



САНКТ-ПЕТЕРБУРГСКИЙ
ГОСУДАРСТВЕННЫЙ
УНИВЕРСИТЕТ

ПРИОРИТЕТНЫЙ
НАЦИОНАЛЬНЫЙ ПРОЕКТ
"ОБРАЗОВАНИЕ"



Проект «Инновационная образовательная среда в классическом университете»

Пилотный проект № 22 «Разработка и внедрение
инновационной образовательной программы «Прикладные математика и физика»»

Физический факультет

Научно-образовательный центр «Электрофизика»

М.А.Павлейно, В.М.Ромаданов

СПЕКТРАЛЬНЫЕ ПРЕОБРАЗОВАНИЯ В MATLAB

Учебно-методическое пособие

Санкт-Петербург
2007 г.

УДК 621.391.26 (075)

- **Рецензенты: доцент кафедры радиофизики,
к.ф.-м.н. Пинегин А.Н.**
- **Печатается по решению методической комиссии
физического факультета СПбГУ**
- **Рекомендовано Ученым советом физического
факультета СПбГУ**

Павлейно М.А., Ромаданов В.М.

СПЕКТРАЛЬНЫЕ ПРЕОБРАЗОВАНИЯ В MATLAB. – СПб., 2007

В учебно-методическом пособии рассматриваются следующие методы анализа сигналов: разложение в тригонометрический ряд, преобразование Фурье непрерывных сигналов, преобразование Фурье дискретных сигналов, алгоритм быстрого преобразования Фурье, оконное преобразование Фурье, непрерывное вейвлет преобразование, дискретное вейвлет разложение, алгоритм быстрого вейвлет преобразования. Каждая тема сопровождается примерами, написанными в MATLAB. Пособие предназначено для студентов 3 курса направления «Прикладные математика и физика». Оно может быть использовано при изучении дисциплины «Методы цифровой обработки сигналов и изображений», при выполнении курсовых и дипломных работ.

Оглавление

1. Введение.....	4
2. Разложение в тригонометрический ряд Фурье	5
2.1. Вещественная форма ряда Фурье.....	5
2.2. Комплексная форма ряда Фурье	7
2.3. Спектр периодической функции	8
3. Интегралы Фурье и Лапласа	24
3.1. Преобразование Фурье.....	24
3.2. Свойства преобразования Фурье.....	26
3.3. Косинусное преобразование	36
3.4. Преобразование Лапласа.....	37
4. Дискретизация сигналов.....	41
4.1. Спектр дискретного сигнала.....	41
4.2. Теорема Котельникова	45
5. Дискретное преобразование Фурье	62
5.1. Вывод матрицы дискретного преобразования Фурье.....	62
5.2. Алгоритм быстрого преобразования Фурье.....	64
5.3. Увеличение разрешения по частоте.....	77
5.4. Растекание спектра	79
6. Оконное преобразование Фурье	88
6.1. Вычисление оконного преобразования Фурье в Matlab	88
6.2. Оконные функции в Matlab	89
7. Wavelet анализ	105
7.1. Непрерывное wavelet преобразование.....	105
7.2. Дискретное вейвлет преобразование	129
7.3. Быстрое вейвлет преобразование.....	144
8. Литература	160

1. Введение

В данном методическом пособии с единых позиций рассмотрены основные типы спектральных преобразований сигналов, представленных как в аналоговой, так и в дискретной форме. Это разложение периодических функций в ряды Фурье, интегральные преобразования Фурье и Лапласа непериодических сигналов, оконное преобразование Фурье. Проанализированы процессы, происходящие при дискретизации непрерывных сигналов. Рассмотрены алгоритмы дискретного и быстрого преобразования Фурье. В последнем разделе приведены сведения о вейвлет преобразовании, которое в последнее время широко применяется при решении различных задач в экспериментальной физике, электротехнике, биологии, медицине.

Особенностью пособия является то, что все теоретические сведения широко иллюстрируются примерами, выполненными в среде Matlab и Simulink. Приведены соответствующие программы и модели. Примеры подобраны таким образом, что они выявляют как основные особенности рассматриваемых преобразований, так и некоторые их «тонкие» детали. Выбор Matlab в качестве среды программирования связан с тем, что этот пакет в настоящее время является несомненным лидером в классе аналогичных программных средств.

2. Разложение в тригонометрический ряд Фурье

2.1. Вещественная форма ряда Фурье

Рассмотрим периодическую функцию $x(t)$ с периодом, равным T : $x(t+nT)=x(t)$, где n – любое целое число. Покажем, что при выполнении определенных условий эта функция может быть представлена в виде суммы, конечной или бесконечной, гармонических функций вида $\cos\left(\frac{2\pi kt}{T} + \phi_k\right)$, период которых совпадает с периодом исходной функции $x(t)$, где k – целое число, ϕ_k – константа. Линейная комбинация таких функций $\sum_{k=0}^N A_k \cos\left(\frac{2\pi kt}{T} + \phi_k\right)$, называемая тригонометрическим полиномом N -го порядка, также будет иметь период, равный T . Таким образом, мы будем решать задачу о разложении периодической функции в тригонометрический ряд:

$$x(t) = \sum_{k=0}^{\infty} A_k \cos\left(\frac{2\pi kt}{T} + \phi_k\right) \quad (1)$$

Отдельное слагаемое этой суммы $A_k \cos\left(\frac{2\pi kt}{T} + \phi_k\right)$ называется k -ой гармоникой функции $x(t)$. Наша задача заключается в том, чтобы подобрать такие коэффициенты A_k и ϕ_k , при которых ряд (1) будет сходиться к заданной функции $x(t)$.

Слагаемые в (1) можно записать в другом виде, раскрыв косинус суммы:

$$x(t) = \frac{a_0}{2} + \sum_{k=1}^{\infty} \left[a_k \cos\left(\frac{2\pi kt}{T}\right) + b_k \sin\left(\frac{2\pi kt}{T}\right) \right] \quad (2)$$

где новые коэффициенты выражаются как $a_k = A_k \cos \phi_k$, $a_0 = 2A_0 \cos \phi_0$ и $b_k = -A_k \sin \phi_k$. Формула (2) называется вещественной формой тригонометрического ряда.

Рассмотрим некоторые свойства тригонометрических функций, входящих в (2). Интеграл от синуса или косинуса, взятый по любому отрезку, длина которого равна периоду T , равен нулю:

$$\int_{\tau}^{\tau+T} \cos\left(\frac{2\pi kt}{T}\right) dt = -\frac{T}{2\pi k} \left[\sin\left(2\pi k + \frac{2\pi k \tau}{T}\right) - \sin\left(\frac{2\pi k \tau}{T}\right) \right] = 0$$

$$\int_{\tau}^{\tau+T} \sin\left(\frac{2\pi kt}{T}\right) dt = \frac{T}{2\pi k} \left[\cos\left(2\pi k + \frac{2\pi k \tau}{T}\right) - \cos\left(\frac{2\pi k \tau}{T}\right) \right] = 0$$
(3)

Докажем ортогональность этих функций. Рассмотрим интегралы по периоду от произведений синусов и косинусов. Для этого воспользуемся следующими формулами:

$$\sin\left(\frac{2\pi kt}{T}\right) \cos\left(\frac{2\pi nt}{T}\right) = \frac{\sin\left(\frac{2\pi(k+n)t}{T}\right) + \sin\left(\frac{2\pi(k-n)t}{T}\right)}{2}$$

$$\sin\left(\frac{2\pi kt}{T}\right) \sin\left(\frac{2\pi nt}{T}\right) = \frac{\cos\left(\frac{2\pi(k-n)t}{T}\right) - \cos\left(\frac{2\pi(k+n)t}{T}\right)}{2}$$

$$\cos\left(\frac{2\pi kt}{T}\right) \cos\left(\frac{2\pi nt}{T}\right) = \frac{\cos\left(\frac{2\pi(k+n)t}{T}\right) + \cos\left(\frac{2\pi(k-n)t}{T}\right)}{2}$$
(4)

Интеграл от такого произведения по периоду будет равен нулю, если $k \neq n$ (свойство ортогональности). Если $k = n$, то интеграл от произведения синуса на косинус будет нулём, а от произведения двух одинаковых функций отличен от нуля.

$$\int_0^T \sin\left(\frac{2\pi kt}{T}\right) \cos\left(\frac{2\pi nt}{T}\right) dt = 0$$

$$\int_0^T \sin\left(\frac{2\pi kt}{T}\right) \sin\left(\frac{2\pi nt}{T}\right) dt = \begin{cases} 0, & n \neq k \\ \frac{T}{2}, & n = k \neq 0 \end{cases}$$

$$\int_0^T \cos\left(\frac{2\pi kt}{T}\right) \cos\left(\frac{2\pi nt}{T}\right) dt = \begin{cases} 0, & n \neq k \\ \frac{T}{2}, & n = k \neq 0 \\ T, & n = k = 0 \end{cases}$$
(5)

Если умножить тригонометрический ряд (2) на $\sin\left(\frac{2\pi mt}{T}\right)$ или $\cos\left(\frac{2\pi mt}{T}\right)$, а потом проинтегрировать по периоду, то из (5) следует, что в сумме должно будет остаться только одно слагаемое, отличное от нуля.

$$\int_0^T x(t) \cos\left(\frac{2\pi nt}{T}\right) dt = \frac{a_n T}{2}$$

$$\int_0^T x(t) \sin\left(\frac{2\pi nt}{T}\right) dt = \frac{b_n T}{2}$$
(6)

Из полученных равенств находим коэффициенты разложения в тригонометрический ряд Фурье.

$$a_k = \frac{2}{T} \int_0^T x(t) \cos\left(\frac{2\pi kt}{T}\right) dt \quad k = 0, 1, 2, \dots$$

$$b_k = \frac{2}{T} \int_0^T x(t) \sin\left(\frac{2\pi kt}{T}\right) dt \quad k = 1, 2, \dots$$
(7)

Можно доказать [4], что тригонометрический ряд будет сходиться равномерно к функции $x(t)$, если сходятся ряды $\sum_{k=1}^{\infty} |a_k|$ и $\sum_{k=1}^{\infty} |b_k|$. Это будет

выполнено, если исходная функция $x(t)$ удовлетворяет условиям Дирихле:

- функция имеет конечное число разрывов первого рода на периоде,
- на периоде можно выделить конечное число отрезков, на которых функция изменяется монотонно.

Заметим, что для любых периодических электрических сигналов условие Дирихле выполняется. В точках разрыва ряд Фурье сходится к полусумме значений функции слева и справа от точки разрыва. В силу равномерной сходимости ряда каждый следующий его член вносит всё меньший вклад в сумму, поэтому функция $x(t)$ может быть приближена с определённой точностью тригонометрическим полиномом порядка N , то есть конечным числом слагаемых.

2.2. Комплексная форма ряда Фурье

Другая, комплексная форма тригонометрического ряда, получается, если записать синусы и косинусы в (2) через комплексные экспоненты.

$$x(t) = \frac{a_0}{2} + \sum_{k=1}^{\infty} \left[a_k \frac{e^{\frac{i2\pi kt}{T}} + e^{-\frac{i2\pi kt}{T}}}{2} + b_k \frac{e^{\frac{i2\pi kt}{T}} - e^{-\frac{i2\pi kt}{T}}}{2i} \right] =$$

$$= \frac{a_0}{2} + \sum_{k=1}^{\infty} \left[\left(\frac{a_k - ib_k}{2} \right) e^{\frac{i2\pi kt}{T}} + \left(\frac{a_k + ib_k}{2} \right) e^{-\frac{i2\pi kt}{T}} \right] =$$

$$= c_0 + \sum_{k=1}^{\infty} \left[c_k e^{\frac{i2\pi kt}{T}} + c_{-k} e^{-\frac{i2\pi kt}{T}} \right] = \sum_{k=-\infty}^{+\infty} c_k e^{\frac{i2\pi kt}{T}}$$
(8)

Коэффициенты вещественной и комплексной формы связаны между собой соотношениями:

$$\begin{aligned} c_k &= \frac{a_k - ib_k}{2} & c_{-k} &= \frac{a_k + ib_k}{2} & c_0 &= \frac{a_0}{2} \\ a_k &= c_k + c_{-k} & b_k &= i(c_k - c_{-k}) \end{aligned} \quad (9)$$

Используя формулы (9), из (7) получим выражения для коэффициентов комплексной формы тригонометрического ряда.

$$\begin{aligned} c_k &= \frac{1}{T} \int_0^T f(t) \left[\cos\left(\frac{2\pi kt}{T}\right) - i \sin\left(\frac{2\pi kt}{T}\right) \right] dt = \frac{1}{T} \int_0^T f(t) e^{-i\frac{2\pi kt}{T}} dt \\ c_{-k} &= \frac{1}{T} \int_0^T f(t) \left[\cos\left(\frac{2\pi kt}{T}\right) + i \sin\left(\frac{2\pi kt}{T}\right) \right] dt = \frac{1}{T} \int_0^T f(t) e^{i\frac{2\pi kt}{T}} dt \\ c_0 &= \frac{1}{T} \int_0^T f(t) dt \end{aligned} \quad (10)$$

Эти коэффициенты могут быть записаны для любого номера k следующим образом

$$c_k = \frac{1}{T} \int_0^T f(t) e^{-i\frac{2\pi kt}{T}} dt \quad (11)$$

Тригонометрический ряд в комплексной форме равномерно сходится к функции $x(t)$, если сходятся ряды $\sum_{k=1}^{\infty} |c_k|$ и $\sum_{k=1}^{\infty} |c_{-k}|$. Это будет выполнено, если исходная функция $x(t)$ удовлетворяет условиям Дирихле.

2.3. Спектр периодической функции

Введем понятие спектра периодической функции. Оно основывается на возможности представления сигнала либо в виде вещественного ряда Фурье (1), либо в виде комплексного ряда (8). Это означает, что вещественные коэффициенты A_k и ϕ_k , или комплексные коэффициенты c_k несут полную информацию о периодической с известным периодом T функции. Набор коэффициентов A_k и ϕ_k называется вещественным спектром сигнала. Коэффициенты A_k несут в себе информацию о том, из каких по амплитуде гармонических сигналов состоит исходный сигнал (частоты отдельных гармоник не зависят от формы сигнала и равны

$\omega_k = \frac{2\pi k}{T}$). Поэтому набор A_k называется амплитудным спектром. Коэффициенты ϕ_k определяют фазы отдельных гармоник, их совокупность называется фазовым спектром.

В результате использования комплексной формы ряда (8) получают комплексный спектр сигнала – набор комплексных коэффициентов c_k . В отличие от вещественного спектра, комплексный спектр определен как для положительных, так и для отрицательных частот. Ниже мы покажем, что модули этих коэффициентов определяют амплитуды гармоник и поэтому могут называться амплитудным спектром, а аргументы (фазовый спектр) определяют начальные фазы гармоник. Из формулы (11) следует, что если функция $x(t)$ вещественна, то $c_{-k} = c_k^*$. Из этого соотношения вытекает свойство четности для амплитудного комплексного спектра и нечетность для фазового.

Посмотрим, как связаны между собой вещественный и комплексный спектры. Запишем ряд (8) в виде

$$x(t) = \sum_{k=-\infty}^{-1} |c_k| e^{i(\omega_k t + \arg c_k)} + c_0 + \sum_{k=1}^{+\infty} |c_k| e^{i(\omega_k t + \arg c_k)} \quad (12)$$

Слагаемые с отрицательными номерами могут быть выражены через слагаемые с положительными номерами, так как $\omega_{-k} = -\omega_k$ и $\arg c_{-k} = -\arg c_k$. Тогда останется только сумма с положительными номерами

$$x(t) = c_0 + \sum_{k=1}^{+\infty} |c_k| \left(e^{i(\omega_k t + \arg c_k)} + e^{-i(\omega_k t + \arg c_k)} \right) \quad (13)$$

После суммирования экспонент с одинаковыми номерами k получим следующее выражение:

$$x(t) = c_0 + 2 \sum_{k=1}^{+\infty} |c_k| \cos(\omega_k t + \arg c_k) \quad (14)$$

Сравнивая ряды (1) и (14), получим искомую связь вещественного и комплексного спектров: $A_k = |c_k|$ и $\phi_k = \arg c_k$.

Так как спектр периодического сигнала состоит из отдельных гармоник, его называют дискретным или линейчатым. Частоты гармоник $\omega_k = \frac{2\pi k}{T}$ обратно пропорциональны периоду T , то есть если, например, увеличить период в 2 раза, то гармоники в спектре станут располагаться в

два раза ближе друг к другу. Для спектров характерно, что чем уже импульс, тем шире его спектр.

Для примера сравним спектры импульсов треугольного и пилообразного сигналов (рис. 1).

$$x_1(t) = \begin{cases} \frac{t}{\tau}, & t \in [0, \tau] \\ \frac{2\tau - t}{\tau}, & t \in [\tau, 2\tau] \\ 0, & t \notin [0, 2\tau] \end{cases} \quad x_2(t) = \begin{cases} \frac{t}{\tau}, & t \in [0, \tau] \\ 0, & t \notin [0, \tau] \end{cases} \quad (15)$$

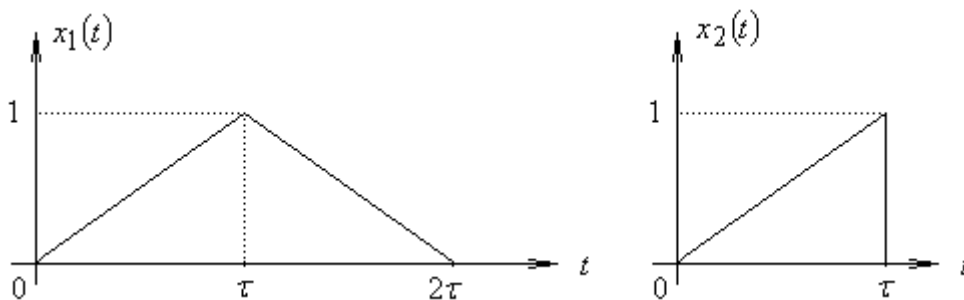


Рис. 1. Пилообразный и треугольный импульс.

Посмотрим, как изменится форма спектра при изменении длительности сигнала, периода повторения и крутизны фронтов. По формуле (11) вычисляем их спектр:

$$c_{1,k} = \frac{1}{T} \int_{-\infty}^{+\infty} x_1(t) e^{-i\omega_k t} dt = \frac{1}{T} \int_0^{\tau} \frac{t}{\tau} e^{-i\omega_k t} dt + \frac{1}{T} \int_{\tau}^{2\tau} \frac{2\tau - t}{\tau} e^{-i\omega_k t} dt = 4 \frac{\sin^2\left(\frac{\omega_k \tau}{2}\right)}{T\tau\omega_k^2} e^{-i\omega_k \tau} \quad (16)$$

$$c_{2,k} = \frac{1}{T} \int_{-\infty}^{+\infty} x_2(t) e^{-i\omega_k t} dt = \frac{1}{T} \int_0^{\tau} t e^{-i\omega_k t} dt = \frac{1 - e^{-i\omega_k \tau} + i\omega_k \tau}{T\tau\omega_k^2} e^{-i\omega_k \tau}$$

Пример спектра периодического сигнала (harmonics.m)

```
clear, clc, close all
```

```
T = 10; % период повторения импульсов
tau = 1; % положение максимума пилообразного импульса
M = 100; % количество рассматриваемых гармоник
k = 1:M; % номера гармоник
w = 2*pi*k/T; % частоты гармоник
% нулевая гармоника (среднее значение сигнала за период)
c1(1) = tau/T;
c1(2:1+M) = 4*sin(w*tau/2).^2/T/tau./w.^2.*exp(-i*w*tau);
f1(1) = 0;
```

```

f1(2:1+M) = w/2/pi; % частоты

% пилообразный импульс в два раза длиннее с тем же периодом
T = 10;
tau = 2; % увеличили длительность
w = 2*pi*k/T;
c2(1) = tau/T;
c2(2:1+M) = 4*sin(w*tau/2).^2/T/tau./w.^2.*exp(-i*w*tau);
f2(1) = 0;
f2(2:1+M) = w/2/pi;

% пилообразный импульс с той же длительностью,
% но большим периодом
T = 20; % увеличили период повторения
tau = 1;
w = 2*pi*k/T;
c3(1) = tau/T;
c3(2:1+M) = 4*sin(w*tau/2).^2/T/tau./w.^2.*exp(-i*w*tau);
f3(1) = 0;
f3(2:1+M) = w/2/pi;

% треугольный импульс с крутым фронтом
T = 10;
tau = 1;
w = 2*pi*k/T;
c4(1) = tau/2/T; % спектр сигнала с крутым фронтом
c4(2:1+M) =
    exp(-i*w*tau).*(1-
    exp(i*w*tau)+i*tau*w)/T/tau./w.^2;
f4(1) = 0;
f4(2:1+M) = w/2/pi;

subplot(4,1,1), stem(f1,abs(c1),'k.')
axis([0 3 0 0.1]), ylabel('|c_1|')
subplot(4,1,2), stem(f2,abs(c2),'k.')
axis([0 3 0 0.2]), ylabel('|c_2|')
subplot(4,1,3), stem(f3,abs(c3),'k.')
axis([0 3 0 0.1]), ylabel('|c_3|')
subplot(4,1,4), stem(f4,abs(c4),'k.')
axis([0 3 0 0.1]), ylabel('|c_4|')
xlabel('f')

```

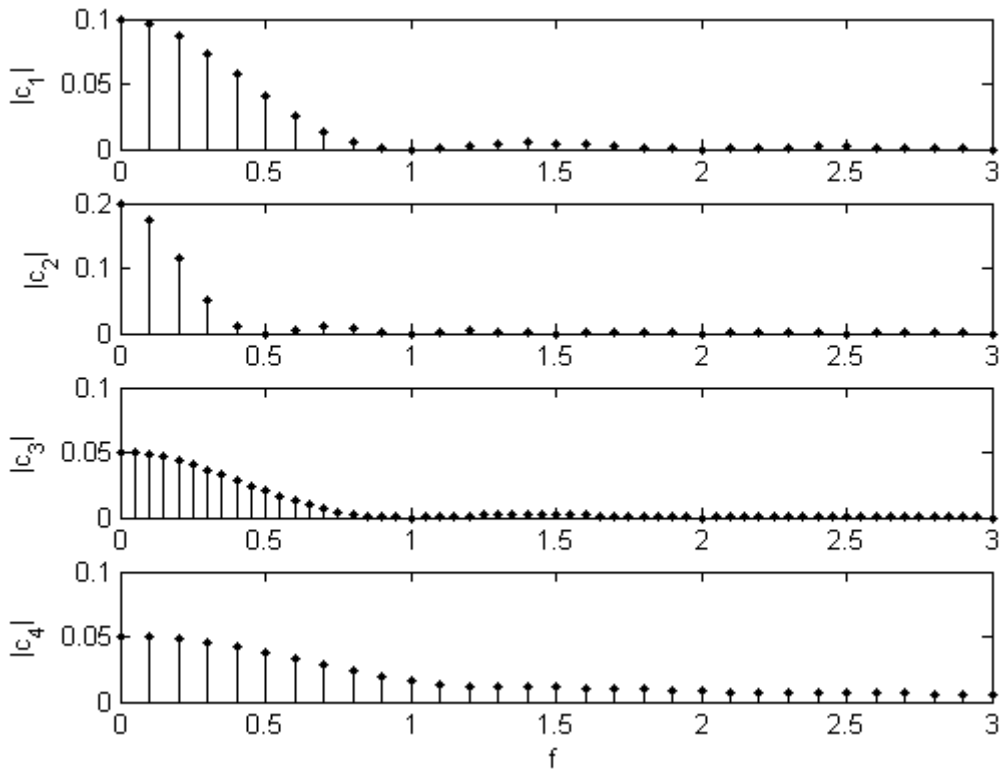


Рис. 2. Амплитуды комплексных гармоник.

Первый график на рис. 2 – это спектр импульса $x_1(t)$ с $T=10$ и $\tau=1$. Второй график – спектр того же импульса, но $\tau=2$. Положения гармоник на втором графике осталось прежние, а ширина спектра уменьшилась. На третьем графике приведен спектр первого импульса, но $T=20$. В результате изменилось расстояние между гармониками, они стали в два раза ближе, а форма огибающей спектра осталась такой же, как и на первом графике. На четвёртом графике показан спектр импульса $x_2(t)$ с крутым фронтом. Спектр такого импульса убывает заметно медленнее, чем спектр импульса, который плавно нарастает и плавно убывает.

