Гистограммы* (*Histogram*)**

Эта кнопка позволяет построить гистограммы всех компонент в совокупности, что бывает удобно для их сравнительной оценки.

Сжатие сигнала

Для этого необходимо нажать кнопку **Compress** справа. Появится новое окно **Wavelet 1-D Compression**, правое меню которого содержит ряд возможностей для задания параметров сжатия. Рассмотрим сначала метод глобальной пороговой обработки (**Global Thresholding**) детализирующих коэффициентов, обеспечивающий быстрое и простое сжатие.

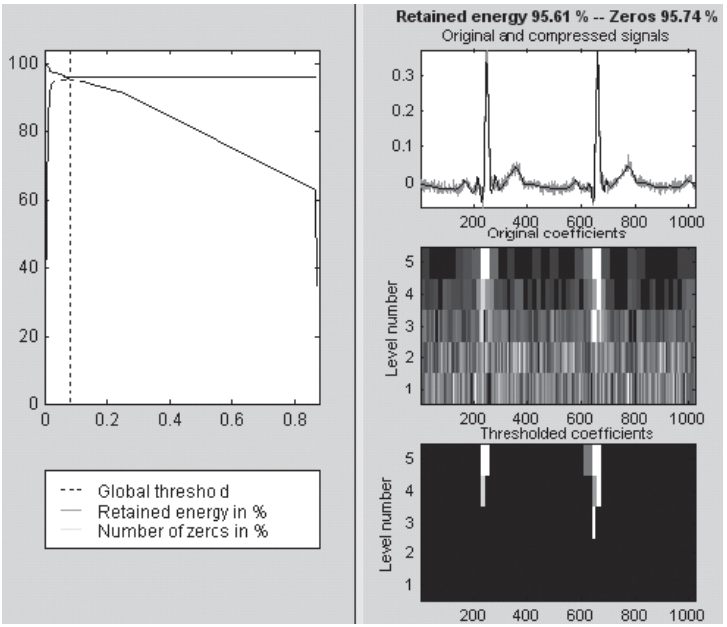


Рис. 4.3.4. Глобальная пороговая обработка

Значение порога устанавливается автоматически. В то же время это значение можно выбрать вручную, используя слайдер, расположенный справа (**Select Global Threshold**), либо вручную левой кнопкой мыши, используя вертикальную желтую штрих-линию в левом верхнем окне. Синяя кривая в этом окне есть график количества нулей (в процентах) в зависимости от значения порога – горизонтальная ось. В верхнем окне справа изображаются графики исходного и сжатого (желтый цвет) сигналов. Два нижних окна справа используются для изображения исходных и оставшихся коэффициентов.

Мы видим (рис. 4.3.4), что при выборе вейвлета sym4 и разложения до 3-го уровня сигнал сжимается почти в 20 раз с сохранением 95,5% энергии сигнала. Несмотря на очень малое значение порога, удалено 95,7% вейвлет-коэффициентов.

Кнопка **Residuals** дает возможность оценить характеристики разности между исходным и сжатым сигналами: график разности, гистограмма, автокорреляционная функция, частотный спектр быстрого преобразования Фурье и числовые статистические характеристики, такие как среднее, мода, медиана, среднеквадратичное отклонение (см. рис. 4.3.3). Все эти величины в случае кардиосигнала имеют диагностическое значение.

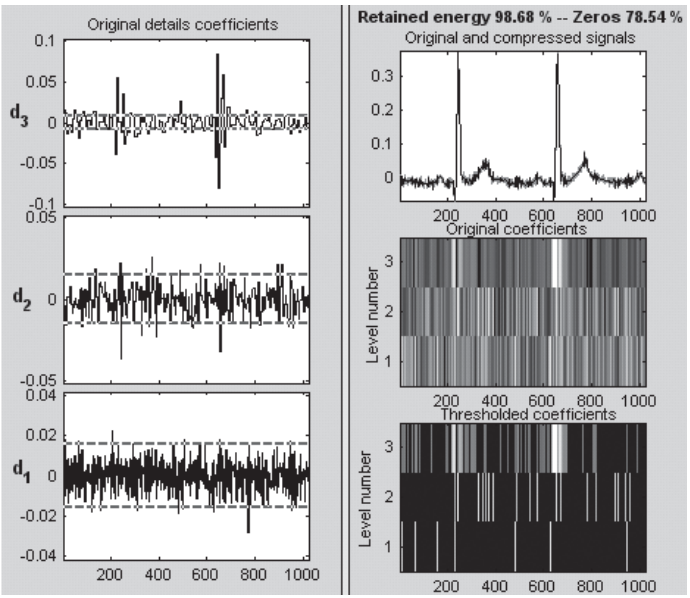


Рис. 4.3.5. Пороговая обработка от уровня к уровню

Более гибким методом является пороговая обработка в зависимости от уровня (**By Level Thresholding**). При выборе этого метода появляется окно, содержащее графики исходного сигнала и детализирующих коэффициентов $\mathbf{d}_1$, $\mathbf{d}_2$, $\mathbf{d}_3$.

Имеется несколько автоматических методов выбора порога в меню **Select Thresholding**. В то же время это значение можно выбрать вруч-

ную отдельно для каждого уровня, если использовать слайдеры, расположенные справа (**Select, Sparsity**), либо вручную левой кнопкой мыши при помощи горизонтальных желтых штрих-линий в левых окнах с изображением вейвлет-коэффициентов. Каждое из этих окон может быть увеличено нажатием кнопки **View Axes**. Кнопка **Interval Depend Threshold Setting**, находящаяся ниже процентных данных, открывает дополнительное окно для установки различных порогов на разных частях одного сигнала.

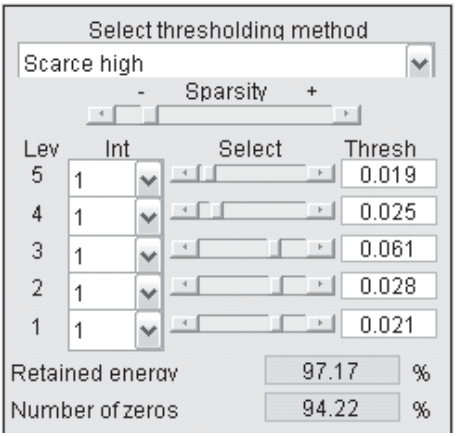


Рис. 4.3.6. Выбор порогов вручную на каждом уровне

Установив для выбранного сигнала ЭКГ разложение до уровня 5 и выбрав вручную пороговые значения, как на рис. 4.2.6, мы видим, что сжатие удаляет большую часть шума, сохраняет почти 100% энергии сигнала и количество оставшихся вейвлет-коэффициентов составляет около 5% от объема исходного сигнала. Сжатая кардиограмма сохранила все свойства, используемые для контурного анализа ЭКГ. Кардиосигналы, полученные при помощи кардиографов высокого разрешения, имеют достаточно большие объемы. Поэтому проблема их хранения является актуальной. В нашем случае сигнал сжат почти в 20 раз, что показывает высокую эффективность вейвлет-методов в обработке и хранении медицинских данных.

Кнопка **Residuals** дает возможность оценить характеристики разности между исходным и сжатым сигналами.

Удаление шума

Для этого необходимо нажать кнопку **De-noise** справа на основном окне **Wavelet 1D**. Появится новое окно **Wavelet 1D De-noise**, правое меню которого содержит ряд возможностей для задания параметров удаления шума. Во-первых, нужно выбрать стратегию пороговой обработки в меню справа **Select Thresholding Method**. Затем необходимо задать тип пороговой обработки: мягкий или жесткий. Далее нужно выбрать тип шума. В этом случае пороговые значения будут определены автоматически. Нажимаем кнопку **De-noise** и получаем сигнал, очищенный от шума. Он изображается в одном окне вместе с исходным. Удаление шума можно провести вручную отдельно для каждого уровня при помощи слайдеров, расположенных справа, либо вручную левой кнопкой мыши при помощи горизонтальных желтых штрих-линий в левых окнах с изображением вейвлет-коэффициентов.

Кнопка **Residuals** дает возможность оценить характеристики разности между исходным и сжатым сигналами.

Отметим очень полезную особенность. Можно левой кнопкой мыши выделить фрагмент сигнала в правом верхнем окне. Затем, если его увеличить кнопкой **XY+** нижнего меню, то мы получим вейвлет-разложение этого фрагмента сигнала и можно провести удаление шума данного фрагмента.

Кнопка **Int. dependent threshold settings** (установка интервальных порогов) выводит дополнительное окно, в котором изображаются компоненты сигнала. Кнопка **Level** позволяет выбрать уровень коэффициентов. На основном поле имеются две горизонтальные линии

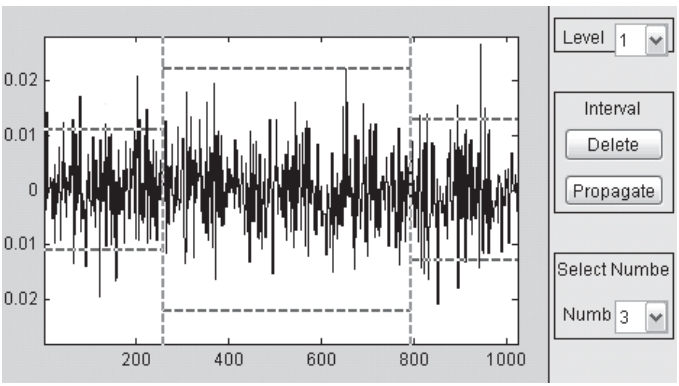


Рис. 4.3.7. Установка интервальных порогов

(пороги), которые можно изменять, удерживая их правой кнопкой мыши. В меню **Select Number of Intervals** можно выбрать количество интервалов, на каждом из которых будет выбираться свое значение порога. Положения границ интервалов (штрих-линия красного цвета) можно установить, удерживая их правой кнопкой мыши.

Полная визуализация процедур сжатия и удаления шума позволяет без труда обработать сигнал. Результаты могут быть сохранены с использованием возможностей обычного Windows-меню.

4.3.2. Одномерный пакетный вейвлет-анализ

Раздел **Wavelet Packet 1D** главного меню предназначен для пакетного вейвлет-разложения и пакетного анализа дискретного сигнала. Для запуска **Wavelet Packet 1-D** нужно вызвать главное меню командой **wavemenu** и нажать кнопку **Wavelet Packet 1D**. Запускается графическая среда, удобная для применения комплекса команд по пакетному анализу сигналов. Окно графического интерфейса содержит обычное меню вверху, специальные кнопки меню справа и внизу, а также основное поле для визуализации результатов анализа.

Верхнее меню. Оно содержит возможности, обычные для Windows-приложений. При помощи меню **File** можно загрузить данные в виде сигнала или уже произведенного разложения. Можно также использовать демонстрационные сигналы.

Кнопки нижнего меню работают точно так же, как и в рассмотренных выше разделах. Они предназначены для более детального рассмотрения графиков.

Правое меню активируется после того, как будет загружен сигнал. Для этого можно выбрать демосигнал либо тот, который необходимо проанализировать.

Вейвлет-разложение

Загрузим, используя меню **File/Example Analysis**, сигнал sumsin.mat из MATLAB-каталога toolbox/wavelet/wavedemo. Появится график сигнала, и справа в верхнем поле будет указана длина выборки. Далее нужно выбрать вейвлет (**Wavelet**) и его порядок, возьмем db4, уровень (**Level**) вейвлет-разложения (по умолчанию 3) и тип энтропии (**Entropy**). Доступные типы энтропии сигнала *s*.

1. Энтропия Шеннона ('shannon') $E1(s) = -\sum_i s_i^2 \log(s_i^2)$.
2. Норма пространства l^p, $p \geq 1$ ('norm'), $E2(s) = \sum_i |s_i|^p = \|s\|_p^p$. Требует задания степени p.

3. Логарифм энергии ('log energy') $E3(s) = \sum_i \log(s_i^2)$ при следующем соглашении: log(0) = 0.
4. Пороговая энтропия ('threshold') $E4(s_i) = 1$, если $|s_i| > \varepsilon$ и 0 – в противном случае, тогда $E4(s)$ есть число элементов сигнала, величина которых больше порогового значения ε. Требует задания порога ε.
5. Энтропия "SURE" (Stein's Unbiased Risk Estimate). Пороговая энтропия, когда величина порога равна $\sqrt{2\log_e(n\log_2(n))}$, где n есть длина выборки.

Для данного примера выберем энтропию 'shannon'. После выбора всех параметров нужно нажать на кнопку **Analyse**, для того чтобы произвести пакетное вейвлет-разложение. Результаты разложения будут изображены на основном поле графического окна (рис. 4.3.8). Вверху слева будет построено дерево пакетного разложения. Узлы дерева являются активными, при выборе мышкой узла дерева в нижнем левом окне изображается график коэффициентов этого узла (см. опцию **Node Action**). Вверху справа изображается график исходного сигнала, а ниже – цветовые изображения коэффициентов в терминальных узлах дерева, блоками, раскрашенными в соответствии с величиной коэффициентов.

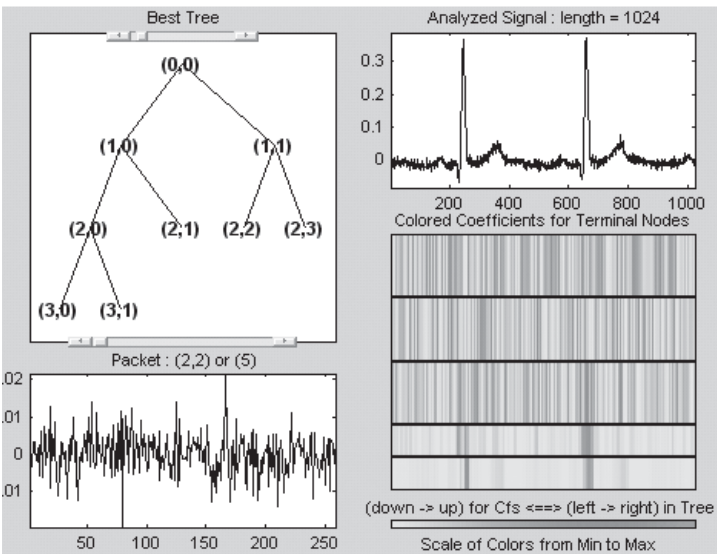


Рис. 4.3.8. Пакетный вейвлет-анализ

Все графические окна могут быть увеличены «во весь экран» четырьмя кнопками в группе **Full Size**.

Возможности раздела для обработки сигнала

Правое меню предназначено для визуализации, обработки и анализа сигнала.

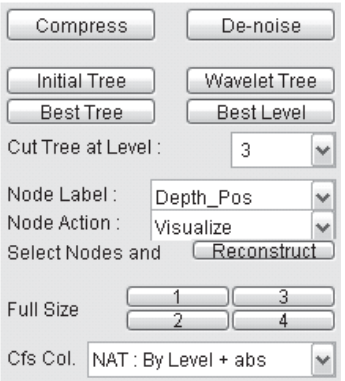


Рис. 4.3.9. Возможности правого меню

1. Сжатие и удаление шума. Опции **Compress** и **De-noise** открывают новые окна, предназначенные для сжатия и удаления шума, вполне аналогичные тем, которые используются для таких же целей в обычном вейвлет-анализе из раздела **Wavelet 1D**.

2. Операции с деревом разложения. Далее идут четыре кнопки, которые позволяют получить следующее:

- **Initial Tree** – начальное дерево разложения;
- **Best Tree** – наилучшее дерево разложения;
- **Wavelet Tree** – дерево вейвлет-разложения (не пакетного);
- **Best Level** – наилучший уровень разложения;
- **Cut Tree at Level** – обрезание ветвей дерева на задаваемом уровне.

3. Выбор меток в узлах дерева (Node Label). Можно выбрать нумерацию парой чисел, уровень – номер на уровне (по умолчанию), либо обычную нумерацию сверху и слева направо, можно указать длину сигнала, энтропию, энергию или тип узла.

4. Выбор различных вариантов работы с данными в узлах дерева (Node Action). Предусмотрены следующие возможности данного меню.

- **Visualize** – визуализация. Если мышкой выбрать узел в дереве, то в нижнем окне появляется соответствующий сигнал.
- **Split/Merge** – расщепление/объединение. Выбранный терминальный узел может быть расщеплен, это увеличит пакетное дерево. Выбор нетерминального узла приводит к объединению всех узлов пакетного дерева ниже выбранного.
- **Recons.** Если выбрать узел в дереве, то в нижнем окне появляется сигнал, восстановленный по одному выбранному узлу.
- **Select On/Off** – выбор включен/отключен. Применяется для того, чтобы можно было выбрать много узлов в пакетном де-

реве. После этого можно восстановить сигнал по данным в выбранных узлах, используя кнопку **Reconstruct** в основном окне. Синтезированный сигнал изображается в нижнем окне. Чтобы очистить все предыдущие выборы, достаточно еще раз щелкнуть по ним мышкой.

- **Statistics** – статистические характеристики данных в узлах дерева. При выборе узла в дереве открывается новое окно, на котором изображены данные в выбранном узле, гистограмма и числовые статистические характеристики.
- **View Col. Cfs.** Эта опция удаляет все цветные изображения коэффициентов и позволяет увидеть изображение коэффициентов только выбранного узла в дереве.

5. Реконструкция сигнала по выбранным узлам (Select Nodes and). После того как выбраны узлы, нужно нажать кнопку **Reconstruct** для восстановления сигнала по выбранным данным.

6. Полный размер Full Size. Четыре кнопки для просмотра графиков в увеличенном виде.

7. Окраска коэффициентов, меню Cfs. Col. Выбор вариантов для цветового изображения коэффициентов в правом нижнем окне в естественном порядке вейвлет-пакетных функций (NAT) или в соответствии с частотой (FRQ).

Напомним, что пакетные вейвлеты имеют три индекса, $W_{j,n,k}$. Индексы j и k имеют тот же смысл, что и в обычных вейвлетах: j – масштабный параметр, k – параметр сдвига. Новый параметр n характеризует частоту колебаний. Таким образом, $W_{j,n,k}$ анализирует флуктуации сигнала вблизи значения $2^j k$, в масштабе 2^j и по частоте в соответствии со значением последнего параметра n. Часто бывает удобно расположить коэффициенты не в соответствии с естественным порядком параметра n, а когда основная частота увеличивается монотонно от низких частот к высоким. Этому варианту и соответствует выбор параметра FRQ.

4.3.3. Непрерывный вейвлет-анализ (Continuous Wavelet 1D)

Раздел **Continuous Wavelet 1D** главного меню предназначен для непрерывного вейвлет-разложения и анализа сигнала. Напомним, что если s – сигнал и ψ – вейвлет, то при непрерывном преобразовании вейвлет-коэффициенты сигнала s, соответствующие масштабному коэффициенту a и положению b, определяются формулой

$$C_{a,b} = \frac{1}{\sqrt{a}} \int_{-\infty}^{\infty} s(t)\, \overline{\psi\left(\frac{t-b}{a}\right)}\, dt.$$

Практически сигнал $s(t)$ задается выборкой значений $\{s_k\}$. Тогда в качестве $s(t)$ мы берем кусочно-постоянную функцию, $s(t) = s(k\Delta t) = s_k$, при $t \in [\, k\Delta t, (k+1)\Delta t)$. Параметр b пробегает значения с шагом Δt, $b = 1{:}\, ls$, $ls = \text{length}(s)$. Масштабный параметр a также меняется, пробегая дискретный ряд значений.

Начало работы

Для запуска **Continuous Wavelet 1D** нужно вызвать главное меню командой **wavemenu** и нажать кнопку **Continuous Wavelet 1D**. Запускается графическая среда, удобная для работы большого комплекса команд по анализу сигналов. Окно графического интерфейса содержит обычное меню вверху, специальные кнопки меню справа и внизу, а также основное поле для визуализации результатов анализа.

Правое меню активируется после того, как будет загружен сигнал. Для этого можно выбрать демосигнал (из каталога MATLAB toolbox/wavelet/wavedemo) либо тот, который необходимо проанализировать. Рассмотрим работу данного комплекса на примере анализа электрокардиограммы. Загрузим сигнал ЭКГ, содержащий два комплекса QRS. Появится график сигнала, и справа в верхнем поле будет указана длина выборки. Выбираем вейвлет (**Wavelet**) и его порядок. Значение по умолчанию периода выборки **Sampling** равно 1 (секунда), мы ставим значение 0.002, поскольку частота дискретизации нашего кардиосигнала равна 500 отсчетов в секунду. Далее устанавливаем параметры изменения масштаба (параметра a) по схеме **Step by Step Mode**: минимальное значение **Min**, шаг **Step**, с которым меняется масштаб, и максимальное значение масштаба **Max**. Изменение масштаба можно задать также по степеням двойки либо вручную, указывая прямо нужные значения масштабов. После этого нажимаем на кнопку **Analyse**, для того чтобы произвести вейвлет-разложение. Результаты разложения будут изображены на основном поле графического окна. Можно изменить выбранные параметры и провести вейвлет-разложение повторно.

На следующем рис. 4.3.10 приведены результаты разложения с использованием гауссова вейвлета gaus 3.

Выбрано изменение масштаба с шагом 1 от значения $a = 1$ до $a = 100$. Вверху – график самого сигнала, ниже – графическое изображение коэффициентов разложения, график средней строки коэф-

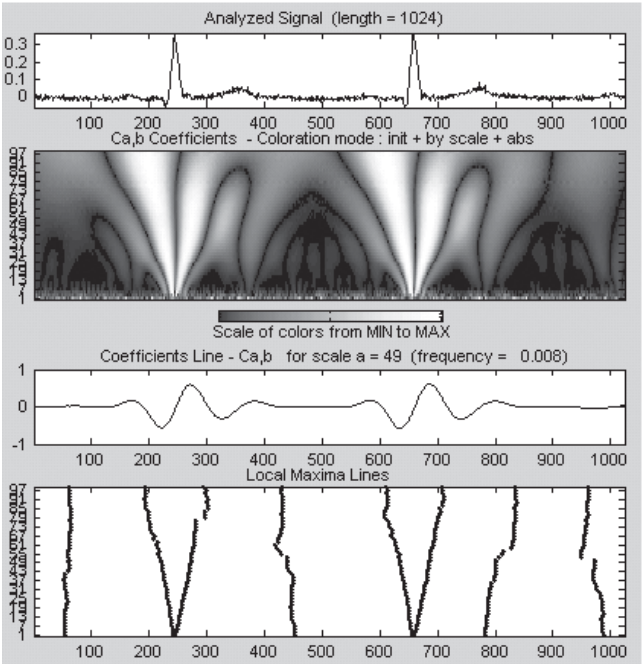


Рис. 4.3.10. Непрерывное вейвлет-разложение

фициентов разложения на уровне $a = 50$ и изображение локальных максимумов коэффициентов на каждом уровне масштаба от 1 до 100. На спектрограмме сигнала отчетливо отражены оба комплекса QRS, зубцы P и T кардиограммы, а также некоторые другие элементы на частотах от 20 Гц до 10 Гц, которые на графике кардиограммы незаметны. Отметим, что средней линии соответствует частота 4 Гц. Самая нижняя частота равна 2 Гц, а высокие частоты от 200 Гц до 50 Гц практически не видны.

Анализ результатов

Имеются следующие дополнительные возможности для изучения полученных результатов.

1. Просмотр изображения коэффициентов сигнала. Способ окраски коэффициентов может быть выбран в меню **Coloration Mode**. На примере выбран способ окраски 1-gray. Важно также выбрать цветовую палитру, количество оттенков цветов и правильное значение яркости. В нормальном режиме цвета масштабируются между мини-

мумом и максимумом коэффициентов. В абсолютном режиме (**abs**) цвета масштабируются между нулем и максимальным абсолютным значением коэффициентов. Подробности относительно способов окраски см. Help MATLAB. На приведенном примере выбран абсолютный тип окраски всех масштабов.

Для более детального рассмотрения картины коэффициентов можно убрать изображения линии коэффициентов и локальных максимумов, используя флажки в правом меню **Selected Axes**. Тогда на экране изображение увеличивается и занимает их место. Кнопка **View Axes** также позволяет увеличить изображение «во весь экран».

Если правой кнопкой мыши отметить точку на изображении коэффициентов, то ее координаты, положение по горизонтали (**X**) и масштаб (**Sca**) отобразятся в окне **Info** (расположенном внизу экрана). Масштаб отсчитывается по вертикали снизу вверх. Масштаб можно выбрать в частотном представлении. Для этого на правой части основного окна нужно выбрать **Frequencies** вместо **Scales**. Тогда позиция мыши дается в терминах местоположения (X) и частоты (Frq) в герцах. Это позволяет интерпретировать масштаб в терминах соответствующей псевдочастоты, которая зависит от вейвлета и периода выборки. Более подробную информацию относительно связи между масштабом и частотой см. в разделе «How to Connect Scale to Frequency?» в MATLAB Help.

Следует иметь в виду, что на спектрограмме отражаются искажения за счет концов сигнала, эти искажения тем сильнее, чем больше масштаб.

2. Просмотр линии вейвлет-коэффициентов. Каждому значению масштаба соответствует линия вейвлет-коэффициентов. Для выбора конкретной линии нужно выбрать значение масштаба, нажимая правой кнопкой мыши на нужный уровень масштаба на изображении коэффициентов. После этого активируются кнопки **New Coefficients Line** и **Refresh Maxima Line**. Если нажать **New Coefficients Line**, то в окне появится график коэффициентов, соответствующих выбранному масштабу. Правая кнопка мыши отмечает точку на графике и выводит ее координаты по горизонтали (**X**) и по вертикали (**Y**) в окне **Info** (расположенном внизу экрана). Для детального просмотра графика линии кнопка **View Axes** позволяет увеличить изображение графика «во весь экран».

3. Просмотр локальных максимумов. Локальные максимумы находятся отдельно на каждой линии коэффициентов. Если выбрано значение масштаба (правой кнопкой мыши на спектрограмме), то при

нажатии **Refresh Maxima Line** в окне появится изображение цепочек локальных максимумов для линий, соответствующих масштабам от начального до выбранного. Правая кнопка мыши отмечает точку на графике и выводит в окне **Info** ее положение по горизонтали **(X)** и масштаб **(Sca)**. Можно увеличить изображение локальных максимумов кнопкой **View Axes**.

4. Разложение фрагмента сигнала. Удерживая левую кнопку мыши, выделяем часть сигнала в самом верхнем окне. Выделенный фрагмент увеличиваем, нажав кнопку **X +** (расположенную внизу экрана), чтобы изменить размер окна только по горизонтали, или **XY +**, если требуется изменить размер окна также и по вертикали.

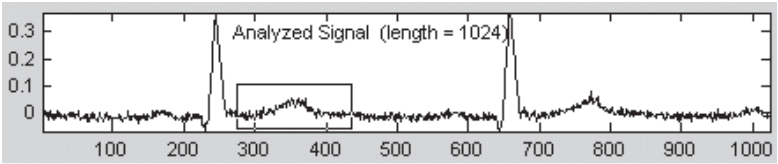


Рис. 4.3.11. Выделение фрагмента сигнала

Программа увеличивает не только выделенный сигнал, но и соответствующие фрагменты спектрограммы, линии вейвлет-коэффициентов и локальных максимумов.

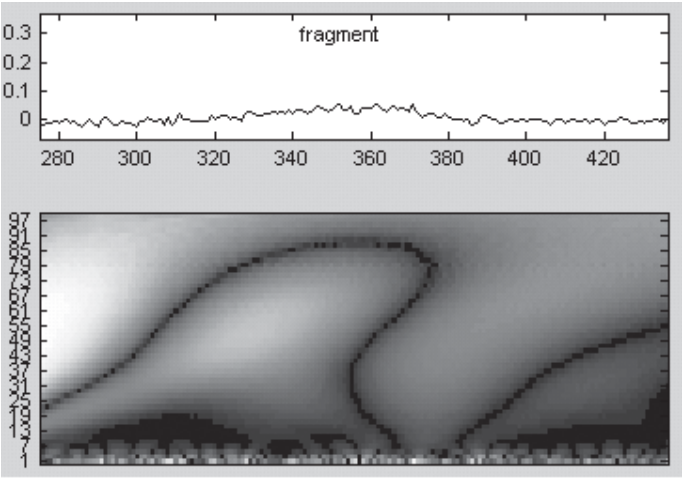


Рис. 4.3.12. Фрагменты сигнала и спектрограммы

Для сравнения получим вейвлет-разложение того же сигнала с использованием вейвлета Хаара.

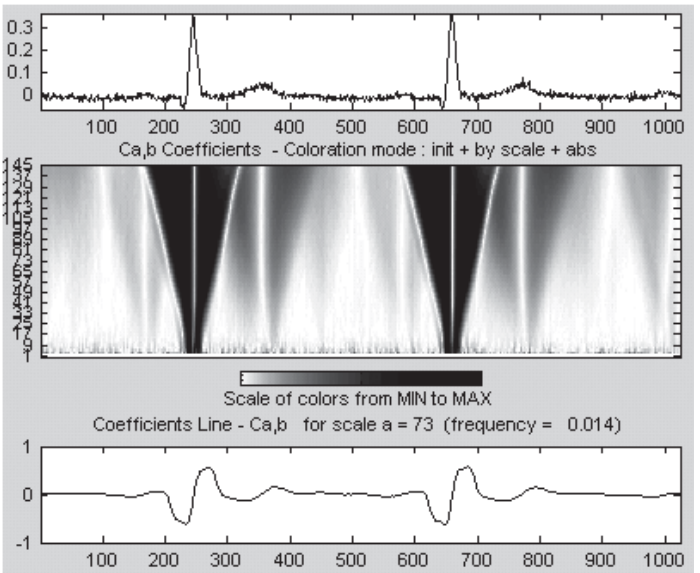


Рис. 4.3.13. Непрерывное вейвлет-разложение с использованием вейвлета Хаара

4.3.4. Комплексный непрерывный вейвлет-анализ (Complex Continuous Wavelet 1D)

Раздел **Complex Continuous Wavelet 1D** главного меню предназначен для непрерывного вейвлет-разложения

$$C_{a,b} = \frac{1}{\sqrt{a}} \int_{-\infty}^{\infty} s(t)\,\overline{\psi\left(\frac{t-b}{a}\right)}\,dt,$$

когда сигнал s и вейвлет ψ являются комплексными функциями. В этом случае коэффициенты являются комплексными числами и характеризуются модулем и аргументом.