Пример

Рассмотрим функцию $x(t)=t^3-t$ на отрезке $[-1,1]$, считаем, что за пределами этого отрезка функция периодически повторяется с периодом $T=2$. Вычислим по формулам (7) коэффициенты разложения.

$$\begin{aligned}
 a_k &= \frac{2}{T} \int_{-1}^1 (t^3 - t) \cos\left(\frac{2\pi kt}{T}\right) dt = 0 \\
 b_k &= \frac{2}{T} \int_{-1}^1 (t^3 - t) \sin\left(\frac{2\pi kt}{T}\right) dt = \frac{12}{\pi^3 k^3} \cos(\pi k) + \left(4 - \frac{12}{\pi^4 k^4}\right) \sin(\pi k)
 \end{aligned}
 \tag{17}$$

Так как функция $x(t)$ в данном примере нечётная, то в разложении присутствуют только синусы. С учетом того, что k – целые числа и поэтому $\cos(\pi k) = (-1)^k$ и $\sin(\pi k) = 0$, выражение для b_k примет вид

$$b_k = (-1)^k \frac{12}{\pi^3 k^3}. \quad (18)$$

Сравним приближения полиномом первого порядка и второго порядка. Первые два коэффициента:

$$b_1 = -\frac{12}{\pi^3} \quad b_2 = \frac{12}{8\pi^3} = -\frac{b_1}{8} \quad (19)$$

Графики этих приближений (рис. 3) строятся с помощью программы `example_1.m`.

Сравнение первого и второго приближения (`example_1.m`)

```
clear, close all, clc

t = -1:0.01:1;
x = t.^3-t; % функция
% коэффициенты разложения
b1 = -12/pi^3;
b2 = -b1/8;
% первое и второе приближение
x1 = b1*sin(2*pi*1*t/2);
x2 = b1*sin(2*pi*1*t/2) + b2*sin(2*pi*2*t/2);

plot(t,x,'k', t,x1,'k--', t,x2,'k:')
legend('x(t)', 'x_1(t)', 'x_2(t)')
```

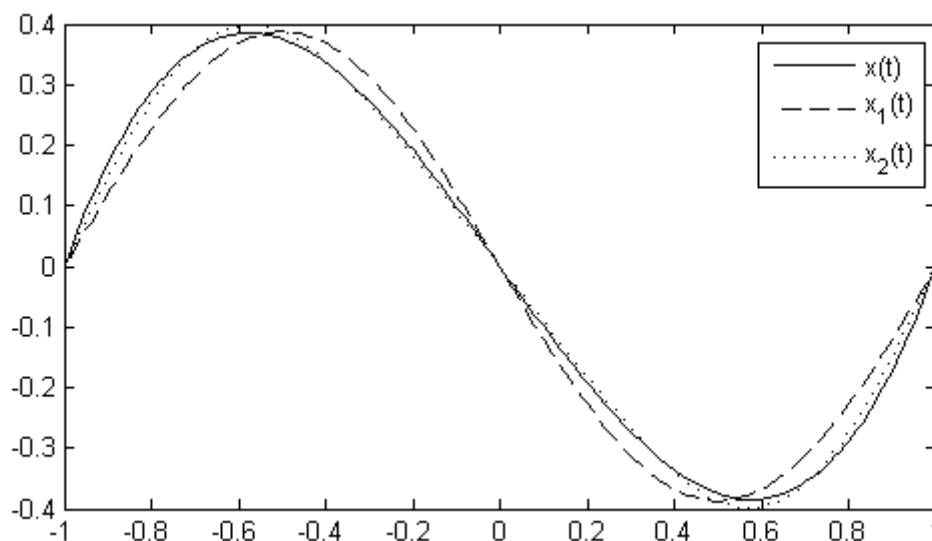


Рис. 3. Сравнение первого и второго приближения.

Видно, что уже с помощью суммы двух гармоник получается результат, достаточно близкий к исходной функции.

Сравним спектры несимметричного и симметричного прямоугольного импульса длительностью τ и периодом повторения T . По формуле (11) вычисляем коэффициенты, полагаем, что амплитуда импульса равна 1. Для несимметричного импульса:

$$c_{1,k} = \frac{1}{T} \int_0^{\tau} e^{-i \frac{2\pi kt}{T}} dt = \left[\frac{1}{\pi k} \sin\left(\frac{\pi k \tau}{T}\right) \right] e^{-i \frac{\pi k \tau}{T}} \quad (20)$$

для симметричного импульса:

$$c_{2,k} = \frac{1}{T} \int_{-\frac{\tau}{2}}^{\frac{\tau}{2}} e^{-i \frac{2\pi kt}{T}} dt = \frac{1}{\pi k} \sin\left(\frac{\pi k \tau}{T}\right) \quad (21)$$

Пример

В рассматриваемом ниже примере, сопровождающимся программой `spectr.m`, выполняются следующие действия:

- сравнивается спектр симметричного и несимметричного прямоугольного импульса,
- строятся графики первых четырёх гармоник спектров импульсов,
- выполняется восстановление импульсов по формуле (14) по нескольким гармоникам.

Все гармоники симметричного импульса – чётные функции, то есть косинусы. Скважность импульса подобрана равной $\frac{1}{3}$, поэтому, как следует из формул (20) и (21), каждая третья гармоника будет с нулевой амплитудой. Аналогично возможны три варианта сдвига по фазе для несимметричного импульса $-\frac{\pi}{3}$, $-\frac{2\pi}{3}$ и $-\pi$.

Пример спектра периодического сигнала (`spectr.m`)

```
clear, clc, close all
```

```
T = 12; % период импульсов
tau = 4; % длительность импульса
N = 15; % количество гармоник для восстановления
k = 0:N; % номера гармоник
w = 2*pi*k/T; % частоты гармоник
t = 0:T/20/N:3*T; % время (три периода)
```

```

% задаём прямоугольные импульсы
u1 = rem(t,T)<tau;           % несимметричный импульс
u2 = rem(t+tau/2,T)<tau;    % симметричный импульс

% коэффициенты разложения несимметричного импульса
c1 = sin(pi*k*tau/T).*exp(-i*pi*k*tau/T)/pi./k;
c1(1) = tau/T; % так как на нулевой частоте было деление на
ноль
% коэффициенты разложения симметричного импульса
c2 = sin(pi*k*tau/T)/pi./k;
c2(1) = tau/T; % так как на нулевой частоте было деление на
ноль
% выделяем амплитудный и фазовый спектр
A1 = abs(c1); % амплитудный спектр
fi1 = angle(c1); % фазовый спектр
A2 = abs(c2);
fi2 = angle(c2);

% восстанавливаем несимметричный импульс
s1(1:length(t)) = c1(1);
for n = 2:N+1
    s1 = s1 + 2*A1(n)*cos(w(n)*t+fi1(n));
end
% восстанавливаем симметричный импульс
s2(1:length(t)) = c2(1);
for n = 2:N+1
    s2 = s2 + 2*A2(n)*cos(w(n)*t+fi2(n));
end

% выводим графики спектров
figure
subplot(2,2,1)
stem(w/2/pi, A1, 'k')
xlabel('f, Гц', 'FontName', 'Arial Cyr')
ylabel('Амплитуда', 'FontName', 'Arial Cyr')
subplot(2,2,3)
stem(w/2/pi, fi1, 'k')
xlabel('f, Гц', 'FontName', 'Arial Cyr')
ylabel('Фаза', 'FontName', 'Arial Cyr')
subplot(2,2,2)
stem(w/2/pi, A2, 'k')
xlabel('f, Гц', 'FontName', 'Arial Cyr')
ylabel('Амплитуда', 'FontName', 'Arial Cyr')
subplot(2,2,4)
stem(w/2/pi, fi2, 'k')
xlabel('f, Гц', 'FontName', 'Arial Cyr')
ylabel('Фаза', 'FontName', 'Arial Cyr')

figure
subplot(2,1,1)
plot(t,u1,'k:', t,s1,'k'), axis([0 max(t) -0.2 1.2])

```

```

xlabel('t, c'), ylabel('u_1(t), s_1(t)'), legend('u_1','s_1')
subplot(2,1,2)
plot(t,u2,'k:', t,s2,'k'), axis([0 max(t) -0.2 1.2])
xlabel('t, c'), ylabel('u_2(t), s_2(t)'), legend('u_2','s_2')

figure
for m=2:5
    subplot(4,1,m-1)
    plot(t,A1(m)*cos(w(m)*t+fi1(m)), 'k', ...
         t,A2(m)*cos(w(m)*t+fi2(m)), 'k:')
    axis([0, T, -0.3, 0.3])
    xlabel('t, c')
    ylabel(['Гармоника ' num2str(m-1)], 'FontName', 'Arial
Cyr')
end

```

В результате выполнения программы на графиках, приведенных на рис. 4 слева строится амплитудный и частотный спектр несимметричного импульса, на графиках справа – для симметричного. Из графиков видно, что амплитудные спектры совпадают, а поведение фазовых спектров отличается. По амплитудным спектрам видно, что амплитуда каждой третьей гармоники равна нулю. На рис. 5 построены графики исходных прямоугольных импульсов и восстановленных по 15-и гармоникам. Исходные импульсы показаны пунктиром.

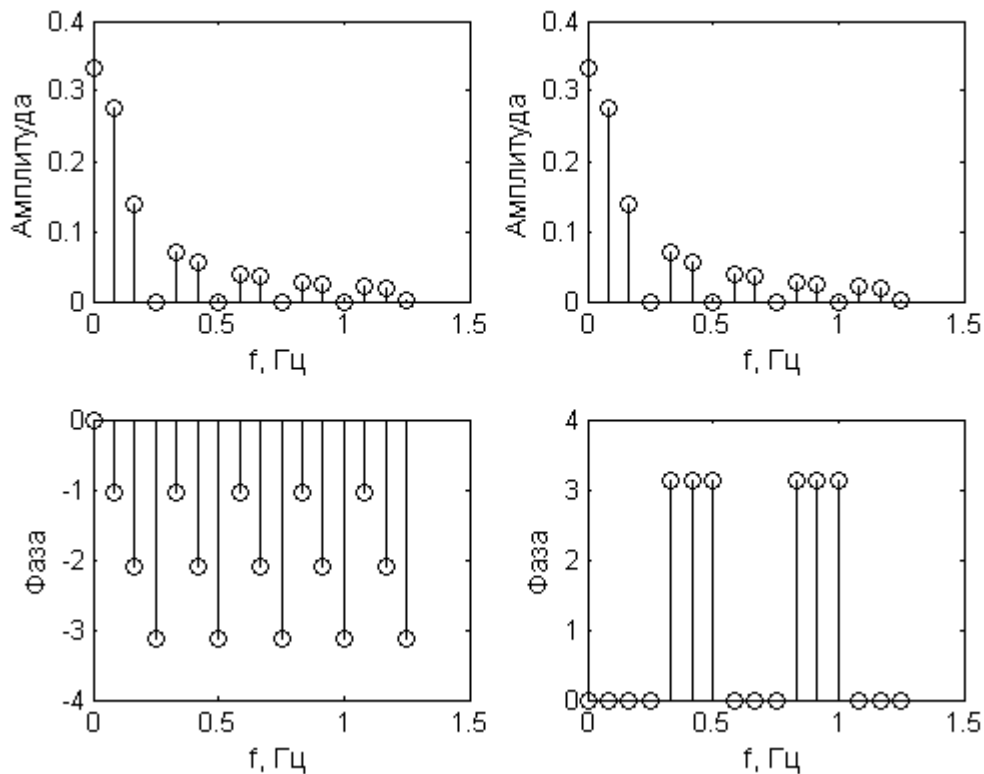


Рис. 4. Спектры несимметричного импульса (слева) и симметричного импульса (справа).

На рис. 6 построены графики первых четырёх гармоник (сплошной линией – для несимметричного импульса, пунктирной – для симметричного). Видно, что третьей гармоники нет, её амплитуда равна нулю. Также видно, что гармоники в спектре симметричного сигнала – косинусы, то есть симметричные функции, в то время как гармоники несимметричного импульса имеют разные фазовые сдвиги.

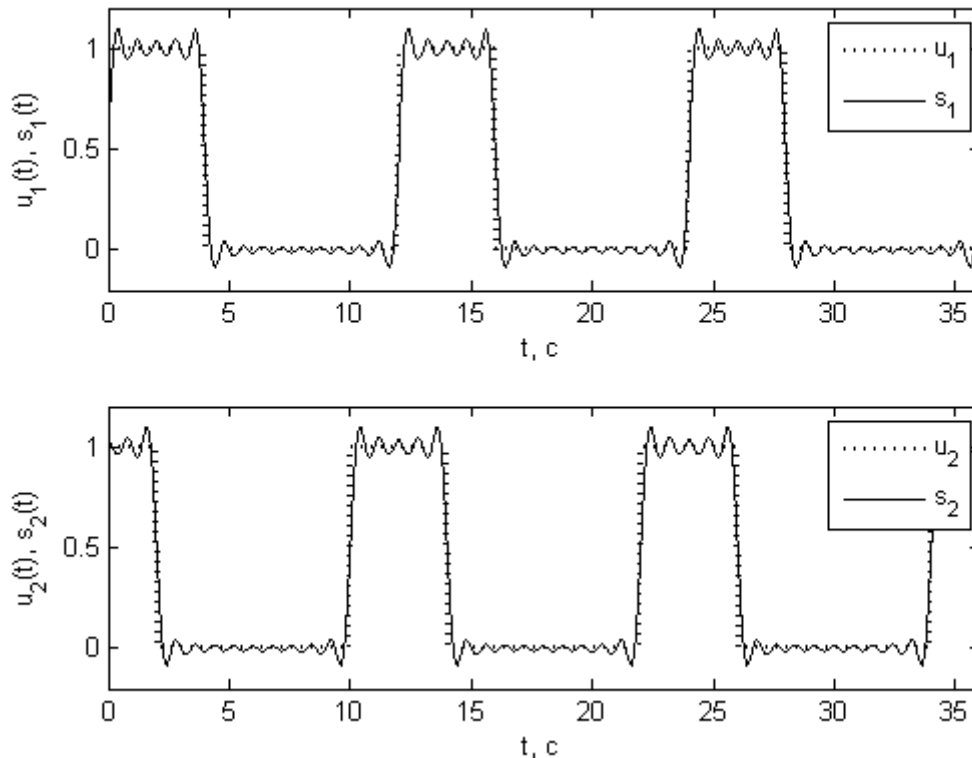


Рис. 5. Восстановленные импульсы по 15 гармоникам.

Если функция, которую раскладывают в ряд Фурье, имеет разрывы первого рода, то в точках разрыва ряд сходится к полусумме значений функции слева и справа от точки разрыва. Если оставить в сумме конечное число слагаемых, то есть выполнять приближение функции тригонометрическим полиномом, то в окрестности точки разрыва будут осцилляции. Это называется эффектом Гиббса.

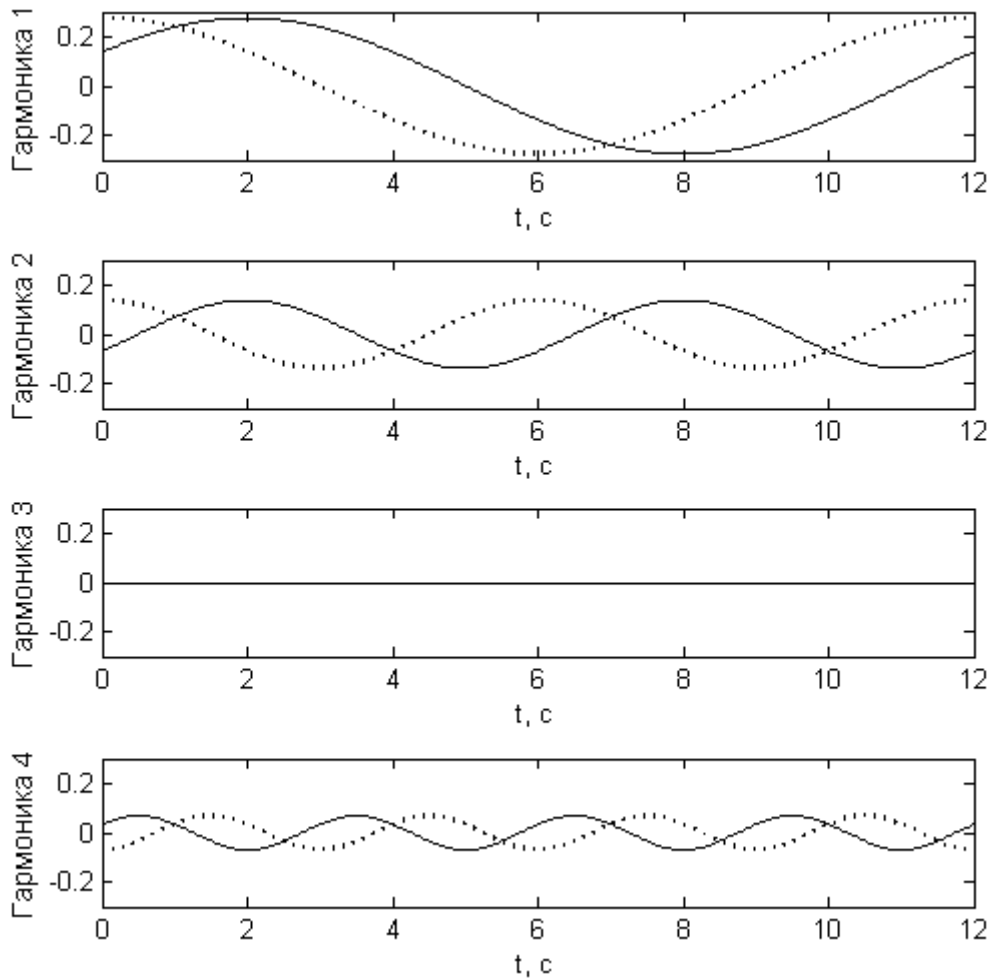


Рис. 6. Гармоники.

Пример

Эффект Гиббса демонстрируется на примере прямоугольного импульса в программе `gibbs.m`. Рассмотрим прямоугольный импульс длительностью τ , повторяющийся с периодом T .

$$x(t) = \begin{cases} 1, & 0 \leq t < \tau \\ 0, & \tau \leq t < T \end{cases} \quad (22)$$

Найдем коэффициенты разложения по формулам (7):

$$\begin{aligned} a_0 &= \frac{2}{T} \int_0^{\tau} dt = \frac{2\tau}{T} \\ a_k &= \frac{2}{T} \int_0^{\tau} \cos\left(\frac{2\pi kt}{T}\right) dt = \frac{1}{\pi k} \sin\left(\frac{2\pi k \tau}{T}\right) \\ b_k &= \frac{2}{T} \int_0^{\tau} \sin\left(\frac{2\pi kt}{T}\right) dt = \frac{1}{\pi k} \left[1 - \cos\left(\frac{2\pi k \tau}{T}\right) \right] \end{aligned} \quad (23)$$

Эффект Гиббса (gibbs.m)

```
clear, clc, close all

T = 10; % период повторения импульсов
tau = 5; % длительность импульса
N1 = 3; % количество оставленных гармоник
N2 = 7; % для сравнения другое количество гармоник
N3 = 50;
t = 0:T/20/max([N1 N2 N3]):T; % вектор с моментами времени
x = t<tau; % прямоугольный импульс длительностью tau
k1 = 1:N1;
k2 = 1:N2;
k3 = 1:N3;

% коэффициенты разложения
a0 = 2*tau/T;
% в первом случае N1 коэффициентов
a1 = sin(2*pi*k1*tau/T)/pi./k1;
b1 = (1-cos(2*pi*k1*tau/T))/pi./k1;
% во втором случае N2 коэффициентов
a2 = sin(2*pi*k2*tau/T)/pi./k2;
b2 = (1-cos(2*pi*k2*tau/T))/pi./k2;
% в третьем случае N3 коэффициентов
a3 = sin(2*pi*k3*tau/T)/pi./k3;
b3 = (1-cos(2*pi*k3*tau/T))/pi./k3;

% вычисляем сумму для первого случая
x1(1:length(t)) = a0/2;
for n = 1:N1
    x1 = x1 + a1(n)*cos(2*pi*n*t/T) + b1(n)*sin(2*pi*n*t/T);
end

% вычисляем сумму для второго случая
x2(1:length(t)) = a0/2;
for n = 1:N2
    x2 = x2 + a2(n)*cos(2*pi*n*t/T) + b2(n)*sin(2*pi*n*t/T);
end

% вычисляем сумму для третьего случая
x3(1:length(t)) = a0/2;
for n = 1:N3
    x3 = x3 + a3(n)*cos(2*pi*n*t/T) + b3(n)*sin(2*pi*n*t/T);
end

subplot(1,2,1)
plot(t,x,'k', t,x1,'k--', t,x2,'k:') % сравниваем результаты
legend('x(t)', 'x_1(t)', 'x_2(t)')
xlabel('t'), ylabel('x(t), x_1(t), x_2(t)')
subplot(1,2,2)
plot(t,x3,'k')
xlabel('t'), ylabel('x_3(t)')
```

Программа строит четыре графика (рис. 7): исходный импульс, и два восстановленных, первый – по 3-м гармоникам, второй – по 7-и, на правом графике – импульс восстановленный по 50-и гармоникам.

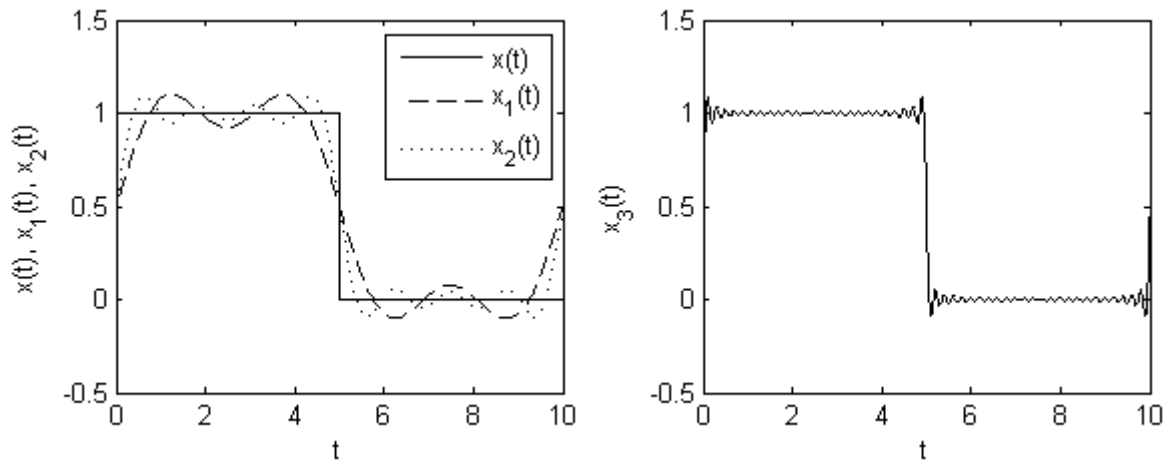


Рис. 7. Эффект Гиббса.

Если функция, раскладываемая в ряд Фурье, обладает свойством чётности или нечётности, то в формулах (7) один из интегралов будет равен нулю. Так как для периодической функции неважно, где располагается отрезок, по которому ведётся интегрирование, а важно лишь, что интеграл берётся по периоду, то эти интегралы можно брать по симметричному относительно начала координат отрезку $\left[-\frac{T}{2}, \frac{T}{2}\right]$. Тогда, если функция чётная, то интеграл с синусом обращается в ноль, так как произведение чётной и нечётной функции есть нечётная функция, а интеграл от нечётной функции по симметричному отрезку равен нулю. Аналогично для нечётной функции интеграл с косинусом равен нулю. Чётные функции раскладываются по косинусам, нечётные – по синусам.

Рассмотрим примеры разложения чётной и нечётной функции. В качестве чётной функции $x_1(t)$ будет симметричный прямоугольный импульс, в качестве нечётной $x_2(t)$ – линейно меняющаяся на периоде функция.

$$x_1(t) = \begin{cases} 1, & -\frac{\tau}{2} \leq t \leq \frac{\tau}{2} \\ 0, & -\frac{T}{2} \leq t < -\frac{\tau}{2}, \frac{\tau}{2} < t \leq \frac{T}{2} \end{cases} \quad (24)$$

$$x_2(t) = \frac{t}{T}, \quad -\frac{T}{2} \leq t \leq \frac{T}{2}$$

Подставим эти функции в формулы для коэффициентов (7). Для чётной функции $x_1(t)$ получим коэффициенты $a_{1,0}$, $a_{1,k}$ и $b_{1,k}$.

$$\begin{aligned}
a_{1,0} &= \frac{2}{T} \int_{-\frac{\tau}{2}}^{\frac{\tau}{2}} dt = \frac{2\tau}{T} \\
a_{1,k} &= \frac{2}{T} \int_{-\frac{\tau}{2}}^{\frac{\tau}{2}} \cos\left(\frac{2\pi kt}{T}\right) dt = \frac{2}{\pi k} \sin\left(\frac{\pi k \tau}{T}\right) \\
b_{1,k} &= \frac{2}{T} \int_{-\frac{\tau}{2}}^{\frac{\tau}{2}} \sin\left(\frac{2\pi kt}{T}\right) dt = 0
\end{aligned} \tag{25}$$

Для нечётной функции $x_2(t)$ аналогично $a_{2,0}$, $a_{2,k}$ и $b_{2,k}$.

$$\begin{aligned}
a_{2,0} &= \frac{2}{T} \int_{-\frac{T}{2}}^{\frac{T}{2}} \frac{t}{T} dt = 0 \\
a_{2,k} &= \frac{2}{T} \int_{-\frac{T}{2}}^{\frac{T}{2}} \frac{t}{T} \cos\left(\frac{2\pi kt}{T}\right) dt = 0 \\
b_{2,k} &= \frac{2}{T} \int_{-\frac{T}{2}}^{\frac{T}{2}} \frac{t}{T} \sin\left(\frac{2\pi kt}{T}\right) dt = \frac{\sin(\pi k)}{\pi^2 k^2} - \frac{\cos(\pi k)}{\pi k} = \frac{(-1)^{k-1}}{\pi k}
\end{aligned} \tag{26}$$

Пример

Пример разложения прямоугольного импульса и линейной функции продемонстрирован в программе `example_2.m`. Отличие этого примера от предыдущего состоит в том, что при вычислении приближенной функции используются либо только косинусы, либо только синусы.

Разложение чётных и нечётных функций (`example_2.m`)

```
clear, clc, close all
```

```
T = 10; % период сигналов
tau = 5; % длительность импульса
N = 3; % количество оставленных гармоник
dt = T/N/20; % шаг по времени
```

```

t = -T/2:dt:T/2; % моменты времени
k = 1:N;

x1 = t>=-tau/2 & t<=tau/2; % чётная функция
x2 = t/T; % нечётная функция

% коэффициенты разложения чётной функции
a10 = 2*tau/T;
a1 = 2*sin(pi*k*tau/T)/pi./k;
% коэффициенты разложения нечётной функции
b2 = (-1).^(k-1)/pi./k;

% вычисление суммы
s1(1:length(t)) = a10/2;
s2(1:length(t)) = 0;
for n=1:N
    s1 = s1 + a1(n)*cos(2*pi*n*t/T); % чётные
    s2 = s2 + b2(n)*sin(2*pi*n*t/T); % нечётные
end

% сравниваем на графиках исходные и приближенные функции
subplot(2,2,1), plot(t,x1,'k', t,s1,'k:') % чётная
subplot(2,2,2), plot(t,x2,'k', t,s2,'k:') % нечётная

% гармоники из которых складывается прямоугольный импульс
subplot(2,2,3), hold on
for n=1:N
    plot(t, a1(n)*cos(2*pi*n*t/T), 'k')
end
hold off

% гармоники из которых складывается линейная функция
subplot(2,2,4), hold on
for n=1:N
    plot(t, b2(n)*sin(2*pi*n*t/T), 'k')
end
hold off

```

В результате строятся четыре графика (рис. 8). На верхних графиках сравниваются исходные функции с приближенными, на нижних графиках приводятся гармоники, из которых складываются приближенные функции. Увеличивая количество гармоник, используемых для приближения, которое хранится в переменной N , можно убедиться, что сумма гармоник сходится к исходной функции.

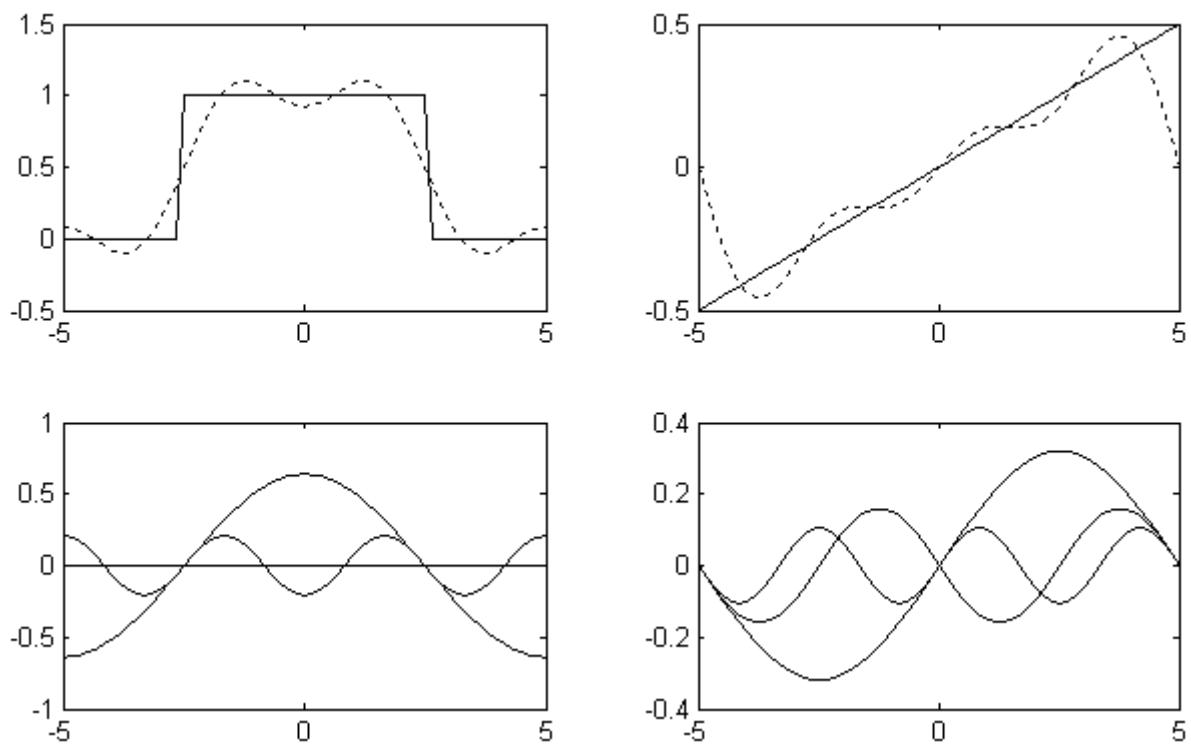


Рис. 8. Разложение чётных и нечётных функций.

3. Интегралы Фурье и Лапласа

3.1. Преобразование Фурье

В предыдущем разделе было рассмотрено разложение периодических сигналов в тригонометрический ряд, то есть в ряд по отдельным гармоникам. Сейчас мы займемся рассмотрением спектральных преобразований для непрерывных сигналов. Пусть $x(t)$ – непрерывно дифференцируемая абсолютно интегрируемая на всей оси функция.

$$\int_{-\infty}^{\infty} |x(t)| dt < \infty \quad (27)$$

Введём новые обозначения в формуле для коэффициентов разложения в комплексной форме (11):

$$\Delta\omega = \frac{2\pi}{T} \quad \omega_k = \Delta\omega k = \frac{2\pi k}{T} \quad (28)$$

Запишем формулы (11) и (8) в этих обозначениях, и в (11) заменим переменную интегрирования t на ξ .

$$x(t) = \sum_{k=-\infty}^{+\infty} c_k e^{i\omega_k t} \quad c_k = \frac{1}{T} \int_{-\frac{T}{2}}^{\frac{T}{2}} x(\xi) e^{-i\omega_k \xi} d\xi \quad (29)$$

Множитель $\frac{1}{T}$ перед интегралом заменим на $\frac{\Delta\omega}{2\pi}$ и подставим выражение для коэффициентов в сумму:

$$x(t) = \frac{\Delta\omega}{2\pi} \sum_{k=-\infty}^{+\infty} \left(\int_{-\frac{T}{2}}^{\frac{T}{2}} x(\xi) e^{-i\omega_k \xi} d\xi \right) e^{i\omega_k t} \quad (30)$$

Выполним предельный переход $T \rightarrow \infty$. При этом $\Delta\omega \rightarrow 0$, и сумму можно заменить интегралом как сумму Римана.

$$x(t) = \frac{1}{2\pi} \int_{-\infty}^{+\infty} \left(\int_{-\infty}^{+\infty} x(\xi) e^{-i\omega\xi} d\xi \right) e^{i\omega t} d\omega \quad (31)$$

Внутренний интеграл называется прямым преобразованием Фурье:

$$X(\omega) = \int_{-\infty}^{+\infty} x(t) e^{-i\omega t} dt \quad (32)$$

Функцию $X(\omega)$ называют также спектром функции $x(t)$. Таким образом, спектр непериодического сигнала – сплошной (в отличие от линейчатого спектра периодического сигнала), он определен на всей оси частот. Обратное преобразование Фурье связывает сигнал с его спектром:

$$x(t) = \frac{1}{2\pi} \int_{-\infty}^{+\infty} X(\omega) e^{i\omega t} d\omega \quad (33)$$

Представим себе случай, когда функция $x(t)$ описывает импульс, ограниченный во времени, длительностью τ , а функция $x_1(t)$ периодическую последовательность этих импульсов с периодом повторения T (рис. 9).

$$x_1(t) = \sum_{n=-\infty}^{+\infty} x(t - nT) \quad (34)$$

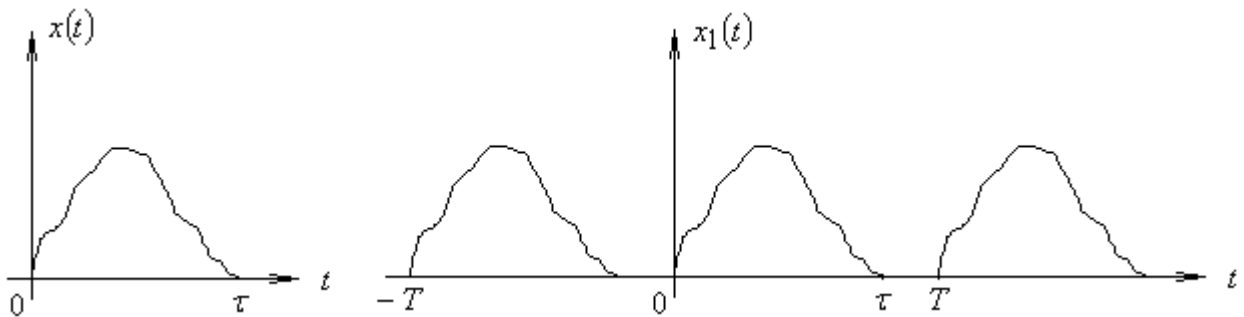


Рис. 9. Непериодический и периодический сигнал.

Между спектром $X(\omega)$ функции $x(t)$ и коэффициентами разложения (11) функции $x_1(t)$ существует определённая связь:

$$c_k = \frac{1}{T} \int_0^{\tau} x(t) e^{-i\omega_k t} dt = \frac{1}{T} X(\omega_k), \quad \omega_k = \frac{2\pi k}{T} \quad (35)$$

То есть спектр одиночного импульса с точностью до константы представляет собой огибающую дискретного спектра последовательности импульсов той же формы.

3.2. Свойства преобразования Фурье

Рассмотрим основные свойства преобразования Фурье.

Линейность. Рассмотрим функции $x_1(t)$ и $x_2(t)$, имеющие спектры $X_1(\omega)$ и $X_2(\omega)$:

$$X_1(\omega) = \int_{-\infty}^{+\infty} x_1(t) e^{-i\omega t} dt \quad X_2(\omega) = \int_{-\infty}^{+\infty} x_2(t) e^{-i\omega t} dt \quad (36)$$

Тогда спектр их линейной комбинации будет:

$$\int_{-\infty}^{+\infty} [ax_1(t) + bx_2(t)] e^{-i\omega t} dt = a \int_{-\infty}^{+\infty} x_1(t) e^{-i\omega t} dt + b \int_{-\infty}^{+\infty} x_2(t) e^{-i\omega t} dt = aX_1(\omega) + bX_2(\omega) \quad (37)$$

Задержка во времени. Считаем, что известен спектр $X(\omega)$ сигнала $x(t)$

$$X(\omega) = \int_{-\infty}^{+\infty} x(t) e^{-i\omega t} dt \quad (38)$$

Рассчитаем спектр сигнала, сдвинутого во времени: $x(t - \tau)$. Обозначим аргумент функции новой переменной $\xi = t - \tau$, тогда $t = \xi + \tau$ и $dt = d\xi$

$$\int_{-\infty}^{+\infty} x(t - \tau) e^{-i\omega t} dt = \int_{-\infty}^{+\infty} x(\xi) e^{-i\omega \xi - i\omega \tau} d\xi = e^{-i\omega \tau} \int_{-\infty}^{+\infty} x(\xi) e^{-i\omega \xi} d\xi = e^{-i\omega \tau} X(\omega) \quad (39)$$

Получили, что задержка сигнала на время τ приводит к умножению спектра на $e^{-i\omega \tau}$.

Изменение масштаба. Считаем, что известен спектр $X(\omega)$ сигнала $x(t)$, как через $X(\omega)$ выражается спектр сигнала $x(at)$. Вводим новую переменную $\xi = at$, делаем замену переменной интегрирования $dt = \frac{d\xi}{a}$.

$$\int_{-\infty}^{+\infty} x(at)e^{-i\omega t} dt = \frac{1}{a} \int_{-\infty}^{+\infty} x(\xi)e^{-i\frac{\omega}{a}\xi} d\xi = \frac{1}{a} X\left(\frac{\omega}{a}\right) \quad (40)$$

Умножение на $e^{i\omega_0 t}$. Как и в предыдущем случае, считаем, что известен спектр $X(\omega)$ сигнала $x(t)$. Найдем спектр этого сигнала, умноженного на $e^{i\omega_0 t}$.

$$\int_{-\infty}^{+\infty} x(t)e^{i\omega_0 t} e^{-i\omega t} dt = \int_{-\infty}^{+\infty} x(t)e^{-i(\omega-\omega_0)t} dt = X(\omega-\omega_0) \quad (41)$$

Таким образом, умножение сигнала на $e^{i\omega_0 t}$ приводит к смещению спектра на ω_0 .

Спектр производной. В данном случае ключевым моментом является абсолютная интегрируемость функции. Из того, что интеграл от модуля функции должен быть ограничен, следует, что на бесконечности функция должна стремиться к нулю. Интеграл от производной функции берётся по частям, получившиеся внеинтегральные слагаемые равны нулю, так как на бесконечности функция стремится к нулю.

$$\begin{aligned} \int_{-\infty}^{+\infty} \frac{dx(t)}{dt} e^{-i\omega t} dt &= \int_{-\infty}^{+\infty} e^{-i\omega t} dx(t) = \underbrace{x(t)e^{-i\omega t} \Big|_{-\infty}^{+\infty}}_0 - \int_{-\infty}^{+\infty} x(t) de^{-i\omega t} = \\ &= i\omega \int_{-\infty}^{+\infty} x(t)e^{-i\omega t} dt = i\omega X(\omega) \end{aligned} \quad (42)$$

Спектр интеграла. Найдем спектр сигнала $g(t) = \int_{-\infty}^t x(\xi)d\xi$. Причём

будем считать, что $\int_{-\infty}^{+\infty} x(t)dt = 0$, то есть у сигнала отсутствует постоянная составляющая. Это требование необходимо, чтобы внеинтегральные слагаемые были равны нулю, когда интеграл берётся по частям.

$$\begin{aligned}
\int_{-\infty}^{+\infty} g(t)e^{-i\omega t} dt &= -\frac{1}{i\omega} \int_{-\infty}^{+\infty} g(t)de^{-i\omega t} = \underbrace{-\frac{g(t)e^{-i\omega t}}{i\omega}}_0 \Big|_{-\infty}^{+\infty} + \frac{1}{i\omega} \int_{-\infty}^{+\infty} \frac{dg(t)}{dt} e^{-i\omega t} dt = \\
&= \frac{1}{i\omega} \int_{-\infty}^{+\infty} x(t)e^{-i\omega t} dt = \frac{1}{i\omega} X(\omega)
\end{aligned} \tag{43}$$

Теорема о свёртке. Известно, что $X(\omega)$ и $G(\omega)$ спектры функций $x(t)$ и $g(t)$ соответственно. Требуется выразить спектр свертки

$$u(t) = \int_{-\infty}^{+\infty} x(\xi)g(t-\xi)d\xi$$

через $X(\omega)$ и $G(\omega)$. Для этого в интеграле Фурье от

свёртки у одной из функций выполним замену переменной $\eta = t - \xi$, тогда в показателе экспоненты можно сделать замену $t = \eta + \xi$. В результате такой замены двукратный интеграл будет равен произведению двух интегралов Фурье.

$$\begin{aligned}
\int_{-\infty}^{+\infty} u(t)e^{-i\omega t} dt &= \int_{-\infty}^{+\infty} \left(\int_{-\infty}^{+\infty} x(\xi)g(t-\xi)d\xi \right) e^{-i\omega t} dt = \\
&= \int_{-\infty}^{+\infty} x(\xi) \left(\int_{-\infty}^{+\infty} g(t-\xi)e^{-i\omega t} dt \right) d\xi = \int_{-\infty}^{+\infty} x(\xi) \left(\int_{-\infty}^{+\infty} g(\eta)e^{-i\omega\eta-i\omega\xi} dt \right) d\xi = \\
&= \int_{-\infty}^{+\infty} x(\xi)e^{-i\omega\xi} d\xi \cdot \int_{-\infty}^{+\infty} g(\eta)e^{-i\omega\eta} dt = X(\omega) \cdot G(\omega)
\end{aligned} \tag{44}$$

Преобразование Фурье свёртки двух сигналов даёт произведение спектров этих сигналов.

Произведение сигналов. Известно, что $X(\omega)$ и $G(\omega)$ – спектры функций $x(t)$ и $g(t)$ соответственно. Требуется выразить спектр произведения $x(t)g(t)$ через спектры $X(\omega)$ и $G(\omega)$. Подставим в интеграл Фурье вместо одного из сигналов, например $g(t)$, его выражение через обратное преобразование Фурье, а потом поменяем порядок интегрирования.

$$\begin{aligned}
\int_{-\infty}^{+\infty} x(t)g(t)e^{-i\omega t} dt &= \frac{1}{2\pi} \int_{-\infty}^{+\infty} x(t) \left(\int_{-\infty}^{+\infty} G(\zeta)e^{i\zeta t} d\zeta \right) e^{-i\omega t} dt = \\
&= \frac{1}{2\pi} \int_{-\infty}^{+\infty} G(\zeta) \left(\int_{-\infty}^{+\infty} x(t)e^{-i(\omega-\zeta)t} dt \right) d\zeta = \frac{1}{2\pi} \int_{-\infty}^{+\infty} G(\zeta)X(\omega-\zeta)d\zeta
\end{aligned} \tag{45}$$

Спектр произведения сигналов есть свёртка спектров этих сигналов.

Приведем ряд примеров вычисления спектров типовых сигналов.

Пример.

Для начала вычислим спектр прямоугольного импульса длительностью τ (рис. 10).

$$u(t) = \begin{cases} 1, & t \in [0, \tau] \\ 0, & t \notin [0, \tau] \end{cases} \quad (46)$$

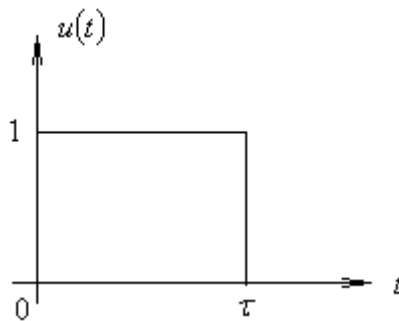


Рис. 10. Прямоугольный импульс.

Преобразование Фурье вычисляется по формуле (32). Подынтегральная функция отлична от нуля только на отрезке $[0, \tau]$:

$$U(\omega) = \int_0^{\tau} e^{-i\omega t} dt = \frac{1 - e^{-i\omega\tau}}{i\omega} = \frac{\tau}{2} \cdot \frac{e^{i\frac{\omega\tau}{2}} - e^{-i\frac{\omega\tau}{2}}}{i\frac{\omega\tau}{2}} \cdot e^{-i\frac{\omega\tau}{2}} = \tau \frac{\sin \frac{\omega\tau}{2}}{\frac{\omega\tau}{2}} e^{-i\frac{\omega\tau}{2}} \quad (47)$$

Спектр является комплексной функцией, обычно в нём выделяют амплитуду и фазу, соответственно амплитудный и фазовый спектр. Для прямоугольного импульса получили, что амплитудный спектр ведёт себя как функция $\frac{\sin x}{x}$, где $x = \frac{\omega\tau}{2}$, а фазовый спектр – линейная функция.

Теперь, используя свойства преобразования Фурье, получим спектры других сигналов, которые могут быть выражены через прямоугольный импульс.

Начнём с симметричного прямоугольного импульса $u_1(t)$ (рис. 11), который может быть представлен как сдвинутый во времени несимметричный $u_1(t) = u\left(t + \frac{\tau}{2}\right)$.

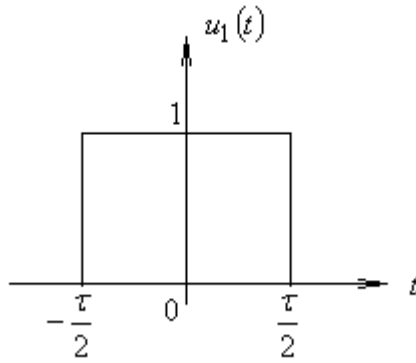


Рис. 11. Симметричный прямоугольный импульс.

Спектр этого сигнала может быть определен, если воспользоваться свойством (39)

$$U_1(\omega) = e^{i\frac{\omega\tau}{2}} U(\omega) = \tau \frac{\sin \frac{\omega\tau}{2}}{\frac{\omega\tau}{2}} \quad (48)$$

Сдвиг во времени изменил фазовый спектр, амплитудный спектр остался неизменным.

Для сравнения рассмотрим также спектр радиоимпульса с частотой заполнения \$\omega_0\$ (рис. 12) и пилообразного импульса (рис. 13).

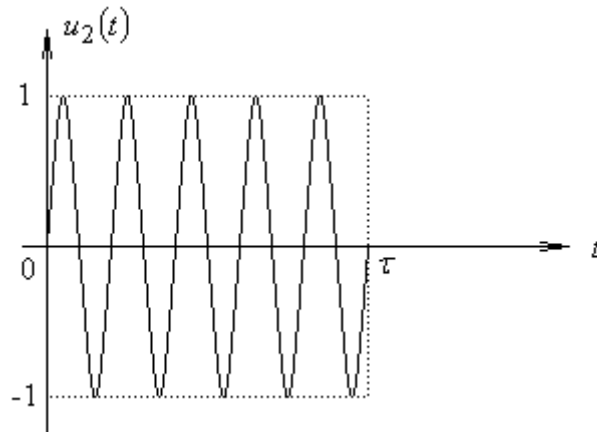


Рис. 12. Радиоимпульс.

Радиоимпульс \$u_2(t)\$ выражается через прямоугольный импульс как \$u_2(t) = u(t)\sin(\omega_0 t) = \frac{1}{2i} [u(t)e^{i\omega_0 t} - u(t)e^{-i\omega_0 t}]\$. Спектр такого сигнала может быть найден при помощи свойств (37) и (41):

$$U_2(\omega) = \frac{1}{2i} [U(\omega - \omega_0) - U(\omega + \omega_0)] \quad (49)$$

Амплитудный спектр пилообразного импульса, в отличие от спектра прямоугольного импульса, убывает заметно быстрее.

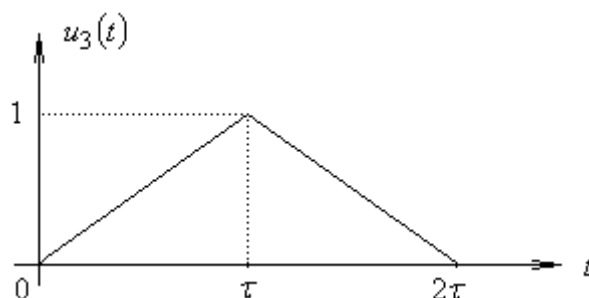


Рис. 13. Пилообразный импульс.