Для запуска данного приложения нужно вызвать главное меню командой **wavemenu** и нажать кнопку **Complex Continuous Wavelet 1-D**. Окно графического интерфейса аналогично окну непрерывного вейвлет-разложения, но содержит вдвое больше окон: два окна для

графиков сигнала, два окна для модулей коэффициентов и для их аргументов, два окна для изображения графиков модулей и их аргументов линии коэффициентов выбранного масштаба и два окна для изображения локальных максимумов линий коэффициентов. Дополнительные три кнопки **Modulus**, **Angle** и **Both** позволяют переключить вид на показ информации только о модулях коэффициентов, только об углах или все вместе. Правое меню активируется после того, как будет загружен сигнал.

Загрузим сигнал ЭКГ, содержащий два комплекса QRS. Выбираем комплексный вейвлет (**Wavelet**) и его параметры. Возможны следующие типы комплексных вейвлетов.

Комплексный гауссов вейвлет cgauwavf – это *p*-я производная комплексной функции Гаусса $\psi(x) = C_p e^{-ix} e^{-x^2}$. Нужно задавать параметр *p*.

Комплексный вейвлет Морле cmorwavf с шириной полосы частот B и центральной частотой C. Он определяется формулой $\psi(x) = \sqrt{(\pi B)^{-1}} e^{i2\pi Cx} e^{-x^2/B}$. Параметры записываются в виде B-C. Например, cmor1-1.5 – комплексный вейвлет Морле шириной полосы частот B=1 и центральной частотой C=1.5.

Комплексный частотный B-сплайновый вейвлет fbspwavf, определенный параметром M, шириной полосы частот B, центральной частотой C и выражением

$$\psi(x) = \sqrt{B}\left(\frac{\sin Bx/M}{Bx}\right)^M e^{i2\pi Cx}.$$

Параметры записываются в виде M-B-C. Например, fbsp 2-1-0.5.

Вейвлет Шеннона shanwavf определен шириной полосы частот B, центральной частотой C и выражением:

$$\psi(x) = \sqrt{B}\,\frac{\sin Bx}{Bx} e^{i2\pi Cx}.$$

Параметры записываются в виде B-C. Например, shan1-1.5.

Значение по умолчанию периода выборки равно 1 (секунда). Далее устанавливаем параметры изменения масштаба. После этого нажимаем на кнопку **Analyse**, для того чтобы произвести вейвлет-разложение. Результаты разложения будут изображены на основном поле графического окна. Можно изменить выбранные параметры и провести вейвлет-разложение повторно.

На следующем рисунке приведены результаты разложения с использованием комплексного гауссова вейвлета порядка 2. Выбрано

изменение масштаба с шагом 1 от значения $a = 1$ до $a = 64$. Достаточно сложно подобрать палитру, чтобы одновременно хорошо были изображены и модули коэффициентов, и их аргументы (углы). В рассматриваемом примере наиболее подходящей оказалась палитра **prism**.

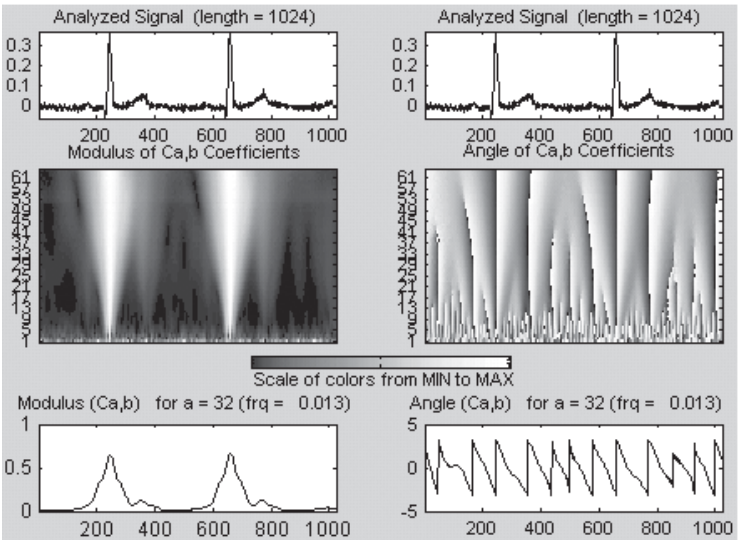


Рис. 4.3.14. Комплексное непрерывное вейвлет-разложение

4.3.5. Непрерывный вейвлет-анализ на основе FFT

Раздел **Continuous Wavelet 1-D (Using FFT)** главного меню предназначен для непрерывного вейвлет-разложения и анализа сигнала на основе дискретного преобразования Фурье. Для этих целей имеется функция cwtft (раздел 3.2), основанная на формуле

$$W_a(m) = \frac{\sqrt{a}}{N\Delta t}\sum_{k=0}^{N-1}\hat{x}\left(\frac{\pi k}{N\Delta t}\right)\overline{\hat{\psi}\left(a\frac{2\pi k}{N\Delta t}\right)}e^{i2\pi km/N},$$

где $x(t)$ – анализируемый сигнал с носителем на промежутке $[0, N]$, $\psi(t)$ – анализирующий вейвлет.

Алгоритм cwtft:

- выполнение дискретного преобразования Фурье (DFT) сигнала при помощи функции fft;

- выполнение дискретного преобразования Фурье DFT-анализирующего вейвлета в соответствующих частотах. Масштабирование DFT-вейвлета в различных масштабах;
- выполнение произведения DFT-сигнала и DFT-вейвлета во всех масштабах;
- обратное DFT для получения CWT-коэффициентов.

Для запуска **Continuous Wavelet 1D (Using FFT)** нужно вызвать главное меню командой **wavemenu** и нажать кнопку **Continuous Wavelet 1D (Using FFT)**. Запускается графическая среда, удобная для работы большого комплекса команд по анализу сигналов. Окно графического интерфейса содержит обычное меню вверху, специальные кнопки меню справа и внизу, а также основное поле для визуализации результатов анализа.

Правое меню активируется после того, как будет загружен сигнал. Загрузим сигнал ЭКГ, содержащий два комплекса QRS. В основном поле появится график сигнала, и справа в верхнем поле будет указана длина выборки. Выбираем вейвлет (**Wavelet**) и его порядок. Значение по умолчанию периода выборки **Sampling** равно 1 (секунда). Далее устанавливаем параметры изменения масштаба. По умолчанию используется диадический выбор масштабов. Тогда вектор масштабов находится по формуле scales = s0*2.^((0:NbSc-1)*ds, где s0 – начальное значение, NbSc – количество уровней масштаба и ds – приращение (рис. 4.3.15). Аналогично задается линейный порядок масштабов.

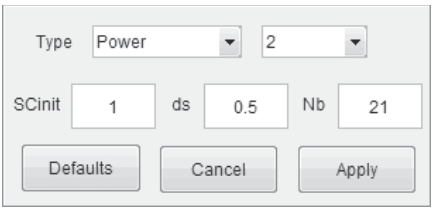


Рис. 4.3.15. Выбор масштабов

Масштабы можно выбрать и «вручную». Сделаем именно так. Для этого в поле **Scales** выберем **Manual** и затем откроем интерфейс **Define Scales** задания параметров для масштабирования. В качестве начального масштаба возьмем s0=1, шаг ds =2, число Nb=50. Среди предлагаемых вейвлетов есть и вещественные, и комплексные. Если вейвлет вещественный, то изображаются модули коэффициентов разложения и вещественные части (рис. 4.3.16). В комплексном случае –

четыре изображения: модуль, угол, вещественная часть и мнимая часть коэффициентов и угла (рис. 4.3.17). Для вейвлет-разложения сигнала нажимаем на кнопку **Analyse**. Результаты будут изображены на основном поле графического окна. Можно изменить выбранные параметры и провести вейвлет-разложение повторно. Выберем комплексный вейвлет Морле и снова разложим сигнал, результаты приведены на рис. 4.3.17.

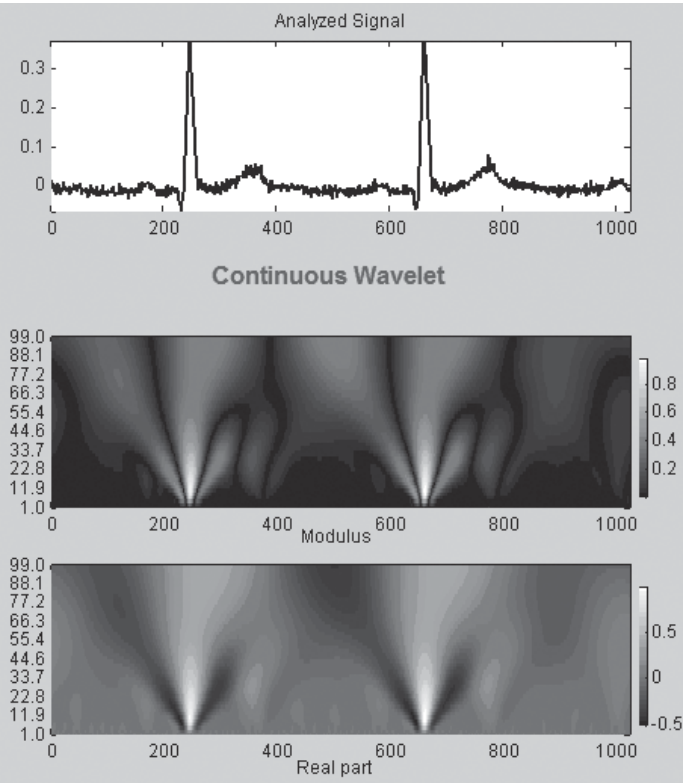


Рис. 4.3.16. Непрерывное вейвлет-разложение FFT

Для восстановления сигнала нужно произвести выбор масштабов. Соответствующие этим уровням масштабов коэффициенты будут участвовать в операции восстановления (синтеза). Если нажать кнопку **Scales Selection**, открывается окно со списком масштабов. Нужные масштабы для востановления можно выбрать правой кнопкой

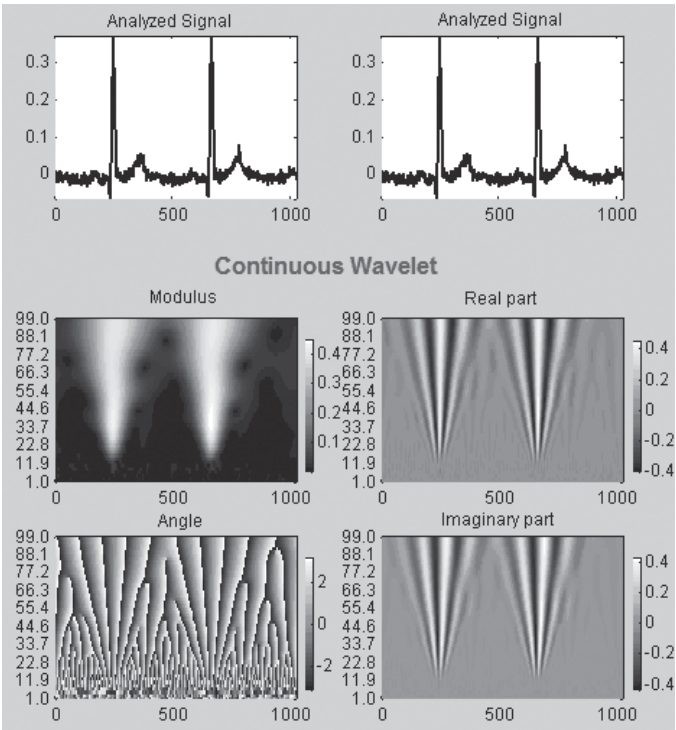


Рис. 4.3.17. Непрерывное вейвлет-разложение FFT комплексным вейвлетом morle

мышки с использованием клавиши **Ctr**. После этого нажать кнопку **Synthesize**.

Отметим, что здесь не предусмотрена пороговая обработка коэффициентов, но зато есть много уровней масштаба, что также открывает большие возможности выбора коэффициентов. Отметим, что даже если выбрать все масштабы, синтезированный сигнал может не совпасть с исходным – выбранных для разложения масштабов может не хватить. Для удовлетворительного восстановления наибольший масштабный коэффициент должен быть больше длины сигнала (если, конечно, период выборки =1). Тогда при линейном порядке масштабов их будет слишком много. Поэтому удобнее выбирать масштабы по степеням двойки (Power). В этом случае вектор масштабов находится по формуле scales = s0*2.^((0:NbSc-1)*ds, где s0 – начальное значение, NbSc – количество уровней масштаба и ds – приращение (рис. 4.3.15).

Выберем значения по умолчанию: начальное значение s0=2, приращение ds=0.5, количество уровней масштаба NbSc=21. В результате разложения мы получили следующую картину.

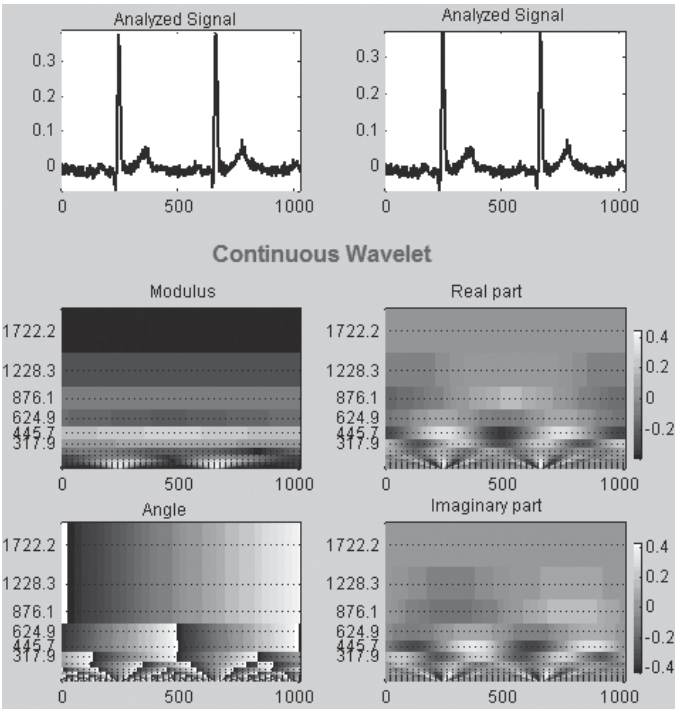


Рис. 4.3.18. Непрерывное вейвлет-разложение FFT комплексным вейвлетом morle

В этом случае выбор всех масштабов дает практически точное восстановление. Кроме того, небольшое количество уровней масштаба облегчает выбор масштабов для восстановления. Полосы на изображениях коэффициентов – это следствие такого выбора масштабов. При линейном типе масштабов отдельные уровни масштабов на изображении коэффициентов почти неразличимы.

Масштабы для восстановления можно выбрать и вручную, используя **Manual Selection of Coefficient**. В этом случае в основном поле будут открываться изображения вейвлет-коэффициентов, в которых правой кнопкой мышки можно выбирать фрагменты коэффициентов

и восстанавливать только по выбранным. Это хорошо демонстрирует следующий тестовый пример.

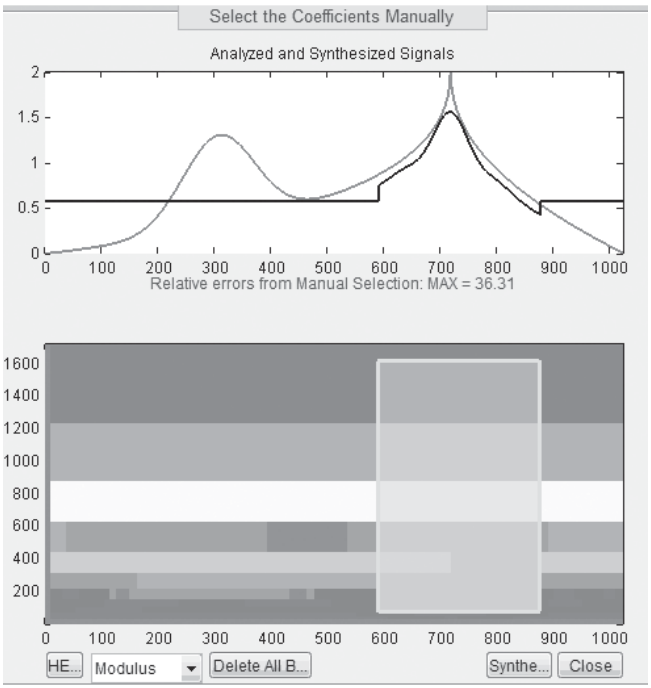


Рис. 4.3.19. Выбор коэффициентов «вручную»

При синтезе мы видим, что частично восстанавливается фрагмент сигнала и только в пределах выбранных коэффициентов. Здесь мы выбрали прямоугольную область модулей коэффициентов. Такая же область является выбранной и на всех других изображениях коэффициентов (углы, вещественные и мнимые части). Другие коэффициенты можно выбрать в раскрывающемся списке внизу.

4.4. Специализированные средства одномерного вейвлет-анализа

Раздел специализированных средств одномерного вейвлет-анализа представлен шестью программами:

- **SWT De-noising 1D** – удаление шума стационарного одномерного сигнала;
- **Density Estimation 1D** – оценка плотности распределения значений сигнала;
- **Regression Estimation 1D** – оценка функции регрессии;
- **Wavelet Coefficients Selection 1D** – выбор вейвлет-коэффициентов;
- **Fractional Brownian Generation 1D** – моделирование дробного броуновского движения;
- **Matching Pursuit 1D** – выполнение подгонки (согласования).

Рассмотрим все эти возможности главного меню **wavemenu**.

4.4.1. Удаление шума стационарного одномерного сигнала (SWT De-noising 1-D)

Статистические свойства стационарных и нестационарных сигналов существенно отличаются, и для их моделирования применяются различные методы. Раздел **SWT De-noising 1-D** предназначен для вейвлет-анализа стационарных сигналов.

Основные понятия

Напомним основные понятия, относящиеся к стационарным сигналам. Как обычно, считаем, что исследуемый сигнал представлен выборкой значений непрерывного процесса.

Случайным процессом называется функция $X(t)$ неслучайного аргумента t, которая при любом фиксированном значении $t = t_0$ является случайной величиной $X = X(t_0)$. Так как при фиксированном t случайный процесс представляет собой случайную величину $x_t = X(t)$, то можно определить вероятностные характеристики в этот момент времени. Таким образом, для случайного процесса функция распределения, плотность, математическое ожидание m_X, дисперсия D_X и другие являются функциями от $t \in \mathbf{R}$. Пусть t_1, t_2 – некоторые фиксированные моменты времени. Тогда $X(t_1)$, $X(t_2)$ – две случайные величины. *Ковариационная функция* случайного процесса X определяется как математическое ожидание $K_x(t_1, t_2) = M[X(t_1)X(t_2)]$, а *корреляционная функция* случайного процесса X есть

$$R_x(t_1, t_2) = M[(X(t_1) - m_x)(X(t_2) - m_x)] = K_x(t_1, t_2) - m_x^2.$$

Определение 1. *Случайный процесс $X(t)$ с конечной дисперсией называется стационарным (в широком смысле), если его математиче-*

ское ожидание постоянно, $M[X(t)] = m_X$, а ковариационная функция $K_x(t_1, t_2)$ зависит только от разности аргументов,

$$K_x(t_1, t_2) = K_x(t_1 - t_2) = K_X(\tau).$$

В этом случае и корреляционная функция R_X зависит только от разности $\tau = t_2 - t_1$, $R_X = R_X(\tau)$. Отметим, что дисперсия стационарного случайного процесса $X(t)$ постоянна, $R_X(0) = D_X$, кроме того, $K_X(0) = D_X + m_X^2$. По своему смыслу ковариационная $K_X(\tau)$ и корреляционная $R_X(\tau)$ функции отражают зависимость случайных величин $X(t)$ и $X(t + \tau)$.

Проверка на стационарность. Первое, что нужно сделать, – посмотреть на график полученных наблюдений. Возможно, он содержит очевидный тренд, также возможно, что разброс значений убывает или возрастает с течением времени. Это свидетельствует о зависимости среднего или дисперсии от времени.

Второе – построить график выборочной автокорреляционной функции (коррелограммы)

$$r_k = \frac{\sum_{t=k+1}^{n}(x_t - \bar{x})(x_{t-k} - \bar{x})}{\sum_{t=1}^{n}(x_t - \bar{x})^2}.$$

Коррелограмма стационарного временного ряда быстро убывает с ростом k.

Часто нестационарный сигнал может быть приведен к стационарному выделением тренда и сезонной составляющей. Применение оператора конечно-разностной производной также может привести к стационарному сигналу.

Примером стационарного случайного процесса является «белый шум» $e(t)$. Для его определения нам потребуются некоторые дополнительные понятия.

Спектральной плотностью $S(\omega)$ стационарного процесса называется преобразование Фурье ковариационной функции $K(\tau)$:

$$S(\omega) = \int_{-\infty}^{\infty} K(\tau) e^{-i\omega\tau} d\tau.$$

Спектральная плотность центрированного процесса $X_0(t) = X(t) - m_X$ будет обозначаться символом $S_0(\omega)$. Из определения корреляционной функции $R(\tau)$ следует, что

$$S_0(\omega) = \int_{-\infty}^{\infty} R(\tau) e^{-i\omega\tau} d\tau.$$

Из формулы $K(\tau) = R(\tau) + m_X^2$ следует $S(\omega) = S_0(\omega) + m_X^2\delta(\omega)$. Отметим также следующее выражение $R(0) = D_X = \frac{1}{2\pi}\int_{-\infty}^{\infty} S_0(\omega)d\omega$.

Функции $S(\omega)$ и $S_0(\omega)$ иногда называют *спектром мощности*, или энергетическим спектром, поскольку они имеют физический смысл энергии. Из взаимных свойств функций $R(\tau)$ и $S_0(\omega)$, связанных преобразованием Фурье, следует, что чем шире носитель спектральной плотности $S_0(\omega)$, тем уже носитель корреляционной функции $R(\tau)$. Последнее означает зависимость случайных величин $X(t)$ и $X(t + \tau)$ только при малых значениях τ. В пределе, когда носитель $S_0(\omega)$, расширяясь, совпадает с **R**, мы получаем, что соответствующий случайный процесс имеет в качестве корреляционной функции δ-функцию, то есть случайные величины $X(t_1)$ и $X(t_2)$ не коррелированны в любые несовпадающие моменты времени t_1 и t_2. Такой процесс можно назвать абсолютно случайным процессом.

Определение 2. *Белым шумом называется стационарный случайный процесс $e(t)$ с нулевым математическим ожиданием и постоянной спектральной плотностью*

$S(\omega) = N_0$.

Данное название отражает тот факт, что такой процесс имеет спектр на всех частотах по аналогии с белым светом, имеющим в видимой части равномерный сплошной спектр. Для белого шума формула $D_X = \sigma^2 = (2\pi)^{-1}\int_{-\infty}^{\infty} S_0(\omega)d\omega$ имеет непонятный физический смысл – дисперсия (средняя мощность такого шума) равна бесконечности $D_X = \infty$. Поэтому иногда требуют постоянства спектра $S(\omega)$ только в широком диапазоне. Понятие белого шума является корректным в случае дискретного сигнала $x_k = X(t_k)$.

Белый шум характеризуется свойствами: нулевое математическое ожидание m, постоянная дисперсия σ^2, нулевая корреляция $R(\tau) = 0$ при $\tau > 0$.

Работа с SWT De-noising 1D

Для запуска приложения нужно вызвать главное меню командой **wavemenu** и выбрать раздел **SWT De-noising 1D**. Запускается графическая среда, удобная для вейвлет-анализа стационарного сигнала и удаления шума. Окно графического интерфейса содержит обычное меню вверху, специальные кнопки меню справа и внизу, а также основное поле для визуализации результатов анализа.

Загрузим кардиосигнал (файл ECG.mat). Отметим, что при загрузке демонстрационного сигнала в меню **File/Example Analysis** имеются две возможности: **Noisy Signals** и **Noisy Signals-Interval dependent Noiste Variance**. Они соответствуют выбору примеров обычного анализа и анализа с установкой интервальных порогов.

В левом верхнем окне появится график сигнала, и в правом меню в верхнем поле будет указана длина выборки. Далее выбираем вейвлет (**Wavelet**), его порядок и уровень (**Level**) вейвлет-разложения (по умолчанию 5, выберем 3). После этого нужно нажать на кнопку **Decompose Signal**, для того чтобы произвести вейвлет-разложение. Результаты разложения, недецимированные аппроксимирующие и детализирующие вейвлет-коэффициенты разложения отображаются на основном поле графического окна слева. После этого активируется все правое меню.

В меню справа можно выбрать различные методы выбора порога для удаления шума. По умолчанию принимается вариант **Fixed form threshold** с мягким (**soft**) способом пороговой обработки. Далее в меню **Select noise structure** нужно определиться с типом шума. По умолчанию берется немасштабированный белый шум (**unscaled white noise**). Слайдеры, расположенные на правой части окна, управляют порогами на каждом уровне.

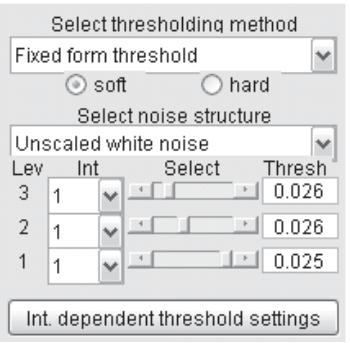


Рис. 4.4.1. Выбор порогов

На графиках детализирующих коэффициентов пороги изображены желтыми горизонтальными пунктирами. Желтые пунктиры можно также перемещать при помощи левой кнопки мыши. Отметим, что аппроксимирующие коэффициенты пороговой обработке не подвергаются.

После выбора всех параметров нажимаем **De-noise** и получаем следующее:

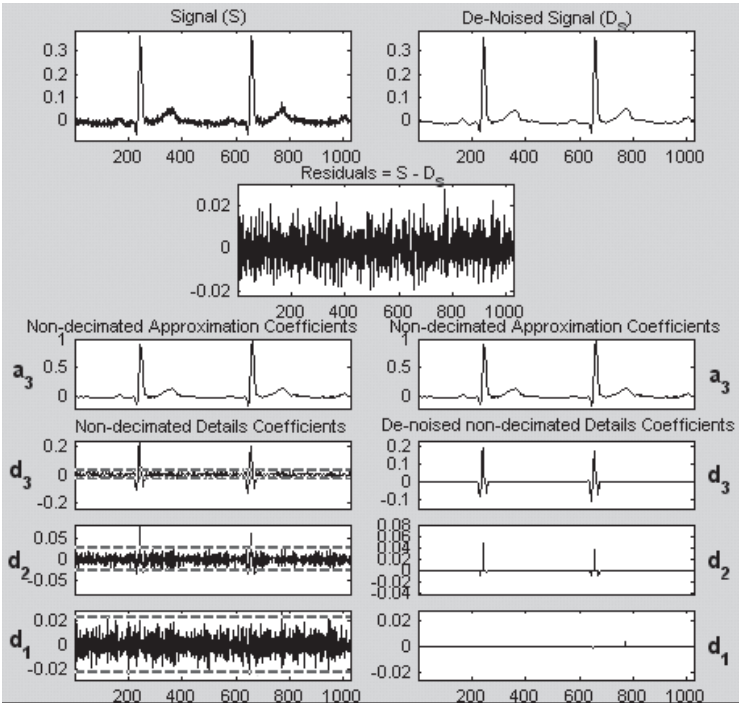


Рис. 4.4.2. Удаление шума

Справа вверху изображается график очищенного от шума сигнала, ниже – графики обработанных коэффициентов. В центре – график шумовой компоненты (разность между исходным сигналом и очищенным). Для исследования статистических свойств этой компоненты можно вызвать специальное окно, нажав кнопку **Residuals**. Как обычно, есть возможность увеличить каждый график, выбрать фрагмент сигнала и провести пороговую обработку коэффициентов с разными порогами на разных интервалах (**Int. dependent threshold settings**).

Используя кнопку **View Axes**, вы можете рассмотреть результат обработки в большем масштабе.

Мы видим, что даже при значениях параметров по умолчанию мы получаем очень хороший результат. Он гораздо лучше, чем при уда-

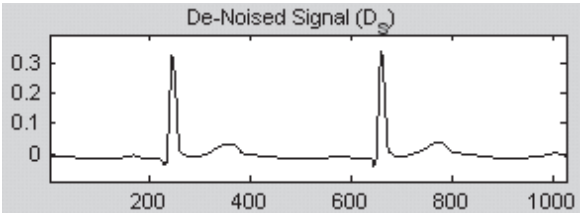


Рис. 4.4.3. Результат удаления шума

лении шума обычным способом в разделе **Wavelet 1D**. Возможно, это является следствием обработки недецимированных коэффициентов. Статистический анализ разности **(Residuals)** подтверждает ее шумовой характер.

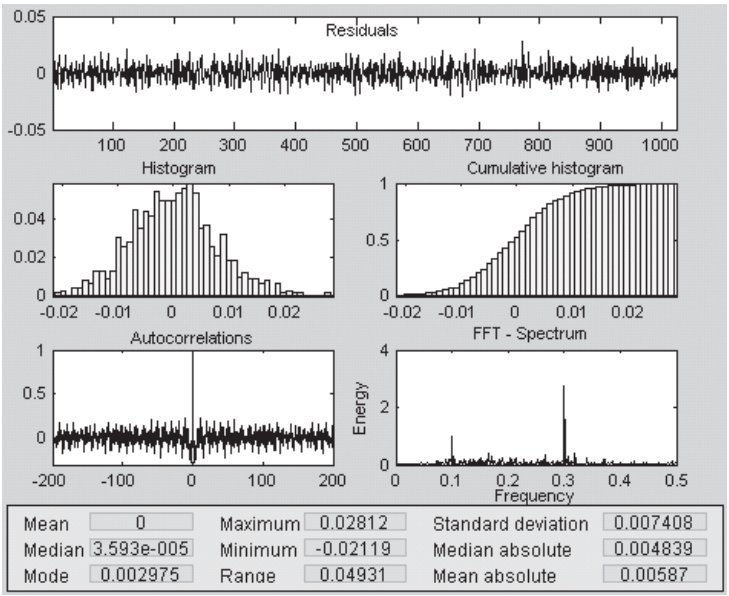


Рис. 4.4.4. Статистические характеристики удаленного шума

Очищенный сигнал и шумовая компонента могут быть сохранены в виде mat-файла. Сглаженный сигнал может использоваться для контурного анализа ЭКГ. Статистические характеристики шума, такие как мода, стандартное отклонение, имеют диагностическую ценность. Кроме того, важны также вид гистограммы и распределение

частот спектра быстрого преобразования Фурье. В частности, вызывают интерес выделяющийся пик на частоте 300 Гц и два пика на частоте 30 и 100 Гц соответственно (сигнал имеет частоту 1000 Гц).