Такой импульс может быть выражен как свёртка прямоугольного импульса длительностью τ с самим собой $u_3(t) = \int_{-\infty}^{+\infty} u(\xi)u(t-\xi)d\xi$. По теореме о свёртке спектр такого импульса

$$U_3(\omega) = U(\omega)U(\omega) = U^2(\omega) \quad (50)$$

Сравнение спектров этих импульсов выполняется в программе example_3.m.

Свойства преобразования Фурье (example_3.m)

```
clear, clc, close all

T = 30; % длительность записи (период сигнала)
tau = 3; % длительность импульса
dt = 0.01; % шаг по времени (период квантования)
w0 = 2*pi*3/tau; % частота заполнения радиоимпульса

t = 0:dt:T; % время (для построения графиков)
N = length(t); % размер массива (количество отсчётов)

% задаём сигналы
u = t<tau; % прямоугольный импульс
u1 = t<tau/2 | t>T-tau/2; % симметричный прямоугольный импульс
u2 = u.*sin(w0*t); % радиоимпульс
u3 = conv(u,u); % пилообразный импульс (свёртка)
u3 = u3(1:length(u))*dt; % столбы сохранить размерность

% определяем частоты
f = -2:0.01:2;
w = 2*pi*f;
% спектр прямоугольного сигнала (определён как функция)
Upuls=inline('tau*sin(w*tau/2)./(w*tau/2).*exp(-
```

```

i*w*tau/2)',...
        'w','tau');

% вычисляем спектры сигналов с помощью свойств
U = Upuls(w, tau);
U1 = exp(i*w*tau/2).*U;
U2 = (Upuls(w-w0, tau)-Upuls(w+w0, tau))/2/i;
U3 = U.^2;

% вывод результатов на графиках
% прямоугольный импульс
figure
subplot(3,1,1)
plot(t,u,'k')
axis([0, T, -0.1 1.1])
xlabel('t, с', 'FontName', 'Arial Cyr')
ylabel('u(t)', 'FontName', 'Arial Cyr')
subplot(3,1,2)
plot(f,abs(U),'k')
axis([-1, 1, 0 tau])
xlabel('f, Гц', 'FontName', 'Arial Cyr')
ylabel('Амплитуда', 'FontName', 'Arial Cyr')
subplot(3,1,3)
plot(f,angle(U),'k')
axis([-1 1 -pi pi])
xlabel('f, Гц', 'FontName', 'Arial Cyr')
ylabel('Фаза', 'FontName', 'Arial Cyr')

% симметричный прямоугольный импульс
figure
subplot(3,1,1)
plot(t,u1,'k')
axis([0, T, -0.1 1.1])
xlabel('t, с', 'FontName', 'Arial Cyr')
ylabel('u_1(t)', 'FontName', 'Arial Cyr')
subplot(3,1,2)
plot(f,abs(U1),'k')
axis([-1, 1, 0 tau])
xlabel('f, Гц', 'FontName', 'Arial Cyr')
ylabel('Амплитуда', 'FontName', 'Arial Cyr')
subplot(3,1,3)
plot(f,angle(U1),'k')
axis([-1 1 -pi pi])
xlabel('f, Гц', 'FontName', 'Arial Cyr')
ylabel('Фаза', 'FontName', 'Arial Cyr')

% радиоимпульс
figure
subplot(3,1,1)
plot(t,u2,'k')
axis([0, T, -1.1 1.1])

```

```

xlabel('t, с', 'FontName', 'Arial Cyr')
ylabel('u_2(t)', 'FontName', 'Arial Cyr')
subplot(3,1,2)
plot(f,abs(U2),'k')
axis([-2, 2, 0 tau/2])
xlabel('f, Гц', 'FontName', 'Arial Cyr')
ylabel('Амплитуда', 'FontName', 'Arial Cyr')
subplot(3,1,3)
plot(f,angle(U2),'k')
axis([-2 2 -pi pi])
xlabel('f, Гц', 'FontName', 'Arial Cyr')
ylabel('Фаза', 'FontName', 'Arial Cyr')

% пилообразный импульс
figure
subplot(3,1,1)
plot(t,u3,'k')
axis([0, T, -0.3 3.3])
xlabel('t, с', 'FontName', 'Arial Cyr')
ylabel('u_3(t)', 'FontName', 'Arial Cyr')
subplot(3,1,2)
plot(f,abs(U3),'k')
axis([-1, 1, 0 3*tau])
xlabel('f, Гц', 'FontName', 'Arial Cyr')
ylabel('Амплитуда', 'FontName', 'Arial Cyr')
subplot(3,1,3)
plot(f,angle(U3),'k')
axis([-1 1 -pi pi])
xlabel('f, Гц', 'FontName', 'Arial Cyr')
ylabel('Фаза', 'FontName', 'Arial Cyr')

```

В результате работы программы будут созданы четыре окна, в каждом из которых по три графика. На верхнем графике изображён сигнал, на нижних графиках амплитудный и фазовый спектр.

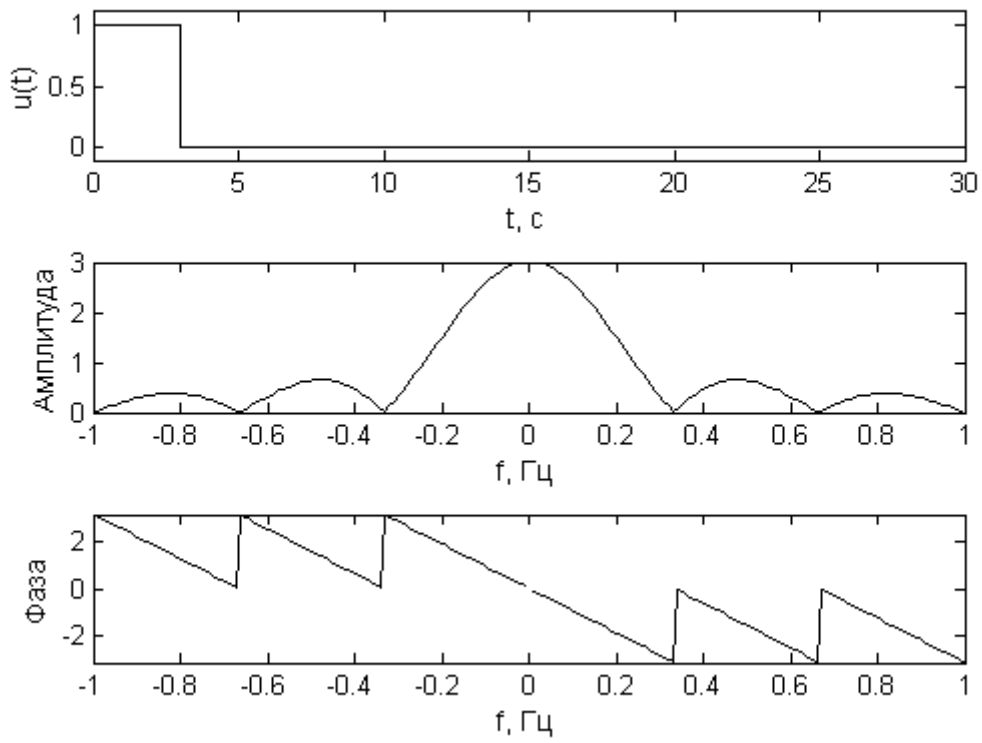


Рис. 14. Прямоугольный импульс.

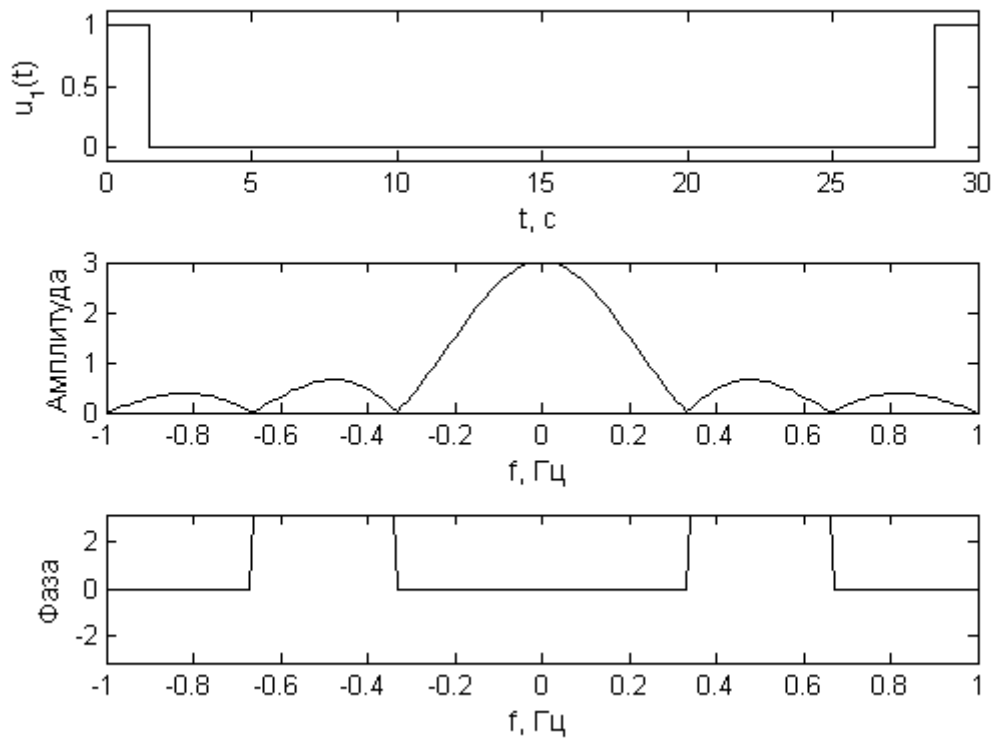


Рис. 15. Симметричный прямоугольный импульс.

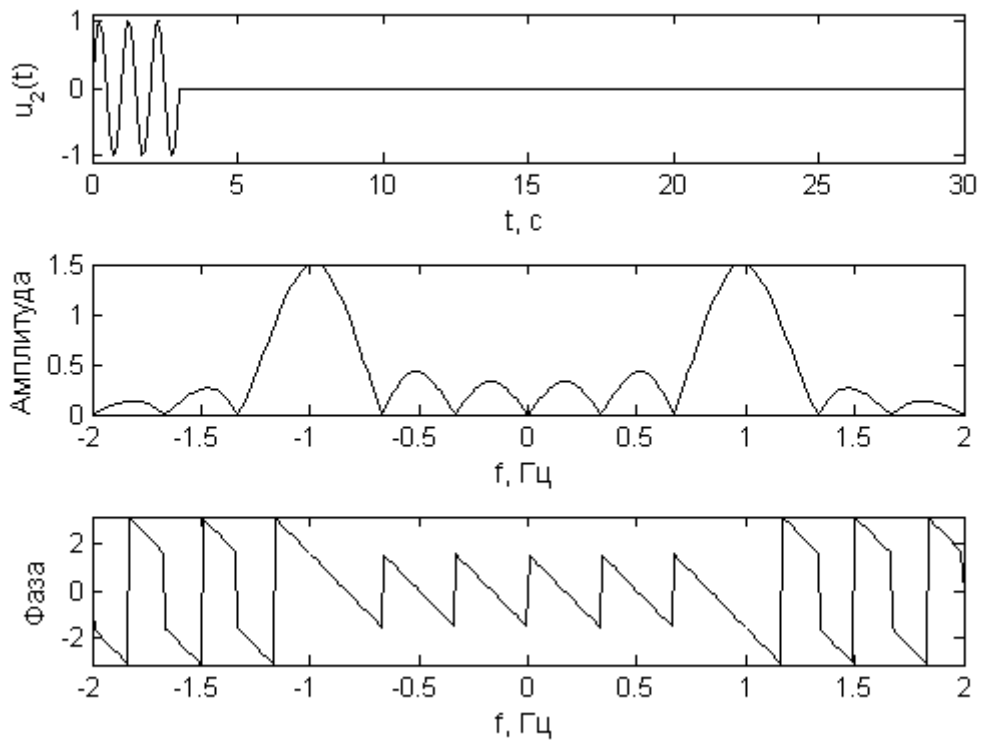


Рис. 16. Радиоимпульс.

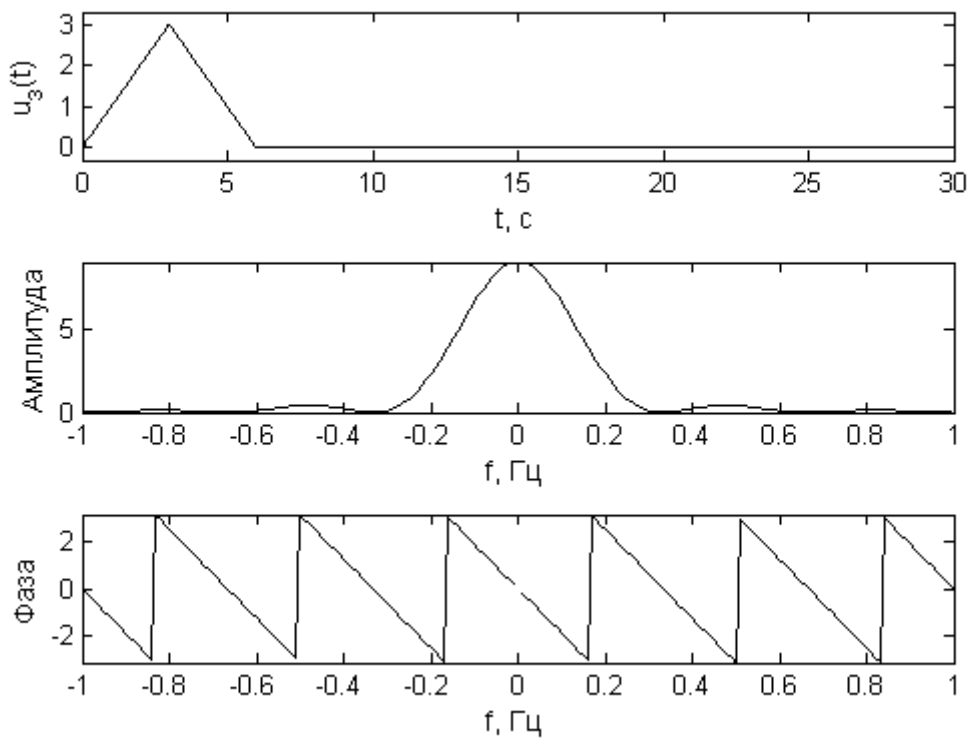


Рис. 17. Пилообразный импульс.

3.3. Косинусное преобразование

Так же как и в случае с разложением в тригонометрический ряд периодических сигналов, для преобразования Фурье можно рассмотреть случай, когда функция $x(t)$ является вещественной и чётной. В этом случае прямое и обратное преобразование Фурье превращаются в прямое и обратное косинусное преобразование. Для этого в интеграле (31) запишем комплексную экспоненту как $e^{i\omega t} = \cos(\omega t) + i \sin(\omega t)$:

$$x(t) = \frac{1}{2\pi} \int_{-\infty}^{+\infty} \left(\int_{-\infty}^{+\infty} x(\xi) e^{-i\omega\xi} d\xi \right) \cos(\omega t) d\omega + \underbrace{\frac{i}{2\pi} \int_{-\infty}^{+\infty} \left(\int_{-\infty}^{+\infty} x(\xi) e^{-i\omega\xi} d\xi \right) \sin(\omega t) d\omega}_0 \quad (51)$$

Второе слагаемое, очевидно, должно быть нулём, так как $x(t)$ – вещественная функция. Теперь выполним аналогичные действия с другой экспонентой и учтём, что функция $x(t)$ – чётная, а синус – нечётная, и интеграл с синусом будет равен нулю.

$$x(t) = \frac{1}{2\pi} \int_{-\infty}^{+\infty} \left(\int_{-\infty}^{+\infty} x(\xi) \cos(\omega\xi) d\xi - i \underbrace{\int_{-\infty}^{+\infty} x(\xi) \sin(\omega\xi) d\xi}_0 \right) \cos(\omega t) d\omega \quad (52)$$

Вещественный спектр является характерным свойством чётных функций. В результате осталось равенство:

$$x(t) = \frac{1}{2\pi} \int_{-\infty}^{+\infty} \left(\int_{-\infty}^{+\infty} x(\xi) \cos(\omega\xi) d\xi \right) \cos(\omega t) d\omega \quad (53)$$

в котором можно изменить пределы интегрирования в обоих интегралах на $[0, +\infty)$, в интеграле по ξ так как функция $x(t)$ – чётная, в интеграле по ω так как косинус – также чётная функция. При этом перед каждым интегралом возникнет множитель 2. Равенство (53) можно разделить на два. Прямое косинусное преобразование

$$X(\omega) = \sqrt{\frac{2}{\pi}} \int_0^{+\infty} x(t) \cos(\omega t) dt \quad (54)$$

и обратное косинусное преобразование, которое по форме в точности совпадает с прямым:

$$x(t) = \sqrt{\frac{2}{\pi}} \int_0^{+\infty} X(\omega) \cos(\omega t) d\omega \quad (55)$$

3.4. Преобразование Лапласа

Преобразование Фурье (32) применимо только к абсолютно интегрируемым функциям (27). Но часто приходится иметь дело с сигналами, которые абсолютной интегрируемостью не обладают, например функция единичного включения (ступенька), гармонические функции. Введём новую функцию $x_1(t) = x(t)e^{-\sigma \cdot t}$. Выбором значения $\sigma > 0$ можно обеспечить абсолютную интегрируемость новой функции, но только если $x(t) = 0$ при $t < 0$ и она растёт не быстрее, чем экспонента. Преобразование Фурье новой функции:

$$X_1(\omega) = \int_0^{+\infty} x_1(t) e^{-i\omega t} dt = \int_0^{+\infty} x(t) e^{-(i\omega + \sigma)t} dt \quad (56)$$

Подставим $X_1(\omega)$ в обратное преобразование Фурье:

$$x_1(t) = \frac{1}{2\pi} \int_{-\infty}^{+\infty} \left(\int_0^{+\infty} x(\xi) e^{-(i\omega + \sigma)\xi} d\xi \right) e^{i\omega t} d\omega \quad (57)$$

Подставим в левую часть $x(t)e^{-\sigma \cdot t}$ вместо $x_1(t)$, перенесём экспоненту $e^{-\sigma \cdot t}$ в правую часть и объединим с экспонентой $e^{i\omega t}$.

$$x(t) = \frac{1}{2\pi} \int_{-\infty}^{+\infty} \left(\int_0^{+\infty} x(\xi) e^{-(i\omega + \sigma)\xi} d\xi \right) e^{(i\omega + \sigma)t} d\omega \quad (58)$$

Теперь выполним замену переменных $s = i\omega + \sigma$, при этом $d\omega = \frac{1}{i} ds$, и мнимую единицу вынесем из-под интегралов. При такой замене переменных пределы интегрирования заменятся на $(\sigma - i\infty, \sigma + i\infty)$. В результате мы получим пару преобразований Лапласа. Прямое преобразование:

$$X(s) = \int_0^{+\infty} x(t)e^{-st} dt \quad (59)$$

и обратное преобразование:

$$x(t) = \frac{1}{2\pi i} \int_{\sigma-i\infty}^{\sigma+i\infty} X(s)e^{st} ds \quad (60)$$

Преобразование Лапласа также является линейным и для него справедливы свойства преобразования Фурье, перечисленные выше, за исключением дифференцирования и интегрирования.

Преобразование Лапласа производной функции:

$$\int_0^{+\infty} \frac{dx(t)}{dt} e^{-st} dt = x(t)e^{-st} \Big|_0^{+\infty} + s \int_0^{+\infty} x(t)e^{-st} dt = sX(s) - x(0) \quad (61)$$

Преобразование Лапласа интеграла функции $g(t) = \int_0^t x(\xi)d\xi$, при $t \leq 0$ $g(t) = 0$, так как аналогичное условие есть для $x(t)$.

$$\int_0^{+\infty} g(t)e^{-st} dt = \underbrace{-\frac{g(t)e^{-st}}{s} \Big|_0^{+\infty}}_0 + \frac{1}{s} \int_0^{+\infty} x(t)e^{-st} dt = \frac{X(s)}{s} \quad (62)$$

Рассмотрим преобразование Лапласа некоторых часто используемых сигналов:

- 1) Дельта-функция Дирака. Характерная особенность дельта-функции, что интеграл от её произведения с любой функцией равен значению этой функции в нуле. Используя это свойство, преобразование Лапласа вычисляется просто:

$$\begin{aligned} \int_{-\infty}^{+\infty} \delta(t) dt &= 1 & \int_{-\infty}^{+\infty} x(t)\delta(t) dt &= x(0) \\ \int_0^{+\infty} \delta(t)e^{-st} dt &= 1 & \Rightarrow & \frac{1}{2\pi i} \int_{\sigma-i\infty}^{\sigma+i\infty} 1 \cdot e^{st} ds = \delta(t) \end{aligned} \quad (63)$$

- 2) Преобразование функции Хевисайда (ступенька, единичный скачок).

$$h(t) = \begin{cases} 0, & t < 0 \\ 1, & t \geq 0 \end{cases} \quad (64)$$

$$\int_0^{+\infty} h(t)e^{-st} dt = \int_0^{+\infty} e^{-st} dt = \frac{1}{s}$$

Для вычисления обратного преобразования Лапласа часто используют метод контурного интегрирования. В обратном преобразовании Лапласа контур интегрирования проходит вдоль мнимой оси. Этот контур может быть изменён на другой, начинающийся и заканчивающийся не на бесконечности, а состоящий из прямого участка, идущего вдоль мнимой оси и замкнутого полуокружностью. По лемме Жордана [5] в пределе при $R \rightarrow +\infty$ интеграл по полуокружности будет стремиться к нулю, если функция, стоящая перед экспонентой, равномерно стремится к нулю при $s \rightarrow \infty$. Контур, замкнутый полуокружностью, перейдёт в контур идущей вдоль мнимой оси.

Рассмотрим интеграл (60) как интеграл по контуру, изображённому на рис. 18. Контур проходит снизу вверх в правой полуплоскости. Если $t < 0$, контур замыкается полуокружностью в правой полуплоскости, если $t > 0$ — в левой. С какой стороны замыкать контур определяется таким образом, чтобы при $R \rightarrow \infty$ экспонента стремилась к нулю, чтобы в пределе интеграл по полуокружности стремился к нулю. Рассмотрим подробнее экспоненту $e^{st} = e^{\text{Re } s \cdot t + i \text{Im } s \cdot t}$, при $t > 0$ экспонента будет убывать в левой полуплоскости $\text{Re } s < 0$.

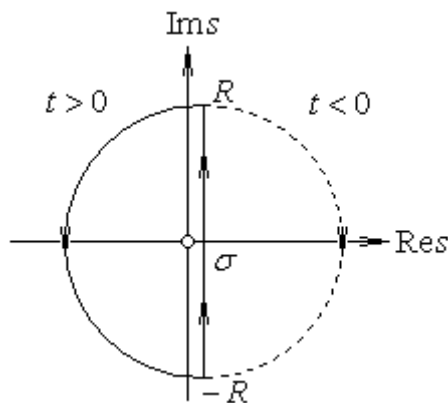


Рис. 18. Контур интегрирования.

Подынтегральная функция имеет полюс в точке $s = 0$, поэтому при $t < 0$ интеграл по контуру, внутри которого нет полюсов, равен нулю, а при $t > 0$ внутри контура будет полюс, и интеграл вычисляется по вычетам [5].

$$h(t) = \frac{1}{2\pi i} \int_{\sigma-i\infty}^{\sigma+i\infty} \frac{1}{s} e^{st} ds = \begin{cases} 0, & t < 0 \\ \operatorname{res}_{s=0} \frac{e^{st}}{s}, & t > 0 \end{cases} = \begin{cases} 0, & t < 0 \\ 1, & t > 0 \end{cases} \quad (65)$$

3) Ступенька с заполнением на частоте ω_0 . В этом случае воспользуемся свойством преобразования Фурье сигнала, умноженного на экспоненту.

$$u(t) = h(t) \sin(\omega_0 t)$$

$$U(s) = \int_0^{+\infty} \sin(\omega_0 t) e^{-st} dt = \frac{1}{2i} \left(\frac{1}{s - i\omega_0} - \frac{1}{s + i\omega_0} \right) = \frac{\omega_0}{s^2 + \omega_0^2} \quad (66)$$

4) Пилообразный импульс. Преобразование Лапласа вычисляется с использованием его свойств. Пилообразный импульс может быть задан как интеграл от линейной комбинации функций Хевисайда.

$$u(t) = \int_0^t [h(\xi) - 2h(\xi - \tau) + h(\xi - 2\tau)] d\xi$$

$$U(s) = \frac{1}{s} \left[\frac{1}{s} - 2 \frac{e^{-s\tau}}{s} + \frac{e^{-2s\tau}}{s} \right] = \frac{1 - 2e^{-s\tau} + e^{-2s\tau}}{s^2} \quad (67)$$

4. Дискретизация сигналов

4.1. Спектр дискретного сигнала

Дискретный сигнал в отличие от непрерывного является последовательностью чисел, соответствующих значениям непрерывного сигнала в определённые моменты времени. Условно дискретный сигнал можно рассматривать как непрерывный сигнал, который в определённые моменты времени принимает какие-то значения, а в остальное время равен нулю. Таким образом, например, дискретный $x_d(t)$ сигнал может быть задан как произведение непрерывного сигнала $x(t)$ на последовательность периодически повторяющихся прямоугольных импульсов $q(t)$ – тактирующих импульсов (рис.19).

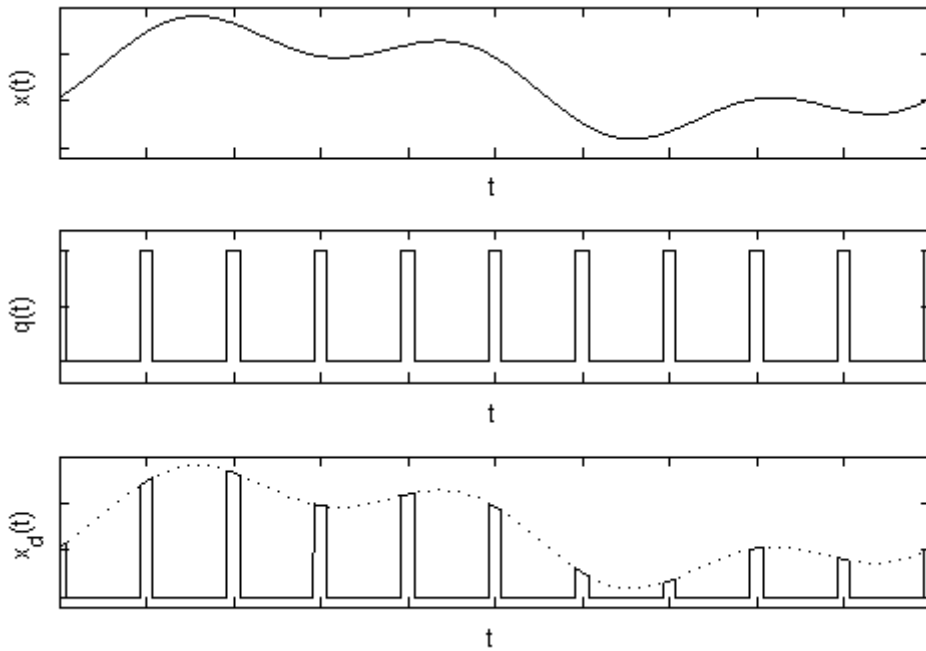


Рис. 19. Дискретизация сигнала.

$$x_d = x(t)q(t) \quad (68)$$

Прямоугольные импульсы имеют длительность τ , период повторения T_s :

$$q(t) = \begin{cases} \frac{T_s}{\tau}, & nT_s - \frac{\tau}{2} \leq t \leq nT_s + \frac{\tau}{2} \\ 0, & nT_s + \frac{\tau}{2} < t < (n+1)T_s - \frac{\tau}{2} \end{cases} \quad (69)$$

Амплитуда импульса выбрана таким образом, чтобы интеграл импульса по периоду равнялся T_s . При этом тактирующие импульсы безразмерны. Разложим последовательность таких импульсов в тригонометрический ряд

$$c_k = \frac{1}{T_s} \int_{-\frac{T_s}{2}}^{\frac{T_s}{2}} q(t) e^{-i\omega_k t} dt = \frac{1}{\tau} \int_{-\frac{\tau}{2}}^{\frac{\tau}{2}} e^{-i\omega_k t} dt = \frac{\sin\left(\frac{\omega_k \tau}{2}\right)}{\frac{\omega_k \tau}{2}} \quad (70)$$

$$q(t) = \sum_{k=-\infty}^{+\infty} c_k e^{i\omega_k t} = \sum_{k=-\infty}^{+\infty} \frac{\sin\left(\frac{\omega_k \tau}{2}\right)}{\frac{\omega_k \tau}{2}} e^{i\omega_k t}$$

Чтобы получить мгновенные отсчёты сигнала $x(t)$, надо устремить к нулю длительность импульсов к нулю: $\tau \rightarrow 0$. Такой тактирующий сигнал назовём идеальным. При этом коэффициенты разложения $q(t)$ в ряд Фурье все будут равны 1.

$$q(t) = \sum_{k=-\infty}^{+\infty} e^{i\omega_k t} \quad (71)$$

Точно такой же вид имеет разложение в ряд Фурье функции:

$$q_0(t) = T_s \sum_{n=-\infty}^{+\infty} \delta(t - nT_s) \quad (72)$$

Коэффициенты разложения в тригонометрический ряд тактирующего сигнала $q_0(t)$:

$$c_k = \frac{1}{T_s} \int_{-\frac{T_s}{2}}^{\frac{T_s}{2}} q_0(t) e^{-i\omega_k t} dt = \frac{1}{T_s} \int_{-\frac{T_s}{2}}^{\frac{T_s}{2}} T_s \delta(t - 0 \cdot T_s) e^{-i\omega_k t} dt = 1 \quad (73)$$

Тогда дискретный сигнал будет иметь вид:

$$x_d(t) = x(t) T_s \sum_{n=-\infty}^{+\infty} \delta(t - nT_s) = T_s \sum_{n=-\infty}^{+\infty} x(nT_s) \delta(t - nT_s) \quad (74)$$

При вычислении преобразования Фурье дискретного сигнала меняем местами операцию суммирования и интегрирования, а потом используем свойство δ -функции

$$X_d(\omega) = \int_{-\infty}^{+\infty} x_d(t) e^{-i\omega t} dt = T_s \sum_{n=-\infty}^{+\infty} \int_{-\infty}^{+\infty} x(t) \delta(t - nT_s) e^{-i\omega t} dt = T_s \sum_{n=-\infty}^{+\infty} x(nT_s) e^{-i\omega nT_s} \quad (75)$$

Спектр дискретного сигнала является периодической функцией. Рассмотрим экспоненту в отдельном слагаемом $e^{-i\omega nT_s}$ как функцию частоты. Её период повторения равен $\frac{2\pi}{nT_s}$. Самый большой период повторения у слагаемых с номерами $n = \pm 1$, и это, соответственно, будет периодом повторения всего спектра. То есть спектр дискретного сигнала имеет период повторения, равный частоте квантования $\omega_s = \frac{2\pi}{T_s} = 2\pi f_s$.

Получим ещё одно представление $X_d(\omega)$. В силу того, что $x_d(t)$ является произведением функций $x(t)$ и $q(t)$, спектр дискретного сигнала $X_d(\omega)$ вычисляется как свёртка спектров непрерывного сигнала $X(\omega)$ и спектра тактирующего сигнала $Q(\omega)$.

$$X_d(\omega) = \frac{1}{2\pi} \int_{-\infty}^{+\infty} X(\Omega) Q(\omega - \Omega) d\Omega \quad (76)$$

Вычислим $Q(\omega)$, используя (71). Так как $q_0(\omega)$ периодическая функция, её спектр дискретный.

$$Q(\omega) = \int_{-\infty}^{+\infty} q_0(t) e^{-i\omega t} dt = \sum_{k=-\infty}^{+\infty} \int_{-\infty}^{+\infty} e^{-i(\omega - \omega_k)t} dt = 2\pi \sum_{k=-\infty}^{+\infty} \delta(\omega - \omega_k) \quad (77)$$

Таким образом, свёртка (76)

$$X_d(\omega) = \sum_{k=-\infty}^{+\infty} \int_{-\infty}^{+\infty} X(\Omega) \delta(\omega - \Omega - \omega_k) d\Omega = \sum_{k=-\infty}^{+\infty} X(\omega - \omega_k) \quad (78)$$

Предположим, что спектр исходного сигнала ограничен частотой ω_{\max} . Из выражения (78) следует, что спектр дискретного сигнала представляет собой периодически повторяющуюся функцию $X(\omega)$.

Пример

Сравним спектры непрерывного сигнала и дискретного сигнала. В качестве сигнала возьмём экспоненциально затухающий импульс.

$$x_1(t) = \begin{cases} e^{-\frac{t}{\tau}}, & t \geq 0 \\ 0, & t < 0 \end{cases} \quad (79)$$

Спектр такого сигнала

$$X_1(\omega) = \int_0^{+\infty} e^{-\frac{t}{\tau}} e^{-i\omega t} dt = \frac{1}{\frac{1}{\tau} + i\omega} \quad (80)$$

Спектр дискретного сигнала будет выражаться через значения исходного сигнала в отдельные моменты времени

$$X_2(\omega) = T_s \sum_{n=-\infty}^{+\infty} x_1(nT_s) e^{-i\omega n T_s} \quad (81)$$

При расчете спектров в Matlab выберем такой промежуток времени, за пределами которого сигнал можно приближено считать равным нулю. Тогда в сумме (81) останутся только слагаемые с номерами от 0 до некоторого номера, соответствующего моменту времени, при котором сигнал можно считать полностью «затухшим». В Matlab нумерация массивов начинается с 1, поэтому под экспонентой будет стоять $n-1$, так как моменту времени $t=0$ соответствует $n=1$.

Спектры дискретного и непрерывного сигналов (dspectr.m)

```
clear, clc, close all

dt = 0.01; % период квантования
t = 0:dt:10; % моменты времени
tau = 0.1;
x = exp(-t/tau); % затухающий импульс
f = -200:0.1:200; % значения частот в Гц
w = 2*pi*f; % круговая частота
X1 = 1./(1/tau+i*w); % спектр непрерывного сигнала
% вычисление спектра дискретного сигнала
X2 = zeros(1, length(f));
for n=1:length(t)
    X2 = X2 + dt*x(n)*exp(-i*w*(n-1)*dt);
end
% сравниваем амплитудные спектры
```

```
subplot(2,1,1), plot(f,abs(X1)), xlabel('f'), ylabel('|X_1|')
subplot(2,1,2), plot(f,abs(X2)), xlabel('f'), ylabel('|X_2|')
```

В результате будет построено два графика. На верхнем графике представлен амплитудный спектр непрерывного сигнала, на нижнем графике – спектр дискретного сигнала (рис. 20).

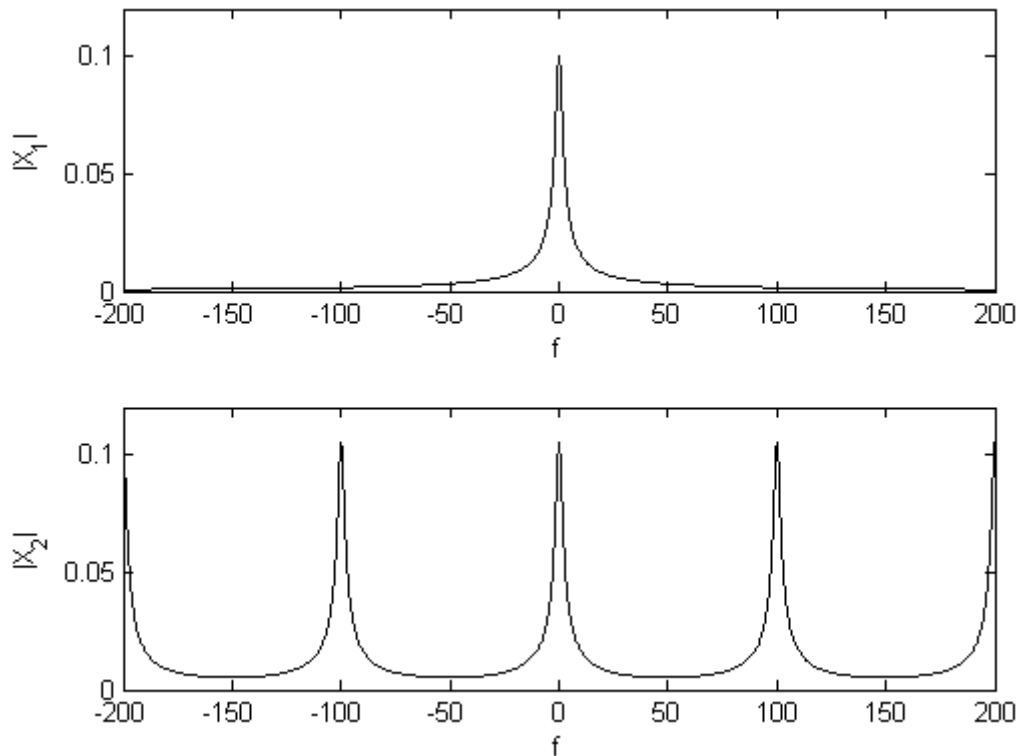


Рис. 20. Амплитудные спектры.

Период квантования был выбран 0,01с, поэтому спектр повторяется каждые 100Гц.

4.2. Теорема Котельникова

Если шаг дискретизации выбран так, что $\omega_{\max} < \frac{\omega_s}{2}$, где $\omega_s = \frac{2\pi}{T_s}$ – частота дискретизации, или частота квантования, то спектральные функции, входящие в (78), не имеют перекрывающихся участков. При этом непрерывный сигнал $x(t)$ может быть полностью восстановлен по дискретному сигналу $x_d(t)$. Чтобы восстановить непрерывный сигнал по дискретному сигналу, надо умножить периодический спектр дискретного сигнала на функцию $\Pi(\omega)$

$$\Pi(\omega) = \begin{cases} 1, & |\omega| \leq \frac{\omega_s}{2} \\ 0, & |\omega| > \frac{\omega_s}{2} \end{cases} \quad (82)$$

Умножение спектра дискретного сигнала на эту функцию даст спектр непрерывного сигнала.

$$X(\omega) = X_d(\omega)\Pi(\omega) \quad (83)$$

На рис. 21 показан пример ограниченного спектра непрерывного сигнала (верхний график) и спектра дискретного сигнала (внизу), штриховой линией показана функция $\Pi(\omega)$.

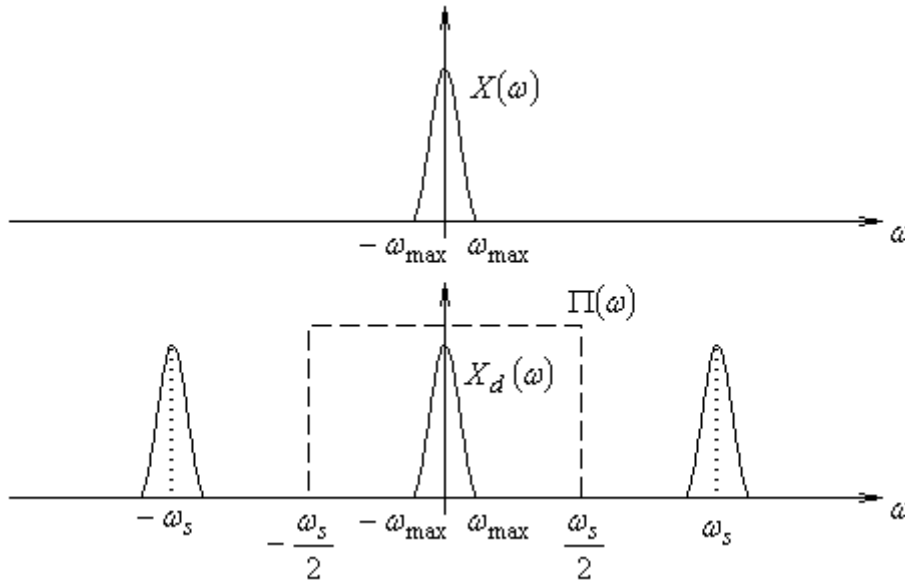


Рис. 21. Иллюстрация к теореме Котельникова.

Вычислим обратное преобразование Фурье функции $\Pi(\omega)$:

$$\Pi(t) = \frac{1}{2\pi} \int_{-\infty}^{+\infty} \Pi(\omega) e^{i\omega t} d\omega = \frac{1}{2\pi} \int_{-\frac{\omega_s}{2}}^{+\frac{\omega_s}{2}} e^{i\omega t} d\omega = \frac{\sin\left(\frac{\omega_s t}{2}\right)}{\pi t} \quad (84)$$

Так как спектр непрерывного сигнала равен произведению двух функций, сам сигнал во временной области будет свёрткой дискретного сигнала (74) с (84). Свёртка легко вычисляется благодаря наличию дельта-функции в подынтегральном выражении. Чтобы привести конечный результат к

симметричному виду, используем равенство $\omega_s = \frac{2\pi}{T_s}$ и занесём в знаменатель множитель T_s , стоящий перед знаком суммы

$$\begin{aligned}
 x(t) &= \int_{-\infty}^{+\infty} x_d(\xi) \Pi(t - \xi) d\xi = T_s \sum_{n=-\infty}^{+\infty} \int_{-\infty}^{+\infty} x(nT_s) \delta(\xi - nT_s) \frac{\sin\left(\frac{\omega_s(t - \xi)}{2}\right)}{\pi(t - \xi)} d\xi = \\
 &= \sum_{n=-\infty}^{+\infty} x(nT_s) \frac{\sin\left(\frac{\pi}{T_s}(t - nT_s)\right)}{\frac{\pi}{T_s}(t - nT_s)}
 \end{aligned} \tag{85}$$

Формула (85) называется рядом Котельникова. Таким образом, справедлива следующая теорема.

Теорема Котельникова: чтобы непрерывный сигнал можно было восстановить по его дискретным отсчётам, необходимо, чтобы частота квантования была выбрана больше удвоенной граничной частоты в спектре сигнала $\omega_s > 2\omega_{\max}$

Пример.

В этом примере демонстрируется восстановление функции с ограниченным спектром по дискретному сигналу с помощью формулы (85). Для восстановления необходимо знать отсчёты дискретного сигнала и период квантования. Причём достаточно знать только отсчёты на одном периоде функции, так как все остальные будут просто их периодическим повторением. В качестве функции с ограниченным спектром используем функцию, состоящую из суммы двух гармонических колебаний с частотами 1Гц и 2Гц.

$$x(t) = \sin(2\pi t) + \frac{1}{2} \sin(4\pi t) \tag{86}$$

Соответственно 2Гц – это верхняя граница спектра. Поэтому для того, чтобы восстановить сигнал, согласно теореме Котельникова, частота квантования должна быть больше 4Гц, то есть на периоде сигнала должно быть больше 4 отсчётов. В примере на периоде сигнала взято 5 отсчётов, по которым восстанавливается исходный сигнал. На графиках (рис. 22) эти отсчёты показаны кружками. Так как в программе нельзя вычислять суммы с бесконечным количеством слагаемых, необходимо ограничиться конечным количеством слагаемых в ряду Котельникова. С ростом номера слагаемые убывают, так как они обратно пропорциональны номеру (85).

При выполнении условия $t = nT_s$ числитель и знаменатель слагаемого ряда одновременно обращаются в ноль. Чтобы в программе не возникало ошибки, это условие проверяется, и если оно выполняется, то вместо вычисления по явной формуле выполняется вычисление по формуле, в которой раскрыта неопределённость $\lim_{\alpha \rightarrow 0} \frac{\sin \alpha}{\alpha} = 1$. В этом примере для хранения отсчётов сигнала используется вектор с пятью элементами. В ряду Котельникова суммирование производится по бесконечному количеству слагаемых. В примере – по конечному количеству, большему, чем количество отсчётов на периоде. Известно, что эти 5 отсчётов будут также периодически повторяться. Поэтому для задания отсчётов с номерами больше 5 и меньше 1 используется операция вычисления модуля от деления нацело – mod. Таким образом, создаётся эффект периодичности дискретного сигнала. В Matlab нумерация начинается не с 0, а с 1, поэтому к остатку от деления прибавляется 1.

Восстановление функции сигнала (kotelnikov.m)

```
clear, clc, close all

% сигнал с ограниченным спектром (две гармоники, период 1)
x = inline('sin(2*pi*t)+0.5*sin(4*pi*t)', 't');

% создание дискретного сигнала из 5 точек на периоде
k = 0:4; % номера отсчётов
Ts = 1/5; % период квантования
t = k*Ts; % моменты времени, соответствующие отсчётам
xd = x(t); % отсчёты

t1 = -1:0.01:3; % рассмотрим функцию на отрезке [-1, 3]

% вычисление ряда Котельникова
x1 = zeros(size(t1)); % для восстановленной функции
N = 20; % количество слагаемых, оставленных в ряде
Котельникова
for m=1:length(t1) % цикл по времени
    for n=-N:N % цикл по слагаемым
        if t1(m)==n*Ts % чтобы избежать деления на ноль
            x1(m) = x1(m) + xd(mod(n,length(xd))+1);
        else
            x1(m) = x1(m) + ...
                xd(mod(n,length(xd))+1) * ...
                sin(pi*(t1(m)-n*Ts)/Ts) / ...
                (pi*(t1(m)-n*Ts)/Ts);
        end
    end
end
end
```

```

% сравнение результатов на графиках
subplot(2,1,1), plot(t1,x(t1),'k', t,xd,'ko')
xlabel('t, c','FontName','Arial Cyr')
ylabel('x(t), x_d(t)','FontName','Arial Cyr')
legend('x(t)', 'x_d(t)')
subplot(2,1,2), plot(t1,x(t1),'k', t1,x1,'k--')
xlabel('t, c','FontName','Arial Cyr')
ylabel('x(t), x_1(t)','FontName','Arial Cyr')
legend('x(t)', 'x_1(t)')

```

В результате работы программы будет построено два графика (рис. 22). На верхнем графике показан исходный сигнал и пять отсчётов, по которым будет происходить восстановление сигнала. Восстановленный сигнал показан на нижнем графике штрихованной линией, сплошной линией – исходный сигнал.

При нарушении условия теоремы Котельникова в результате оцифровки сигнала в спектре сигнала произойдут изменения, которые не позволят восстановить исходный сигнал. Спектр дискретного сигнала образован периодически повторяющимся спектром непрерывного сигнала – см. (78). Если частота квантования ω_s будет меньше удвоенной граничной частоты спектра ω_{\max} , то в спектре дискретного сигнала будут перекрываться

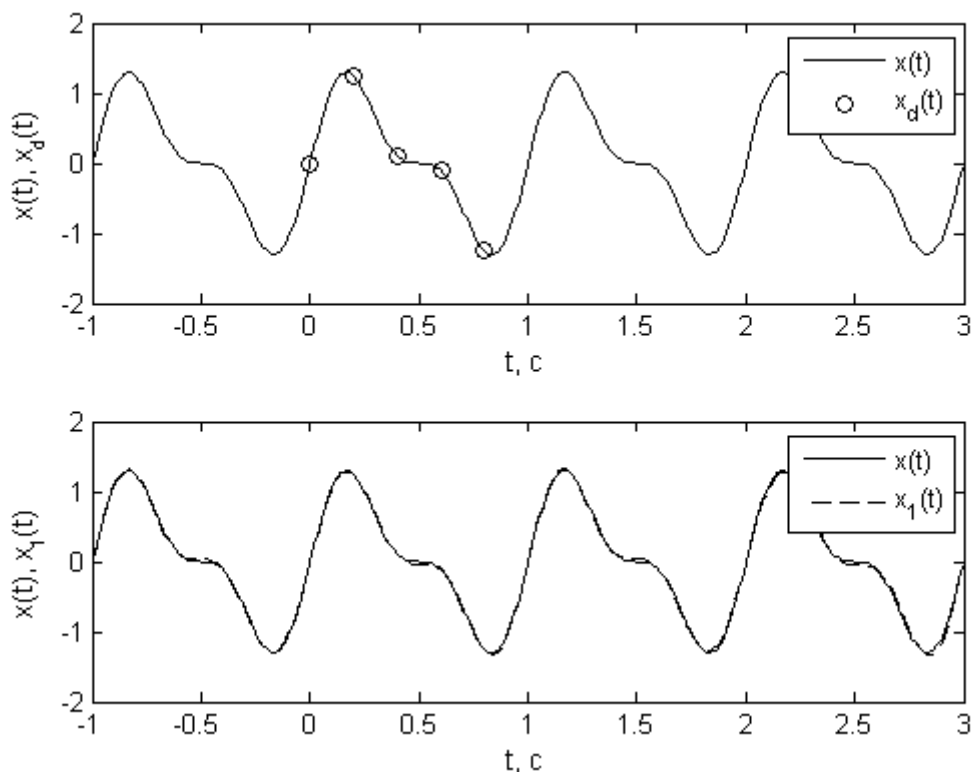


Рис. 22. Восстановление сигнала.