Отметим в заключение, что если мы загрузим тестовый сигнал, используя меню **File/Example Analysis/ Noisy Signals-Interval dependent Noiste Variance**, то сразу открывается окно с графиком вейвлет-коэфициентов для выбора интервалов и задания на них разных порогов.

В конце работы можно сохранить сигнал, очищенный от шума в виде mat-файла.

4.4.2. Оценка плотности (Density Estimation 1D)

Раздел **Density Estimation 1D** главного меню предназначен для оценки плотности распределения значений изучаемого сигнала при помощи вейвлет-разложения.

Идея алгоритма

Напомним, что случайной величиной называют такую величину, которая в результате испытаний принимает значение, которое зависит от случайных обстоятельств и заранее не может быть предсказано. Непрерывная случайная величина X может принимать любые значения некоторого промежутка $[a, b]$, или любые числовые значения. Функцией распределения случайной величины X называется функция $F(x)$, равная вероятности того, что случайная величина X примет значение, меньшее, чем x, $F(x) = P\{X < x\}$, $x \in \mathbf{R}$. Для непрерывной случайной величины X можно ввести функцию плотности распределения вероятности случайной величины как производную функции распределения, $p(x) = F'(x)$. Плотность вероятности является очень важной характеристикой случайной величины. Она позволяет найти функцию распределения и числовые характеристики, такие как математическое ожидание и дисперсия. Примером плотности распределения вероятности является нормальное распределение

$$p(x) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{(x-m)^2}{2\sigma^2}},$$

где m и σ^2 – параметры распределения (математическое ожидание и дисперсия). Данное распределение играет особую роль и применяется в статистике гораздо шире других распределений.

Предположим, что нам известны результаты наблюдений неизвестной случайной величины s, представленные в виде конечной выборки $s_1, s_2, ..., s_n$. Требуется определить тип функции распределе-

ния и найти параметры функции распределения. Чтобы получить представление о законе распределения случайной величины на основе наблюдений ее значений, поступают следующим образом. Область значений случайной величины разбивается на m одинаковых интервалов длиной Δx. Вероятность попадания в i-й интервал равна относительной частоте попадания в этот интервал $\Delta P_i = n_i/n$, здесь n_i – количество значений, попавших в i-й интервал, а n – общее число значений. Затем вычисляется относительная плотность точек в каждом интервале: $p_i = \Delta P_i/\Delta x = n_i/(n\Delta x)$. Эта величина является оценкой плотности распределения. Найденные таким образом значения p_i изображают в виде гистограммы. Выбор количества интервалов зависит от величины выборки n, в Matlab берется $m = n/4$. По виду гистограммы делают предположение о типе функции распределения случайной величины. Сопоставляя гистограмму с графиками различных теоретических распределений и подбирая наиболее близкое, делают вывод о характере распределения неизвестной случайной величины.

Если сигнал имеет большую длину n, то и количество разбиений m может быть большим. В этом случае относительная плотность значений в каждом интервале $p_i = n_i/(n\Delta x)$ представляет самостоятельный, достаточно сложный сигнал. Хотя гистограмма дает информацию относительно плотности распределения, ее сложный вид не позволяет сделать предположение о типе функции распределения. Тогда можно использовать вейвлет-разложение и пороговую обработку коэффициентов для удаления шума в сигнале p_i и последующей оценки плотности распределения. В этом заключается основная идея комплекса программ **Density Estimation 1-D**. Существует несколько методов оценки плотности. Оценочные методы на основе вейвлетов ведут себя не хуже других, а иногда и лучше. В случае, когда плотность $p(x)$ имеет нерегулярности типа точек разрыва или точек разрыва производной, оценки на основе вейвлетов предпочтительнее.

Процедура оценки состоит из следующих шагов:

1. Выбираем число m промежутков разбиения. По умолчанию считается, что $m = n/4$. Пусть интервалы разбиения занумерованы по порядку индексами $i = 1, 2, ..., m$. Пусть n_i – количество значений, попавших в i-й интервал, и $p_i = n_i/(n\Delta x)$. Поскольку значения n и Δx фиксированы, то удобнее будет анализировать данные n_i вместо плотности p_i.
2. Делаем вейвлет-разложение сигнала n_i.
3. Делаем пороговую обработку вейвлет-коэффициентов одним из методов, используемых в процедурах удаления шума.

4. Восстанавливаем сигнал $\hat{n}(i)$ по обработанным коэффициентам и получаем оценку $\hat{p}(i) = \hat{n}(i)/(n\Delta x)$ плотности распределения $p(x)$.

В принципе, процедура оценки может быть проведена следующим образом. Рассмотрим наш сигнал $s_1, s_2, ..., s_n$ как выборку значений случайной величины с плотностью распределения $p = p(x)$. Эта плотность $p(x)$ является неизвестной, и мы хотим ее оценить. Предполагаем, что функция $p(x)$ лежит в пространстве $L^2(\mathbf{R})$. Пусть $\varphi(x)$ и $\psi(x)$ – ортогональные масштабирующая функция и вейвлет. Пусть J – целое число, масштаб разложения. Возьмем разложение функции $p(x)$:

$$p(x) = \sum_k a_{J,k}\varphi_{J,k}(x) + \sum_{j=-\infty}^{J}\sum_k d_{J,k}\psi_{J,k}(x),$$

где

$$a_{J,k} = \int_R \varphi_{J,k}(x)p(x)dx, \quad d_{j,k} = \int_R \psi_{J,k}(x)p(x)dx.$$

Поскольку $p(x)$ есть плотность распределения случайной величины, то последние интегралы имеют вероятностный смысл. А именно $a_{J,k} = \int_R \varphi_{J,k}(x)p(x)dx$ есть математическое ожидание $M[\varphi_{J,k}(s)]$ случайной величины $\varphi_{J,k}(s)$. Такой же смысл имеют и коэффициенты $d_{J,k}$. Как известно, математическое ожидание оценивается средним значением. Тогда получаем следующие оценки коэффициентов:

$$\hat{a}_{J,k} = \frac{1}{n}\sum_{i=1}^{n}\varphi_{J,k}(s_i), \quad \hat{d}_{J,k} = \frac{1}{n}\sum_{i=1}^{n}\psi_{j,k}(s_i).$$

Если j_0 – уровень разложения, то оценка плотности распределения задается формулой

$$\hat{p}(x) = \sum_k \hat{a}_{J,k}\varphi_{J,k}(x) + \sum_{j=J-j_0}^{J}\sum_k \hat{d}_{J,k}\psi_{J,k}(x).$$

Ввиду конечности числа наблюдений n суммирование по k является конечным. В последнем представлении делаем еще пороговую обработку коэффициентов, удаляя значения $\hat{d}_{j,k}$, меньшие некоторого порогового значения T. После такой обработки получается окончательное выражение для оценки $\hat{p}(x)$ плотности распределения $p(x)$.

С вычислительной точки зрения, в изложенной процедуре трудно использовать быстрые алгоритмы, потому что при вычислении $\varphi_{J,k}(s_i)$ и $\psi_{J,k}(s_i)$ значения s_i неравномерно распределены. Однако эта проблема в Matlab преодолена. Подробности можно найти в Help Matlab.

Работа с Density Estimation 1D

Для запуска приложения нужно вызвать главное меню командой **wavemenu** и нажать кнопку **Density Estimation 1D**. Запускается графическая среда, удобная для построения гистограммы и последующей обработки коэффициентов для оценки плотности распределения. Окно графического интерфейса содержит обычное меню вверху, специальные кнопки меню справа и внизу, а также основное поле для визуализации результатов анализа.

Загрузим сигнал ЭКГ, содержащий два комплекса QRS (файл ECG.mat). Выбираем вейвлет (**Wavelet**), его порядок и уровень (**Level**) вейвлет-разложения 3 (по умолчанию 5). Затем нужно выбрать количество **Nb bins** разбиений промежутка значений для построения гистограммы. Наименьшее количество считается равным 64, и для такого разбиения справа строится гистограмма. По умолчанию считается, что $m = n/4$. Мы можем выбрать большее число разбиений. После этого нужно нажать на кнопку **Decompose**, для того чтобы произвести вейвлет-разложение. Результаты разложения будут изображены на основном поле графического окна слева (рис. 4.4.5). После этого активируется все правое меню.

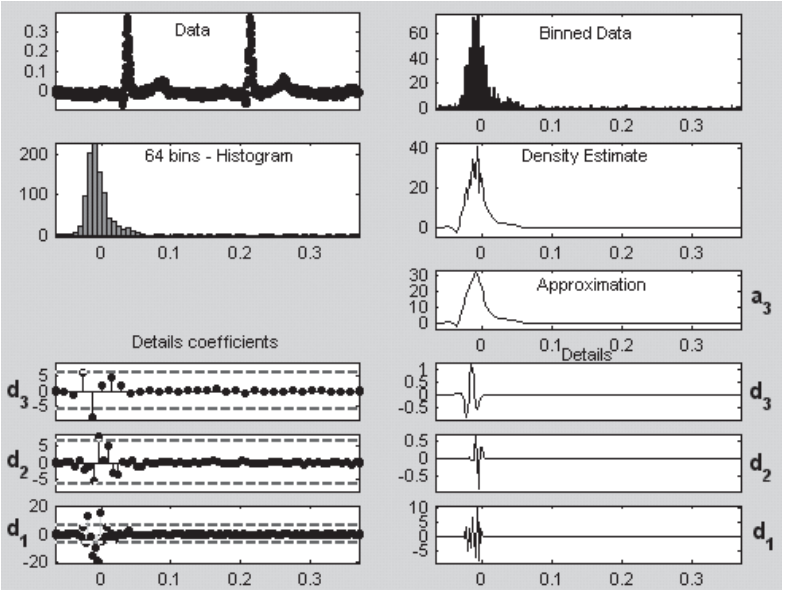


Рис. 4.4.5. Оценка плотности

Пороговая обработка коэффициентов. Для этого необходимо воспользоваться возможностями средней части правого меню, см. рис. 4.4.1. В меню **Select thresholding method** есть возможность выбрать несколько режимов автоматического выбора порогов. По умолчанию выбирается **Global thresholding** с мягкой пороговой обработкой. Расположенные ниже слайдеры управляют выбором порога вручную. Кроме того, имеется возможность левой кнопкой мыши менять уровни порогов на графиках слева. Эти уровни изображены горизонтально желтыми штрих-линиями.

Продолжим работу, нажав кнопку **Estimate**. Может оказаться, что полученная в результате оценки плотность представляет собой очень нерегулярную кривую во втором окне справа. Она складывается из сигналов, полученных из обработанных коэффициентов, эти элементарные сигналы изображены справа ниже. Тогда можно экспериментировать с различными стратегиями определения порогового значения, выбирая соответствующие варианты в меню **Select thresholding method**, или непосредственно, перемещая желтые строки на графиках коэффициентов левой кнопкой мыши.

Кнопки нижнего меню работают аналогично, для более детального рассмотрения сигнала, коэффициентов или фрагментов.

Установка интервальных порогов. Кнопка **Int. dependent threshold settings** выводит дополнительное окно, в котором изображаются вейвлет-коэффициенты. Кнопка **Level** позволяет выбрать уровень коэффициентов. На основном поле имеются две горизонтальные линии (пороги), которые можно изменять, удерживая их правой кнопкой мыши. В меню **Select Number of Intervals** можно выбрать количество интервалов, на каждом из которых будет выбираться свое значение порога. Положения границ интервалов (штрих-линия красного цвета) можно установить, удерживая их правой кнопкой мыши (см. рис. 4.4.2).

4.4.3. Оценка регрессии (Regression Estimation 1D)

Раздел **Regression Estimation 1D** главного меню предназначен для оценки функции регрессии.

Основные понятия

Регрессия – зависимость среднего значения какой-либо величины от некоторой другой величины или от нескольких величин. В отличие от функциональной зависимости $Y = f(X)$, когда каждому зна-

чению независимой переменной x соответствует одно определенное значение y величины Y, при регрессионной связи одному значению x величины X могут соответствовать в зависимости от случая различные значения y_i величины Y. Зависимость среднего значения чисел y_i от значения x и является регрессией в статистическом понимании этого термина. Таким образом, *уравнение регрессии* $y = f(x)$ определено следующим образом: $f(x)$ есть математическое ожидание $M[Y | X = x]$ случайной величины Y при условии, что случайная величина X принимает значение x.

Задачи регрессии возникают в самых обычных практических вопросах. Цель состоит в том, чтобы получить модель зависимости между одной случайной переменной Y и одной или более переменными X и дать объяснение части изменчивости Y за счет изменения X. Модель принимается в виде $Y = f(X) + e$. Функция f представляет основную часть зависимости. Оставшаяся часть e считается шумом. Переменная Y называется объясняемой (зависимой) переменной, а переменная X – объясняющей переменной, или регрессором. Приведенное выше уравнение называется регрессионным уравнением. Основная задача заключается в нахождении уравнения регрессии по выборочным данным переменных X и Y.

Выбор *модели регрессии* определяется предположением о форме зависимости $Y = f(X)$. Самый простой случай – линейная регрессия $Y = aX + b + e$. Когда f является полностью неизвестной, задачу нелинейной регрессии называют непараметрической задачей, и она может быть решена при помощи статистических методов, или методов, основанных на вейвлетах. В статистических приложениях, когда для точного определения регрессии $Y = f(X)$ нет достаточных сведений о форме совместного распределения вероятностей, возникает задача приближенного нахождения уравнения регрессии. Выбирается класс функций (линейные, полиномиальные, экспоненциальные), и в этом классе ищется функция $y = T(x)$, которая дает наилучшее представление величины Y, в том смысле, что она минимизирует математическое ожидание $M[(Y - T(x))^2]$ квадратов отклонений. Найденная методом наименьших квадратов функция $T(x)$ является приближением к истинной функции регрессии $f(x)$.

В данном разделе ищется зависимость $Y = f(X)$ с не заданным заранее типом функции. Поэтому речь идет об удалении шума e в модели $Y = f(X) + e$. В таких задачах вейвлеты применяются достаточно эффективно.

В зависимости от статуса объясняющей переменной X различают две схемы регрессии: классическую схему и стохастическую.

Классическая схема регрессии (Fixed-Design Regression). Это случай, когда значения X детерминированы, предопределены, как, например, дни недели, возраст и т. п. Обычно в этом случае значения переменной X расположены равномерно.

Стохастическая схема регрессии (Stochastic Design Regression). Это случай, когда значения X являются результатами измерения, обработки или случайно выбраны. Обычно значения X располагаются нерегулярно. Эта структура является более общей, так как она включает анализ отношений между переменной Y и общей, возможно случайной, переменной X.

Оценка функции регрессии f на основе вейвлетов. Основная идея заключается в сведении общей проблемы регрессии к классической модели регрессии. Основные шаги следующие.

1. Преобразуем данные (X, Y) в данные (Xb, Yb), используя процедуру разбиения области значений на малые промежутки. Значения Xb равномерно распределены. Для каждого промежутка i определяем

$$Yb(i) = \frac{\text{сумма}\{Y(j),\ \text{таких что}\ X(j)\ \text{лежит в}\ i\text{-м промежутке}\}}{\text{число}\{Y(j),\ \text{таких что}\ X(j)\ \text{лежит в}\ i\text{-м промежутке}\}}$$

   с соглашением $0/0 = 0$.
2. Делаем вейвлет-разложение сигнала Yb, используя быстрые алгоритмы. Здесь предполагается, что Xb-данные есть 1, 2, ..., nb, где nb – число промежутков.
3. Делаем пороговую обработку вейвлет-коэффициентов одним из методов, описанных для удаления шума.
4. Восстанавливаем оценку f_1 функции f из обработанных вейвлет-коэффициентов, используя быстрые алгоритмы.
5. Перемасштабируем результирующую функцию f_1, преобразовывая 1, 2..., nb в данные Xb, и интерполируем f_1 в каждом промежутке, чтобы найти оценку $f(x)$.

В техническом плане задача регрессии решается так же, как и оценка плотности. Основные различия касаются модели. Есть и другое различие с оценкой плотности распределения: мы имеем здесь две переменные X и Y вместо одной в схеме плотности.

Работа с Regression Estimation 1D

Для запуска приложения нужно вызвать главное меню командой **wavemenu** и нажать кнопку **Regression Estimation 1D**. Запускается графическая среда, удобная для оценки функции регрессии. Окно

графического интерфейса содержит обычное меню вверху, специальные кнопки меню справа и внизу, а также основное поле для визуализации результатов анализа.

Для анализа можно загрузить данные двух типов: по классической схеме регрессии (**Data for Fixed Design Regression**) и по стохастической схеме (**Data for Stochastic Design Regression**). Они отличаются тем, что в классической схеме определяющая переменная является детерминированной и значения расположены равномерно.

Загрузим демонстрационный пример Example III более общей стохастической регрессии (файл ex3nsto.mat из MATLAB-каталога toolbox/wavelet/wavedemo).

Вверху появится изображение точек на плоскости с координатами (x_i, y_i) данных X и Y, справа эти точки образуют график. В правом меню в верхнем поле будет указана длина выборки. Поскольку файл демонстрационный, его анализ уже будет проведен. Однако есть возможность самостоятельно выбрать вейвлет (**Wavelet**), его порядок и уровень (**Level**) вейвлет-разложения (по умолчанию 5), а также количество **Nb bins** разбиений промежутка значений для построения гистограммы определяющего сигнала X. По умолчанию считается, что число промежутков равно четверти от длины сигнала. После этого нужно нажать на кнопку **Decompose**, для того чтобы произвести вейвлет-разложение. Результаты разложения будут изображены на основном поле графического окна слева. После этого активируется все правое меню.

В меню справа можно выбрать различные методы выбора порога для удаления шума. По умолчанию принимается вариант **Fixed form threshold** с мягким (**soft**) способом пороговой обработки. Далее в меню **Select noise structure** нужно определиться с типом шума. По умолчанию берется немасштабированный белый шум (**unscaled white noise**). Слайдеры, расположенные на правой части окна, управляют порогами на каждом уровне.

На графиках детализирующих коэффициентов пороги изображены желтыми горизонтальными пунктирами. Желтые пунктиры можно также перемещать при помощи левой кнопки мыши. Отметим, что аппроксимирующие коэффициенты пороговой обработке не подвергаются.

После выбора всех параметров нажимаем **Estimate**. Результаты отражены на рис. 4.4.6.

Видно, что шум удален. Желтым цветом изображен график регрессии на фоне графика исходных массивов и в отдельном окне. Он складывается из сигналов справа внизу, построенных по оставшимся коэффициентам. Для исследования статистических свойств шу-

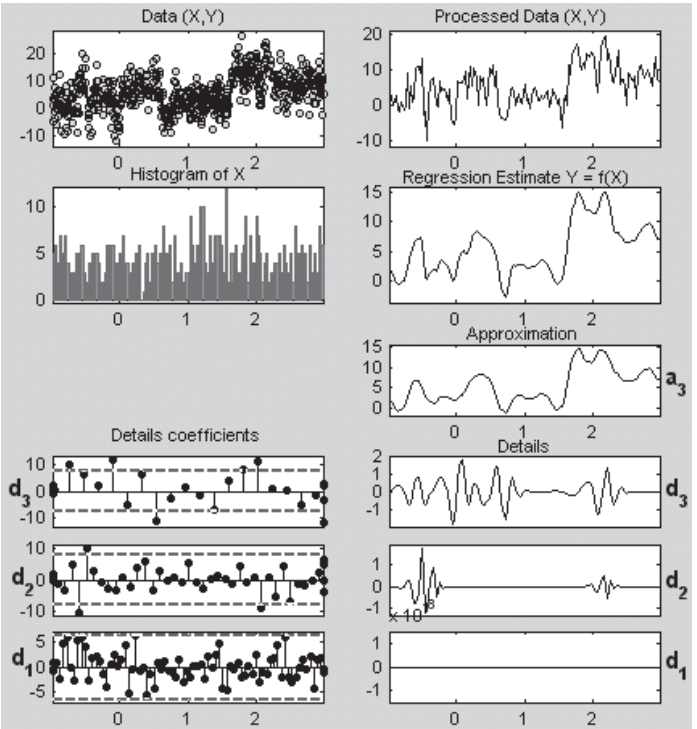


Рис. 4.4.6. Оценка регрессии

мовой компоненты можно вызвать специальное окно, нажав кнопку **Residuals**. Как обычно, есть возможность увеличить каждый график, выбрать фрагмент сигнала и провести пороговую обработку коэффициентов с разными порогами на разных интервалах **(Int. dependent threshold settings)**. Используя кнопку **View Axes**, можно рассмотреть результат обработки в большем масштабе.

4.4.4. Выбор вейвлет-коэффициентов сигнала (Wavelet Coefficients Selection 1D)

Раздел **Wavelet Coefficients Selection 1D** предназначен для более тонкого выбора вейвлет-коэффициентов при анализе сигнала. Для запуска этого приложения нужно вызвать главное меню командой **wavemenu** и нажать кнопку **Wavelet Coefficients Selection 1D**. Можно применить следующие стратегии выбора вейвлет-коэффициентов:

- глобальный выбор самых больших коэффициентов (в абсолютном значении);
- поуровневый выбор самых больших коэффициентов;
- автоматический пошаговый выбор самых больших коэффициентов;
- выбор коэффициентов вручную.

Для работы этого раздела необходимо включить симметричный режим расширения при помощи команды dwtmode('sym') в командном окне Matlab.

Загрузим данные, используя меню **File/Load Signal**. После этого появится диалоговое окно, в котором можно выбрать демонстрационные сигналы, а также свой файл с данными. Выберем используемый нами ранее сигнал ECG.mat.

Выполнение вейвлет-разложения. В правом меню **Wavelet** выберем вейвлет db3 и уровень разложения 5 в меню **Level**, затем нажмем кнопку **Analyze**.

В двух окнах слева будут изображены: оригинальный сигнал, его вейвлет-коэффициенты приближения A5 и детализирующие коэффициенты от D5 сверху до D1 внизу. В двух окнах справа будет изображен синтезированный сигнал (который на первом шаге совпадает с исходным, поскольку все коэффициенты сохраняются), ниже отображаются выбранные коэффициенты (рис. 4.4.7).

Выбор самых больших коэффициентов глобально. В меню справа выберем опции **Global** и **Select All**. Ниже справа найдем столбец **Kept (Сохраняемое)**. Последняя строка показывает общее количество коэффициентов: 1045. Это немного больше, чем число наблюдений, которое равно 1024. Можно задать число выбранных самых больших коэффициентов, напечатав номер вместо 1045 или используя слайдер. Напечатаем 40 и нажмем **Enter**. Числа выбранных наибольших коэффициентов от уровня к уровню меняются (но не могут модифицироваться, когда **Global** является текущим методом выбора). Теперь нажмем кнопку **Apply**. Отобразятся оставшиеся наибольшие коэффициенты и восстановленный по ним сигнал. При этом первоначальный сигнал можно отобразить совместно с синтезированным, если выбрать внизу опцию **Show Original Signal**. Все коэффициенты аппроксимации сохранялись. Можно ослабить это ограничение, выбирая другую опцию в меню **App. cfs** (Аппроксимирующие коэффициенты). Выберем опцию **Unselect** и щелкнем **Apply**. Ни один из коэффициентов аппроксимации не сохраняется.

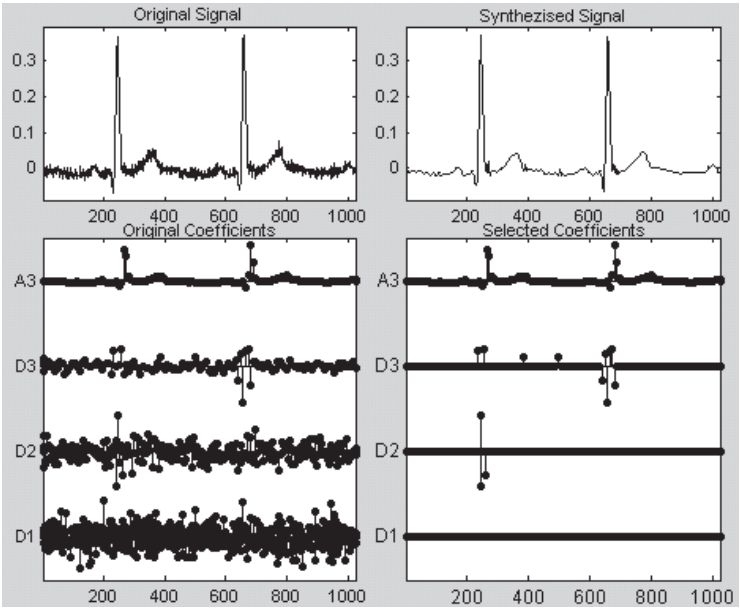


Рис. 4.4.7. Разложение сигнала

В меню **App. Cfs** можно выбрать опцию **Selectable**. Тогда выбор будет производиться среди всех коэффициентов, в том числе и аппроксимирующих. Некоторые из коэффициентов аппроксимации будут сохраняться.

Выбор самых больших коэффициентов на каждом уровне. В меню **Define Selection method** нужно выбрать опцию **By Level**. Можно задать число самых больших коэффициентов на каждом уровне, напечатав эти числа в соответствующих полях, или сделать это, используя слайдер.

Выбор коэффициентов вручную. Из меню **Define Selection method** выберем опцию **Manual**. Вначале никакие коэффициенты не сохраняются, и синтезируемый сигнал является нулевым. Однако появляется возможность выбора коэффициентов индивидуально двойным нажатием каждого левой кнопкой мыши. Цвет выбранных коэффициентов переключается с зеленого на желтый – для детализирующих и с синего на желтый – для аппроксимирующих. Выбранные коэффициенты появляются в окнах слева. Нажмем **Apply** и получим синтезированный сигнал. Можно выбрать или снять выбор

набора коэффициентов следующим образом. Выделите и удержите левой кнопкой мыши некоторую область, содержащую коэффициенты (можно сразу на нескольких уровнях), а затем нажмите кнопки **Select/Unselect** в зависимости от того, что нужно – выбрать или отменить выбор. Для удобства можно увеличить поле выбора коэффициентов при помощи кнопки **View Axes**.

Автоматический пошаговый выбор самых больших коэффициентов. В меню **Define Selection method** выберем опцию **Stepwise movie** (Пошаговое движение). Левое меню слегка меняется. Появляются три поля, в которых нужно выбрать минимальное количество коэффициентов, шаг добавления коэффициентов и максимальное число коэффициентов. Выбрав все эти параметры, нажимаем **Start**. Тогда видно в динамике, как меняется синтезированный сигнал при пошаговом добавлении коэффициентов.

Синтезированный сигнал может быть сохранен на диске как mat-файл. В конце работы необходимо вернуть режим расширения к исходному режиму добавления нулей при помощи команды dwtmode('zpd').

4.4.5. Моделирование дробного броуновского движения (Fractional Brownian Generation 1D)

Броуновское движение (винеровский процесс) есть однородный гауссов процесс $X(t)$ с независимыми приращениями. Это одна из математических моделей хаотического перемещения взвешенных в жидкости или газе мелких частиц. С математической точки зрения броуновское движение определяется как непрерывный по времени и по значениям случайный процесс $X(t)$, обладающий свойствами:

$$X(0) = 0, \quad M[X(t) - X(s)] = 0, \quad D[X(t) - X(s)] = t - s, s \le t,$$

где M – математическое ожидание и D – дисперсия.

Броуновское движение обладает следующими характерными свойствами. Броуновская траектория нигде не дифференцируема. При выходе из какой-либо точки x эта траектория за сколь угодно малое время δ с вероятностью 1 бесконечное число раз пересекает уровень x (возвращаясь в исходную точку). Рассматриваемая на фиксированном отрезке $[0, t]$, эта траектория имеет тенденцию достигать экстремальных значений вблизи концевых точек.

Дробное броуновское движение с показателем Херста $0 < H < 1$ есть непрерывный по времени и по значениям гауссов процесс $X(t)$ с нулевым средним значением, такой что $X(0) = 0$ и

$$D[X(t) - X(s)] = v|t - s|^{2H},$$

где v – положительная константа. Он является обобщением обычного броуновского движения, соответствующего показателю $H = 0.5$, производная которого есть белый шум. Дробное броуновское движение является нестационарным процессом.

Синтез дробного броуновского движения при помощи вейвлетов можно сделать в главном вейвлет-меню при помощи приложения **Fractional Brownian Generation 1D** (см. также раздел 3.3.1). Для запуска этого приложения нужно вызвать главное меню командой **wavemenu** и нажать кнопку **Fractional Brownian Generation 1D**. Появится окно приложения, на правой части которого имеется возможность выбора вейвлета, уровня разложения, длины сигнала, индекса фрактальности H (показатель Херста). После уточнения этих параметров достаточно нажать кнопку **Generate**. В основном окне приложения появляются два графика. Первый – синтезированное броуновское движение, а ниже – приращения первого порядка, см. рис. 4.4.8. Статистические данные приращений можно узнать на следующей странице, которую можно вызвать кнопкой **Statistiks**.

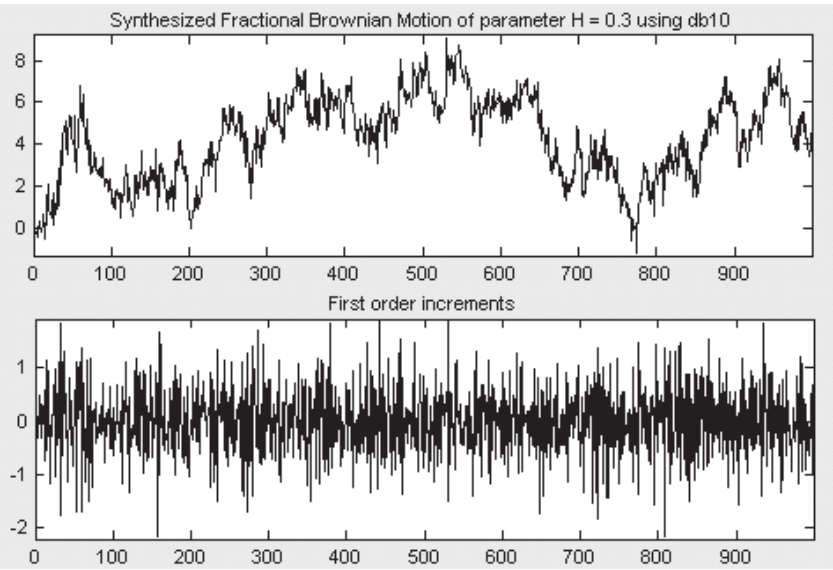


Рис. 4.4.8. Дробное броуновское движение и первые приращения

4.4.6. Выполнение подгонки (Matching Pursuit 1D)

Данное приложение позволяет для исследуемого сигнала подобрать базис, в котором он разлагается с наименьшим числом коэффициентов, становится разреженным.