периодически повторяющиеся спектры непрерывного сигнала, из которых он состоит. В местах перекрытия они складываются. Эта ситуация показана на рис. 23. Штрихованными линиями показаны повторяющиеся спектры непрерывного сигнала, из которых складывается спектр дискретного сигнала, показанный сплошной линией.

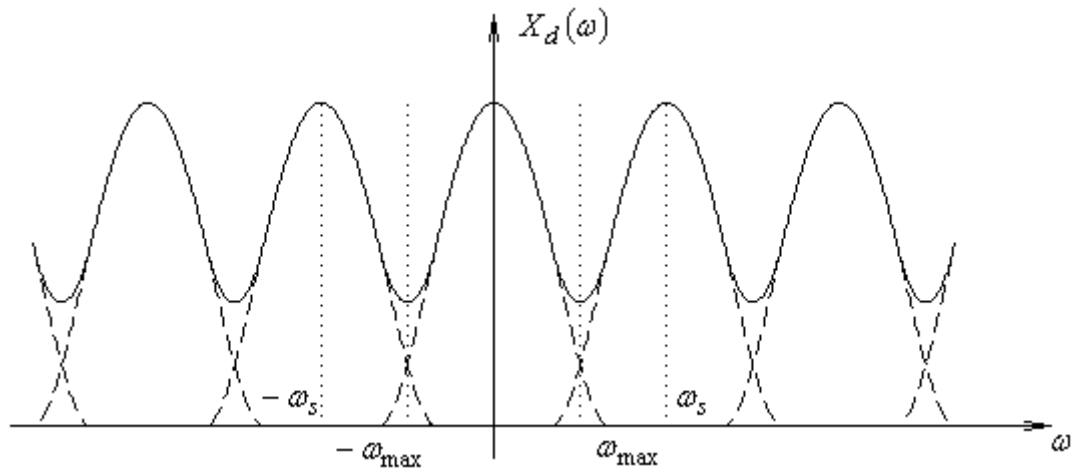


Рис. 23. Перекрытие спектров.

Пример.

В этом примере демонстрируется, что происходит с сигналом в результате дискретизации, если было нарушено условие теоремы Котельникова. Рассмотрим сигнал, состоящий из двух гармоник:

$$x_1(t) = \sin(4\pi t) + \sin(10\pi t) \quad (87)$$

Период сигнала $T = 1\text{с}$. Так как сигнал периодический, то спектр будет дискретным. Наиболее простой способ представить сигнал в виде комплексного тригонометрического ряда – это записать синусы через комплексные экспоненты

$$x_1(t) = \frac{i}{2}e^{-i2\pi \cdot 5 \cdot t} + \frac{i}{2}e^{-i2\pi \cdot 2 \cdot t} - \frac{i}{2}e^{i2\pi \cdot 2 \cdot t} - \frac{i}{2}e^{i2\pi \cdot 5 \cdot t} \quad (88)$$

Из этой формулы видно, что в комплексном спектре сигнала присутствуют гармоники с частотами $\pm 2\text{Гц}$ и $\pm 5\text{Гц}$. Так как сигнал вещественный, то комплексная амплитуда гармоники с положительной частотой равна комплексно-сопряжённой комплексной амплитуде гармоники с соответствующей отрицательной частотой. Граничная частота спектра равна 5Гц . По теореме Котельникова частота квантования должна быть выбрана больше 10Гц . Пусть частота квантования будет равна 8Гц . Посмотрим, каким будет спектр дискретного сигнала, и какому непрерывному сигналу он будет соответствовать. Спектр дискретного

сигнала будет образован периодическим повторением через каждые 8Гц спектра непрерывного сигнала, как показано на рис. 24.

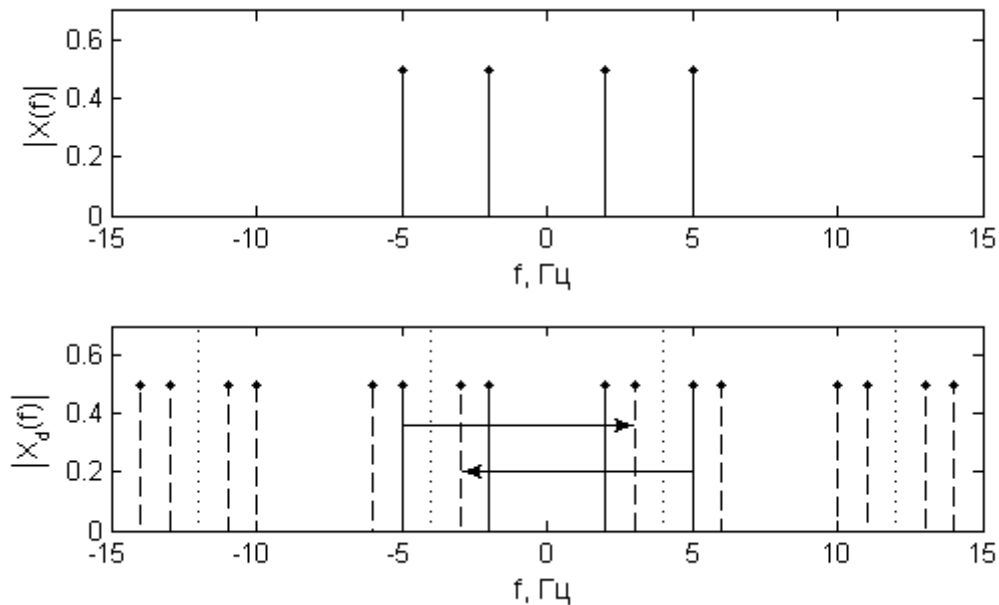


Рис. 24. Амплитудные спектры непрерывного и дискретного сигналов.

Пунктирными линиями на рисунке выделен диапазон частот $[-4\text{Гц}, 4\text{Гц}]$. На верхнем графике приведён спектр непрерывного сигнала, на нижнем графике – спектр дискретного сигнала. В спектре дискретного сигнала гармоники, находящиеся на тех же местах, что и в спектре непрерывного сигнала, показаны сплошными линиями, а все остальные, образованные их периодическим повторением, показаны штриховыми линиями. Как видно из рис. 21, дискретный сигнал будет соответствовать непрерывному сигналу, спектр которого попал в диапазон частот $[-4\text{Гц}, 4\text{Гц}]$. В результате периодического повторения спектра в этом диапазоне оказались гармоники с частотами $5\text{Гц}-8\text{Гц} = -3\text{Гц}$ и $-5\text{Гц}+8\text{Гц} = 3\text{Гц}$, это соответствие гармоник на рисунке показано стрелками. Непрерывный сигнал, которому соответствует дискретный сигнал:

$$x_2(t) = \frac{i}{2}e^{2\pi i \cdot 3 \cdot t} + \frac{i}{2}e^{2\pi i \cdot 2 \cdot t} - \frac{i}{2}e^{2\pi i \cdot 2 \cdot t} - \frac{i}{2}e^{-2\pi i \cdot 3 \cdot t} = \sin(4\pi t) - \sin(6\pi t) \quad (89)$$

Программа `kotelnikov2.m` строит графики (рис. 25) сигналов $x_1(t)$ и $x_2(t)$, и на этих графиках точками наносит отсчёты дискретного сигнала.

Искажение спектров сигнала (kotelnikov2.m)

```
clear, clc, close all

Fs = 8; % частота квантования
T = 1; % период сигнала
dt = 1/Fs; % период квантования
t = 0:dt:T; % моменты времени, соответствующие отсчётам
t1 = 0:0.01:T; % чтобы строить непрерывный сигнал

% исходный сигнал
x1 = inline('sin(2*pi*2*t)+sin(2*pi*5*t)', 't');
% сигнал полученный в результате наложения спектра
x2 = inline('sin(2*pi*2*t)-sin(2*pi*3*t)', 't');

% сравнение на графиках дискретного сигнала (кружки)
% и непрерывных сигналов
subplot(3,1,1), plot(t1,x1(t1),'k',t,x1(t),'ko')
xlabel('t'), ylabel('x_1(t)')
subplot(3,1,2), plot(t1,x2(t1),'k',t,x2(t),'ko')
xlabel('t'), ylabel('x_2(t)')
subplot(3,1,3), plot(t1,x1(t1),'k',t1,x2(t1),'k',t,x1(t),'ko')
xlabel('t'), ylabel('x_1(t), x_2(t)')
```

На третьем графике видно, что отсчёты находятся в точках пересечения двух графиков. Для сигналов с более сложным спектром, состоящим из большего количества гармоник, искажения будут происходить по такому же принципу.

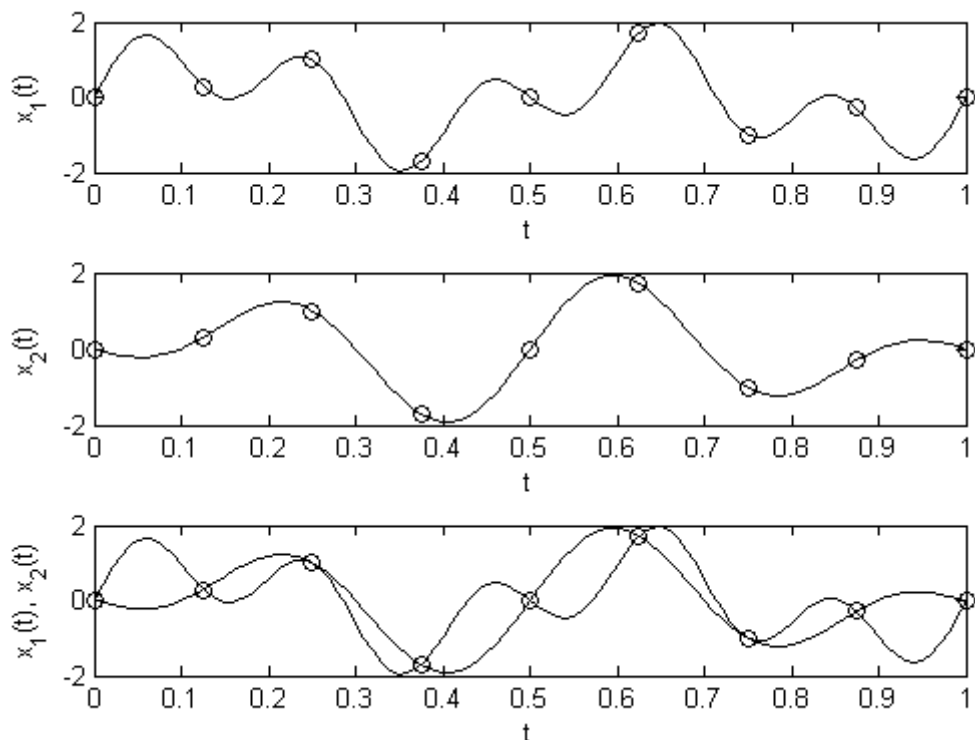


Рис. 25. Искажение сигнала при нарушении теоремы Котельникова.

В некоторых случаях при оцифровке сигналов частоту квантования можно выбирать, нарушая теорему Котельникова. При этом из оцифрованного сигнала по-прежнему можно восстановить необходимую информацию (но не сам сигнал целиком). Некоторые из таких случаев показаны в следующих примерах.

Пример.

Предположим, что оцифровывается амплитудно-модулированный сигнал. При этом необходимо получить без искажений только его огибающую. Как при оцифровке исказится несущее колебание – значения не имеет. Амплитудно-модулированный сигнал задан формулой:

$$x(t) = \left(1 + \frac{1}{2} \sin(2\pi f_1 t)\right) \sin(2\pi f_0 t) \quad (90)$$

Пусть частота несущего колебания $f_0 = 68$ Гц, а частота огибающей сигнала $f_1 = 1$ Гц. Период такого сигнала $T = 1$ с. Чтобы получить спектр сигнала, разложим его в тригонометрический ряд, записав синусы через экспоненты.

$$x(t) = -\frac{1}{8} e^{-2\pi i(f_0 + f_1)t} + \frac{i}{2} e^{-2\pi i f_0 t} + \frac{1}{8} e^{-2\pi i(f_0 - f_1)t} + \frac{1}{8} e^{2\pi i(f_0 - f_1)t} - \frac{i}{2} e^{2\pi i f_0 t} - \frac{1}{8} e^{2\pi i(f_0 + f_1)t} \quad (91)$$

Спектр состоит из 6 гармоник, показанных на рис. 26 сплошными линиями. Граничная частота спектра 69 Гц.

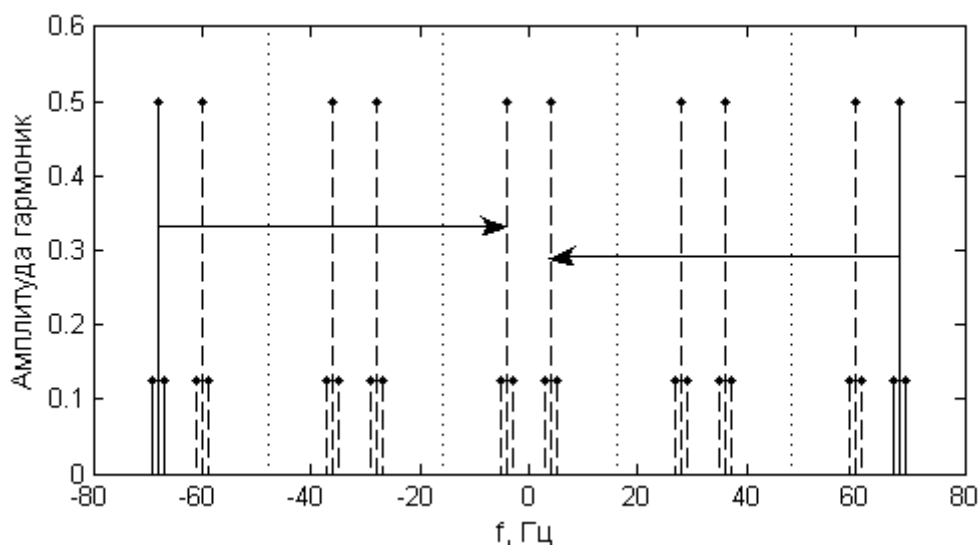


Рис. 26. Амплитудный спектр дискретного амплитудно-модулированного сигнала.

По теореме Котельникова, чтобы восстановить форму сигнала, частоту квантования следует выбирать больше 138Гц. Но если требуется восстановить только форму огибающей, то частоту квантования можно выбрать меньшей. Спектр дискретного сигнала (рис. 26) образован периодическим повторением спектра непрерывного сигнала. В данном случае частота квантования была выбрана 32Гц, поэтому на рисунке повторение спектра, показанное штриховой линией, происходит через каждые 32Гц. Восстановленный сигнал будет также амплитудно-модулированным, с той же огибающей, но с другой несущей частотой. Например, если выбрать частоту квантования 32Гц, то дискретный сигнал будет соответствовать сигналу с несущей частотой $f_0 = 4\text{Гц}$ и боковыми гармониками $f_0 - f_1 = 3\text{Гц}$ и $f_0 + f_1 = 5\text{Гц}$, как показано стрелками на рис. 26. То есть частота огибающей останется по-прежнему $f_1 = 1\text{Гц}$. Этот пример выполнен в программе kotelnikov3.m.

Амплитудно-модулированный сигнал (kotelnikov3.m)

```
clear, clc, close all

Fs = 32; % частота квантования
f1 = 1; % частота огибающей
f0 = 68; % частота несущего колебания
T = 2; % длительность сигнала

dt = 1/Fs; % период квантования
t = 0:dt:T; % время
% амплитудно-модулированный сигнал
x = (1+0.5*sin(2*pi*f1*t)).*cos(2*pi*f0*t);
% для сравнения выбираем другую частоту квантования
% чтобы выполнялась теорема Котельникова
Fs = 1380; dt = 1/Fs;
t1 = 0:dt:T;
% тот же самый амплитудно-модулированный сигнал
x1 = (1+0.5*sin(2*pi*f1*t1)).*cos(2*pi*f0*t1);
% сравниваем на графиках оба случая
subplot(2,1,1), plot(t1,x1,'k')
xlabel('t'), ylabel('x_1(t)')
subplot(2,1,2), plot(t,x,'k',t,1+0.5*sin(2*pi*t),'k--')
xlabel('t'), ylabel('x(t)')
```

В результате будут получены два графика (рис. 27). На верхнем графике неискажённый сигнал с несущей частотой 68Гц и огибающей 1Гц, а на нижнем графике сигнал, полученный после оцифровки с частотой квантования 32Гц, на этом же графике приведена огибающая.

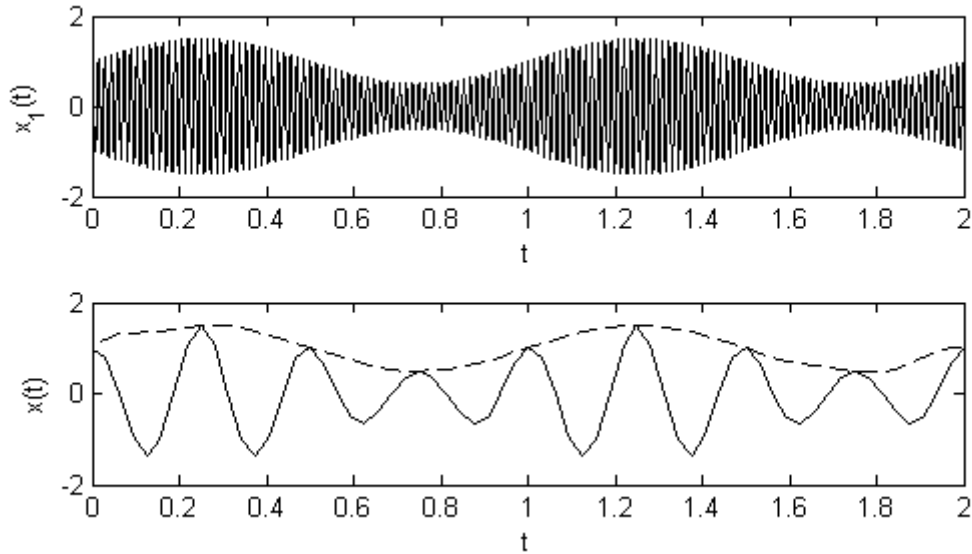


Рис. 27. Изменение несущей частоты при оцифровке.

Пример

Этот пример аналогичен предыдущему, но выполнен в Simulink, и в нём демонстрируется один из способов амплитудного детектирования. В качестве амплитудно-модулированного сигнала используется тот же сигнал (90), но с другими частотами. Частота несущего колебания $f_0 = 880\text{Гц}$, частота огибающей $f_1 = 10\text{Гц}$, частота квантования 400Гц . Восстановление амплитуды сигнала продемонстрировано в модели **kotelnikov3_1.mdl** (рис. 28).

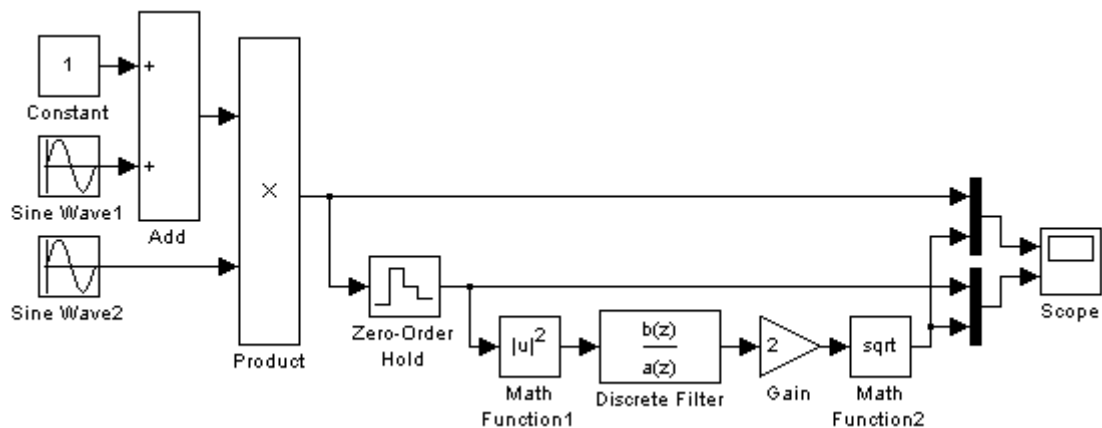


Рис. 28. Модель **kotelnikov3_1.mdl**.

Для работы этого примера в рабочей области Matlab должны находиться определённые переменные. Эти переменные создаются с помощью callback-функции модели PostLoadFcn, которая автоматически выполняется сразу после загрузки модели.

PostLoadFcn функция

```
fs = 400; % частота квантования
f0 = 880; % несущая частота
f1 = 10; % частота огибающей
[b, a] = butter(3, 0.3); % коэффициенты фильтра
```

С помощью блоков **Constant**, **Sine Wave 1**, **Sine Wave 2**, **Add** и **Product** генерируется амплитудно-модулированный сигнал (90). В блоке **Zero-Order Hold** происходит квантование сигнала с частотой, заданной в переменной f_s , а блоки **Math Function 1**, **Discrete Filter**, **Gain**, **Math Function 2**, производят детектирование сигнала с помощью квадратичного детектора.

Квадратичный детектор работает следующим образом. Пусть есть амплитудно-модулированный сигнал $x(t)$ с огибающей $A(t)$ и несущей частотой f_0 , причём высокочастотное заполнение может быть с произвольной начальной фазой φ_0 :

$$x(t) = A(t)\sin(2\pi f_0 t + \varphi_0) \quad (92)$$

Сначала сигнал возводится в квадрат, квадрат синуса можно выразить через косинус удвоенного угла:

$$u^2(t) = A^2(t)\sin^2(2\pi f_0 t + \varphi_0) = \frac{A^2(t)}{2} - \frac{A^2(t)}{2}\cos(2\pi \cdot 2f_0 \cdot t + 2\varphi_0) \quad (93)$$

Возведение в квадрат дало низкочастотное слагаемое и высокочастотное слагаемое с удвоенной несущей частотой, которое можно отфильтровать, пропустив сигнал через фильтр нижних частот. Для этой цели были найдены коэффициенты дискретного фильтра, хранящиеся в переменных a и b , которые задаются в параметрах блока **Discrete Filter**. Команда `butter` вычисляет эти коэффициенты, первый аргумент команды задаёт порядок дискретного фильтра, второй параметр – отношение частоты отсечки к частоте Найквиста, то есть половине частоты квантования. Можно построить АЧХ этого фильтра:

АЧХ дискретного фильтра

```
z = 0:0.1:pi;
h = freqz(b, a, z);
semilogx(z*fs/2/pi, abs(h))
```

Функция `freqz` вычисляет частотные характеристики дискретного фильтра. Первые два аргумента – коэффициенты фильтра, третий аргумент – вектор со значениями в интервале от 0 до π . В этот вектор определяет

частоты, в которых будет вычислен коэффициент передачи. Чтобы получить частоты надо умножить его на частоту квантования и разделить на 2π . Чтобы построить АЧХ, используем команду `abs(h)`.

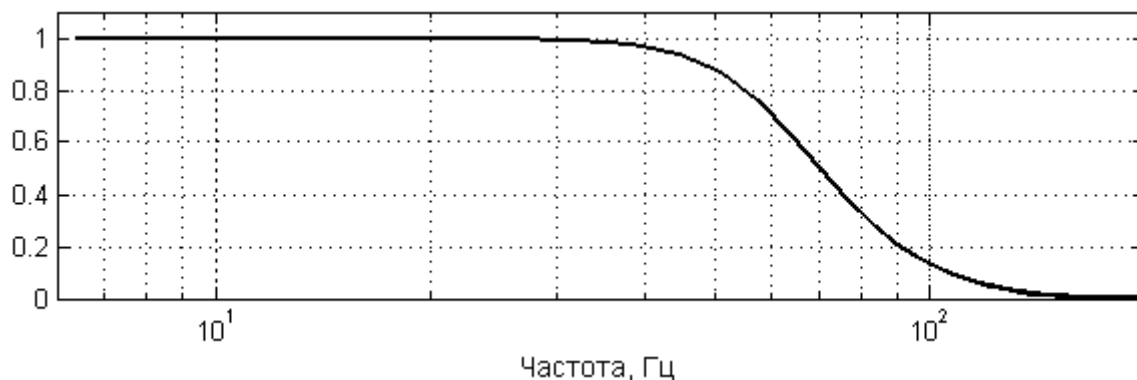


Рис. 29. АЧХ дискретного фильтра.

Как будет показано дальше, после оцифровки несущая частота станет 80 Гц, соответственно фильтр должен подавлять частоту 160 Гц. После фильтрации в (93) останется только первое слагаемое, чтобы восстановить амплитуду, его надо умножить на 2 и извлечь корень.

В настройках **Simulation->Configuration Parameters...** задать **Stop time** – $4/f_1$, **Solver** – `Discrete`, **Max Step Size** – $1/10/f_0$. В настройках блоков заданы параметры (табл. 1):

Блок	Параметр	Значение
Sine Wave 1	Amplitude	0.5
	Frequency	$2*\pi*f_1$
Sine Wave 2	Frequency	$2*\pi*f_0$
	Phase	$2*\pi*rand$
Zero-Order Hold	Sample time	$1/f_s$
Math-Function 1	Function	$magnitude^2$
Discrete Filter	Numerator coefficients	b
	Denominator coefficients	a
	Sample time	-1
Gain	Gain	2
Math Function 2	Function	<code>sqrt</code>

Табл. 1. Параметры блоков модели **example2**

Для восстановления формы сигнала без потерь необходимо было бы выбрать частоту квантования $f_s > 2f_0$, но в этом примере частота квантования этому условию не удовлетворяет. Также как и в предыдущем примере происходят изменения в спектре сигнала (рис. 30).

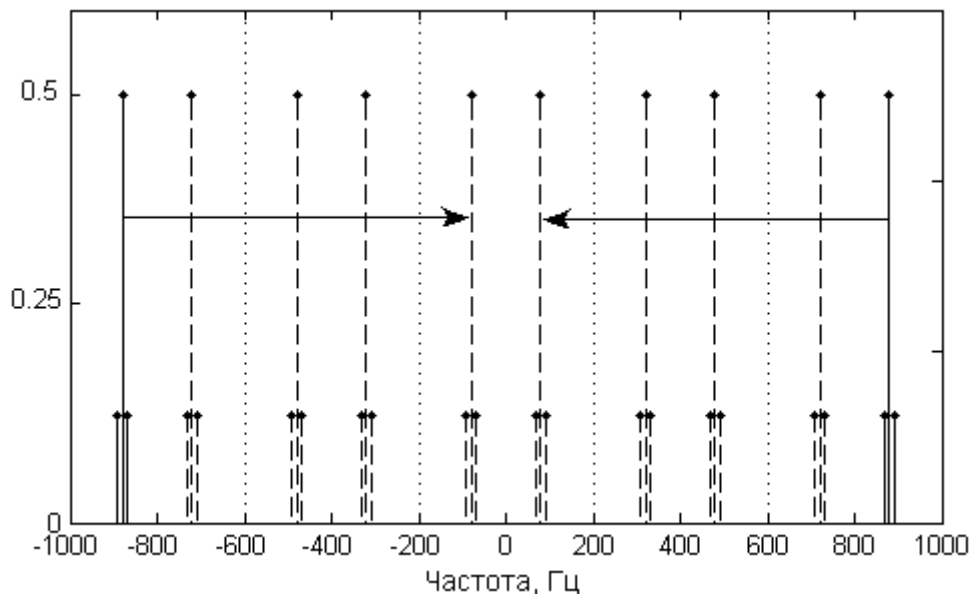


Рис. 30. Изменения в спектре сигнала.

Сплошной линией показан спектр исходного сигнала, пунктирной линией отмечена половина частоты квантования. Спектр оцифрованного сигнала периодически повторяется, эти гармоники показаны штрихованной линией. В сигнале, оцифрованном с частотой 400 Гц будут наблюдаться гармоники, попавшие в интервал частот ± 200 Гц, то есть несущая частота 80 Гц и две гармоники 70 Гц и 90 Гц соответствующие амплитудной модуляции гармоническим сигналом с частотой 10 Гц. Несущая частота изменилась, а расстояние между гармониками осталось прежним, поэтому закон модуляции остался прежним.

Пример

При оцифровке периодических сигналов период квантования может быть выбран больше, чем период сигнала. Пусть период сигнала T , период квантования $T_s = nT + \Delta t$, где n — это количество пропущенных периодов сигнала, а $\Delta t = \frac{T}{m}$, где m — количество отсчётов на период. В результате дискретизации получится сигнал той же формы, но растянутый во времени. На практике такой приём позволяет оцифровать периодические сигналы, для которых нельзя подобрать частоту квантования АЦП, удовлетворяющую теореме Котельникова. Для АЦП всегда существуют ограничения для выбора частоты квантования и других параметров,

вызванные их техническими характеристиками. В этом примере оцифровывается сигнал с периодом $T = 1\text{с}$. Спектр сигнала состоит из двух гармоник 1Гц и 2Гц. По теореме Котельникова частота квантования должна быть больше 4Гц, то есть период квантования меньше 0,25с. В примере сигнал оцифровывается с периодом квантования 0,05с и 1,05с. В первом случае условие теоремы Котельникова выполнены, и на периоде получается 20 отсчётов, такой выбор сделан, чтобы построить график сигнала по отсчётам. Во втором случае при оцифровке пропускается один период, но за счёт периодичности получается сигнал той же формы, но периодом 20с.

Оцифровка периодических сигналов (kotelnikov4.m)

```
clear, clc, close all

% сигнал с периодом 1с и максимальной частотой в спектре 2Гц
x = @(t) 0.75-0.5*cos(2*pi*t)-0.25*cos(4*pi*t);

% по теореме котельникова период квантования должен быть
% меньше 0.25с (больше 4-х отсчётов на период)
% оцифровываем сигнал с разными периодами квантования
t1 = 0:0.05:20; % 20 отсчётов на период и 20 периодов
t2 = 0:1.05:20; % нарушили теорему Котельникова (1 период)
t3 = 0:1.05:400; % нарушили теорему Котельникова (20 периодов)

subplot(2,1,1), plot(t1,x(t1),'k', t2,x(t2),'ko-')
xlabel('t'), ylabel('x(t)')
subplot(2,1,2), plot(t3,x(t3),'k')
xlabel('t'), ylabel('x(t)')
```

Программа построит два графика (рис. 31), на верхнем графике показан исходный сигнал и оцифрованный, отсчёты оцифрованного сигнала показаны кружками. Из графика видно, что период оцифрованного сигнала в 20 раз больше чем период исходного, а форма – такая же. Чтобы убедиться, что форма оцифрованного сигнала повторяет форму исходного сигнала, построим график (нижний график на рис. 31) оцифрованного сигнала на отрезке времени в 20 раз больше чем на верхнем графике.

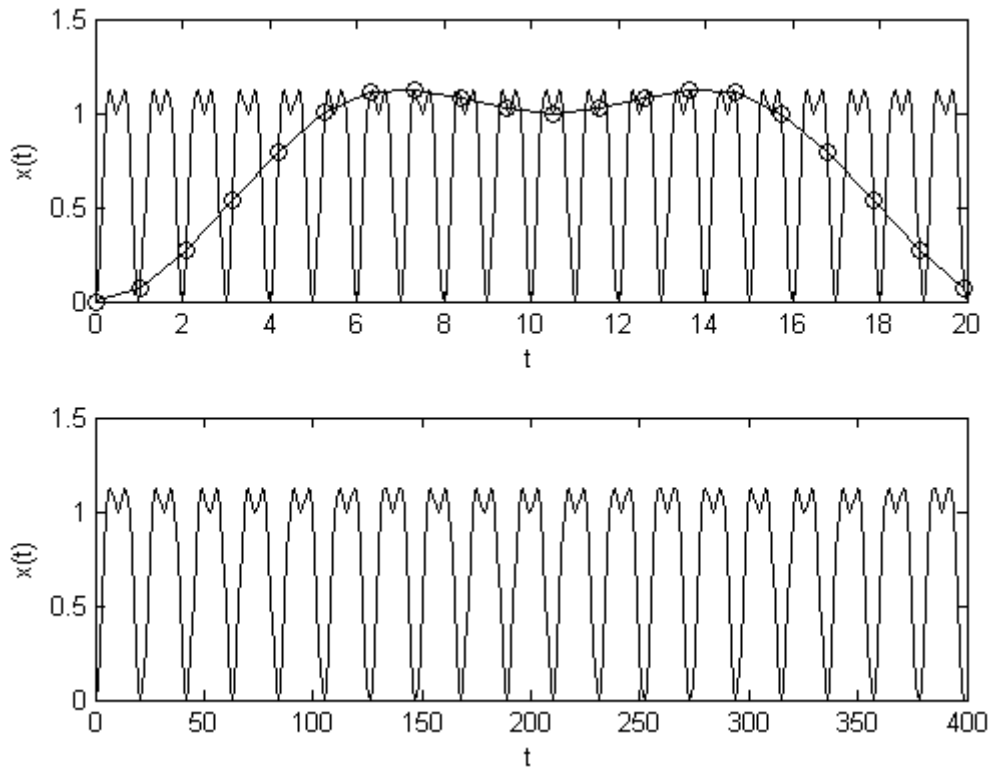


Рис. 31. Оцифровка периодического импульса.

Пример

В этом примере выполняются те же действия, что и в предыдущем, но моделирование происходит в Simulink. Сигнал создаётся с помощью блоков Constant и Sine Wave из библиотеки Simulink\Sources и блока Add из библиотеки Simulink\Math. Квантование сигнала осуществляется с помощью блоков Zero-Order Hold из библиотеки Simulink\Discrete. В схеме (рис.6) два таких блока подключены в схему параллельно, один осуществляет оцифровку с периодом 0,05с, другой 1,05с.

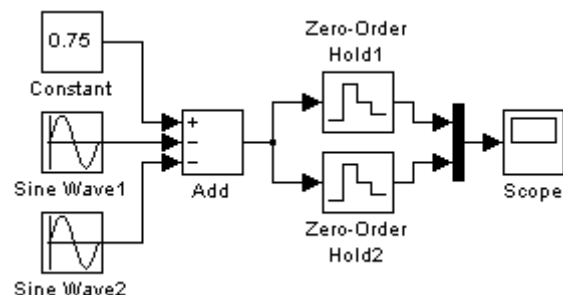


Рис. 6. Модель kotelnikov4_1.mdl.

Параметры блоков модели перечислены в таблице 3. Если задать время моделирования 20с, то на осциллографе получится график, аналогичный верхнему графику на рис. 31.

Блок	Параметр	Значение
Constant	Constant value	0.75
Sin Wave 1	Frequency	$2 \cdot \pi$
	Phase	$\pi/2$
Sin Wave 2	Frequency	$4 \cdot \pi$
	Phase	$\pi/2$
Zero-Order Hold 1	Sample time	0.05
Zero-Order Hold 2	Sample time	1.05

Табл. 3 Параметры блоков модели `kotelnikov4_1.mdl`

5. Дискретное преобразование Фурье

5.1. Вывод матрицы дискретного преобразования Фурье

Ранее было показано, что спектр периодического сигнала дискретный, то есть сигнал может быть разложен по определённым гармоникам. Дискретный сигнал имеет периодический спектр. Дискретный периодический сигнал будет иметь дискретный периодический спектр. Дискретный сигнал представляется в виде последовательности значений сигнала $x_n = x(t_n)$ в фиксированные моменты времени $t_n = nT_s$. У периодического дискретного сигнала значения x_n периодически повторяются через определённое количество отсчётов N , то есть для любого n выполняется $x_n = x_{n+N}$. Разложим исходный непрерывный сигнал в ряд Фурье:

$$x(t) = \sum_{k=-\infty}^{+\infty} c_k e^{i\omega_k t}, \quad \omega_k = \frac{2\pi k}{T} \quad (94)$$

Вычислим значения этого ряда в моменты времени t_n и учтём, что на периоде сигнала T находится N отсчётов, то есть $T = NT_s$.

$$x_n = \sum_{k=-\infty}^{+\infty} c_k e^{i\omega_k t_n} = \sum_{k=-\infty}^{+\infty} c_k e^{i \frac{2\pi k}{T} n T_s} = \sum_{k=-\infty}^{+\infty} c_k e^{\frac{2\pi i}{N} kn} \quad (95)$$

В получившейся сумме можно привести подобные слагаемые, так как каждое N -ое значение комплексной экспоненты одинаково:

$$e^{\frac{2\pi i}{N}(k+mN)n} = e^{\frac{2\pi i}{N}kn} \cdot e^{i2\pi mn} = e^{\frac{2\pi i}{N}kn} \quad (96)$$

$$x_n = \sum_{k=0}^{N-1} A_k e^{\frac{2\pi i}{N}kn}, \quad A_k = \sum_{s=-\infty}^{+\infty} c_{k+sN}$$

Комплексные экспоненты в разложении в ряд Фурье образуют ортонормированный базис функции, это свойство используется для нахождения коэффициентов разложения A_k . Определим скалярное произведение дискретно заданных функций x_n и y_n следующим образом:

$$(x, y) = \frac{1}{N} \sum_{n=0}^{N-1} x_n \bar{y}_n \quad (97)$$

где чертой обозначено комплексное сопряжение. Умножив скалярно выражение для x_n в (96) на $e^{-\frac{2\pi i}{N}mn}$ можно найти все коэффициенты A_k .

$$\frac{1}{N} \sum_{n=0}^{N-1} x_n e^{-\frac{2\pi i}{N}mn} = \sum_{k=0}^{N-1} A_k \left[\frac{1}{N} \sum_{n=0}^{N-1} e^{\frac{2\pi i}{N}(k-m)n} \right] \quad (98)$$

Сумма в квадратных скобках может быть сосчитана как сумма геометрической прогрессии, которая вычисляется по формуле:

$$b_m = b_0 q^m, \quad S_N = \sum_{n=0}^{N-1} b_0 q^n = b_0 \frac{1-q^N}{1-q} \quad (99)$$

Сравнив сумму в квадратных скобках в (98) с (99):

$$b_0 = 1, \quad q = e^{\frac{2\pi i}{N}(k-m)}$$

$$\frac{1}{N} \sum_{n=0}^{N-1} e^{\frac{2\pi i}{N}(k-m)n} = \frac{1}{N} \frac{1 - e^{2\pi i(k-m)}}{1 - e^{\frac{2\pi i}{N}(k-m)}} = \delta_{km} = \begin{cases} 1, & m = k \\ 0, & m \neq k \end{cases} \quad (100)$$

Числитель дроби всегда равен нулю, а знаменатель обращается в ноль, только когда $k = m$. В этом случае возникшая неопределённость раскрывается по правилу Лопиталья. Если подставить в (98) вместо квадратных скобок символ Кронекера δ_{km} , в сумме по k останется только A_m .

$$A_m = \frac{1}{N} \sum_{n=0}^{N-1} x_n e^{-\frac{2\pi i}{N}nm} \quad (101)$$

Формула (101), определяющая коэффициенты разложения в тригонометрический ряд Фурье, называется прямым дискретным преобразованием Фурье, а формула (96) обратным дискретным преобразованием. Эти формулы симметричны и отличаются только знаком под экспонентой и постоянным множителем перед знаком суммы. Обычно

дискретное преобразование Фурье сигнала, заданного отсчётами в виде вектора x_n из N элементов, вычисляется по формуле:

$$X_m = \sum_{n=0}^{N-1} x_n e^{-\frac{2\pi i}{N}nm} = \sum_{n=0}^{N-1} x_n W_N^{mn}, \quad W_N = e^{-\frac{2\pi i}{N}} \quad (102)$$

Обратное преобразование Фурье по формуле:

$$x_n = \frac{1}{N} \sum_{m=0}^{N-1} X_m e^{\frac{2\pi i}{N}mn} = \frac{1}{N} \sum_{m=0}^{N-1} X_m W_N^{-mn} \quad (103)$$

Формула (102) может быть представлена произведением матрицы на вектор:

$$\begin{pmatrix} X_0 \\ X_1 \\ \vdots \\ X_{N-1} \end{pmatrix} = \begin{pmatrix} W_N^{00} & W_N^{01} & \dots & W_N^{0(N-1)} \\ W_N^{10} & W_N^{11} & \dots & W_N^{1(N-1)} \\ \vdots & \vdots & \ddots & \vdots \\ W_N^{(N-1)0} & W_N^{(N-1)1} & \dots & W_N^{(N-1)(N-1)} \end{pmatrix} \begin{pmatrix} x_0 \\ x_1 \\ \vdots \\ x_{N-1} \end{pmatrix} \quad (104)$$

5.2. Алгоритм быстрого преобразования Фурье

Для вычисления каждого X_m производится N операций умножения и $N-1$ операций сложения, то есть для вычисления всего вектора u количество операций сложения и умножения будет $N(2N-1)$, величина порядка $O(N^2)$. При этом ещё необходимо затратить дополнительное время для вычисления всех элементов матрицы. Использование дискретного преобразования Фурье не эффективно, так как время, затрачиваемое на вычисления, пропорционально квадрату количества отсчётов.

Количество операций, затрачиваемых на вычисление суммы в (102), можно значительно сократить, если количество элементов x_n может быть представлено как произведение целых чисел $N = N_1 N_2$. Этот алгоритм называется алгоритмом **быстрого преобразования Фурье** (был разработан математиками J W Cooley и J W Tukey в 1965 году). Наиболее эффективный вариант, когда N есть 2 в какой-нибудь степени, тогда количество операций, требуемых для быстрого преобразования Фурье, порядка $O(N \ln N)$. В этом случае $N = 2M$.

В формуле (102) разобьём слагаемые на две суммы, в одной будут слагаемые с чётными номерами, в другой – с нечётными:

$$X_m = \sum_{n=0}^{\frac{N}{2}-1} x_{2n} W_N^{2nm} + \sum_{n=0}^{\frac{N}{2}-1} x_{2n+1} W_N^{(2n+1)m} \quad (105)$$

Преобразуем каждую из этих сумм, воспользовавшись следующими свойствами W_N .

$$W_N^{2nm} = e^{-\frac{2\pi i}{N} 2nm} = e^{-\frac{2\pi i}{N/2} nm} = W_M^{nm} \quad (106)$$

$$W_N^{(2n+1)m} = e^{-\frac{2\pi i}{N} (2n+1)m} = e^{-\frac{2\pi i}{N/2} nm} \cdot e^{-\frac{2\pi i}{N} m} = W_N^m \cdot W_M^{nm}$$

Подставив (106) в (105) получим:

$$X_m = \sum_{n=0}^{M-1} a_n W_M^{nm} + W_N^m \sum_{n=0}^{M-1} b_n W_M^{nm}, \quad a_n = x_{2n}, \quad b_n = x_{2n+1} \quad (107)$$

Каждая из сумм в полученном выражении – это дискретное преобразование Фурье (105) сигнала, состоящего из половины элементов исходного сигнала. Каждая из этих сумм содержит M операций умножения и $M-1$ операций сложения. Отметим также ещё одно свойство W_N , позволяющее сократить в два раза количество операций:

$$W_N^{m+M} = e^{-\frac{2\pi i}{N} \left(m + \frac{N}{2}\right)} = e^{-i\pi} \cdot e^{-\frac{2\pi i}{N} m} = -W_N^m \quad (108)$$

Для вычисления всех X_m по формуле (107) суммы необходимо вычислять для номеров n только от 0 до $M-1$, а для нахождения оставшихся X_m с номерами от M до $N-1$ использовать свойство (108). Общее количество операций для вычисления X_m по формуле (107) порядка $O(M^2)$, то есть в четыре раза меньше, чем по формуле (102).

Если N является степенью двойки, то и M также будет чётным числом, поэтому каждую сумму в (107) можно разложить таким же образом ещё на две, и так далее до тех пор, пока не дойдём до дискретных преобразований Фурье двухточечных сигналов.

Разберём подробно алгоритм вычисления быстрого преобразования Фурье на примере сигнала, состоящего из $N=4$ отсчётов. Дискретное преобразование такого сигнала (105) разбиваем на две суммы с чётными и нечётными слагаемыми:

$$X_m = \sum_{n=0}^3 x_n W_4^{nm} = \sum_{n=0}^1 x_{2n} W_2^{nm} + W_4^m \sum_{n=0}^1 x_{2n+1} W_2^{nm} \quad (109)$$

Каждая из сумм есть преобразование Фурье двухточечного сигнала. Выпишем, чему равны W_N .

$$\begin{aligned} W_2^{0:0} = W_2^{0:1} = W_2^{1:0} = 1, \quad W_2^{1:1} = e^{-\pi i} = -1 \\ W_4^0 = 1, \quad W_4^1 = e^{-\frac{i\pi}{2}} = -i, \\ W_4^2 = -W_4^0 = -1, \quad W_4^3 = -W_4^1 = i \end{aligned} \quad (110)$$

Подставив (110) в (109), получим:

$$\begin{aligned} X_0 &= (x_0 + x_2) + (x_1 + x_3) \\ X_1 &= (x_0 - x_2) - i(x_1 - x_3) \\ X_2 &= (x_0 + x_2) - (x_1 + x_3) \\ X_3 &= (x_0 - x_2) + i(x_1 - x_3) \end{aligned} \quad (111)$$

Фактически формулы (111) – это те же самые формулы, что и (104). Отличие только в том, что слагаемые сгруппированы так, чтобы было видно, за счёт чего можно сократить количество операций. Заметим, что каждое выражение в скобках повторяется по два раза, каждое из которых достаточно вычислить один раз. Сравним выражения для X_0 и X_2 , а также для X_1 и X_3 . Эти пары выражений отличаются только знаком между скобками. На примере сигнала из четырёх отсчётов преимущество не так заметно, но для большего количества отсчётов за счёт свойства, описываемого выражением (108), можно сэкономить количество операций, вычислив только половину всех коэффициентов W_N^m перед второй суммой в (107). Перепишем формулы (111), введя новые обозначения, так чтобы при вычислении производилось минимум математических операций.

$$\begin{aligned} A_0 &= x_0 + x_2 & A_1 &= x_0 - x_2 \\ B_0 &= x_1 + x_3 & B_1 &= x_1 - x_3 \\ W_4^0 &= e^{-\frac{2\pi}{4} \cdot 0} & W_4^1 &= e^{-\frac{2\pi}{4} \cdot 1} \\ X_0 &= A_0 + W_4^0 \cdot B_0 & X_1 &= A_1 - W_4^1 \cdot B_1 \\ X_2 &= A_0 - W_4^0 \cdot B_0 & X_3 &= A_1 + W_4^1 \cdot B_1 \end{aligned} \quad (112)$$

Общее количество операций в (112) равно 12, если не считать для простоты взятие экспоненты. Для сравнения, вычисление дискретного преобразования Фурье как произведение матрицы на вектор по формуле (102) или (104), требует 28 операций (для каждого из 4-х X_m 4 операции умножения элемента вектора на элемент матрицы и 3 операции сложения $4 \cdot (4+3)=28$).

Обобщим результаты, полученные на примере сигнала из четырёх отсчётов, на сигнал, в котором количество отсчётов равно 2 в степени больше 2. В алгоритме быстрого преобразования Фурье можно выделить три основные операции:

1. Последовательное разделение сигнала на 2, по чётным и нечётным индексам до уровня двухточечных сигналов.
2. Двухточечное преобразование Фурье.
3. Последовательное объединение результатов.

Принцип разделения сигнала по чётным и нечётным номерам показан в (105)-(107). Отметим, что каждый из новых сигналов a_n и b_n в (107) также может быть разделён по тому же принципу на два сигнала. Это деление необходимо продолжать до тех пор, пока каждый из новых сигналов состоит больше чем из двух отсчётов.

Двухточечное преобразование Фурье выполняется быстрее всего, так как это просто сумма и разность отсчётов сигнала:

$$\begin{aligned} y_0 &= x_0 + x_1 \\ y_1 &= x_0 - x_1 \end{aligned} \quad (113)$$

Отметим, что операцию двухточечного преобразования Фурье необходимо выполнить 2^{p-1} раза, если сигнал состоит из 2^p отсчётов.

Операция последовательного объединения результатов на самом простом примере показана в (112). Обобщим её на случай с большим числом отсчётов. Для этого первую сумму в (107) заменим на A_m , вторую сумму – на B_m , также будем считать, что сигналы A и B состоят каждый из M отсчётов, а сигнал X состоит из $2M$ отсчётов. Тогда операция объединения результатов выражается формулами:

$$\begin{aligned} X_m &= A_m + W_{2M}^m B_m & m \in [0, M-1] \\ X_{m+M} &= A_m - W_{2M}^m B_m \end{aligned} \quad (114)$$

Покажем на примере сигнала из $N=8$ отсчётов последовательно описанные выше операции (рис. 32).