Иногда, для более точного представления сигнала $f(x)$, бывает удобно расширить базис пространства сигналов $L^2(\mathbf{R})$, то есть добавить еще некоторые элементы. Полученная система $\{\varphi_k(x)\}$ будет полной, но избыточной. Возможно, это будет фрейм. В данном разделе такую систему функций принято называть словарем, а отдельные элементы $\varphi_k(x)$ называются атомами словаря. Предполагается, что все они имеют единичную норму. Процедура подгонки заключается в определенной серии операций по нахождению главных составляющих сигнала. Она производится в несколько этапов:

1. Находим атом словаря $\varphi_p(x)$, для которого скалярное произведение с $f(x)$ будет максимальным по модулю.
2. Вычисляем скалярное произведение $(f,\varphi_p)_{L^2}$ и находим остаток $R^1(f) = f - (f,\varphi_p)_{L^2}\varphi_p$.
3. Применяем шаги 1 и 2 для получаемых остатков. На m-ом шаге получаем остаток $R^{m+1}(f) = R^m(f) - (R^m(f),\varphi_k)_{L^2}\varphi_k$.
4. Заканчиваем процедуру подгонки, когда остаток $R^{m+1}(f)$ будет мал в определенном смысле.

Подбирая такое разложение сигнала, мы делаем сигнал более разреженным, то есть представленным меньшим числом коэффициентов.

Реально сигнал задается выборкой значений $Y = \{y_k\}$, $k = 1, ..., N$. Тогда мы в качестве пространства сигналов должны рассматривать $\mathbf{R}^N$. Возьмем словарь Φ в этом пространстве, состоящий из $M > N$ атомов. Они образуют полную, но избыточную систему нормальных векторов (столбцов) пространства $\mathbf{R}^N$. Будем рассматривать словарь Φ как матрицу M на N. Тогда скалярные произведения сигнала Y с атомами словаря находятся как произведение $\Phi^T \cdot Y$ транспонированной матрицы Φ^T на вектор-столбец Y. Пусть максимум модуля достигается для p-го атома (столбца) φ_p. Тогда $R^1(Y) = Y - (\varphi_p,Y)\varphi_p$, где (φ_p,Y) – обычное скалярное произведение в $\mathbf{R}^N$.

На втором этапе, если мы выбрали еще один атом φ_q для остатка $R^1(Y)$, мы проектируем остаток $R^1(Y)$ на пространство, порожденное векторами φ_p и φ_q, и из остатка $R^1(Y)$ вычитаем эту проекцию: $R^2(Y) = R^1(Y) - P(R^1(Y))$. Остаток $R^2(Y)$ получается ортогональным

атомам-векторам φ_p и φ_q. Образуем матрицу Ψ из векторов-столбцов атомов φ_p и φ_q. Тогда оператор проектирования P находится по общей формуле $P = \Psi(\Psi^T\Psi)^{-1}\Psi^T$. Дальнейшие шаги по нахождению $R^{m+1}(Y)$ совершенно понятны.

Для запуска приложения нужно вызвать главное меню командой **wavemenu** и выбрать раздел **Matching Pursuit 1D**. Запускается графическая среда, удобная для исследования сигнала и выбора базиса. Окно графического интерфейса содержит обычное меню вверху, специальные кнопки меню справа и внизу, а также основное поле для визуализации результатов анализа.

Загрузим тестовый сигнал **Cuspmax**. Отметим, что при загрузке демонстрационного сигнала он сразу подвергается анализу, и в основном окне мы получаем уже результаты. Рассмотрим сначала элементы управления процессом подгонки правой панели (рис. 4.4.9):

- название сигнала – в самом верху;
- раздел выбора словаря для нашего сигнала. В него уже включены элементы по умолчанию (их описание см. ниже). Есть возможность удалить некоторые блоки словаря и добавить новые;

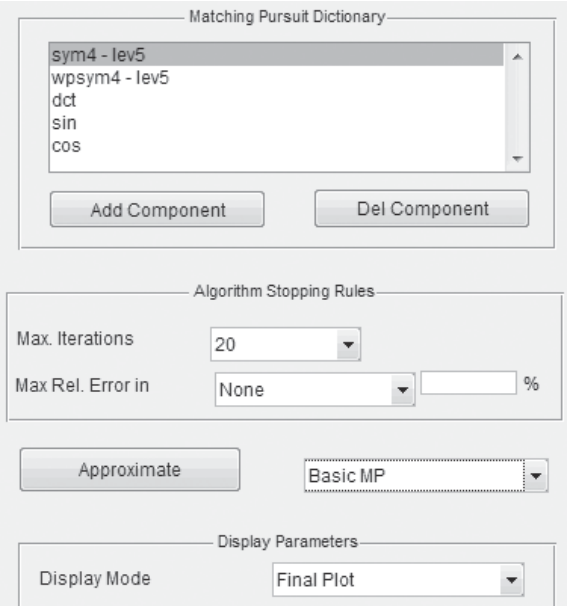


Рис. 4.4.9. Возможности управления процессом подгонки

- выбор условий для остановки процесса;
- выбор метода аппроксимации;
- выбор метода отображения процесса подстройки.

По умолчанию загружается словарь, состоящий из следующих групп:

- sym4-lev5 – комплект векторов, значений функций вейвлет-базиса симлетов порядка 4 до уровня масштабирования 5;
- wp sym4-lev5 – комплект векторов, значений функций пакетного вейвлет-базиса симлетов порядка 4 до уровня масштабирования 5;
- dct – унитарное дискретное косинусное преобразование массива длины N. Ортонормированный базис состоит из элементов:

$$\varphi_k(n) = \begin{cases} \dfrac{1}{\sqrt{N}} & k = 0 \\ \sqrt{\dfrac{2}{N}} \cos\dfrac{\pi(2n+1)k}{2N} & k = 1,2,\ldots,N-1 \end{cases};$$

- sin – подсловарь синусов. Он состоит из следующих столбцов: $\varphi_k(t) = \sin(2\pi kt)$, $k = 1, 2, ..., [N/2]$, $0 \le t \le 1$, где t есть линейно распределенный N-точечный вектор;
- cos – подсловарь косинусов. Он состоит из следующих столбцов: $\varphi_k(t) = \cos(2\pi kt)$, $k = 1, 2, ..., [N/2]$, $0 \le t \le 1$, где t есть линейно распределенный N-точечный вектор;
- 'poly – полиномиальный подсловарь. Он состоит из следующих столбцов: $p_n(t) = t^{n-1}$, $n = 1, 2, ..., 20$, $0 \le t \le 1$, где t есть линейно распределенный N-точечный вектор;
- 'RnIdent' – подсловарь смещенных символов Кронекера. Он состоит из следующих столбцов: $\varphi_k(n) = \delta(n - t)$, $k = 0, 1, ..., N$. Это просто стандартный базис пространства $\mathbf{R}^N$.

По умолчанию задана остановка процесса после 20 итераций (тогда сигнал будет задаваться всего 20 коэффициентами при длине в 1024). Для запуска процесса подгонки нужно нажать кнопку **Approximate**. В основном окне отображаются результаты (рис. 4.4.10 и 4.4.11).

В левой части (рис. 4.4.10) отображаются исходный сигнал и его приближение (после 20 итераций – как у нас было задано). Ниже приведены оценки точности приближения в процентах и график остатка $R^{20}(Y)$ (Residual). Еще ниже приведены графики оценок качества приближения после каждого этапа процесса.

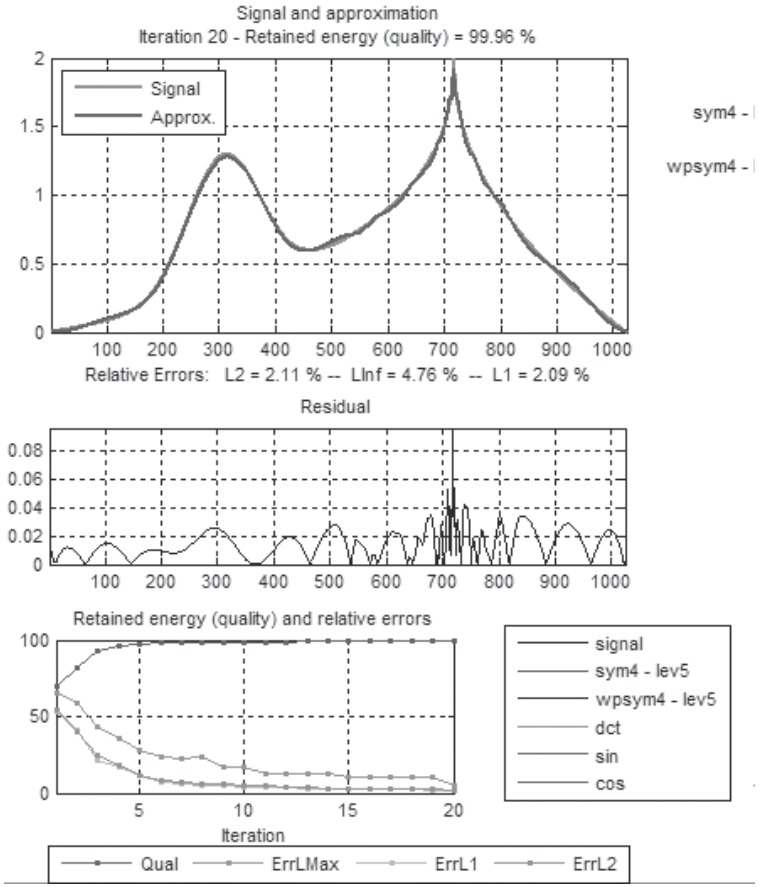


Рис. 4.4.10. Характеристики качества приближения

В правой части основного окна (рис. 4.4.11) отображаются число и индексы коэффициентов разложения по предложенному базису. Мы видим, например, что косинус представлен в сигнале тремя слагаемыми, а синус – одним. Ниже приведены соответствующие компоненты сигнала – его проекции на выбранные подсловари. Например, проекция на подсловарь синусов представлена графиком одной волны синуса, а проекция на подсловарь косинусов представлена более сложным графиком – суммой трех косинусов. Внизу окна компонент предусмотрена возможность (Highligthed) выделения компонент, относящихся к определенным подсловарям.

На правой панели внизу две кнопки (More on Components, More on Residuals), открывающие возможности получения дополнительной информации о коэффициентах и об остатке.

Отметим в заключение, что сигнал представлен всего 20 коэффициентами при относительной L2-ошибке 2.11%. Степень сжатия – в 51 раз!

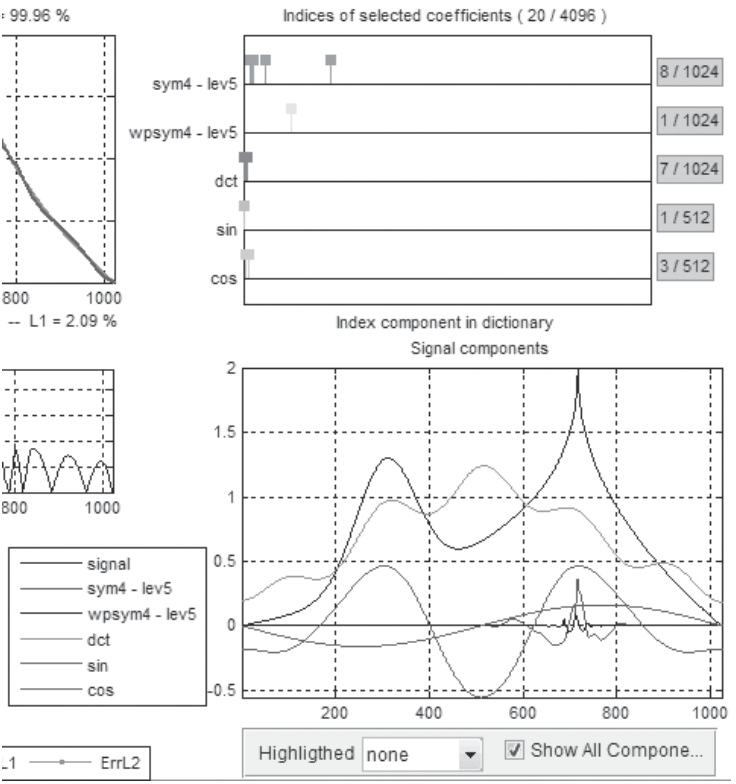


Рис. 4.4.11. Коэффициенты и компоненты приближения

4.5. Двумерный вейвлет-анализ

Раздел двумерного вейвлет-анализа представлен двумя программами:

- **Wavelet 2D** – двумерный дискретный вейвлет-анализ;
- **Wavelet Packet 2D** – двумерный пакетный вейвлет-анализ.

Рассмотрим все эти возможности главного меню **wavemenu**.

4.5.1. Двумерный дискретный вейвлет-анализ (Wavelet 2D)

Раздел **Wavelet 2D** главного меню предназначен для вейвлет-разложения и анализа изображений. Для запуска приложения нужно вызвать главное меню командой **wavemenu** и нажать кнопку **Wavelet 2D**. Запускается графическая среда, удобная для работы большого комплекса команд по анализу изображений.

Вейвлет-разложение. Загрузим изображение, используя меню **File/Load Image**. Выберем файл wbarb.mat из MATLAB-каталога toolbox/wavelet/wavedemo. Далее выбираем вейвлет **(Wavelet)**, его порядок и уровень **(Level)** вейвлет-разложения (по умолчанию 2). После этого нужно нажать на кнопку **Analyse**, для того чтобы произвести вейвлет-разложение. Результаты разложения будут изображены на основном поле графического окна (рис. 4.5.1).

По умолчанию результаты анализа представлены в «квадратном режиме» **(View Mode: Square)**. Этот режим включает четыре различных окна. В верхнем левом – оригинальное изображение. Ниже –

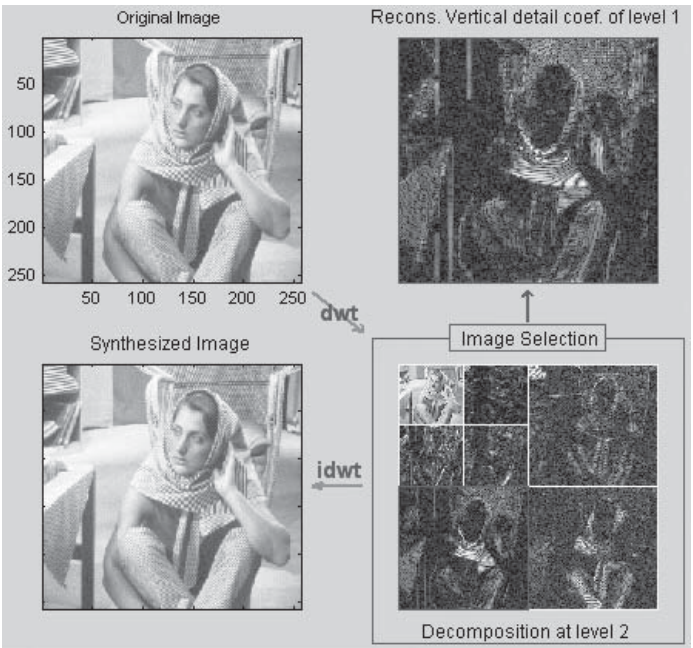


Рис. 4.5.1. Вейвлет-разложение изображения



Рис. 4.5.2. Схема расположения вейвлет-коэффициентов

восстановленное изображение после обработки коэффициентов. Внизу справа – вейвлет-разложение, представленное в виде блочной матрицы. В самом верхнем левом блоке изображаются коэффициенты приближения *A*2, далее в трех смежных блоках – детализирующие коэффициенты горизонтальные *H*2, диагональные *D*2 и вертикальные *V*2 в том порядке, как изображено на рис. 4.5.2.

Детализирующие коэффициенты первого уровня *H*1, *D*1 и *V*1 расположены вокруг указанного выше блока коэффициентов третьего уровня в аналогичном порядке. Хорошо видно, что размер коэффициентов разложения уменьшается в два раза при переходе на более высокий уровень.

Верхнее правое окно предназначено для более детального изображения отдельного блока коэффициентов. Для этого нужно выбрать мышкой блок коэффициентов (тогда он выделяется зеленой рамкой) и нажать кнопку **Visualize** из группы **Operations on selected image**. Кнопка **Full Size** открывает выбранные коэффициенты «во весь экран». Кнопка **Reconstruct** восстанавливает изображение по коэффициентам выбранного блока. Снять выделение блока можно, если по этому блоку щелкнуть еще раз.

Группа кнопок **Full Size** дает возможность открыть «во весь экран» каждое из четырех окон основного поля. Это можно сделать также при помощи **View Axes**.

Режим просмотра в виде дерева (View Mode: Tree). В этом случае вейвлет-коэффициенты будут представлены в виде дерева (рис. 4.5.3).

В первом ряду $\mathbf{L_1}$ идут коэффициенты разложения 1-го уровня, во втором ряду $\mathbf{L_2}$ идут коэффициенты разложения 2-го уровня. Дополнительно изображены аппроксимирующие коэффициенты 1-го уровня. Все кнопки работают так же, как и в первом случае. Группа кнопок **Full Size** дает возможность открыть «во весь экран» каждое из трех верхних окон основного поля.

Анализ фрагментов. Удерживая левую кнопку мыши, вы можете выделить часть изображения. Кнопками нижнего меню (например, **XY+**) можно увеличить выбранную часть и одновременно получить вейвлет-разложение этой части. Вернуться к исходному положению можно кнопкой **History <-**.

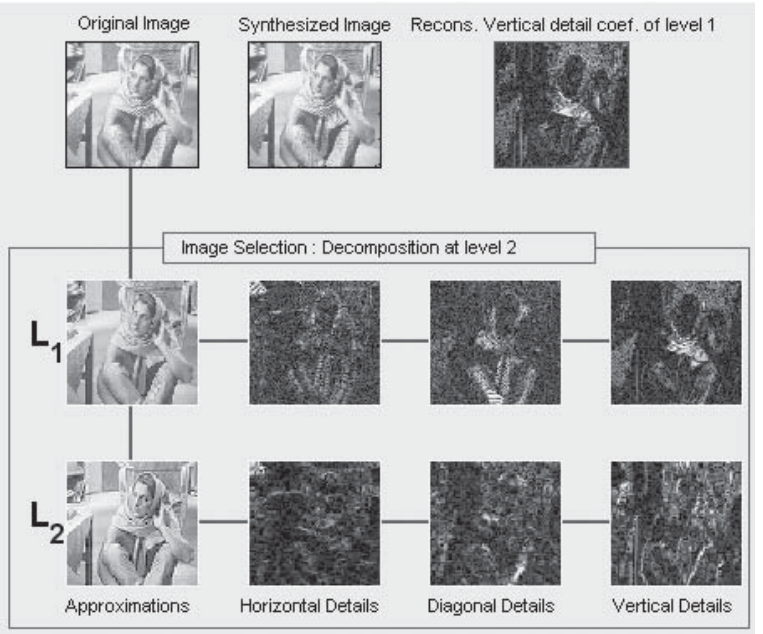


Рис. 4.5.3. Вейвлет-разложение в виде дерева

Статистический анализ, удаление шума. Группа кнопок **Statistic**, **Compress**, **Histogram**, **De-noise** позволяет получить статистические характеристики изображения, сжать изображение или очистить от шума. При нажатии на каждую из этих кнопок вызывается дополнительное окно, содержащее дополнительные функции анализа, аналогичные тем, которые были предусмотрены для одномерного сигнала. Отличие заключается только в том, что значения коэффициентов отображаются теперь в виде гистограммы. Для этого область значений коэффициентов разбивается на m одинаковых интервалов длиной Δx и вычисляется относительная плотность точек в каждом интервале $p_k = n_k/(n\Delta x)$ – отношение относительных частот попадания в этот интервал к длине интервалов. Эта величина является оценкой плотности распределения. Найденные таким образом значения p_k изображают в виде гистограммы – над каждым k-м интервалом строится прямоугольник с высотой p_k. Число промежутков разбиения задается в поле **Number of Bins**.

4.5.2. Двумерный пакетный вейвлет-анализ

Раздел **Wavelet Packet 2D** главного меню предназначен для пакетного вейвлет-разложения и анализа изображений. Для запуска приложения нужно вызвать главное меню командой **wavemenu** и нажать кнопку **Wavelet Packet 2D**. Запускается графическая среда, удобная для работы большого комплекса команд по анализу изображений.

Вейвлет-разложение. Загрузим изображение, используя меню **File/Load Image**. Выберем файл wbarb.mat из MATLAB-каталога toolbox/wavelet/wavedemo. Далее выбираем вейвлет **(Wavelet)**, его порядок, уровень **(Level)** вейвлет-разложения (по умолчанию 2) и тип энтропии **(Entropy)**. После этого нужно нажать на кнопку **Analyse**, для того чтобы произвести пакетное вейвлет-разложение. Результаты разложения будут изображены на основном поле графического окна (рис. 4.5.4).

Вверху слева будет построено дерево пакетного разложения. Узлы дерева являются активными, при выборе мышкой узла дерева в нижнем левом окне изображается график коэффициентов этого узла (см.

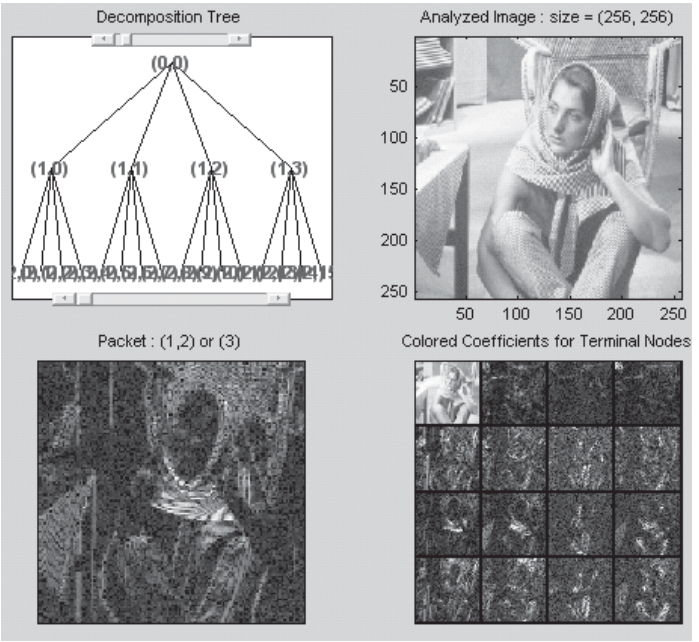


Рис. 4.5.4. Пакетное вейвлет-разложение изображения

опцию **Node Action**). Вверху справа изображается исходный образ, а ниже – изображения коэффициентов в терминальных узлах дерева блоками. Если разбить матрицу на четыре блока, то в левом верхнем блоке будут изображены коэффициенты разложения узла (1, 0), в правом верхнем – коэффициенты разложения узла (1, 1), в правом нижнем – коэффициенты разложения узла (1, 2) и в левом нижнем – коэффициенты разложения узла (1, 3). Все графические окна могут быть увеличены «во весь экран» четырьмя кнопками в группе **Full Size**.

Работа с этим приложением мало отличается от уже разобранной выше работы с одномерными сигналами.

4.6. Специализированные средства двумерного вейвлет-анализа

Раздел специализированных средств двумерного вейвлет-анализа представлен четырьмя программами:

- **SWT De-noising 2D** – удаление шума стационарного одномерного сигнала;
- **Wavelet Coefficients Selection 2D** – выбор вейвлет-коэффициентов изображения;
- **Image Fusion** – слияние (наложение) изображений;
- **True Compressipn 2D** – полное сжатие с использованием вейвлетов.

Рассмотрим все эти возможности главного меню **wavemenu**.

4.6.1. Удаление шума изображения (SWT De-noising 2D)

Раздел **SWT De-noising 2D** главного меню предназначен для вейвлет-разложения и удаления шума в изображении. Для запуска приложения нужно вызвать главное меню командой **wavemenu** и нажать кнопку **SWT De-noising 2D**. Запускается графическая среда, удобная для вейвлет-анализа изображения и удаления шума.

Загрузим изображение, используя меню **File/Load Image**. Выберем файл noiswom.mat из MATLAB-каталога toolbox/wavelet/wavedemo. Далее выбираем вейвлет (**Wavelet**), его порядок и уровень (**Level**) вейвлет-разложения (по умолчанию 2). После этого нужно нажать на кнопку **Decompose Image**, для того чтобы произвести вейвлет-разложение. Результаты разложения будут изображены на

основном поле графического окна. В левом верхнем окне появится исходное изображение, справа – очищенное от шума. Ниже будут отображены гистограммы детализирующих коэффициентов вейвлет-разложения. Эти гистограммы расположены рядами в соответствии с уровнями разложения.

Можно выбрать различные варианты для стратегий удаления шума. По умолчанию принимаются мягкая пороговая обработка и немасштабированный белый шум (**Fixed form soft thresholding** и **Unscaled white noise**). Слайдеры, расположенные справа, управляют уровнями порогов, вертикальные линии на гистограммах, обозначенные желтыми пунктирами, также позволяют выбирать пороги. Для выполнения очистки от шума нужно нажать кнопку **De-noise**. Результаты этой операции представлены на рис. 4.6.1.

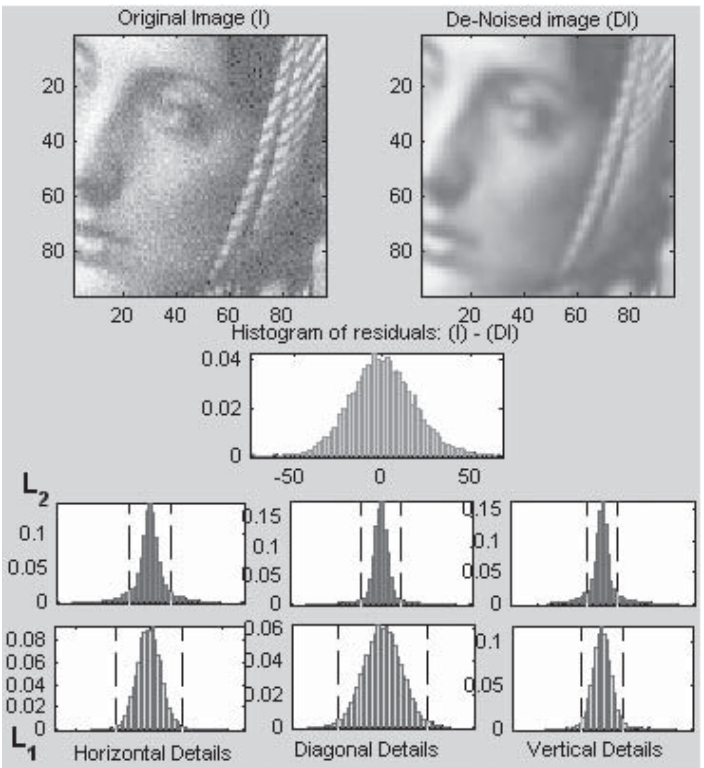


Рис. 4.6.1. Очистка от шума изображения

Если очищенное изображение не удовлетворяет, нужно выбрать другой метод пороговой обработки, другие пороги или другой вейвлет и уровень разложения.

4.6.2. Выбор вейвлет-коэффициентов изображения (Wavelet Coefficients Selection 2D)

Раздел **Wavelet Coefficients Selection 2D** предназначен для более тонкого выбора вейвлет-коэффициентов при анализе изображений. В отличие от гистограмм коэффициентов, здесь будут отображаться сами коэффициенты, окрашенные в соответствии с их значениями. Для запуска этого приложения нужно вызвать главное меню командой **wavemenu** и нажать кнопку **Wavelet Coefficients Selection 2D**. Открывается окно, которое содержит обычное меню вверху, а также специальные кнопки меню справа и внизу и слайдеры для выбора коэффициентов. Можно использовать следующие стратегии выбора вейвлет-коэффициентов:

- глобальный выбор самых больших коэффициентов в абсолютном значении (**Global**);
- поуровневый выбор самых больших коэффициентов (**By Level**);
- автоматический пошаговый выбор самых больших коэффициентов (**Stepwise movie**).

Для работы этого раздела необходимо включить симметричный режим расширения при помощи команды **dwtmode('sym')** в командном окне Matlab. Работа с данным разделом вполне аналогична предыдущему случаю одномерного сигнала.

Загрузим данные, используя меню **File/Load Image**. После этого появится диалоговое окно, в котором можно выбрать также демонстрационные сигналы. Выберем mat-файл noiswom.mat из MATLAB-каталога toolbox/wavelet/wavedemo.

Выполнение вейвлет-разложения. В правом меню **Wavelet** выберем вейвлет `db3` и уровень разложения 3 в меню **Level**, затем нажмем кнопку **Analyze**. Результаты представлены на рис. 4.6.2.

В двух окнах слева будут изображены: оригинальный сигнал и его вейвлет-коэффициенты. В левом верхнем углу изображаются коэффициенты приближения *A*3, далее в трех смежных блоках – детализирующие коэффициенты горизонтальные *H*3, диагональные *D*3 и вертикальные *V*3 по часовой стрелке. Детализирующие коэффициенты второго уровня *H*2, *D*2 и *V*2 расположены вокруг указанного выше

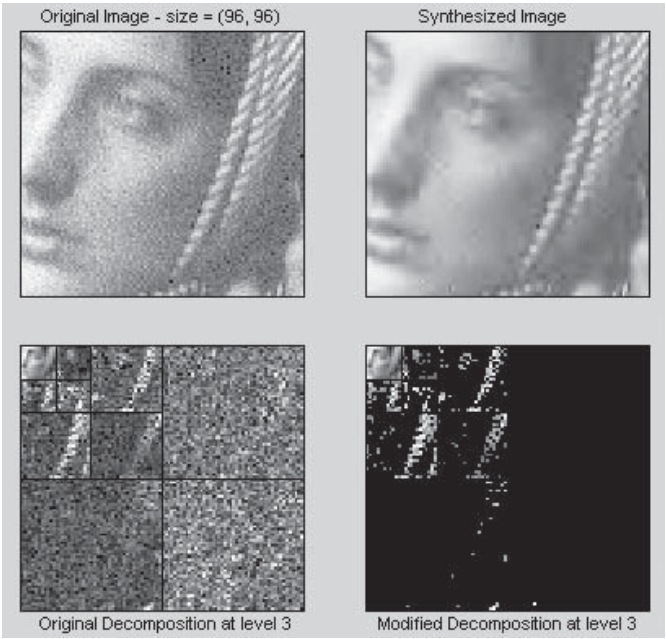


Рис. 4.6.2. Разложение изображения

блока коэффициентов третьего уровня в аналогичном порядке, далее идут коэффициенты первого уровня *H*1, *D*1 и *V*1. Хорошо видно, что размер коэффициентов разложения уменьшается в два раза при переходе на более высокий уровень.

Справа будет изображен синтезированный сигнал, ниже отображаются выбранные коэффициенты.

Режимы выбора аппроксимирующих коэффициентов можно выбрать в меню **App. Cfs**:

- **Select All** – выбраны все аппроксимирующие коэффициенты;
- **Selectable** – аппроксимирующие коэффициенты участвуют в выборе наравне с другими;
- **Unselect** – не выбраны (удалены) все аппроксимирующие коэффициенты.

Число самых больших коэффициентов на каждом уровне можно задать, напечатав числа в соответствующих полях или используя слайдер. Правой кнопкой мыши можно выделить фрагмент изображения и кнопкой нижнего меню **XY+** получить его разложение. К первоначальному разложению можно вернуться стрелкой **<-** из

History нижнего меню. Разность между исходным сигналом и синтезированным можно оценить дополнительно, используя кнопку **Residuals**.

Синтезированный сигнал может быть сохранен на диске в различных форматах. В конце работы необходимо вернуть режим расширения к исходному режиму добавления нулей при помощи команды **dwtmode('zpd')**.

4.6.3. Слияние двух изображений (Image Fusion)

Слияние (наложение) изображений и массивов можно сделать как из командной строки MATLAB при помощи функций **wfusimg** и **wfusmat**, так и при помощи приложения **Image Fusion** главного меню пакета Wavelet Toolbox. Для запуска этого приложения нужно вызвать главное меню командой **wavemenu** и нажать кнопку **Image Fusion**. Появится окно приложения, на правой части которого имеется возможность выбора вейвлета, уровня разложения и методов слияния. После уточнения этих параметров нужно загрузить два изображения одинакового размера, затем нажать кнопку **Decompose** для вейвлет-разложения изображений, выбрать метод выбора коэффициентов аппроксимации и деталей, нажать кнопку **Apply** для исполнения слияния.

Идея *слияния изображений* заключается в следующем. После вейвлет-разложения изображений делается выбор коэффициентов для нового изображения из коэффициентов аппроксимации и детализации оригинальных изображений. Затем по выбранным коэффициентам восстанавливается изображение обычным методом вейвлет-восстановления.

Возможно несколько *методов выбора* коэффициентов для нового изображения из двух, соответствующих двум оригинальным изображениям:

- 'max' – выбор максимального коэффициента;
- 'min' – выбор минимального коэффициента;
- 'mean' – выбор среднего значения из двух коэффициентов;
- 'img1' – выбор коэффициента первого изображения;
- 'img2' – выбор коэффициента второго изображения;
- 'rand' – случайный выбор из двух коэффициентов.

Кроме того, имеется еще несколько параметрических методов выбора. Поскольку коэффициенты вейвлет-разложений изображений составлены из матриц, поясним эти методы на матрицах. Пусть $A(i,j)$ и $B(i,j)$ представляют коэффициенты вейвлет-разложений первого и

второго изображений соответственно и C(i, j) – коэффициенты вейвлет-разложения для объединенного изображения. Тогда доступные параметрические, зависящие от параметра $0 \le t \le 1$, методы выбора коэффициентов следующие:

- 'linear' – линейный, тогда C = A* t + B*(1 – t), $0 \le t \le 1$;
- 'UD_fusion' – слияние сверху вниз с параметром $t \ge 0$. Для определения этого метода задается линейно распределенный массив

  ```
  x = linspace(0,1,size(A,1));
  ```

  определяется степенная функция $P = x^t$:

  ```
  P = x.^t;
  ```

  Тогда каждая строка матрицы C вычисляется по формуле

  ```
  C(i,:) = A(i,:)*(1-P(i)) + B(i,:)*P(i);
  ```

  Этот метод обеспечивает плавное перетекание сверху вниз первого изображения во второе (в соответствии с функцией x^t);
- 'DU_fusion' – слияние снизу вверх;
- 'RL_fusion' – слияние справа налево;
- 'UserDEF' – слияние слева направо;
- 'UserDEF' – определенный пользователем метод, тогда вместо параметра задается функция пользователя Func.Name, такая что C = Func.Name(A, B).

Меню **File** обеспечивает возможность выбора серии демонстрационных примеров. Выберем два изображения **fond_bou** и **mask** и проведем слияние при помощи вейвлета sym4, уровня разложения 2, выбором коэффициентов аппроксимации по методу 'UD_fusion', а детализирующих коэффициентов – по методу 'max', результат показан на рис. 4.6.3.

Кнопка **Inspect Fusion Tree** открывает новое окно, в котором изображена схема дерева разложения. Щелкая мышкой по узлам этого дерева, внизу основного окна мы получаем изображение вейвлет-коэффициентов соответствующих изображений. Открывающийся список **Nodal Label** позволяет изменить способ обозначения терминальных узлов. Открывающийся список **Nodal Action** позволяет увидеть коэффициенты в узлах дерева (**Visualize**), восстановить изображения по коэффициентам только этого узла (**Reconstruct**) либо еще раз сделать разложение коэффициентов аппроксимации или свернуть это разложение (**Split/Merge**).

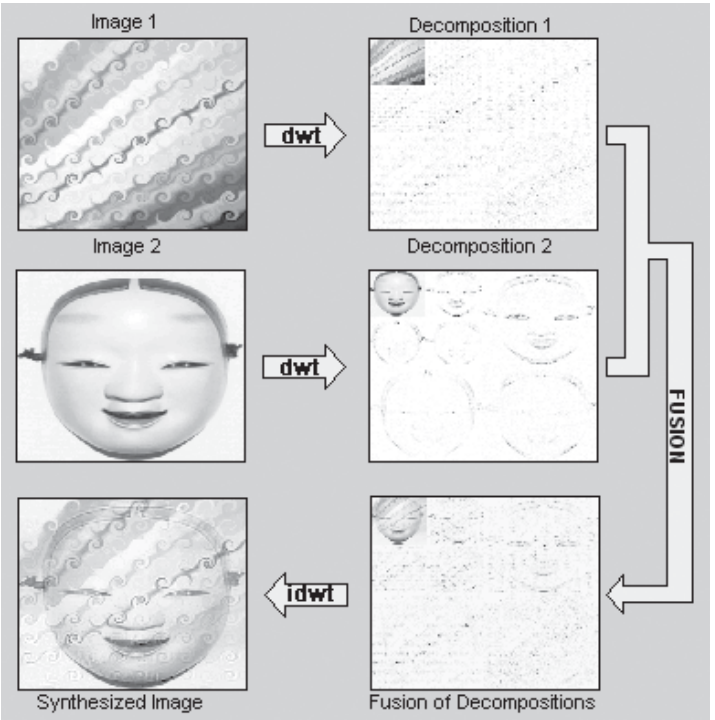


Рис. 4.6.3. Слияние двух изображений методами UD_fusion – max

4.6.4. Полное сжатие с использованием вейвлетов (True Compression 2D)

Раздел **True Compression 2D** предназначен для сжатия сигнала с использованием вейвлет-разложения и пороговой обработки коэффициентов совместно с процедурой квантования и кодирования. Полная каскадная последовательность сжатия включает фазы вейвлет-обработки, квантования, кодирования и раскодирования. Рисунок 4.6.4 показывает чисто вейвлет-сжатие (левая часть) и подлинное сжатие с использованием дополнительных процедур. Отметим, что формат JPEG2000 также использует вейвлет-разложение совместно с процедурой квантования и кодирования коэффициентов. Напомним, что квантование (или округление) производится по следующей формуле:

$$q = sign(y)\left[\frac{|y|}{\Delta}\right]\Delta,$$

где y – исходное значение коэффициента, $sign(y)$ – знак коэффициента, Δ – значение шага округления и $[x]$ – целая часть числа. Мертвая зона квантователя – это интервал диапазоном 2Δ около нуля, она дает большее количество нулей на выходе. Идеи кодирования несравнимо сложнее, см. об этом на сайте http://rain.ifmo.ru/cat/view.php/.

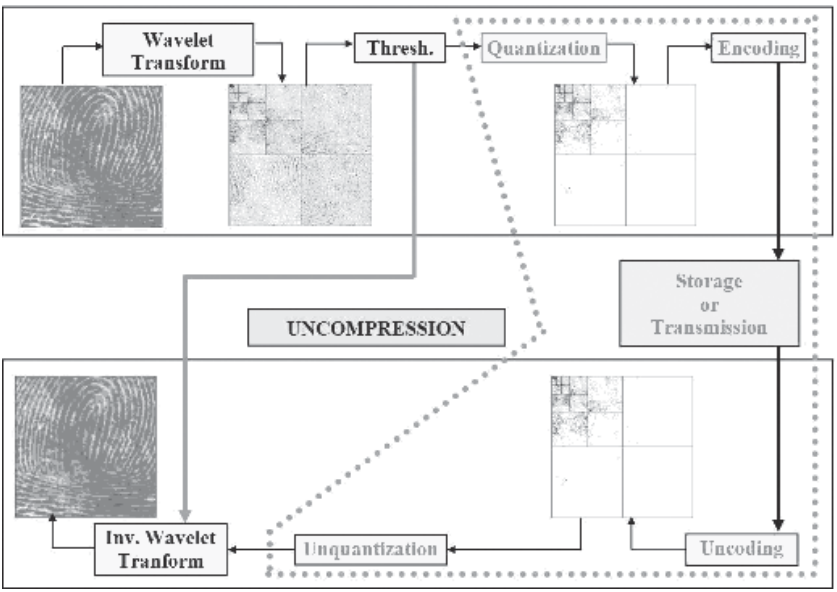


Рис. 4.6.4. Схема подлинного сжатия
(рисунок из документации MATLAB)

Методы сжатия разбиваются на две группы. Первую группу представляют методы пороговой обработки коэффициентов (СТМ-методы). В MATLAB предложены три метода, которые производятся каскадно, простыми шагами с выполнением пороговой обработки (глобальной или уровневой) и кодирования квантованием. Используется либо фиксированный, либо метод Хаффмана в зависимости от выбора метода. О кодировании Хаффмана см., напр., http://ru.wikipedia.org/wiki/Код_Хаффмана и приведенные там источники.

Методы пороговой обработки коэффициентов (СТМ-методы):

- 'gbl_mmc_f' – глобальная пороговая обработка коэффициентов и фиксированное кодирование;
- 'gbl_mmc_h' – глобальная пороговая обработка коэффициентов и фиксированное кодирование Хаффмана;
- 'lvl_mmc' – субполосная пороговая обработка коэффициентов и фиксированное кодирование Хаффмана.

Доступны более сложные методы, которые комбинируют вейвлет-разложение и квантование. Это основной принцип прогрессивных методов. Прогрессивность позволяет во время декодирования получить изображение, разрешение которого постепенно увеличивается. Кроме того, возможно получить ряд степеней компрессии, основанных на длине сохраненного кода. Это сжатие обычно вызывает потери информации, но данный вид алгоритмов включает также сжатие без потерь. Такие методы основаны на нескольких идеях. Во-первых, можно использовать вейвлет-разложение, для того чтобы добиться разреженности (большого количества нулевых коэффициентов) и классические методы кодирования. Идея, имеющая решающее значение для использования вейвлетов в сжатии изображений, состоит в том, чтобы использовать фундаментально древовидную структуру вейвлет-разложения. Определенные коды, разработанные с 1993 до 2000 года, используют эту идею, в частности алгоритм кодирования EZW вложенного нуль-дерева, введенный Шапиро (Shapiro J. M. (1993). Embedded image coding using zerotrees of wavelet coefficients. – IEEETrans. Signal Proc. – Vol. 41. – No. 12. – P. 3445–3462). Алгоритм EZW комбинирует пошаговую пороговую обработку и последовательное квантование, сосредоточиваясь на более эффективном способе кодирования коэффициентов изображения, чтобы минимизировать отношение компрессии. Две разновидности SPIHT и STW являются усовершенствованными версиями первоначального алгоритма EZW.

Полный обзор этих прогрессивных методов можно найти в работе: Walker J. S. Wavelet-Based Image Compression. – University of Wisconsin, Eau Claire, Wisconsin, USA, 1999.

Прогрессивные методы кодирования:

- 'ezw' – метод Шапиро вложенного нуль-дерева (Embedded Zero tree Wavelet);
- 'spiht', 'spiht_3d' – метод разбиения множества в иерархических деревьях (Set Partitioning In Hierarchical Trees);
- 'stw' – метод пространственно ориентированного вейвлет дерева (Spatial-orientation Tree Wavelet);

- 'wdr' – метод редукции вейвлет разности (Wavelet Difference Reduction);
- 'aswdr' – метод редукции адаптивно сканируемой вейвлет разности (Adaptively Scanned Wavelet Difference Reduction).

Рассмотрим на примере работу приложения **True Compression 2D**. Для запуска нужно вызвать главное меню командой **wavemenu** и нажать кнопку **True Compression 2D**. Открывается окно, которое содержит обычное меню вверху, основное окно для отображения результатов, а также специальные кнопки меню справа и внизу.

Используя меню **File/Example**, загрузим тестовое изображение **Mask**, с методом сжатия GBL_MMC_F? вейвлет Хаара, уровень разложения 4. Сразу начинается процесс анализа, предусмотренный этим тестовым примером, и выводятся результаты и параметры компрессии (рис. 4.6.5 и 4.6.6).

Числовые результаты компрессии:

- MSE – среднеквадратичная ошибка (Mean Square Error). Вычисляется по формуле $MSE = \frac{1}{mn}\sum_{i=0}^{m-1}\sum_{j=0}^{n-1}|X(i,j) - X_c(i,j)|^2$, где X – исходный сигнал, а X_c – сжатый;
- Max Error – максимальная ошибка $\max_{ij}|X(i,j) - X_c(i,j)|$;
- L2 Norm Ratio – отношение L2-норм сжатого сигнала и исходного;
- PSNR – мера ошибки в децибелах (Peak Signal to Noise Ratio), $PSNR = 10\log_{10}\left(\frac{255^2}{MSE}\right)$. Чем выше PSNR, тем лучше качество сжатого или восстановленного изображения. Типичные значения для сжатия с потерями изображения – между 30 и 50 децибелами и когда PSNR больше, чем 40 децибелов, тогда эти два изображения неразличимы;
- BPP – коэффициент Bit-Per-Pixel – это число битов, необходимых для хранения одного пикселя изображения. Для серых (grayscale) исходных изображений BPP равно 8, а для полноцветных начальное значение BPP равно 24 (RGB – это три матрицы);
- Comp. Ratio – коэффициент сжатия (CR). Он означает, что сжатый образ хранится, используя только CR% от размера хранения начального образа.

Параметры компрессии задаются в среднем разделе правой панели (рис. 4.6.5). Прежде всего нужно задать методы компрессии. Для каждого метода свои параметры. Для данного примера выбраны параметры, показанные на рис. 4.6.6. В предложенных полях можно уста-

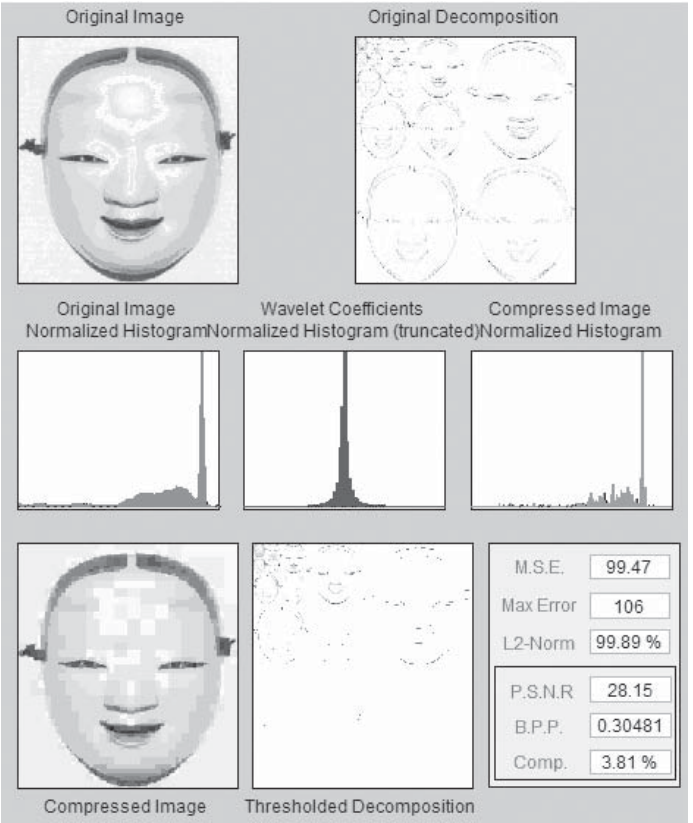


Рис. 4.6.5. Результаты компрессии

навливать желаемые параметры. Эти параметры не являются независимыми: изменение одного параметра приводит к изменению других. После корректировки параметров нужно нажать кнопку **Compress** и получить результат с новыми характеристиками качества. Отметим, что для прогрессивных методов компрессии нужно задавать количество циклов. Чем больше количество циклов, тем лучше качество сжатого сигнала, но хуже характеристики сжатия.

Кнопка **Inspect Wavelet Trees** открывает новое окно для получения дополнительной информации о структуре мультиразложения сжатого образа. Открывающееся вейвлет-дерево имеет активные узлы, при выборе которых внизу отображаются коэффициенты разложения (рис. 4.6.7).

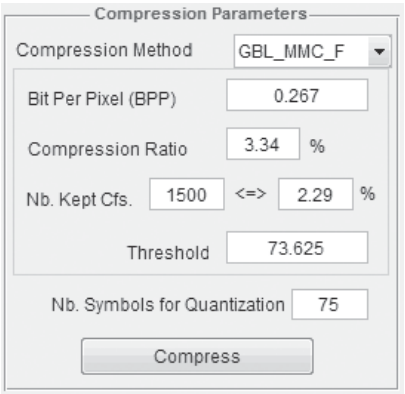


Рис. 4.6.6. Параметры компрессии

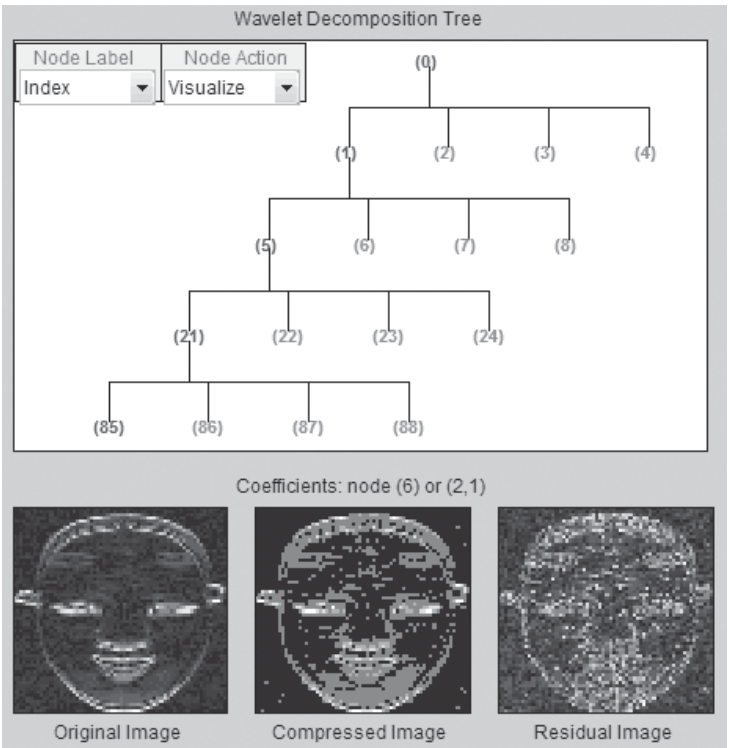


Рис. 4.6.7. Дерево разложения, узел (6)

4.7. Трехмерный вейвлет-анализ (Wavelet 3D)

Раздел **Wavelet 3D** главного меню предназначен для вейвлет-разложения и анализа трехмерных сигналов. Такой сигнал можно представлять себе как тело, но актуальнее считать третьим измерением время, а трехмерный сигнал – как видеозапись, или серию, слайдов. Для изучения подобного сигнала из командной строки можно использовать функции (см. раздел 3.3.3):

- **dwt3** – одноуровневое дискретное 3D-вейвлет-преобразование;
- **idwt3** – одноуровневое обратное дискретное 3D-вейвлет-преобразование;
- **wavedec3** – многоуровневое 3D-вейвлет-разложение;
- **waverec3** – многоуровневое 3D-вейвлет-восстановление.

В данном разделе рассмотрим вейвлет-анализ трехмерного сигнала при помощи графического интерфейса пользователя `wavemenu`.

Трехмерный аналоговый сигнал задается функцией $s(x, y, z)$ трех переменных. Алгоритм разложения сигнала $s(x, y, z)$ аналогичен тому, что применяется в двумерном случае. Пусть $\varphi(x)$ – масштабирующая (одномерная) вейвлет-функция и $\psi(x)$ – вейвлет. Как известно, они порождают базисные функции $\varphi_{j,n}(x)$ и $\psi_{j,n}(x)$. Трехмерный сигнал $s(x, y, z)$ раскладывается по базисным в $L^2(\mathbf{R}^3)$ функциям:

$$\varphi_{j,n}(x)\varphi_{j,m}(y)\varphi_{j,k}(z),$$
$$\varphi_{j,n}(x)\varphi_{j,m}(y)\psi_{j,k}(z), \quad \varphi_{j,n}(x)\psi_{j,m}(y)\varphi_{j,k}(z), \quad \psi_{j,n}(x)\varphi_{j,m}(y)\varphi_{j,k}(z),$$
$$\varphi_{j,n}(x)\psi_{j,m}(y)\psi_{j,k}(z), \quad \psi_{j,n}(x)\varphi_{j,m}(y)\varphi_{j,k}(z), \quad \psi_{j,n}(x)\psi_{j,m}(y)\varphi_{j,k}(z),$$
$$\psi_{j,n}(x)\psi_{j,m}(y)\psi_{j,k}(z).$$

Получается 8 групп коэффициентов. Они организованы в трехмерный 2×2×2 массив ячеек. Ячейка dec{i, j, k}, $i, j, k = 1, 2$, содержит коэффициенты, которые получены при помощи низкочастотного фильтра, соответствующего функции $\varphi(x)$, если i, j или $k = 1$, и получены при помощи высокочастотного фильтра, соответствующего функции $\psi(x)$, если i, j или $k = 2$. Например, ячейка dec{1, 1, 1} содержит аппроксимирующие коэффициенты, а ячейка dec{1, 1, 2} содержит детализирующие коэффициенты, соответствующие вейвлетам $\varphi_{j,n}(x)$ $\varphi_{j,m}(y)\psi_{j,k}(z)$. Группы коэффициентов разложения в MATLAB обозначаются также символами: AAA, AAD, ADA, ADD, DAA, AAD, DDA, DDD, смысл которых понятен. Напомним, что все размеры массивов коэффициентов разложения уменьшаются вдвое, по сравнению с размерами исходного массива.

При изучении данного раздела очень полезно рассмотреть демонстрационный пример Wavelet Analysis for 3D Data, представленный в Help\Wavelet Toolbox\Wavelet Toolbox Examples (функция wavelet3ddemo.m), и примеры трехмерного вейвлет-анализа в главном вейвлет-меню wavemenu. Трехмерный анализ проводится по следующей схеме:

- загрузка трехмерных данных;
- анализ трехмерных данных;
- выбор и просмотр срезов;
- определение движения срезов;
- создание полного 3D-образа;
- импорт и экспорт информации.

Рассмотрим возможности графического интерфейса пользователя. Для запуска приложения нужно вызвать главное меню командой `wavemenu` и нажать кнопку **Wavelet 3D**. Запускается графическая среда, удобная для работы большого комплекса команд по анализу трехмерных сигналов.

Загрузим тестовый трехмерный сигнал, используя меню **File/Example Analysis**. Выберем пример MRI, который моделирует снимки МРТ головы. Сразу начинается процесс анализа, предусмотренный этим тестовым примером. Чтобы этого избежать, можно загрузить этот же сигнал из файла wmri.mat из MATLAB-каталога toolbox/wavelet/wavelet.

После загрузки сигнала открывается первый срез (z = 1) загруженного тела (рис. 4.7.1) и отображаются его размеры. Блоки панели справа позволяют управлять процессом анализа. В верхней части панели справа можно выбрать вейвлеты по всем трем осям, уровень

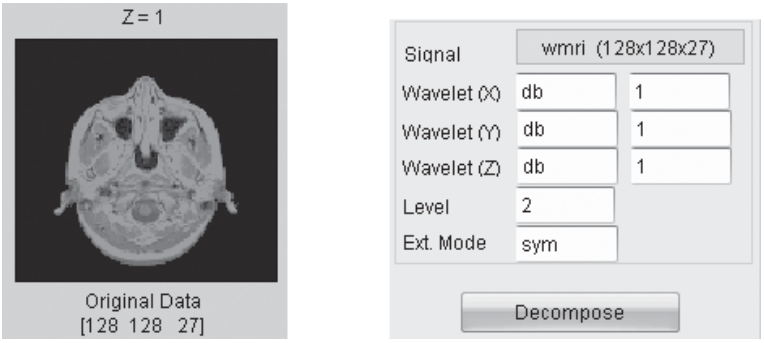


Рис. 4.7.1. Начальный слайд и выбор вейвлетов для разложения

разложения и моду расширения (рис. 4.7.1). Нажмем **Decompose**. В результате мы получаем снимки для $z = 1$ (рис. 4.7.2):

- верхний ряд – три изображения для $z = 1$: исходный сигнал, восстановленный из коэффициентов аппроксимации старшего уровня разложения и восстановленный из всех детализирующих коэффициентов;
- два нижних ряда – все 8 типов коэффициентов разложения верхнего уровня (в нашем случае 2).

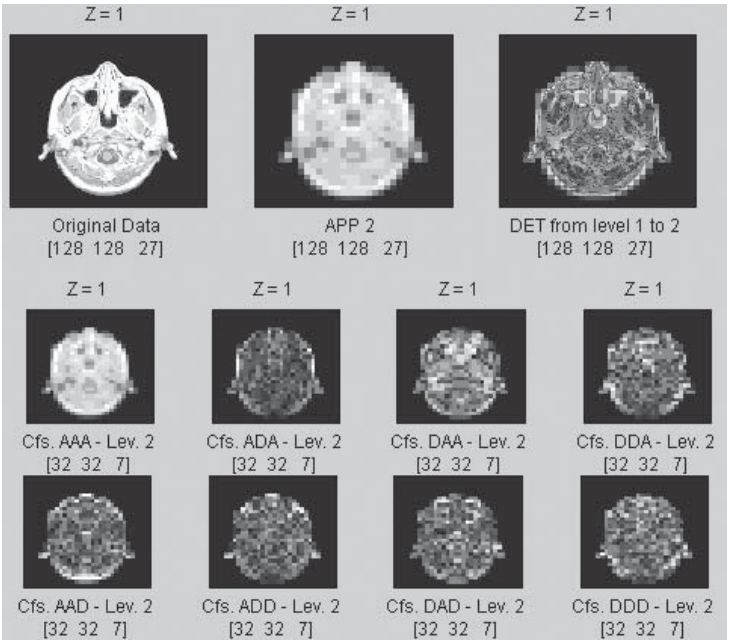


Рис. 4.7.2. Результаты разложения

На правой панели активируются раздел управления просмотром результатов разложения (рис. 4.7.3):

- текстовое поле для выбора направления представления срезов (по умолчанию Z = const);
- текстовое поле для выбора уровня отображаемых коэффициентов;
- два слайдера для выбора отображаемого уровня среза Z = *n*, один для верхнего ряда срезов, а другой – для нижнего ряда коэффициентов;

- кнопка Slice Movie – для просмотра коэффициентов для всех срезов Z = *n* «в движении»;
- текстовое поле для выбора трехмерного просмотра сигнала и результатов разложения. Отметим, что дается трехмерное изображение, восстановленное по выбранным группам коэффициентов (рис. 4.7.4).

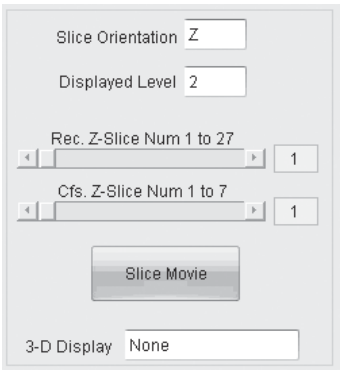


Рис. 4.7.3. Управление просмотром результатов разложения

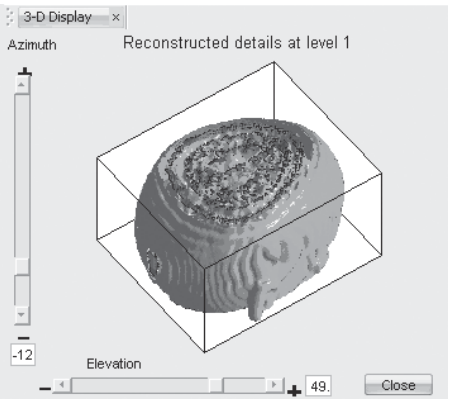


Рис. 4.7.4. Трехмерное представление восстановленных деталей уровня 1

Для примера выберем в окне **3D Display** опцию DET1. Тогда производится восстановление сигнала только по всем коэффициентам деталей (7 групп) первого уровня разложения. Результат отображается в трехмерном виде с возможностями вращения. Для этого служат два движка слева и внизу графического окна (рис. 4.7.4).

Все результаты можно сохранить в MAT-файле.

4 .8. Мультисигналы (Multiple 1D)

Раздел вейвлет-анализа мультисигналов представлен тремя программами:

- **Multisignal Analysis 1D** – анализ мультисигнала;
- **Multivariate Denoising** – многовариантное удаление шума;
- **Multiscale Princ. Comp. Analysis** – многомасштабный анализ главных компонент.

Рассмотрим все эти возможности главного меню **wavemenu**.

4.8.1. Вейвлет-анализ мультисигнала

Мультисигналом будем называть набор из нескольких сигналов одинаковой длины. Например, это могут быть строки матрицы, представляющей изображение, числовые данные нескольких каналов кардиосигнала.

Пакет расширения Wavelet Toolbox имеет ряд функций (см. раздел 3.3.4), специально созданных для анализа мультисигнала. Их можно использовать как из командной строки, так и через графический интерфейс главного вейвлет-меню (wavemenu). Покажем на примере исследование мультисигнала в главном вейвлет-меню. Для открытия программы анализа нужно выполнить команду `wavemenu` и затем выбрать кнопку **Multisignal Analysis 1D** для запуска программы анализа мультисигнала.

Открывающееся основное окно графического интерфейса разделено на четыре панели. Правая панель – обычная, для выбора параметров, как и в других окнах GUI. Три новые панели (рис. 4.8.1):

- **Visualization of Selected Data** – для визуализации выбранных данных;
- **Information on Selected Data** – для информации о выбранных сигналах;
- **Selection of Data** – для выбора отдельных сигналов мультисигнала.

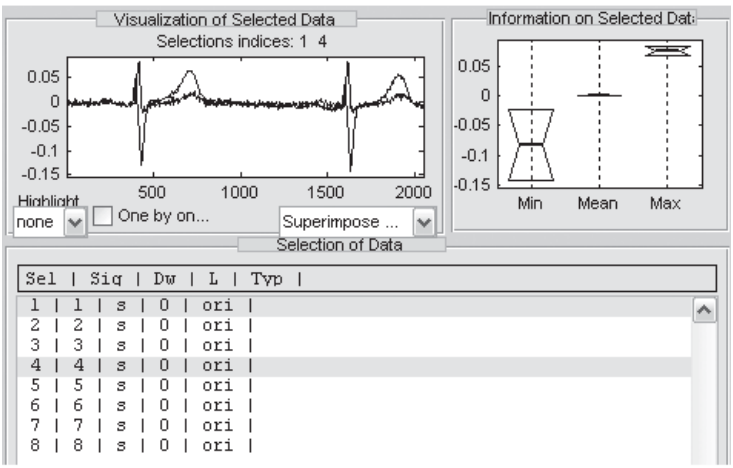


Рис. 4.8.1. Загруженный сигнал ECG для анализа

Загрузим сигнал, используя меню **File**. Для загрузки возьмем MAT-файл ECG.mat, содержащий 8 каналов кардиосигнала, каждый длины 2048.

Панель **Selection of Data** содержит список загруженных сигналов для выбора. Вначале доступны только первоначально загруженные сигналы. Первый сигнал является выбранным при загрузке. Его график отображается в панели **Visualization of Selected Data**, а на панели **Information on Selected Data** – информация о сигнале: максимальное, минимальное значения и среднее. Далее можно генерировать и добавлять новые сигналы к списку, раскладывая, сжимая или очищая от шума оригинальные сигналы.

На панели **Selection of Data** можно выбрать несколько сигналов. Это делается мышкой, при нажатой клавише **Ctrl** или **Shift** (кнопка **Select ALL** внизу панели **Selection of Data** выделяет, а кнопка **Clear** очищает все сигналы).

Выберем сигналы 1 и 4. Графики отобранных сигналов (1, 4) отображаются совместно в панели **Visualization of Selected Data**. Панель **Information on Selected Data** содержит графики минимумов, средних и максимумов этих сигналов.

Графики отображаются совместно и накладываются друг на друга. Иногда это неудобно. Поэтому есть возможность выбора различных видов представления графиков. Для того чтобы отобразить все графики отдельно, нужно внизу панели **Visualization of Selected Data** изменить режим представления на **Separate Mode**. Кроме того, есть возможность один из сигналов выделить более жирно и другим цветом. Для этого используется открывающийся список **Highlight Sel** в нижнем левом углу панели **Visualization of Selected Data**.

Разложение мультисигнала. Для выбора режимов разложения обратимся к правой верхней панели (рис. 4.8.2). Проведем разложение по строкам, при помощи вейвлета db2 и до уровня 3 с модой расширения sym. После выбора параметров нужно нажать кнопку **Decompose**.

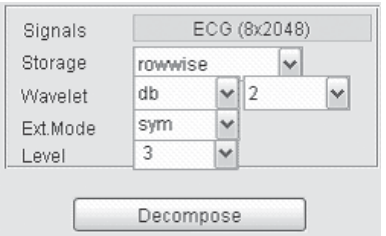


Рис. 4.8.2. Панель выбора параметров разложения

В результате происходит разложение всех оригинальных сигналов, и сигнал 1 автоматически выбирается (рис. 4.8.3).

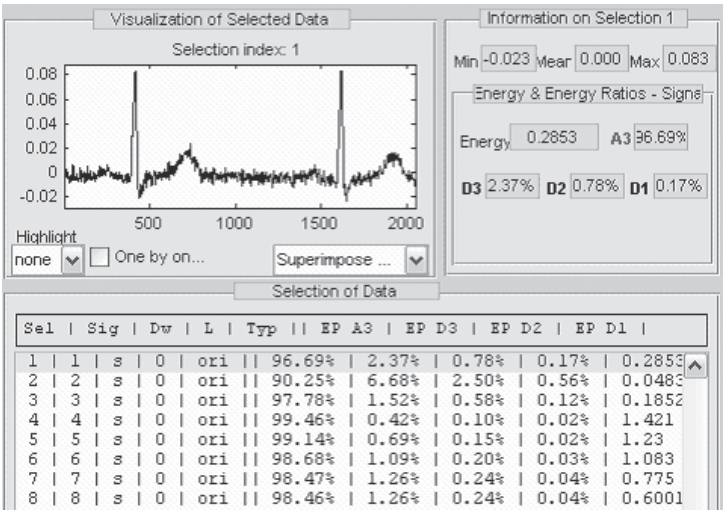


Рис. 4.8.3. Разложение сигнала

В панели **Selection of Data** добавляется новая информация для каждого оригинального сигнала – процент от энергии вейвлет-компонентов (**D1**, **D2**, **D3** и **A3**) и полная энергия сигнала (L^2-норма). Панель **Information on Selected Data** содержит информацию относительно отобранного сигнала: **Min**, **Mean**, **Max** – и распределение энергии сигнала по компонентам разложения.

Так как оригинальные сигналы разложены и появляются новые объекты, правая панель **Selection of Data Sets** обновляется (рис. 4.8.4), в ней появляются новые имена массивов, полученных после разложения.

Эта панель позволяет выбрать новые полученные сигналы для просмотра их графиков и информации. При этом список в правом столбце – для выбора вейвлет-коэффициентов, а левый – для выбора компонент сигнала, восстановленных только по указанному набору коэффициентов. Отметим, что в списке справа векторы могут иметь различные длины, но только компоненты одной и той же длины могут быть отобраны вместе (например, **APP1** и **DET1**).

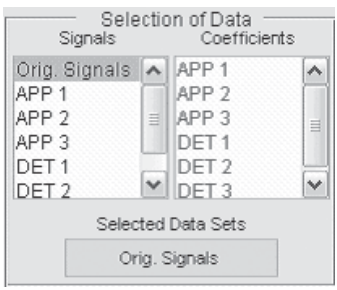


Рис. 4.8.4. Панель выбора массивов

В панели **Visualization of Selected Data** в списке режимов представления сигналов появляется несколько дополнительных возможностей. В частности, если выбрать режим **Full Dec Mode**, то отображаются (совместно) компоненты разложения отобранных сигналов (рис. 4.8.5).

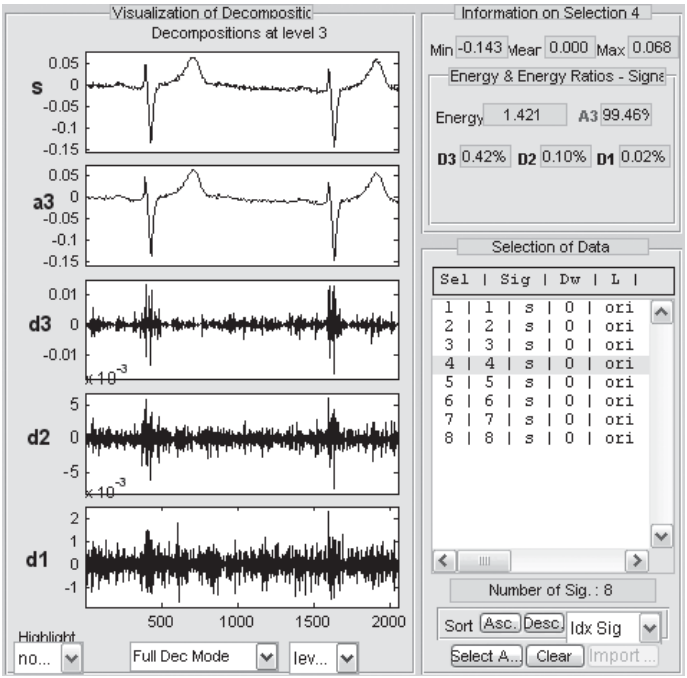


Рис. 4.8.5. Компоненты разложения 4-го сигнала

Отметим несколько других режимов визуализации. Режим **Full Dec Mode (Cfs)** – для отображения коэффициентов декомпозиций отобранных сигналов. Режим представления **Stem Mode** – для отображения коэффициентов декомпозиций в виде стеблей. Режим **Tree Mode** – для отображения разложения в виде дерева с активными узлами.

Как уже отмечалось, в панели **Selection of Data Sets** можно выбрать несколько различных вейвлет-компонент и коэффициентов. Например, при нажатой клавише **Ctrl** выбрать мышкой в списке слева в панели **Selection of Data Sets** элементы **Orig. Signals**, **APP 2** и **DET 2** для выбора этих трех наборов сигналов. Общее количество отобранных данных **(Number of Sig.)** отображается в панели **Selection of**

Data: три набора по 8 сигналов каждый – в общей сложности 24 сигнала. Отметим, что в таблице признаки каждого выбираемого сигнала обновлены: символом **a** отмечается компонента аппроксимации, символом **d** – компонента детализации **s** – для сигнала. Отобранные сигналы отсортированы в порядке возрастания сначала индекса сигнала и порядка выбора уровня разложения. Метод сортировки можно изменить. Например, нажатие на кнопку **Asc** в панели **Sort** устанавливает сортировку. Сначала идут три выбранные компоненты разложения первого сигнала, затем 3 компоненты разложения второго сигнала и т. д.

Статистика, кластеризация, удаление шума и сжатие мультисигнала. Для выбора этих процедур в правой части окна имеются четыре кнопки. При нажатии каждой из них открывается отдельное окно для соответствующего анализа мультисигнала.

Для примера рассмотрим сжатие мультисигнала. Нажимаем кнопку **Compress**. Появляется окно **Compression** (рис. 4.8.6). Отметим,

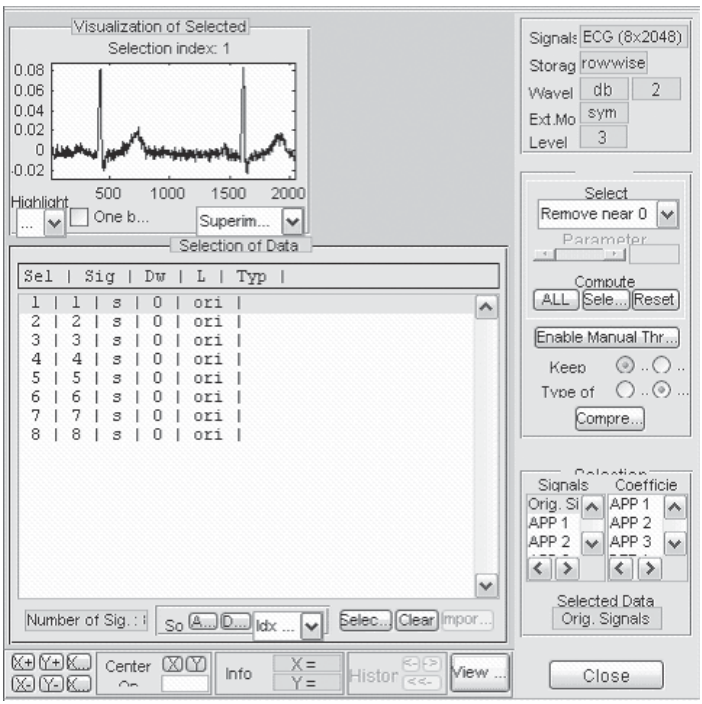


Рис. 4.8.6. Окно сжатия сигнала

что все оригинальные сигналы сжимаются при нажатии кнопки **Compress**.

Перед сжатием нужно выбрать правило для вычисления порогов на панели **Thresholding** (рис. 4.8.7). Это можно сделать выбором метода в открывающемся списке **Select Compression Method** и подтверждениемп применения этого метода для выбранных сигналов кнопкой **Selected**. Кнопка **ALL** обеспечивает применение процедуры сжатия ко всем сигналам.

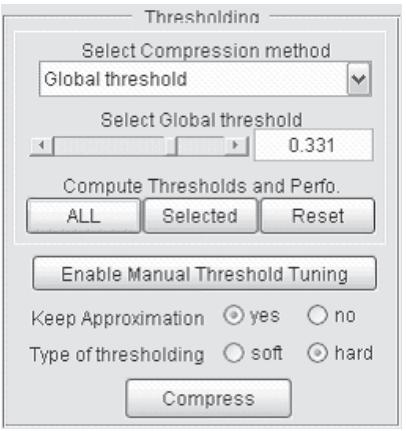


Рис. 4.8.7. Панель **Thresholding**

На панели **Selection of Data** отображаются пороги для коэффициентов деталей каждого уровня (**ThrD1**, **ThrD2**, **ThrD3**), отношение энергии (**En. Rat.**) сжатого сигнала к исходному и проценты нулевых коэффициентов (**NbZ Rat.**) (рис. 4.8.8).

Нажимаем кнопку **Compress** внизу панели **Thresholding**.

Sig	ThrD1	ThrD2	ThrD3	En. Rat.	NbZ Rat.
1	0.028	0.028	0.028	96.69%	87.45%
2	0.018	0.018	0.018	90.25%	87.45%
3	0.012	0.012	0.012	97.78%	87.45%
4	0.031	0.031	0.031	99.46%	87.45%
5	0.041	0.041	0.041	99.14%	87.45%
6	0.058	0.058	0.058	98.68%	87.45%
7	0.047	0.047	0.047	98.47%	87.45%
8	0.032	0.032	0.032	98.46%	87.45%

Рис. 4.8.8. Пороги и проценты сжатия

Ниже, в панели **Selection of Data Sets**, можно выбрать новые наборы данных: исходные сигналы, компоненты приближения и деталей, сжатый сигнал и его компоненты разложения и аналогичные данные для коэффициентов (префикс CMP- означает, что соответствующий массив относится к сжатому сигналу). Для примера выберем оригинальные и сжатые сигналы. Они отображаются в списке панели **Selection of Data**, всего 16 сигналов. Отметим сигналы под номерами 1 и 9 – это первый сигнал, исходный и сжатый. Они совместно отображаются на панели визуализации. Выделим более жирным и другим цветом сжатый сигнал 9. Результат показан на рис. 4.8.9.

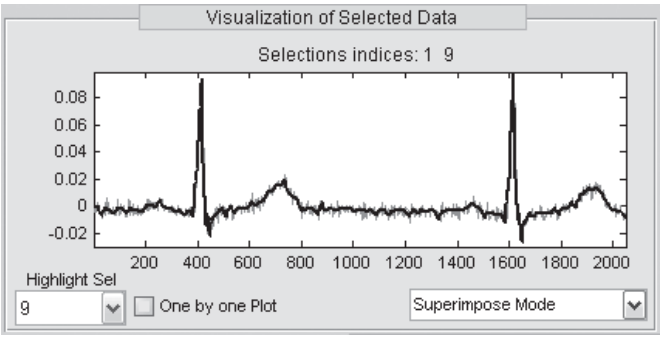


Рис. 4.8.9. Исходный и сжатый сигналы

Остальные программы: статистика, кластеризация сигналов и удаление шума мультисигнала – работают аналогично.

4.8.2. Многовариантное удаление шума (Multivariate Denoising)

В этом разделе излагаются методы удаления шума многомерного сигнала пакета Wavelet Toolbox, в которых объединяются вейвлет-методы и анализ главных компонент.

Многомерный дискретный сигнал X можно рассматривать как последовательность точек $\{X_k\}$ в n-мерном пространстве $\mathbf{R}^n$ или как набор из n одномерных сигналов, составленных из координат X_{ik} последовательности $\{X_k\}$. Для удаления шума многомерного сигнала можно использовать методы одномерного вейвлет-анализа, примененные отдельно для каждой i-й компоненты $\{X_{ik}\}$ многомерного сигнала. В то же время имеются специальные методы анализа многомерных сигналов. Достаточно распространенным является анализ главных компо-

нент (Principal Components Analysis, PCA) многомерного временного ряда [Ра], [ДЖ].

Пакет Wavelet Toolbox предлагает методы удаления шума многомерного сигнала, основанные на одновременном использовании одномерного вейвлет-анализа и анализа главных компонент (PCA). Напомним сначала кратко теоретические основы PCA.

Анализ главных компонент

Рассмотрим вещественный конечный n-мерный сигнал $X = \{X_k\}$, $k = 1, 2, ..., N$, составленный из n координатных сигналов $\{X_{ik}\}$. Образуем матрицу $X = \{X_{ik}\}$ из n строк и N столбцов (каждая строка представляет отдельный одномерный сигнал).

Основная идея метода *главных компонент* заключается в том, чтобы из матрицы X размера n на N получить какие-нибудь ее алгебраические характеристики, которые потом можно будет использовать для анализа сигнала. Предлагается следующий подход. Умножим матрицу X на транспонированную, получим квадратную симметричную матрицу $\Sigma = X \cdot X^T$ порядка n. Она имеет вещественные собственные числа $\lambda_1 \geq \lambda_2 \geq ... \geq \lambda_n \geq 0$ и ортонормированную систему собственных векторов $\{v_1, v_2, ..., v_n\}$ (адаптированный базис). Тогда симметричная матрица Σ раскладывается в виде произведения $\Sigma = V\Lambda V^T$ диагональной матрицы Λ собственных чисел и ортогональной матрицы V собственных векторов-столбцов $v_1, v_2, ..., v_n$:

$$\Lambda = \begin{pmatrix} \lambda_1 & 0 & \cdots & 0 \\ 0 & \lambda_2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \lambda_n \end{pmatrix}, \quad V = \begin{pmatrix} V_{11} & V_{21} & \cdots & V_{n1} \\ V_{12} & V_{22} & \cdots & V_{n2} \\ \vdots & \vdots & \ddots & \vdots \\ V_{1n} & V_{2n} & \cdots & V_{nn} \end{pmatrix}.$$

Это равносильно диагонализации ковариационной матрицы Σ за счет перехода к другому базису $v_i = \sum_{j=1}^{n} V_{ij} e_j$, $e_j = \sum_{s=1}^{n} V_{sj} v_s$.

Как уже отмечалось, многомерный дискретный сигнал X можно рассматривать как последовательность точек $\{X_k\}$ в n-мерном пространстве $\mathbf{R}^n$ или как набор из n одномерных сигналов, составленный из координат X_{ik} последовательности $\{X_k\}$ в стандартном базисе $\{e_1, e_2, ..., e_n\}$ пространства $\mathbf{R}^n$. Перейдем теперь к новому базису $\{v_1, v_2, ..., v_n\}$. Ясно, что координаты точек $\{X_k\}$ изменятся: $X_k = (Y_{1k}, ..., Y_{nk})^T$. Найдем эти координаты:

$$X_k = \sum_{j=1}^{n} X_{jk} e_j = \sum_{j,s=1}^{n} X_{jk} V_{sj} v_s = \sum_{j,s=1}^{n} V_{sj} X_{jk} v_s = \sum_{s=1}^{n} Y_{sk} v_s,$$

где $Y_{sk} = \sum_{j=1}^{n} V_{sj} X_{jk}$.

Таким образом, новые компоненты $\{Y_s\}$ сигнала X в адаптированном базисе находятся по формуле $Y_{sk} = \sum_{j=1}^{n} V_{sj} X_{jk}$, то есть представляют собой матрицу вида $Y = V \cdot X$. Эти новые компоненты $\{Y_s\}$ сигнала X называют главными компонентами. Все они соответствуют собственным числам $\lambda_1 \geq \lambda_2 \geq ... \geq \lambda_n \geq 0$. Считается, что чем меньше собственное число λ_s, тем меньше значения в сигнале X имеет соответствующая компонента $Y_s = \{Y_{sk}\}$. При малых λ_s ее можно не учитывать, что приводит к выделению в сигнале главных компонент. Этот метод позволяет получить: n числовых характеристик, это собственные числа $\lambda_1 \geq \lambda_2 \geq ... \geq \lambda_n \geq 0$ ковариационной матрицы Σ; адаптированный базис собственных векторов $\{v_1, v_2, ..., v_n\}$ и главные компоненты сигнала. После удаления «неглавных компонент», которые оценены как несущественные (шумовые), остаются только главные компоненты $(Y_1, ..., Y_i, 0, ..., 0)^T$, которые образуют новую матрицу $\tilde{Y}$. Из этой матрицы $\tilde{Y}$ и из матрицы V восстанавливается новая матрица сигнала по формуле $\tilde{X} = V^T \tilde{Y}$. Этот сигнал можно считать многомерным сигналом, очищенным от шумовых компонент методом PCA.

Изложим метод главных компонент более аккуратно. Пусть $X = \{X_k\}$ – конечный n-мерный сигнал, $k = 1, 2, ..., N$, составленный из n координатных сигналов $\{X_{ik}\}$. Образуем матрицу $X = \{X_{ik}\}$ из n строк и N столбцов (каждая строка представляет отдельный одномерный сигнал). Вычислим средние арифметические и стандартные отклонения по строкам матрицы X:

$$\overline{X}_i = \frac{1}{N} \sum_{k=1}^{N} X_{ik}, \quad S_i = \sqrt{\frac{1}{N} \sum_{k=1}^{N} (X_{ik} - \overline{X}_i)^2}.$$

Обозначим через X^* матрицу, полученную из X центрированием по строкам и нормированием по стандартным отклонениям S_i, $X^*_{ik} = (X_{ik} - \overline{X}_i) / S_i$. Образуем выборочную ковариационную матрицу Σ размера n на n как произведение X^* и транспонированной:

$$\Sigma = \frac{1}{N} X^* (X^*)^T, \quad \Sigma_{ki} = \frac{1}{N} \sum_{j=1}^{N} \frac{1}{S_k S_i} (X_{kj} - \overline{X}_k)(X_{ij} - \overline{X}_i).$$

Найдем собственные числа $\lambda_1 \geq \lambda_2 \geq ... \geq \lambda_n \geq 0$ и собственные векторы симметричной матрицы Σ. Другими словами, найдем ее разложение в виде произведения $\Sigma = V \Lambda V^T$ диагональной матрицы Λ собственных чисел и ортогональной матрицы V собственных векторов-столбцов:

$$\Lambda = \begin{pmatrix} \lambda_1 & 0 & \cdots & 0 \\ 0 & \lambda_2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \lambda_n \end{pmatrix}, \quad V = \begin{pmatrix} V_{11} & V_{21} & \cdots & V_{n1} \\ V_{12} & V_{22} & \cdots & V_{n2} \\ \vdots & \vdots & \ddots & \vdots \\ V_{1n} & V_{2n} & \cdots & V_{nn} \end{pmatrix}.$$

Это равносильно диагонализации ковариационной матрицы Σ за счет перехода к другому базису.

Главными компонентами $Y_1, ..., Y_n$ многомерного сигнала X называются строки следующей матрицы $Y = V \cdot X^*$. В этом случае собственные числа имеют смысл выборочных дисперсий главных компонент $Y_1, ..., Y_n$. Отметим, что сумма собственных чисел равна n, поэтому величина $(\lambda_i/n)100\%$ имеет смысл доли дисперсии в процентах, которая используется для оценки доли информации, которую несет главная компонента Y_i.

Практически это используется так, что если доля $(\lambda_i/n)100\%$ главной компоненты оценивается как малая, то считается, что соответствующая главная компонента Y_i несет мало информации и может быть удалена как шумовая компонента. После удаления компонент сигнала, которые оценены как несущественные, остаются только главные компоненты ($Y_1, ..., Y_i, 0, ..., 0$), которые образуют новую матрицу $\tilde{Y}$. Из этой матрицы $\tilde{Y}$ и из матрицы V восстанавливается новая матрица сигнала по формуле $\tilde{X}^* = V^T\tilde{Y}$. Затем производятся децентрирование и денормализация матрицы $\tilde{X}^*$. Полученная в итоге матрица $\tilde{X}$ представляет многомерный сигнал, очищенный от шумовых компонент методом PCA.

Многовариантное удаление шума многомерного сигнала

Функция **wmulden** позволяет использовать как вейвлет-разложение, так и анализ главных компонент для удаления шума сигнала. Эта функция может использоваться и в командной строке, и при помощи графического интерфейса пользователя (wavemenu). Описание функции wmulden и пример ее использования можно найти в разделе 3.10.5 и в документации MATLAB. Здесь мы рассмотрим инструментальные средства графического интерфейса главного вейвлет-меню.

Для открытия графического интерфейса **Multivariate De-Noising** достаточно вызвать главное вейвлет-меню командой `wavemenu` и выбрать компонент **Multivariate Denoising**.

Загрузим, используя меню **File ⇒ Load Signals**, тестовый массив в виде MAT-файла ex4mwden.mat из каталога toolbox/wavelet/

wmultsig1d. Этот сигнал есть матрица, содержащая четыре столбца, где каждый столбец – сигнал для удаления шума. Все сигналы – это зашумленные версии простых комбинаций двух оригинальных сигналов. Первый есть "Blocks", который является нерегулярным, второй – "HeavySine", который является регулярным, за исключением двух точек. Другие два сигнала – сумма и разность между оригинальными сигналами. Многомерный гауссов белый шум, показывающий сильную пространственную корреляцию, добавлен к результирующим четырем сигналам. Приведем графики (рис. 4.8.10) двух первых исходных и зашумленных сигналов.

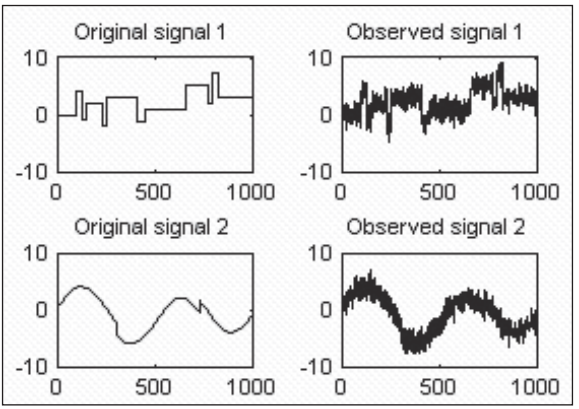


Рис. 4.8.10. Оригинальные сигналы «Blocks» и «HeavySine»

После загрузки сигнала на основном окне будет три столбца: первый содержит графики зашумленных сигналов, во втором столбце будут изображаться коэффициенты разложения, третий столбец – для графиков очищенных сигналов.

Рассматриваемый пример иллюстрирует применение двух методов удаления шума. Сначала выполним удобное изменение базиса, чтобы справиться с пространственной корреляцией, и удаление шума в новом базисе. Затем используем анализ главных компонент (PCA), чтобы применять в своих интересах отношения между сигналами, что приводит к дополнительному удалению шума.

Выполним вейвлет-разложение и диагонализируем ковариационную матрицу. Панель справа вверху предназначена для выбора параметров вейвлет-разложения. Используем значения параметров по

умолчанию. Затем нажимаем кнопку **Decompose and Diagonalize**. Во втором столбце окна отображаются вейвлет-коэффициенты аппроксимации и детализации каждого сигнала в исходном базисе (рис. 4.8.12). Выберем **Noise Adapted Basis** для отображения сигналов и их коэффициентов в адаптированном базисе.

Чтобы увидеть подробную информацию об этом новом базисе, нужно нажать **More on Noise Adapted Basis**. Появляющееся новое окно отображает шумовую оценочную ковариационную матрицу и соответствующие собственные векторы и собственные значения (рис. 4.8.11).

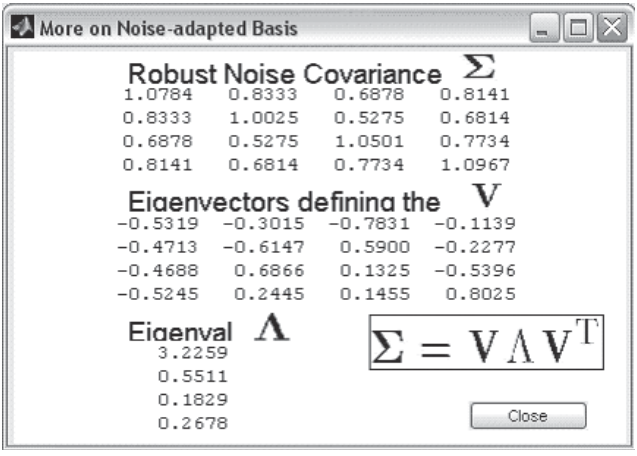


Рис. 4.8.11. Ковариационная матрица и элементы нового базиса

Собственные векторы определяют изменение базиса, собственные значения – дисперсии некоррелированных шумов в новом базисе.

Метод удаления шума многомерного сигнала, предложенный ниже, интересен, если шумовая ковариационная матрица далека от диагональной, что имеет место в этом примере.

Для того чтобы настроить алгоритм удаления шума, имеется множество опций. Однако мы будем использовать их значения по умолчанию: fixed form thresholding, soft, PCA и предложенные числа сохраненных основных компонентов. В этом случае значения по умолчанию для PCA приводят к сохранению всех компонентов.

Выберем **Original Basis** для возвращения к исходному базису и затем нажмем **De-noise**.

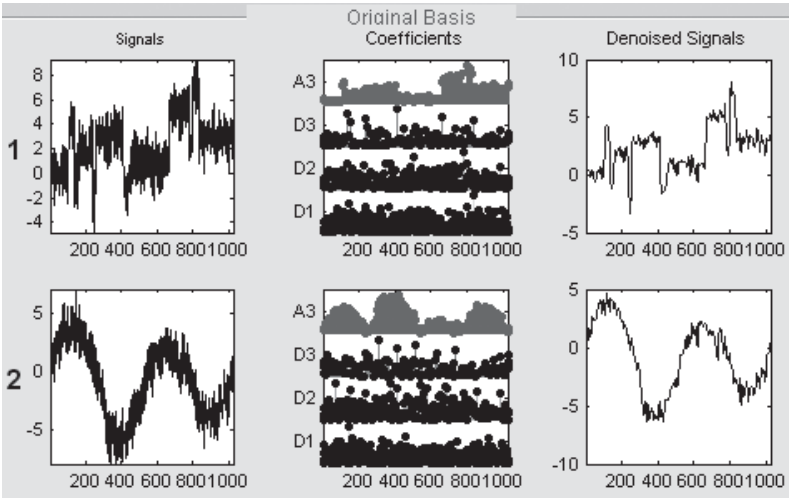


Рис. 4.8.12. Графики двух первых сигналов и их очищенных версий

Оба из двух первых сигналов достаточно правильно восстановлены. Результаты удовлетворительны, но они могут быть улучшены с учетом более подробной информации о главных компонентах (до сих пор использовались все 4 главные компоненты). Нажмем для этого **More on Principal Components**.

Новое окно (рис. 4.8.13) отображает информацию для выбора числа компонентов, которые следует оставить для PCA-приближения и для финального PCA после вейвлет-восстановления. Мы видим, что только две первые главные компоненты представляют интерес.

Закроем окно **More on Principal Components** и выберем число 2 главные компоненты для параметра **Nb. of PC for APP** и число 2 главные компоненты для параметра **Nb. of PC for final PCA** и затем нажмем **De-noise**.

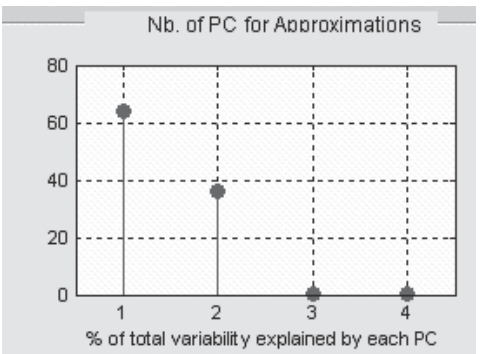


Рис. 4. 8.13. Информативность главных компонент

Результаты лучше (рис. 4.8.14), чем полученные ранее. Первый сигнал, который является нерегулярным, также достаточно хорошо восстанавливается. Второй сигнал, который является более регулярным, гораздо лучше очищен от шума после этой второй стадии PCA. Можно получить подробную информацию об удаленном шуме, нажимая **Residuals**.

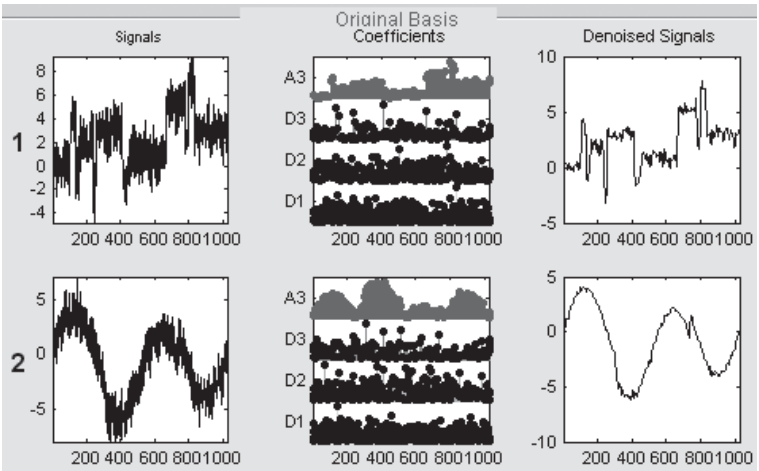


Рис. 4.8.14. Удаление шума с применением PCA

Сохранение результатов. Очищенные от шума сигналы можно сохранить на диске в виде MAT-файла при помощи пунктов меню **File ⇒ Save De-Noised Signals**, или **File ⇒ Save De-Noised Signals and Parameters**. В последнем случае сохраняются две структуры DEN_Params и PCA_Params и матрица очищенного сигнала.

Структура PCA_Params содержит параметры изменения базиса при PCA и имеет вид:

```
PCA_Params
PCA_Params =
    NEST: {[4x4 double]  [4x1 double]  [4x4 double]}
     APP: {[4x4 double]  [4x1 double]  [2]}
     FIN: {[4x4 double]  [4x1 double]  [2]}
```

Поле PCA_Params.NEST{1} содержит матрицу преобразования базиса. Поле PCA_Params.NEST{2} содержит собственные значения, и поле PCA_Params.NEST{3} – оценочная шумовая матрица ковариа-

ции. Поле PCA_Params.APP{1} содержит матрицу преобразования базиса, поле PCA_Params.APP{2} содержит собственные значения, и PCA_Params.APP{3} – число сохраненных основных компонент для приближений.

Та же самая структура используется для PCA_Params.FIN для конечного PCA.

Структура DEN_Params содержит параметры удаления шума в диагональном базисе:

```
DEN_Params =
     thrVAL: [4.8445 2.0024 1.1536 1.3957 0]
    thrMETH: 'sqtwolog'
    thrTYPE: 's'
```

Пороги представлены параметром thrVAL. Метод порогового значения дается параметром thrMETH, и режим порогового значения – параметром thrTYPE.

4.8.3. Многомасштабный анализ главных компонент

В предыдущем разделе анализ главных компонент применялся к многомерному сигналу *X* совместно с вейвлет-удалением шума. Однако при вейвлет-разложении сигнала *X* получаются новые многомерные сигналы: это матрицы вейвлет-коэффициентов аппроксимации и детализации и матрицы вейвлет-компонентов сигнала. Естественно применить анализ главных компонент и для обработки этих новых многомерных сигналов с последующим вейвлет-восстановлением. Выбором главных компонент вейвлет-коэффициентов можно упростить сигнал.

В этом разделе покажем особенности многомасштабного вейвлет-анализа многомерного сигнала с применением анализа *главных компонент* не только к самому сигналу, но и к вейвлет-коэффициентам и вейвлет-компонентам многомерного сигнала *X*. Для проведения такого анализа пакет Wavelet Toolbox содержит функцию **wmspca**. Эта функция может использоваться и в командной строке, и при помощи графического интерфейса пользователя wavemenu. Описание функции wmspca и примеры ее применения можно найти в разделе 3.5.2. Здесь мы рассмотрим инструментальные средства графического интерфейса главного вейвлет-меню.

Для открытия графического интерфейса **Multiscale Principal Components Analysis** достаточно вызвать главное вейвлет-меню командой `wavemenu` и выбрать компонент **Multiscale Princ. Comp. Analysis**.

Загрузим, используя меню **File** ⇒ **Load Signals**, тестовый массив в виде MAT-файла ex4mwden.mat из каталога toolbox/wavelet/wmultsig1d. Этот сигнал есть матрица, содержащая четыре столбца, где каждый столбец – сигнал для удаления шума. Все сигналы – это зашумленные версии простых комбинаций двух оригинальных сигналов. Первый есть `"Blocks"`, который является нерегулярным, и второй – `"HeavySine"`, который является регулярным, кроме двух точек (рис. 4.8.15). Другие два сигнала – сумма и разность между оригинальными сигналами. Многомерный гауссов белый шум, показывающий сильную пространственную корреляцию, добавлен к результирующим четырем сигналам. Приведем графики (рис. 4.8.15) двух первых исходных и зашумленных сигналов.

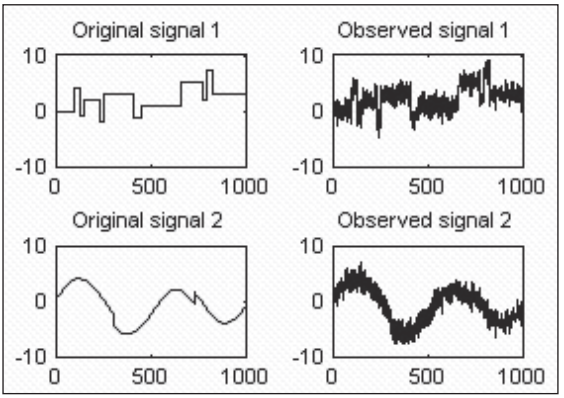


Рис. 4.8.15. Оригинальные сигналы

После загрузки этого тестового примера на основном окне будет три столбца: первый содержит графики зашумленных сигналов, во втором столбце будут изображаться коэффициенты разложения, третий столбец – для графиков очищенных сигналов.

Выберем параметры вейвлет-разложения, предложенные по умолчанию (правая верхняя панель). Затем нажимаем кнопку **Decompose and Diagonalize**. Во втором столбце окна отображаются вейвлет-коэффициенты аппроксимации и детализации каждого сигнала в исходном базисе (рис. 4.8.18).

Чтобы увидеть подробную информацию о новом базисе собственных векторов, нужно нажать **More on Noise Adapted Basis**. Появляющееся новое окно отображает собственные векторы и собственные

значения для многомерных сигналов, полученных при разложении. Выбрать конкретный сигнал можно в открывающемся списке внизу данного окна (рис. 4.8.16). Отметим, что для коэффициентов аппроксимации сумма собственных значений не равна $n = 4$, что говорит об использовании анализа главных компонент без центрирования сигнала.

Для выполнения простого многомасштабного PCA нужно выбрать параметры PCA в правой панели (рис. 4.8.17). Начальные значения

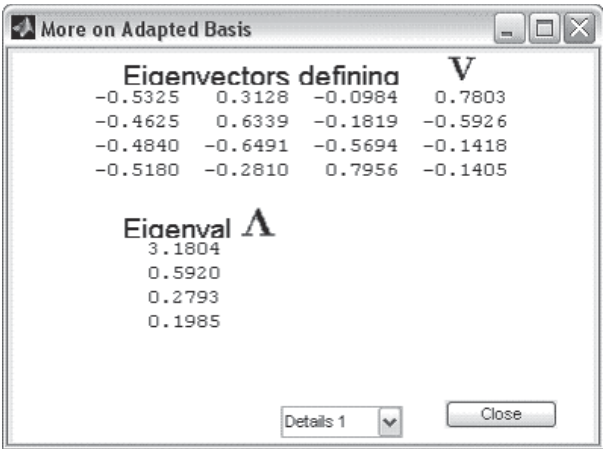


Рис. 4.8.16. Собственные векторы и собственные значения коэффициентов разложения

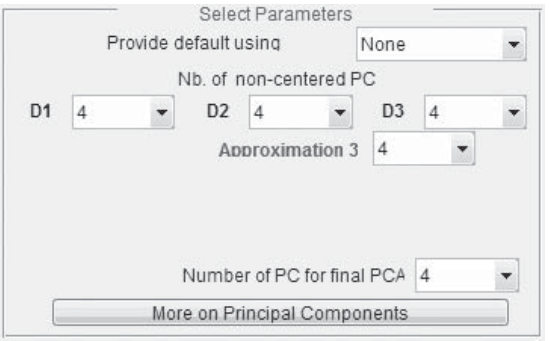


Рис. 4.8.17. Панель выбора числа главных компонент

для PCA предполагают сохранение всех компонентов. Выберем метод **Kaiser** в открывающемся списке **Provide default using** (рис. 4.8.17) и число главных компонент 1 – для всех детализирующих коэффициентов и 2 – для финального PCA, а затем нажмем кнопку **Apply**.

В результате сигнал стал немного лучше. Приведем графики двух первых сигналов (рис. 4.8.18).

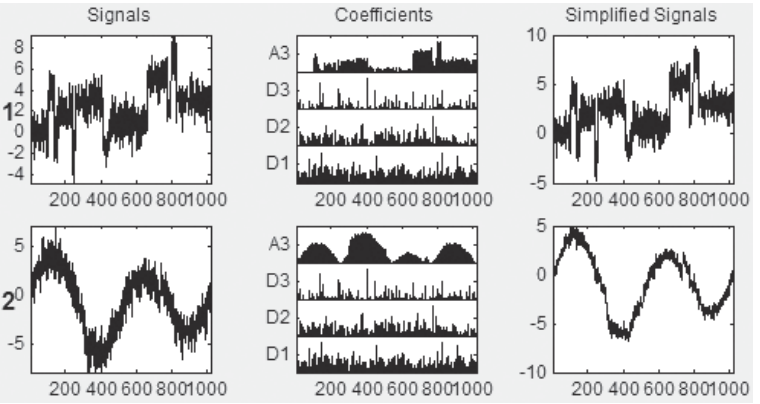


Рис. 4.8.18. Графики двух первых сигналов и их очищенных версий

Можно улучшить полученный результат, сохраняя меньше основных компонентов. Как известно, детализирующие коэффициенты на низких уровнях представляют шумы, и их удаление дает хороший эффект. Имея в виду это обстоятельство, установим нулевое значение числа главных компонент **Nb. of non-centered PC** для коэффициентов **D1** и **D2** и нажмем снова **Apply**. Полученные результаты значительно лучше (рис. 4.8.19). Можно получить подробную информацию об удаленном шуме, нажимая кнопку **Residuals**.

Сохранение результатов. Многомасштабный PCA позволяет сохранять упрощенные сигналы на диске в виде MAT-файла при помощи пунктов меню **File ⇒ Save Simplified Signals** или **Save Simplified Signals and Parameters**. В последнем случае сохраняются структура PCA_Params и матрица упрощенного сигнала. Структура PCA_Params:

```
PCA_Params =
1x7 struct array with fields:
```

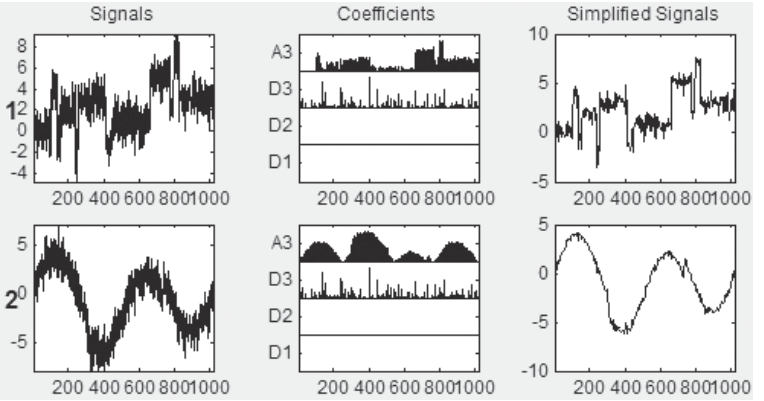


Рис. 4.8.19. Графики сигналов
при другом выборе главных компонент

```
pc
variances
npc
```

Массив структур PCA_Params имеет длину d + 2 (здесь d – максимальный уровень декомпозиции, такой что поле PCA_Params(d).pc является матрицей основных компонентов). Столбцы сохранены в порядке убывания дисперсий. Поле PCA_Params(d).variances – это вектор дисперсий главных компонент, и PCA_Params(d).npc – вектор чисел сохраненных главных компонент.

4.9. Проектирование вейвлетов для непрерывного вейвлет-преобразования (New Wavelet for CWT)

Во многих задачах анализа сигналов требуется найти (идентифицировать) достаточно характерные всплески, например фрагмент кардиосигнала, указывающий на какую-либо паталогию. Для обнаружения таких колебаний при помощи непрерывного вейвлет-преобразования желательно использовать вейвлет, который имеет вид, похожий на этот всплеск. Однако имеются некоторые проблемы. Вейвлет для непрерывного вейвлет-преобразования должен иметь

нулевой интеграл, а характерная волна, которую мы хотим обнаружить, может иметь ненулевой интеграл. Для решения этой проблемы Wavelet Toolbox предлагает функцию **pat2cwav**, которая находит достаточно регулярное приближение с нулевым интегралом к указанной волне методом наименьших квадратов (описание функции pat2cwav см. в разделе 3.1.3).

Раздел **New Wavelet for CWT** позволяет не только спроектировать новый вейвлет, но и протестировать его на точность обнаружения образца волны. Для запуска этого приложения нужно вызвать главное меню командой **wavemenu** и нажать кнопку **New Wavelet for CWT**. Появится окно приложения, на правой части которого имеется возможность выбора метода приближения, степени полинома и граничных условий. Загружаем сигнал PattQRS.mat – фрагмент комплекса QRS кардиосигнала, полученный в примере из параграфа 3.1.3. Далее выбираем параметры: метод аппроксимации 'Oth. to Const', граничные условия 'Continuous' – и производим поиск адаптированного вейвлета, нажав кнопку **Approksimate**. В рабочем окне слева внизу появляется изображение адаптированного вейвлета и исходного образца. Можно попробовать выбрать другие параметры, с тем чтобы приближение вейвлетом было достаточно удовлетворительным.

Приложение **New Wavelet for CWT** дает возможность проверить, насколько эффективно работает построенный вейвлет для обнаружения волн типа образца. С этой целью можно создать новый сигнал, который получается растяжением, сдвигом и сложением двух волн $F(t)$ образца, в виде $F((t - 20)/8) + \mathrm{sqrt}(2)*F((t - 40)/4)$ с наложением шумовой компоненты N и треугольного тренда T. Для создания тестового сигнала можно также наложить друг на друга эти две волны, то есть рассмотреть сдвиги на один и тот же отрезок времени, но с разными коэффициентами сжатия, $F((t - 20)/8) + 2*F((t - 20)/4)$. Выбор этих параметров тестового сигнала можно сделать в правой нижней части окна приложения, сдвиг и растяжение задаются автоматически. Затем нажимаем кнопку **Run**, и программа безошибочно определяет сдвиги по времени $t = 20$, $t = 40$ искомых волн и коэффициенты их сжатия $a = 8$ и $a = 4$ по максимумам коэффициентов непрерывного вейвлет-преобразования. Результаты отражены на рис. 4.9.1.

Программа дает возможность сравнить результаты обнаружения волн адаптированным вейвлетом с результатами их обнаружения обычными вейвлетами. Для этого нужно нажать на кнопку **Compare** и

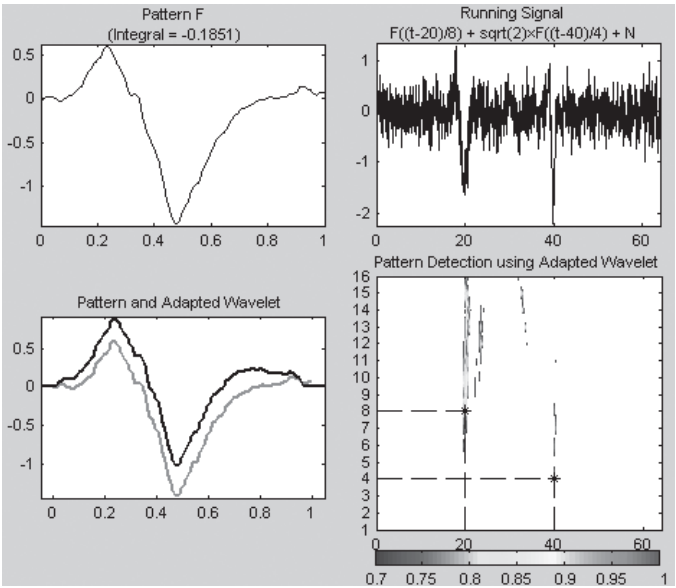


Рис. 4.9.1. Построение и тестирование адаптированного вейвлета

перейти к следующему окну программы. В новом окне можно выбрать вейвлет для сравнения. Выберем вейвлет db2, также напоминающий волну QRS. Для наглядности исключим шумовую компоненту и выберем порог для отображения коэффициентов непрерывного вейвлет-преобразования равным 0.85. Нажимая кнопку **Run**, мы получаем требуемое сравнение, см. рис. 4.9.2. Оба вейвлета хорошо определяют сдвиг по времени волн $t = 20$ и $t = 40$, но коэффициент масштабирования гораздо лучше определен адаптированным вейвлетом.

Отметим, что изображения карт cwt можно определить переключателем справа **Coefficiend Display**.

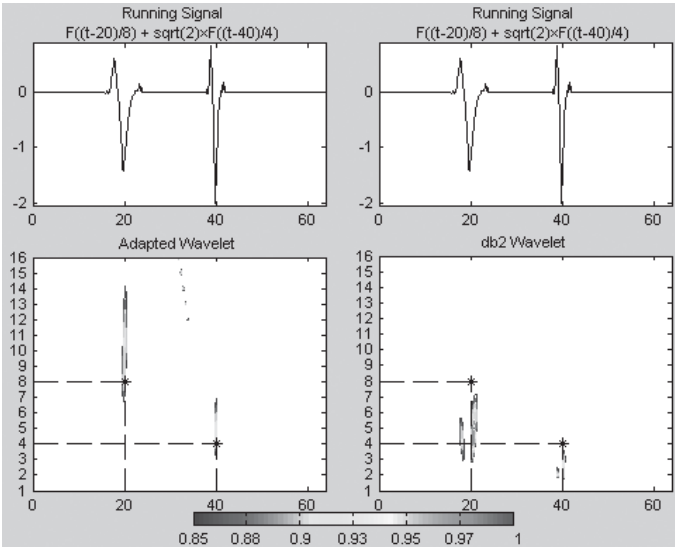


Рис. 4.9.2. Сравнение эффективности обнаружения волн

Приложение
Список функций Wavelet Toolbox



Приведем список функций MATLAB для работы с вейвлетами.

Функция	Краткое описание
addlift	Добавление элементарных шагов схемы лифтинга
allnodes	Нумерация узлов дерева
appcoef	Вычисление коэффициентов аппроксимации одномерного сигнала
appcoef2	Нахождение коэффициентов аппроксимации двумерного сигнала
bestlevt	Дерево пакетного вейвлет-разложения наилучшего уровня
besttree	Наилучшее дерево пакетного вейвлет-разложения
biorfilt	Набор фильтров биортогонального вейвлета
biorwavf	Масштабирующие фильтры биортогонального вейвлета
bswfun	Вычисление и графики биортогональных функций
centfrq	Центральная частота вейвлета
cfs2wpt	Построение пакетного дерева из коэффициентов
cgauwavf	Комплексный гауссов вейвлет
chgwdeccfs	Изменение 1D-коэффициентов декомпозиции мультисигнала
cmorwavf	Комплексный вейвлет Морле
coifwavf	Масштабирующий фильтр вейвлета «койфлет»
cwt	Непрерывное вейвлет-преобразование
cwtext	Непрерывное одномерное вейвлет-преобразование с параметрами расширения массива
cwtft	Непрерывное вейвлет-преобразование, использующее FFT-алгоритм
cwtftinfo	Информация о допустимых вейвлетах для CWT на основе FFT
conofinf	Конус влияния
dbaux	Вейвлет-фильтр Добеши
dbwavf	Масштабирующий фильтр Добеши
ddencmp	Параметры по умолчанию для удаления шума или сжатия (1D, 2D)

Функция	Краткое описание
depo2ind	Переиндексация узлов дерева в «порядковый номер»
detcoef	Вычисляет детализирующие коэффициенты одномерного сигнала
detcoef2	Нахождение детализирующих коэффициентов двумерного сигнала
displs	Отображение схемы лифтинга
drawtree	Графическое представление дерева
dtree	Конструктор класса DTREE
dwt	Одноуровневое дискретное одномерное вейвлет-преобразование
dwt2	Одноуровневое дискретное двумерное вейвлет-преобразование
dwt3	Одноуровневое дискретное 3D-вейвлет-преобразование
dwtmode	Мода расширения дискретного вейвлет-преобразования
dyaddown	Диадическая децимация
dyadup	Диадическое добавление нулей
entrupd	Замена функции энтропии в пакетном анализе
fbspwavf	Комплексный B-сплайновый вейвлет
filt2ls	Преобразование четверки фильтров в схеме лифтинга LS
gauswavf	Гауссов вейвлет
get	Содержание поля объекта WPTREE
icwtft	Обратное непрерывное вейвлет-преобразование
icwtlin	Обратное непрерывное вейвлет-преобразование для линейно распределенных масштабов
idwt	Обратное дискретное одномерное вейвлет-преобразование
idwt2	Обратное дискретное двумерное вейвлет-преобразование
ilwt	Обратное одномерное вейвлет-преобразование лифтинга
ilwt2	Обратное двумерное вейвлет-преобразование лифтинга
idwt3	Одноуровневое обратное дискретное 3D-вейвлет-преобразование
indwt2	Обратное недецимированное 2D-вейвлет-преобразование
ind2depo	Переиндексация «порядковых номеров» узлов дерева в 2-индексы
instdfft	Обратное нестандартное 1D-быстрое преобразование Фурье
intwave	Первообразная вейвлет-функции psi
isnode	Индикатор существования узла
istnode	Определение индексов терминальных узлов
iswt	Обратное дискретное вейвлет-преобразование стационарного сигнала
iswt2	Двумерное обратное вейвлет-преобразование стационарного сигнала

Функция	Краткое описание
laurmat	Конструктор для класса LM матриц Лорана
laurpoly	Конструктор для класса LP полиномов Лорана
leaves	Определение индексов терминальных узлов
liftwave	Нахождение схемы лифтинга LS, связанной с вейвлетом
liftfilt	Применение элементарных шагов лифтинга на четверке фильтров
ls2filt	Преобразование схемы лифтинга LS в четверку фильтров
localmax	Определение цепочек локальных максимумов коэффициентов
lsinfo	Информация о схеме лифтинга
lwt	Одномерное вейвлет-преобразование лифтинга
lwt2	Двумерное вейвлет-преобразование лифтинга
lwtcoef	Извлечение или восстановление вейвлет-коэффициентов LWT
lwtcoef2	Двумерный вариант функции lwtcoef
mexihat	Вейвлет «мексиканская шляпа»
meyer	Вейвлет Мейера
meyeraux	Вспомогательная функция вейвлета Мейера
morlet	Вейвлет Морле
mdwtcluster	Кластеризация 1D-мультисигнала
mdwtdec	Вейвлет-разложение мультисигнала
mdwtrec	Вейвлет-восстановление мультисигнала и извлечение коэффициентов
mswcmp	1D-вейвлет-сжатие мультисигнала
mswcmpscr	1D-вейвлет-сжатие мультисигнала
mswcmptp	Вычисление порогов и энергии 1D-сжатия мультисигнала
mswden	1D-вейвлет-удаление шума мультисигнала
mswthresh	Выполнение 1D-пороговой обработки мультисигнала
ndwt2	Недецимированное 2D-вейвлет-преобразование
nodeasc	Влияние (предки) узла
nodedesc	Потомки узла
nodejoin	Перестройка узла
nodepar	Родитель узла
nodesplt	Расщепление (разложение) узла
noleaves	Определение нетерминальных узлов
nstdfft	Нестандартное 1D-быстрое преобразование Фурье
ntnode	Число терминальных узлов
ntree	Конструктор класса NTREE
orthfilt	Набор фильтров ортогонального вейвлета
pat2cwav	Создание вейвлета для cwt по образцу
plot	Дерево пакетного вейвлет-разложения

Функция	Краткое описание
qmf	Квадратурный зеркальный фильтр
rbiowavf	Масштабирующие фильтры биортогонального вейвлета
read	Считывание данных в терминальных узлах дерева
readtree	Чтение пакетного дерева разложения из фигуры
scal2frq	Пересчет масштабирующей переменной в частоту
set	Содержание поля объекта WPTREE
shanwavf	Вейвлет Шеннона
swt	Дискретное вейвлет-преобразование стационарного сигнала
swt2	Двумерное дискретное вейвлет-преобразование стационарного сигнала
symaux	Масштабирующий фильтр вейвлета «симлет»
symwavf	Масштабирующий фильтр вейвлета «симлет»
thselect	Выбор порога для удаления шума 1D
tnodes	Определение терминальных узлов
treedpth	Глубина дерева
treeord	Порядок дерева
upcoef	Прямое восстановление из коэффициентов
upcoef2	Прямое восстановление из 2D-вейвлет-коэффициентов
upwlev	Одноуровневая реконструкция одномерного вейвлет-разложения
upwlev2	Одноуровневая реконструкция 2D-вейвлет-разложения
wavedec	Многоуровневое одномерное вейвлет-разложение
wavedec2	Многоуровневое двумерное вейвлет-разложение
wavedec3	Многоуровневое 3D-вейвлет-разложение
wavedemo	Вызов GUI для просмотра демонстрационных примеров
wavefun	Масштабирующая и вейвлет-функции
wavefun2	2D-масштабирующая и вейвлет-функции
waveinfo	Информация о вейвлете
wavemenu	Вызов главного вейвлет-меню GUI
wavemngr	Вейвлет-менеджер
wavenames	Имена вейвлетов, доступные для LWT
waverec	Многоуровневое одномерное вейвлет-восстановление
waverec2	Многоуровневое двумерное вейвлет-восстановление
waverec3	Многоуровневое 3D-вейвлет-восстановление
wavsupport	Носитель вейвлета
wave2lp	Полиномы Лорана, ассоциированные с вейвлетом
wbmpen	Штрафной порог для 1D- или 2D-удаления шума
wcodemat	Расширенное псевдоцветное масштабирование матрицы
wcoher	Вейвлет-когерентность

Функция	Краткое описание
wcompress	Подлинное сжатие с использованием вейвлетов
wdcbm	Трешолдинг 1D-сигналов по стратегии Birge-Massart
wdcbm2	Трешолдинг 2D-сигналов по стратегии Birge-Massart
wdcenergy	Распределение энергии 1D-декомпозиции мультисигнала
wden	Автоматическое удаление шума 1D
wdencmp	Удаление шума и сжатие при помощи вейвлетов (1D, 2D)
wenergy	Распределение энергии сигнала по коэффициентам разложения
wenergy2	2D-вариант функции wenergy
wentropy	Вычисляет энтропию сигнала
wextend	Расширение вектора или матрицы
wfbm	Генерирование дробного броуновского движения
wfbmesti	Оценка параметров дробного броуновского движения
wfilters	Фильтры вейвлета
wfusimg	Слияние (наложение) двух изображений
wfusmat	Слияние (наложение) двух матриц или массивов
wkeep	Выбор части вектора или матрицы
wmaxlev	Максимальный уровень разложения
wmpalg	Процесс подгонки (matching pursuit)
wmpdictionary	Словарь для процесса подгонки
wmspca	Многомасшабный анализ главных компонент
wmulden	Многовариантное удаление шума
wnoise	Тестовые 1D-сигналы, в том числе с шумом
wnoisest	Оценка шума 1D-вейвлет-коэффициентов
wp2wtree	Извлечение дерева разложения из пакетного дерева
wpbmpen	Штрафной порог удаления шума вейвлет-пакета
wpcoef	Коэффициенты пакетного вейвлет-преобразования
wpcutree	Обрезание пакетного дерева
wpdec	Пакетное вейвлет-разложение одномерного сигнала
wpdec2	Пакетное вейвлет-разложение двумерного сигнала
wpdencmp	Удаление шума и сжатие при помощи вейвлет-пакетов
wpfun	Вейвлет-пакетные функции
wpjoin	Отрезание ветвей дерева
wprcoef	Восстановление вейвлет-пакетных коэффициентов
wprec	Пакетное вейвлет-восстановление 1D
wprec2	Пакетное вейвлет-восстановление 2D
wpsplt	Расщепление вейвлет-пакета
wpspectrum	Вейвлет-пакетный спектр
wpthcoef	Пороговая обработка пакетных вейвлет-коэффициентов (1D, 2D)

Функция	Краткое описание
wptree	Конструктор класса WPTREE
wpviewcf	Цветной график вейвлет-пакетных коэффициентов
wrcoef	Восстановление сигнала по одной ветви вейвлет-коэффициентов
wrcoef2	2D-восстановление по одной ветви вейвлет-коэффициентов
wrev	Обращение (порядка) вектора
write	Запись значений в терминальные узлы дерева
wscalogram	Скалограмма непрерывного вейвлет-преобразования
wtbo	Конструктор класса WTBO
wthcoef	Пороговая обработка коэффициентов
wthcoef2	Двумерная пороговая обработка
wthresh	Мягкая или жесткая пороговая обработка (1D, 2D)
wthrmngr	Менеджер установки порога
wtreemgr	Менеджер NTREE
wvarchg	Поиск варианта изменения точек (сигнала)

Список литературы

[АСЧ] *Архипов Г. И., Садовничий В. А., Чубариков В. Н.* Лекции по математическому анализу. – М.: Высшая школа, 2000.

[Ас] *Астафьева Н. М.* Вейвлет-анализ: основы теории и примеры применения // УФН. – 1996. – Т. 166. – № 11. – С. 1145–1170. URL: http://www.ufn.ru/ufn96/ufn96_11/ Russian/r9611a.pdf.

[БР] *Барут А., Рончка Р.* Теория представлений групп и ее приложения. – Т. 1. – М.: Мир, 1980.

[Бл] *Блаттер К.* Вейвлет-анализ. Основы теории. – М.: Техносфера, 2004. – 273 с.

[Вл] *Владимиров В. С.* Обобщенные функции в математической физике. – М.: Наука, 1976.

[В] *Воробьев А. С.* Электрокардиография. Новейший справочник. – М.: Эксмо; СПб.: Сова, 2003. – 560 с.

[ГМШ] *Гельфанд И. М., Минлос Р. А., Шапиро З. Я.* Представления группы вращений и группы Лоренца. – М.: Физматгиз, 1958.

[ДК1] *Дейнеко Ж. В., Кириченко Л. О.* Оценивание самоподобия стохастического временного ряда методом вейвлет-анализа // Радіоелектронні і комп'ютерні системи. – Х.: НАКУ «ХАІ», 2009. – № 4. – С. 99–105.

[ДК2] *Дейнеко Ж. В., Кириченко Л. О.* Оценивание параметра Херста для временных рядов с трендом методом вейвлет-преобразования // Системи управління, навігації та зв'язку. – 2010. – Вип. 4 (16). – С. 85–89.

[Дб] *Добеши И.* Десять лекций по вейвлетам. – М.; Ижевск: РХД, 2001.

[ДЖ] *Данилов Д.Л., Жиглявский А.А.* (ред.) Главные компоненты временных рядов: метод «Гусеница». – СПб.: Изд-во СПбГУ, 1997.

[ДИН] *Дремин И. М., Иванов О. В., Нечитайло В. А.* Вейвлеты и их использование // УФН. – 2001. – Т. 171. – № 5. – С. 465– 501. URL: http://www.ufn.ru/ufn01/ufn01_/Russian/ r015a.pdf.

[ДНФ] *Дубровин Б. А., Новиков С. П., Фоменко А. Т.* Современная геометрия. Методы и приложения. – М.: Наука, 1986.

[ЖШ] *Желобенко А. П., Штерн А. И.* Представления групп Ли. – М.: Наука, 1983.

[За] *Захаров В. Г.* Вейвлет-анализ: теория и приложения. Часть 1: Непрерывное вейвлет-преобразование. – Пермь: ПГУб, 2003. – 100 с.

[ИЧЩ] *Истомина Т. В., Чувыкин Б. В., Щеголев В. Е.* Применение теории Wavelets в задачах обработки информации. – Пенза: Изд-во Пензенского госуниверситета, 2000.

[К] *Казаков Д. В.* Разработка методики для идентификации значимых признаков кардиосигнала на основе оценки статистических и стохастических характеристик вейвлет-преобразования // Глобальный научный потенциал. Биотехнологии и медицина. – СПб. – 2012. – № 4 (13). – С. 23–27.

[ККМ] *Каттани К., Кудрейко А. А., Мигранов Н. Г.* Гармоническое вейвлет-решение уравнения типа Кортевега-деФриза // Вестник Челябинского государственного университета. – 2009. – № 8 (146): Физика. – Вып. 4. – С. 28–31.

[ККМОП] *Кириллов Д. С., Короб О. В., Митин Н. А., Орлов Ю. Н., Плешаков Р. В.* Распределения показателя Херста нестационарного маркированного временного ряда // Препринты ИПМ им. М. В. Келдыша. – 2013. – № 11. – 16 с. URL: http://library.keldysh.ru/preprint.asp?id=2013-11O.

[КОГЭ] *Клионский Д. М., Орешко Н. И., Геппенер В. В., Экало А. В.* Адаптивные методы цифровой обработки сигналов. – СПб.: Изд-во СПбГЭТУ «ЛЭТИ», 2010.

[КФ] *Колмогоров А. Н., Фомин С. В.* Элементы теории функций и функционального анализа. – М.: Наука, 1976.

[Кт] *Компьютера*, 1998. – № 8 (236) (сборник статей по вейвлетной тематике). URL: http://www.computerra.ru/offline/ 1998/236/.

[Ко] *Кострикин А. И.* Введение в алгебру. – М.: Наука, 1977.

[Л] *Лукашенко Т. П.* Всплески на топологических группах // Изв. РАН. – Сер. Мат. – Т. 58. – 1994. – № 3. – С. 88–102.

[М] *Малла С.* Вэйвлеты в обработке сигналов. – М.: Мир, 2005.

[Ми] *Мищенко Е. В.* О нахождении границ Рисса сплайн-базиса с помощью тригонометрических полиномов // Вестник КемГУ. – 2011. – № 3/1 (47). – С. 269–274.

[На] *Нарасимхан Р.* Анализ на действительных и комплексных многообразиях. – М.: Мир, 1971.

[Но] *Новиков Л. В.* Основы вейвлет-анализа сигналов: учеб. пособие. – СПб.: Изд-во ООО «МОДУС+», 1999. – 152 с.

[НС] *Новиков И. Я., Стечкин С. Б.* Основы теории всплесков // Успехи математических наук. – 1998. – Т. 53. – № 6 (324). – С. 53–128.

[НПС] *Новиков И. Я., Протасов В. Ю., Скопина М. А.* Теория всплесков. – М.: Физматлит, 2005. – 616 с.

[Пе] *Петухов А. П.* Введение в теорию базисов всплесков: учеб. пособие. – СПб.: Изд-во СПбГТУ, 1999. – 132 с.

[По1] *Подкур П. Н.* О построении некоторых типов вейвлетов с коэффициентом масштабирования N // Электронный научный журнал «Исследовано в России». – 2007. – № 14. – С. 128–138. URL: http://zhurnal.ape.relarn.ru/articles/ 2007/014.pdf.

[По2] *Подкур П. Н.* О построении вейвлетов с коэффициентом масштабирования N на основе B-сплайнов // Электронный научный журнал «Исследовано в России». – 2007. – № 93. – С. 965–974. URL: http://zhurnal.ape.relarn.ru/articles/2007/093.pdf.

[По3] *Подкур П. Н.* Масштабирующие функции и вейвлеты с коэффициентом масштабирования $N > 2$: дис. ... канд. физ.-мат. наук. – Кемерово, 2007. – 233 стр.

[По4] *Подкур П.Н.* Масштабирующие функции и вейвлеты с коэффициентом масштабирования $N > 2$. – Германия: LAP LAMBERT Academic Publishing, 2010. – 172 с.

[По5] *Подкур П. Н.* О высокочастотных компонентах кардиосигнала // Седьмая Всероссийская научно-практическая конференция «Новые достижения в развитии электрокардиографии», ГУ НИИ кардиологии Томского научного центра СО РАМН Тюменский кардиологический центр. – Тюмень, 24–26 мая 2005. – 3 с.

[ППС] *Подкур М. Л., Подкур П. Н., Смоленцев Н. К.* Программирование в среде Borland C++ Builder с математическими библиотеками MATLAB C/C++. – М.: ДМК Пресс, 2006. – 496 с.: ил., CD.

[ПК] *Попечителев Е. П., Кореневский Н. А.* Электрофизиологическая и фотометрическая медицинская техника. – М.: Высшая школа, 2002. – 303 с.

[Ра] *Рао С. Р.* Линейные статистические методы и их применения. – М.: Наука, 1968.

[СДС] *Столниц Э., ДеРоуз., Салезин Д.* Вейвлеты в компьютерной графике. Теория и приложения. – М.; Ижевск: РХД, 2002.

[См1] *Смоленцев Н. К.* Основы теории вейвлетов. Вейвлеты в MATLAB. – М.: ДМК Пресс, 2008.

[См2] *Смоленцев Н. К.* MATLAB: программирование на Visual C#, Borland JBuilder, VBA: учеб. курс. – М.: ДМК Пресс; СПб.: Питер, 2009.

[СЦ1] *Смоленцев Н. К., Цыганкова Т. И.* Диагностические возможности вейвлет-анализа высокочастотных компонент кардиосигнала // Материалы III Всерос. научно-практ. конференции «Информационные технологии и математическое моделирование». – Анжеро-Судженск, 2004. – С. 99–101.

[СЦ2] *Смоленцев Н. К., Цыганкова Т. И.* О высокочастотных компонентах кардиосигнала // Труды IV Всерос. научно-практ. конференции «Информационные недра Кузбасса». – Кемерово, 3–4 февраля 2005. – С. 231–232.

[Со] *Соболев С. Л.* Введение в теорию кубатурных формул. – М.: Наука, 1974.

[СВ] *Стейн И., Вейс Г.* Введение в гармонический анализ на евклидовых пространствах. – М.: Мир, 1974.

[С] *Сычев В. В.* Электронные корреляции и поляронные эффекты в двухэлектронных системах: дис. ... канд. физ.-мат. наук, 05.13.18. – Пущино, 2004.

[У] *Уэлстид С.* Фракталы и вейвлеты для сжатия изображений в действии. – М.: Триумф, 2003.

[Фар] *Фарков Ю. А.* Ряды Фурье и основы вейвлет-анализа. – М.: РГГРУ, 2007. – 111 с.

[Фе] *Федер Е.* Фракталы. – М.: Мир, 1991.

[Фр] *Фрейзер М.* Введение в вэйвлеты в свете линейной алгебры. – М.: БИНОМ, 2010.

[Чуи] *Чуи К.* Введение в вейвлеты. – М.: Мир, 2001.

[Я] *Яковлев А. Н.* Введение в вейвлет-преобразования: учеб. пособие. – Новосибирск: Изд-во НГТУ, 2003. – 104 с. URL: http://window.edu.ru/resource/328/29328/files/ nstu68.pdf.

[AAG] *Ali S. T., Antoine J.-P., Gazeau J.-P.* Coherent States, Wavelets and their Generalizations. – Springer, New York and Berlin, 1999.

[Alp] *Alpert B. K.* A class of bases in L2 for the sparse representation of integral operators. – SIAM J. Math. Anal. – Vol. 24. – 1993. – P. 246–262.

[An] *Antoine J.-P.* Wavelet analysis of signals and images: A grand tour. Revista Ciencias Matematicas (La Habana). – V. 18. – P. 113–143, 2000.

[AJV] *Antoine J.-P., Jacques L., Vandergheynst P.* Penrose tilings, quasicrystals, and wavelets. – Wavelet Applications in Signal and Image Processing VII, P. 28–39; Proc. SPIE, V. 3813, 1999.

[AMV] *Antoine J.-P., Murenzi R., Vandergheynst P.* Directional wavelet revisited: Cauchy wavelets and symmetry detection in patterns // Appl. Comput. Harmon. Anal. – 1999. – V. 6. – P. 314–345.

[AV1] *Antoine J.-P., Vandergheynst P.* Wavelets on 2-D sphere: A group-theoretical approach // Appl. Comput. Harmon. Anal. – 1999. – V. 7. – P. 1–30 (Preprint, UCL-IPT-97-17, Louvain-la-Neuve, Belgium. – 1997, 30 P.)

[AV2] *Antoine J.-P., Vandergheynst P.* Wavelets on the n-sphere and other manifolds // Journal of Mathematical Physics. – 1998. – V. 39. – P. 3987–4008 (Preprint, UCL-IPT-97-19, Louvain-la-Neuve, Belgium. – 1997, 28 P.)

[Ba] *Battle G.* A block spin construction of ondelettes. Part I: Lemarier functions // Comm. Math. Phys. – 1987. – Vol. 110. – P. 601–615.

[Bog] *Bogdanova I.* Wavelets on non-euclidean manifolds (dissertation) // Thèse EPFL, no 3409 (2005), 145 P. URL: http://library.epfl.ch/theses/?nr=3409, http://biblion.epfl.ch/EPFL/theses/2005/3409/EPFL_TH3409.pdf.

[BJ99] *Bratteli O., Jorgensen P. E. T.* Iterated function systems and permutation representations of the Cuntz algebra // Mem. Amer. Math. Soc. – Vol. 139. – No. 663. – 1999 (arXiv:funct-an/9612002v1).

[BJ] *Bratteli O., Jorgensen P. E. T.* Wavelet filters and infinite-dimensional unitary groups // arXiv.org: math.FA/0001171. – 2000. – 30 P.

[BVAJM] *Bogdanova I., Vandergeynst P., Jacques L., Morvidone M., Antoine J.-P.* Stereographic Frames of Wavelets on the Sphere // Applied and Computational Harmonic Analysis. – Vol. 19. – Nr. 2. P. 223–252. – 2005. ITS Technical Report No. 2004.001. URL: http://citeseer.ist.psu.edu/bogdanova04stereographic.html.

[Cat10] *Cattani C.* Shannon Wavelets for the Solution of Integrodifferential Equations. Mathematical Problems in Engineering. – Vol. 2010. – Article ID 408418. – 22 pages, doi:10.1155/2010/408418.

[Cat08] *Cattani C.* Shannon Wavelets Theory. Mathematical Problems in Engineering. Vol. 2008. – Article ID 164808. – 24 pages, doi:10.1155/2008/164808.

[Cat-Kud08] *Cattani C., Kudreyko A.* On the Discrete Harmonic Wavelet Transform. Mathematical Problems in Engineering. – Vol. 2008. – Article ID 687318. – 7 pages, doi:10.1155/2008/687318

[Cat03] *Cattani C.* Harmonic wavelet solutions of the Schrödinger equation. International Journal of Fluid. Mechanics Research. – 2003. – Vol. 30. – No. 5. – P. 1–10.

[Co] *Cohen A.* Ondelettes, analyses multiresolutions et filtres miroir en quadrature // Ann. Inst. H.Poincare, Anal. Non. Lineare, 7. – P. 439–159.

[DGHM] *Donovan G., Geronimo J. S., Hardin D. P. and Massopust P. R.* Construction of orthogonal wavelets using fractal interpolation functions. SIAM J. Math. Anal. – Vol. 27. –1996. – P. 1158–1192.

[DJ] *Dutkay D. E., Jorgensen P. E. T.* Wavelets on fractals // Rev. Mat. Iberoamericana. – 2006. – Vol. 22. – P. 131–180, arXiv:math.CA/0305443 v4, 2004, 47 P.

[Da] *Daubechies I.* The wavelet transform, time-frequency localization and signal analysis // IEEE Trans. Inform. Theory. – 1990. – Vol. 36 – P. 961–1005.

[FO] *Fabec R., Ólafsson G.* The Continuous Wavelet Transform and Symmetric Spaces // Acta Applicandae Math. –2003. – Vol. 77. – P. 41–69, arXiv.org: math.FA/0111100, 2001, 26 P.

[Fr] *Franklin Ph.* A set of continuous orthogonal functions // Math. Ann. – 1928. – Vol. 100. – P. 522–529.

[GP1] *Grassberger P., Procaccia I.* Characterization of strange attractors. *Phys. Rev. Lett*. 50, 346–349 (1983).

[GP2] *Grassberger P., Procaccia I.* Measuring the strangeness of strange attractors. *Physica* D 9, 189–208 (1983).

[GLLWW1] *Guo K., Labate D., Lim W., Weiss G. and Wilson E. N.* The theory of wavelets with composite dilations. In Harmonic analysis and applications, Appl. Numer. Harmon. Anal. – P. 231–250. Birkhauser Boston, Boston, MA, 2006.

[GLLWW2] *Guo K., Labate D., Lim W., Weiss G. and Wilson E. N.* Wavelets with composite dilations and their MRA properties. Appl. Comput. Harmon. Anal. – Vol. 20 (2). – P. 202–236. – 2006.

[HKPT] *Härdle W., Kerkyacharian G., Picard D., Tsybakov A.* Wavelets, Approximation and Statistical Applications. Seminaire Paris-Berlin. 1997. URL: http://www.quantlet.de/ scripts/wav/html. Рус. перевод: *Хардле В., Крекьячарян Ж., Пикар Д., Цыбаков А.* Вейвлеты, аппроксимация и статистические приложения. URL: http://www.matlab.ru.

[Her] *Herve L.* Multi-resolution analysis of multiplicity d. Application to dyadic interpolation. Comput. Harmonic Anal. – Vol. 1. – 1994. – P. 299–315.

[IM] *Ishikawa Y., Mochimaru F.* Wavelet Theory-Based Analysis of High-Frequency, High-Resolution Electrocardiograms: A New Concept for Clinical Uses // Progress in Biomedical Research. – 2002. – Vol. 7. – No. 3. – P. 179–184.

[Is] *Ishikawa Y.* Wavelet Analysis for Clinicial Medicine. Chapter 6: SAECG (Signal Averaged ECG) which was seen from Wavelet Analysis – Supplement – original color images. URL: http://www.uinet.or.jp/~ishiyasu/ch6/index.html.

[J03] *Jorgensen P. E. T.* Matrix Factorizations, Algorithms, Wavelets // Notices Amer. Math. Soc. – Vol. 50. – No. 8. – 2003. – P. 880– 894. URL: http://www.math.uiowa.edu/~jorgen/fea-jorgensen.pdf.

[Kei] *Keinert F.* Wavelets ang Multiwavelets. CRC Press Company. – Charman&Hall/CRC, 2004.

[La] *Lawton W.* Tight frames of compactly supported wavelets // J. Math. Phys. – V. 31. – P. 1898–1901.

[Le] *Lemarie P. G.* Une nouvelle base d'ondelettes de L^2(R^n) // J. de Math. Pures et App. – Vol. 67. – 1988. – P. 227–263.

[L] *Lessig C.* Orthogonal and Symmetric Haar Wavelets on the Sphere. URL: http://www.dgp. toronto.edu/~lessig/soho/data/msc_lessig_small.pdf. 2007.

[LWW13] *Labate D., Weiss G. and Wilson E.* Wavelets // Notices Amer. Math. Soc. – Vol. 60. – No. 1. – 2013. – P. 66–76. URL: http://dx.doi.org/10.1090/noti927.

[Ma] *Mallat S.* Multiresolution approximation and wavelet orthonormal basis of L^2(R) // Trans. AMS. – Vol. 315. – 1989. – P. 69–87.

[McE] *McEwen J.* Wavelets on the sphere. New methodologies and cosmological applications. http://www.mrao.cam.ac.uk/~jdm57/research/talks/cosmolunch_jun07.pdf, 2007.

[MF] *Mochimaru F., Fujimoto Y.* Detecting the Fetal Electrocardiogram byWavelet Theory-Based Methods // Progress in Biomedical Research. – 2002. – Vol. 7. – No. 3. – P. 185–193.

[Newl] *Newland D. E.* Harmonic Wavelet Analysis. Proc. R. Soc. Lond. A. – 1993. – Vol. 443. – P. 203–225. URL: http://rspa.royalsocietypublishing.org/content/443/1917/203#related-urls.

[Newl-2] *Newland D. E.* Harmonic and musical wavelets. Proc. R. Soc. Lond. A. – 1994. – Vol. 444. – P. 605–620.

[OS] *Ólafsson G., Speegle D.* Groups, Wavelets, and Wavelet Sets // Электронное издание: arXiv.org:math.FA/0307317, 2003. – 21 p.

[PS1] *Podkur P. N., Smolentsev N. K.* Construction of some types wavelets with coefficient of scaling N // arXiv.org: math.FA/0612573. – 2006. – 19 p.

[PS2] *Podkur P. N., Smolentsev N. K.* About construction of orthogonal wavelets with compact support and with scaling coefficient *N* // arXiv.org, arXiv:0705.4150v1 [math.FA]. – 2007. – 15 p.

[RA] *Roşca D., Antoine J.-P.* Locally Supported Orthogonal Wavelet Bases on the Sphere via Stereographic Projection. Math. Probl. in Engineering. – Vol. 2009 (2009). – 13 p. doi: 10.1155/2009/124904.

[SLBD] *Simons F. J., Loris I., Brevdo E., Daubechies I. C.* Wavelets and wavelet-like transforms on the sphere and their application to geophysical data inversion. Proc. of SPIE. – Vol. 8138. – 2011. – P. 81380X-1-15. doi: 10.1117/12.892285. URL: http://www.princeton.edu/ geosciences/ people/simons/pdf/SPIE-2011.pdf.

[SMAN] *Starck J.-L., Moudden Y., Abrial P., and Nguyen M.* Wavelets, ridgelets and curvelets on the sphere. A&A. – Vol. 446. – P. 1191–1204 (2006). DOI: 10.1051/0004-6361:20053246.

[StV] *Strela V.* Multiwavelets: Theory and Applications. Massachusetts institute of technology. Submitted to the Department of Mathematics in partial fulfillment of the requirements for the degree of Doctor of Philosophy in Mathematics. – 99 p.

[RT] *Ruelle D., Takens F.* On the nature of turbulence // Comm. Math. Phys. 20, 167 (1971).

[Ta] *Takahashi R.* Sur les representation unitaires des groupes de Lorentz generalizes // Bull. Soc. Math. France, 91. – 1963. – P. 289–433.

[Tak] *Takens F.* Detecting strange attractors in turbulence. In: Dynamical Systems and Turbulence. Lecture Notes in Mathematics, edited by D. A. Rand L. S. Young. – Heidelberg: Springer-Verlag, 366–381 (1981).

[Va] *Vaidyanathan, P. P.* Multirate Digital Filters, Filter Banks, Polyphase Networks, and Applications: A Tutorial // Proceedings of the IEEE. – Vol. 78. – No. 1. – 1990. – P. 65–93.

Предметный указатель

R

RS-анализ, 483

A

Анализ главных
компонент, 446, 449, 595
  многомасштабный, 454, 602
Аттрактор, 484

Б

Базис
  безусловный, 103
  дуальный, 166
  Рисса, 103
  Хаара, 76
Белый шум, 546
Броуновское движение, 377
  дробное, 378

B

Вейвлет
  Гаусса, 328
  для непрерывного
  преобразования, 186
  Мейера, 132, 327
  мексиканская шляпа, 328
  Морле, 329
  ортогональный, 97
  с матрицей
  масштабирования, 280
  с нулевыми моментами, 145
  Хаара, 74
  Шеннона, 331
  Шеннона–Котельникова, 127
Вейвлет-анализ, 113
Вейвлет-восстановление,
118, 313
  N-кратномасштабное
  ортогональное, 208
Вейвлет-пакетное
разложение, 122
Вейвлет-пакеты, 121, 512
Вейвлет-преобразование, 113
  n-мерное непрерывное, 194, 195
  быстрое, 115
  двумерное непрерывное, 197
  дискретное, 362
  непрерывное, 186, 345
Вейвлет-разложение
  N-кратномасштабное, 207
  пакетное, 417
Вейвлеты, 96
  Батла–Лемарье, 138, 143
  биортогональные, 166
  двумерные, 198
  двумерные направленные, 199
  комплексные, 329, 348
  ортогональные
  с параметром N, 247
  пакетные, 417

Вейвлеты с масштабом N
  Кантора, 222
  сплайновые, 228
  степени числа два, 242
  Хаара, 212
  Шеннона, 217, 218
Выборка, 15
Выборочная дисперсия
несмещенная, 521
Выборочное среднее, 521

Г

Гармонический вейвлет, 296
Гистограмма, 522
Группа аффинная, 190, 192

Д

Децимация, 49
Децимация N-адическая, 209
Дисперсия, 521
  выборочная, 521

И

Изображение
  бинарное, 382
  палитровое, 383
  полноцветное, 383
  полутоновое, 383
Импульсная функция
отклика, 39
Импульсная характеристика, 44
  бесконечная, 45
  конечная, 45

К

Каркас, 104
Койфлет, 164
Конус влияния, 357
Корреляционная
размерность, 486
Корреляция, 19
  последовательностей, 36
Коэффициент усиления, 39
Коэффициенты
  аппроксимации, 111
  аппроксимирующие, 363
  детализирующие, 112, 363
    вертикальные, 387
    горизонтальные, 387
    диагональные, 387
  разложения, 111
Кратномасштабный анализ, 89

Л

Лифтинг, 341

М

Масштабирующая функция
  двойственная, 169
  с коэффициентом N, 201
  сплайновая
  с коэффициентом N, 226
Масштабный
коэффициент, 186
Математическое
ожидание, 521
Матрица фильтров
  восстановления, 211
  разложения, 211
Медиана, 521
Методы слияния, 397, 576
Многовариантное удаление
шума, 449, 597
Множество Кантора, 221
Мода случайной величины, 521
Мода расширения, 361
Модели регрессии, 555
Момент
  начальный, 14
  центральный, 14

Н

Носитель
  компактный, 13
  функции, 13

О

Ортонормированная система, 14
  полная, 14

П

Первообразная вейвлета, 325
Передаточная функция, 39
Периодизация, 299
Периодические вейвлеты, 301
Пиковая частота, 39
Плотность распределения вероятности, 520
Показатель Херста, 482
Полифазная компонента, 229
Полифазная матрица, 229
Полупрямое произведение, 190
Пороговая обработка, 431
Представление унитарное, 192
Преобразование Габора, 29
Преобразование Фурье, 19
  n-мерное, 189
  дискретное, 33
  обратное, 20
  оконное, 28
  оцифрованного сигнала, 32
  последовательности, 35
  по циклической группе, 229, 250
Продолжение сигнала, 515
Пространство вейвлетов, 96

Р

Равенство Парсеваля, 15
Разложение
  N-кратномасштабное ортогональное, 203
  кратномасштабное, 104, 166, 167, 168
    орторгональное, 89, 93
  полифазное, 229
Размерность фазового пространства, 485
Регрессия, 554

С

Свертка
  последовательностей, 35
  функций, 19
Сигнал
  аналоговый, 14
  многомерный, 262
Сигнал в многомерном пространстве, 262
Симлеты, 160
Скалограмма, 345, 351
Слияние
  изображений, 397, 576
  массивов, 399
Случайная величина, 520
Случайный процесс, 544
  стационарный, 544
Создание вейвлетов
  для CWT, 339
  методами лифтинга, 341
Спектральная плотность, 545
Спектральная факторизация, 156
Спектр мощности, 546
Сплайн
  базисный, 139
  порядка n, 138
Среднее квадратичное отклонение, 521
Стандартное отклонение
  выборочное, 478, 521
  несмещенное, 521
Странный аттрактор, 484

Т

Теорема
  Безу, 60
  Котельникова–Шеннона, 30
  Пэли–Винера, 26
Трешолдинг
  жесткий, 431
  мягкий, 431

У

Уравнение масштабирующее, 77
Уравнение регрессии, 555
Условия ортогональности, 93

Ф

Фаза фильтра, 39
Фильтр, 39
  высокочастотный, 47
  квадратурный, 337
  масштабирующий, 77, 332
  низкочастотный, 47
  рекурсивный, 46
  с линейной фазой, 160
  транспонированный, 39
  цифровой, 45
Фильтр восстановления
  высокочастотный, 335
  низкочастотный, 335
Фильтр разложения
  высокочастотный, 335
  низкочастотный, 335
Фильтры вейвлетов, 332
Формула
  Котельникова–Шеннона, 31
  Планшереля, 21
  суммирования Пуассона, 28
Фрактальная размерность, 486
Фрейм, 104
Функция
  автокорреляционная, 24
  Гаусса, 29
  интегрируемая с квадратом, 15
  ковариационная, 544
  корреляционная, 544
  лифтинга, 338, 343
  масштабирующая, 77
  масштабирующая Мейера, 128
  ограниченной шириной
  полосы, 26
  распределения, 520
  Хаара, 67
  целая аналитическая, 25
  частотная, 80, 308, 309, 310
  экспоненциального типа, 26
  экспоненциально
  убывающая, 26

Ц

Цветовая карта, 383

Ч

Частота, 14
  вейвлета центральная, 324
  круговая, 14
  Найквиста, 30
Частотная функция сплайна
с коэффициентом N, 227

Э

Энергетический спектр, 480
Энтропия, 426
  логарифм энергии, 426, 479
  норма l^p, 426, 479
  пороговая, 426, 479
  Шеннона, 426, 479, 504
Энтропия Шеннона, 124

Книги издательства «ДМК Пресс» можно заказать в торгово-издательском холдинге «АЛЬЯНС БУКС» наложенным платежом, выслав открытку или письмо по почтовому адресу: 123242, Москва, а/я 20 или по электронному адресу: **orders@alians-kniga.ru**.

При оформлении заказа следует указать адрес (полностью), по которому должны быть высланы книги; фамилию, имя и отчество получателя. Желательно также указать свой телефон и электронный адрес.

Эти книги вы можете заказать и в интернет-магазине: **www.alians-kniga.ru**.

Оптовые закупки: тел. (499) 725-54-09, 725-50-27; электронный адрес **books@alians-kniga.ru**.

Смоленцев Николай Константинович

Основы теории вейвлетов
Вейвлеты в MATLAB

Главный редактор *Мовчан Д. А.*
dmkpress@gmail.com
Корректор *Синяева Г. И.*
Верстка *Чаннова А. А.*
Дизайн обложки *Мовчан А. Г.*

Подписано в печать 09.09.2013. Формат 60×90 1/16.
Гарнитура «Петербург». Печать офсетная.
Усл. печ. л. 39,25. Тираж 100 экз.

Веб-сайт издательства: www.dmkpress.com