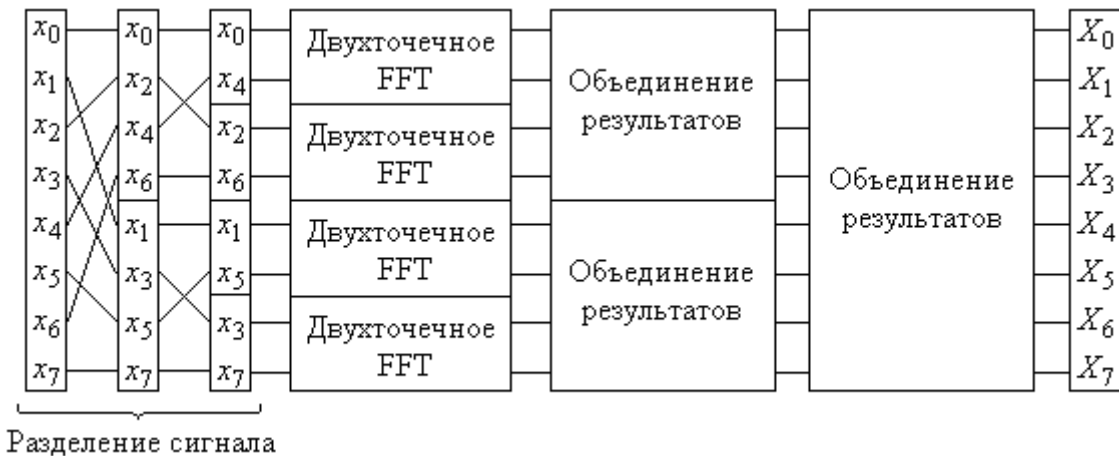


Рис. 32. Быстрое преобразование Фурье сигнала из 8 отсчётов.

Операции двухточечного преобразования Фурье производятся по формулам (113), операция объединения результатов – формулы (114). Первый этап, на котором из исходного сигнала перестановкой элементов получают новый сигнал, называется bit-reversal sorting. Это название произошло от того, что в отсортированном сигнале номера элементов массива, если записать их в двоичной форме, являются номерами исходных элементов, записанных в зеркальном отображении.

0	000	000	0
1	001	100	4
2	010	010	2
3	011	110	6
4	100	001	1
5	101	101	5
6	110	011	3
7	111	111	7

Существуют различные способы отсортировать элементы массива, основанные на операциях с отдельными битами целых чисел. Такая сортировка массива особенно эффективно может быть выполнена, если быстрое преобразование Фурье выполняется на аппаратном уровне.

В Matlab вычисление быстрого преобразования Фурье выполняется с помощью встроенной команды `fft`. Как было сказано выше, для выполнения быстрого преобразования Фурье необходимо, чтобы количество отсчётов в сигнале было представимо как произведение целых чисел, не обязательно одних только 2, поэтому быстрое Фурье может быть выполнено для сигнала с любым количеством отсчётов. Если количество отсчётов является простым числом, то есть делится нацело только на единицу и на самого себя, то сократить количество операций при

вычислении дискретного преобразования (102) невозможно. Команда `fft` использует алгоритм для произвольного количества отсчётов.

Пример.

В примере описывается, как вычислить быстрое преобразование Фурье, не используя встроенной функции, если количество сигналов равно степени двойки. В файле `my_fft.m` описаны две функции. Имя первой функции совпадает с именем файла, её можно вызывать из других программ, расположенных в рабочей папке. Вторая функция, описанная в этом же файле, является вспомогательной, её можно вызывать только в пределах этого файла. В этом примере не будут рассматриваться вопросы, касающиеся осуществления предварительной сортировки элементов массива с отсчётами сигнала, ограничимся лишь разделением массива на два по чётным и нечётным номерам, как это сделано в формуле (107). Как было отмечено выше, если в формуле (107) число M также является степенью 2, то каждую из двух сумм в этой формуле можно разбить на две по чётным и нечётным номерам. Так же было отмечено, что каждая из этих сумм является дискретным преобразованием Фурье от сигнала, состоящего из половины отсчётов исходного сигнала. Это наталкивает на мысль использовать в качестве формулы для вычисления быстрого преобразования Фурье формулу (107), а для вычисления сумм в этой формуле также использовать (107), так как это тоже преобразования Фурье и M есть степень двойки, так как N тоже степень двойки и $N = 2M$. Вторая функция, описанная в этом же файле, будет выполнять операцию объединения результатов по формулам (114).

Вычисление быстрого преобразования Фурье (`my_fft.m`)

```
function y = my_fft(x)
N = length(x); % количество отсчётов
if N > 2
% Если N>2 то каждую из сумм в (107) можно разбить ещё на две
% строка ниже это формула (107), в которой каждая из сумм
% вычисляется также с помощью формулы (107), а результаты
% объединяются по формулам (114) с помощью команды
fft_combine,
% описанной ниже.
    y = fft_combine( my_fft( x(1:2:N-1) ), my_fft( x(2:2:N) )
);
else
% Если сигнал состоит из двух отсчётов, то выполняется
% двухточечное преобразование Фурье по формулам (113), на этом
% рекуррентный вызов функции my_fft заканчивается.
    y(1) = x(1) + x(2);
    y(2) = x(1) - x(2);
end
```

```

% Объединение результатов по формулам (114)
function y = fft_combine(x1, x2)
N = length(x1);
W = exp(-pi*i/N);
for m = 1:N
    y(m) = x1(m) + W^(m-1) * x2(m);
    y(N+m) = x1(m) - W^(m-1) * x2(m);
end

```

Сравним работу функции `my_fft` с функцией `fft` на примере с последовательностью прямоугольных импульсов. В командном окне задаём прямоугольный импульс, состоящий из 4 отсчётов, вся запись длительностью 32 отсчёта.

Сравнение `my_fft` и `fft`

```

% задаём импульс
u(1:4) = 1;    u(5:32) = 0;
% вычисляем преобразование Фурье
sp1 = my_fft(u);
sp2 = fft(u);
% сравниваем результаты на графиках
figure
subplot(2,2,1), stem(abs(sp1))
subplot(2,2,2), stem(angle(sp1))
subplot(2,2,3), stem(abs(sp2))
subplot(2,2,4), stem(angle(sp2))

```

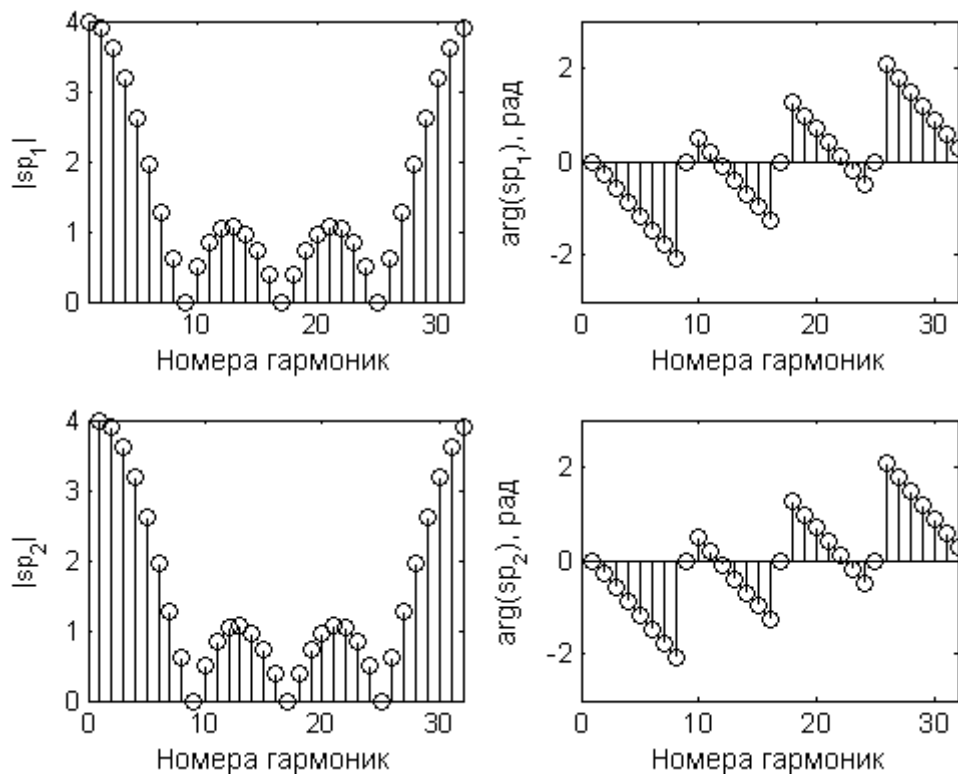


Рис. 33. Сравнение результатов `my_fft` (сверху) и `fft` (снизу).

В результате получим четыре графика (рис. 33). Два верхних графика – это результат, полученный с помощью `my_fft`, нижние – с помощью `fft`.

Данный пример вычисления быстрого преобразования Фурье не оптимален с точки зрения расхода памяти и, возможно, скорости выполнения. Эти два критерия зависят от того, на каком языке и под какую платформу создана программа. Операция вызова функции, как правило, связана с выделением определённой области памяти, которая освобождается после выхода из функции. При рекуррентном вызове функции происходит многократное выделение памяти. Другая причина большого расхода памяти заключается в том, что при вызове функции параметры передаются в неё путём копирования. В других языках программирования (C++, Pascal) есть возможность передавать в функцию лишь только адрес памяти, где хранятся элементы вектора (массива), избегая многократного копирования данных.

Разберём подробнее, как в Matlab вычисляется быстрое преобразование Фурье. Если набрать в командном окне `doc fft`, откроется окно со справкой по команде `fft`. В справке приведены следующие формулы:

$$\begin{aligned} X(k) &= \sum_{j=1}^N x(j) W_N^{(j-1)(k-1)} \\ x(j) &= \frac{1}{N} \sum_{k=1}^N X(k) W_N^{-(j-1)(k-1)} \\ W_N &= e^{-\frac{2\pi i}{N}} \end{aligned} \quad (115)$$

Под обозначениями $x(j)$ и $X(k)$ понимается x_j и X_k соответственно. Нумерация элементов массивов в Matlab начинается с 1, поэтому в этих формулах в отличие от (102) и (103) индексы в суммах меняются от 1 до N , а степени экспоненты содержат -1. Теперь выясним каким частотам соответствуют значения X_k . В Matlab дискретный сигнал представляется в виде вектора (одномерного массива) элементы которого – это значения сигнала в определённые моменты времени $x_j = x(t_j)$, $t_j = (j-1)T_s$, $j = 1, 2, \dots, N$. В этом векторе хранится один период сигнала, состоящий из N отсчётов. То есть период сигнала T может быть записан как $T = NT_s$. Спектр периодического сигнала дискретный и частоты гармоник, с учётом того, что нумерация начинается с 1, будут $\omega_k = \frac{2\pi(k-1)}{T}$, $k = 1, 2, \dots, N$. Спектр же дискретного сигнала периодичен, и его период повторения равен

$\omega_s = \frac{2\pi}{T_s}$. Но обычно принято рассматривать спектр на симметричном отрезке частот $\left[-\frac{\omega_s}{2}, \frac{\omega_s}{2}\right]$. Так как спектр дискретного сигнала периодичен, то спектральные составляющие, расположенные на отрезке частот $\left[-\frac{\omega_s}{2}, 0\right]$ – такие же, как и в диапазоне $\left[\frac{\omega_s}{2}, \omega_s\right]$. Поэтому достаточно разделить вектор с комплексными амплитудами X_k посередине на две части и поменять их местами. Эту операцию выполняет команда `fftshift`. На рис. 34 показаны амплитуды гармоник, кружками отмечены гармоники, которые будут переставлены справа налево. Осталось выяснить, как будет разбиваться на две части вектор X_k командой `fftshift`, если количество элементов вектора N чётное и нечётное. Команда `fftshift` разбивает вектор на две части от 1 до `ceil(N/2)` и от `ceil(N/2)+1` до N и меняет их местами (`ceil` – округление в большую сторону). Существуют более сложные варианты вызова команды `fftshift` для многомерных массивов с указанием, по каким индексам их переворачивать, но принцип такой же, как и для одномерного массива. Тогда, чтобы получить номера частот ω_k для первой половины, достаточно вычесть 1, так как элементы массива нумеруются с 1, а частоты начинаются с 0.

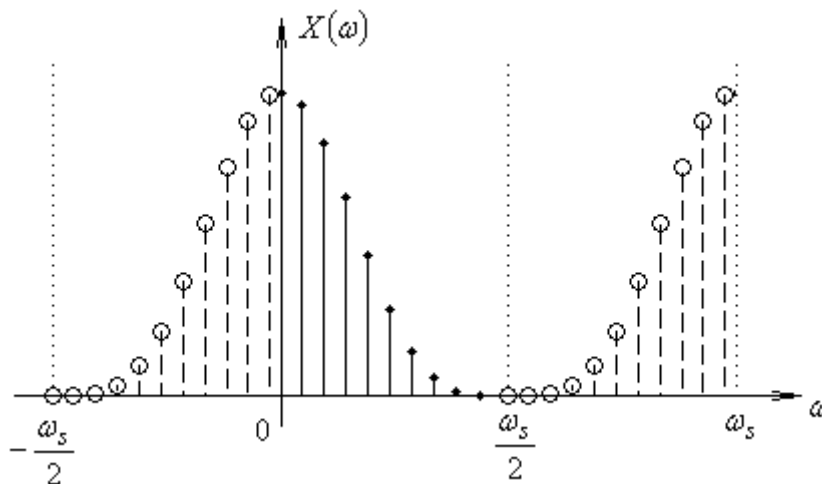


Рис. 34. Перестановка элементов массива.

Для второй половины надо также вычесть 1 и ещё N , так как вторую половину перенесли в отрицательные частоты, а спектр дискретного сигнала повторяется периодически каждые N гармоник. Тогда номера гармоник будут начинаться с `ceil(N/2)-N` и заканчиваться номером `ceil(N/2)-1`.

Чтобы из результата `fft` получить амплитуды гармоник необходимо разделить результат быстрого преобразования Фурье на количество отсчётов N . Сравним формулы прямого и обратного дискретного преобразования Фурье (102) и (103). В формуле обратного преобразования (103) перед знаком суммы стоит множитель $\frac{1}{N}$, поэтому амплитудой гармоники с номером m будет $\frac{X_k}{N}$, то есть всё, на что умножается экспонента.

Пример.

В этом примере демонстрируется применение функции `fft` для вычисления спектра прямоугольных импульсов с периодом повторения 0,02с и скважностью 10%. На периоде импульсов находится 100 отсчётов, соответственно длительность самого импульса 10 отсчётов. Сначала с помощью `fft` вычисляется быстрое преобразование Фурье, потом применяется команда `X1 = fftshift(X)`, чтобы переставить элементы массива `X`. Частота измеряется в герцах, поэтому нет множителя 2π . Значения частот при построении графика «перевернутого» спектра `X1` вычисляются с помощью команды `f1 = (ceil(N/2)-N:ceil(N/2)-1)/T`.

Пример быстрого преобразования Фурье (`ifft_example.m`)

```
clear, clc, close all

N = 100; % количество отсчётов
T = 0.02; % период сигнала
x(1:10)=1; x(11:N)=0; % прямоугольный импульс 0.1 периода

X = fft(x); % быстрое преобразование Фурье
f = (0:N-1)/T; % частоты соответствующие гармоникам

X1 = fftshift(X); % переставляем местами половинки спектра
% частоты соответствующие гармоникам
f1 = (ceil(N/2)-N:ceil(N/2)-1)/T;

% результат fft
subplot(3,1,1), stem(abs(X),'k.')
xlabel('Номера гармоник', 'FontName','Arial Cyr')
% амплитудный спектр как функцию частоты
subplot(3,1,2), stem(f,abs(X)/N,'k.')
xlabel('f, Гц', 'FontName','Arial Cyr')
% амплитудный спектр с переставленными гармониками
subplot(3,1,3), stem(f1,abs(X1)/N,'k.')
xlabel('f, Гц', 'FontName','Arial Cyr')
```

В результате будут построены три графика (рис. 35). На первом графике приведен результат, который возвращает функция `fft`, взятый по модулю, построенный относительно номеров элементов вектора. На втором графике построен амплитудный спектр как функция частоты. Этот график отличается от первого тем, что все значения X_k уменьшены в N раз. На втором графике видно, что среднее значения прямоугольных импульсов с скважностью 10% за один период равно 0,1. На третьем графике приведен амплитудный спектр, построенный на отрезке $\left[-\frac{f_s}{2}, \frac{f_s}{2}\right]$, частота квантования $f_s = \frac{N}{T}$.

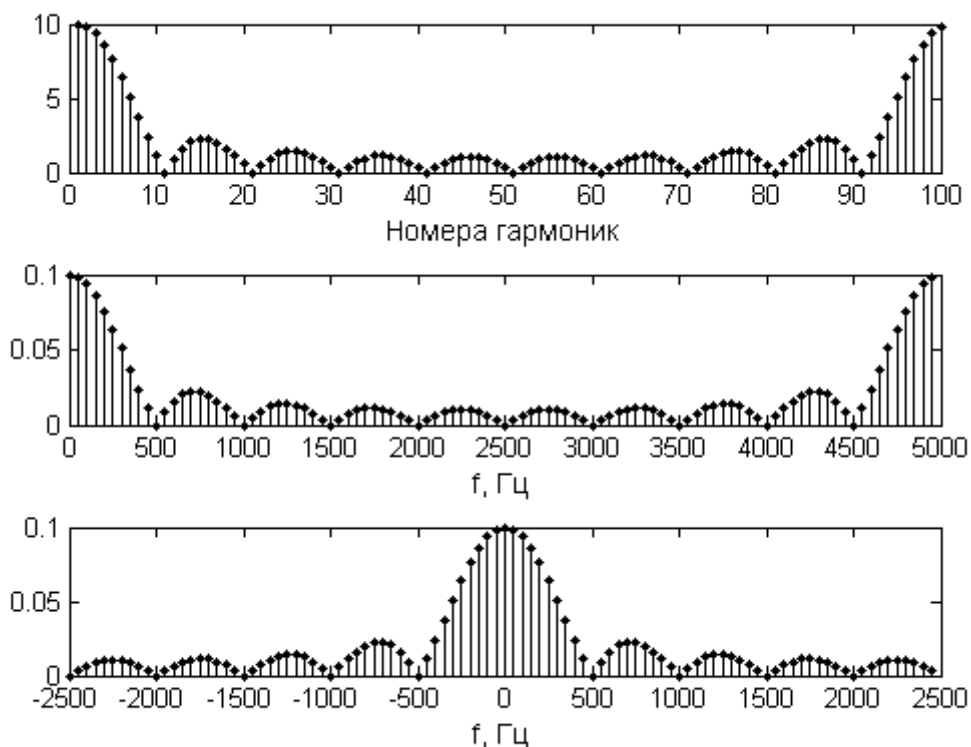


Рис. 35. Пример использования `fft`.

Обратное преобразование Фурье вычисляется с помощью команды `ifft`.

Пример

В этом примере демонстрируется использование дискретного обратного преобразования Фурье. В рассмотренном примере, который сопровождался программой `example_3.m`, использовался пилообразный импульс, который может быть получен как свёртка прямоугольного импульса самим с собой. Спектр пилообразного импульса по теореме о свёртке равен спектру прямоугольного импульса, возведённого в квадрат. В примере сначала с помощью `fft` получается спектр прямоугольного

импульса, потом, используя свойства преобразования Фурье, получается спектр пилообразного импульса $X_1 = X.^2$, из которого с помощью `ifft` синтезируется пилообразный импульс. Для сравнения этот же импульс получается из прямоугольного с помощью свёртки $x_2 = \text{conv}(x, x)$.

Пример обратного преобразования Фурье (`ifft_example.m`)

```
clear, clc, close all

N = 50; % количество отсчётов
M = 10; % длительность прямоугольного импульса в отсчётах
T = 0.01; % период повторения импульсов
t = (0:(N-1))*T/N; % время
% прямоугольный импульс
x(1:N) = 0; x(5:5+M) = 1;

X = fft(x); % спектр прямоугольного импульса
X1 = X.^2; % спектр пилообразного импульса
x1 = ifft(X1); % обратное преобразование Фурье
% для сравнения получаем тот же импульс с помощью свёртки
x2 = conv(x,x); % свёртка
x2 = x2(1:N); % чтобы было N отсчётов

subplot(2,1,1), plot(t,x,'k'), xlabel('t'), ylabel('x(t)')
subplot(2,1,2), plot(t,x1,'k',t,x2,'ko')
xlabel('t'), ylabel('x_1(t), x_2(t)'),
legend('x_1(t)', 'x_2(t)')
```

На верхнем графике (рис. 36) показан прямоугольный импульс, на нижнем графике синтезированный треугольный импульс двумя способами, сплошной линией показан результат `ifft`, кружками – свёртки.

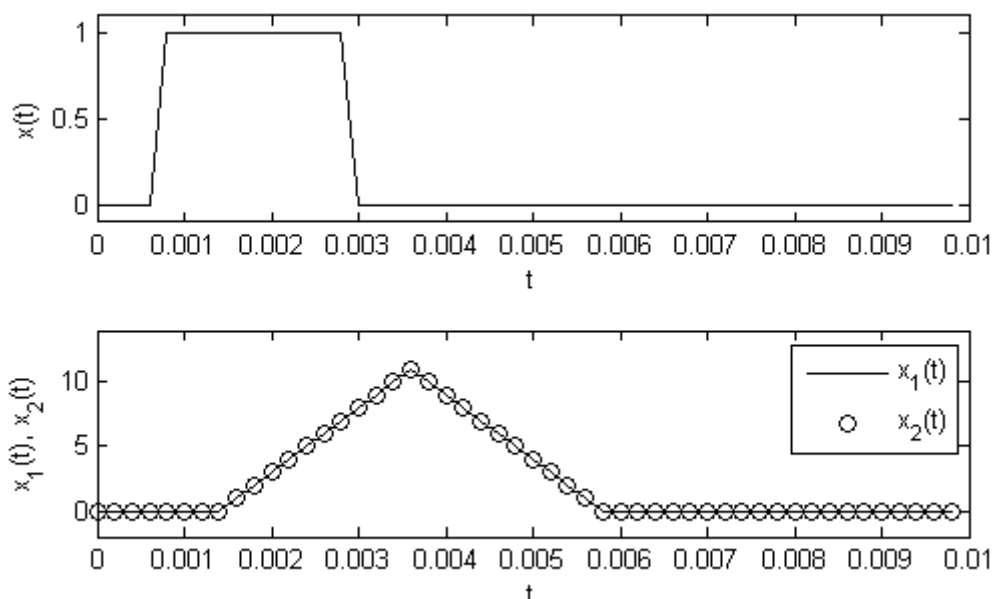


Рис. 36. Синтез пилообразного импульса.

В документации Matlab описано несколько вариантов вызова функции `fft`:

- `fft(x)` выполняет дискретное преобразование Фурье, вычисляя сумму (102) с помощью алгоритма быстрого преобразования Фурье.
- `fft(x,n)` выполняет дискретное преобразование Фурье, используя n отсчётов из вектора x . Если вектор x состоит из меньшего числа отсчётов, то в конец сигнала добавляется необходимое количество нулевых отсчётов.
- `fft(x,[],dim)` выполняет дискретное преобразование Фурье многомерного массива x , параметр `dim` указывает номер индекса массива по которому выполняется преобразование. Такой вариант вызова функции может быть полезен, когда в одной переменной хранятся сразу несколько дискретных сигналов. По умолчанию в Matlab считается, что сигналы находятся в столбцах матрицы, то есть первый индекс двумерного массива чисел – это номер отсчёта, второй индекс – номер сигнала. Если задать параметр `dim` равным 2, то второй индекс массива x будет означать номер отсчёта, а первый – номер сигнала.
- `fft(x,n,dim)` этот вариант объединяет в себе возможности двух предыдущих.

Пример

С помощью программы `specters.m` можно построить спектры некоторых сигналов (прямоугольного, треугольного, трапеции, отрезок синуса), а также спектр сигнала, заданного пользователем. Для задания параметров сигнала и изучения спектров пользователю предоставляется удобный графический интерфейс (рис. 37). В зависимости от выбранного сигнала в рамке **Выбор сигнала** будет отображаться набор соответствующих этому сигналу параметров. Для последнего варианта, в котором пользователь сам задаёт сигнал, появляется поле для ввода формулы, при этом время в формуле обозначается переменной t . Параметр **Частота квантования** определяет количество отсчётов на период сигнала, и соответственно количество гармоник в спектре. На графиках амплитудного и фазового спектра (второй и третий график) по горизонтальной оси откладываются номера гармоник. В программе можно выполнить синтез сигнала с помощью обратного преобразования Фурье. Количество гармоник, используемых для синтеза сигнала, задаётся в поле **Число учитываемых гармоник**.

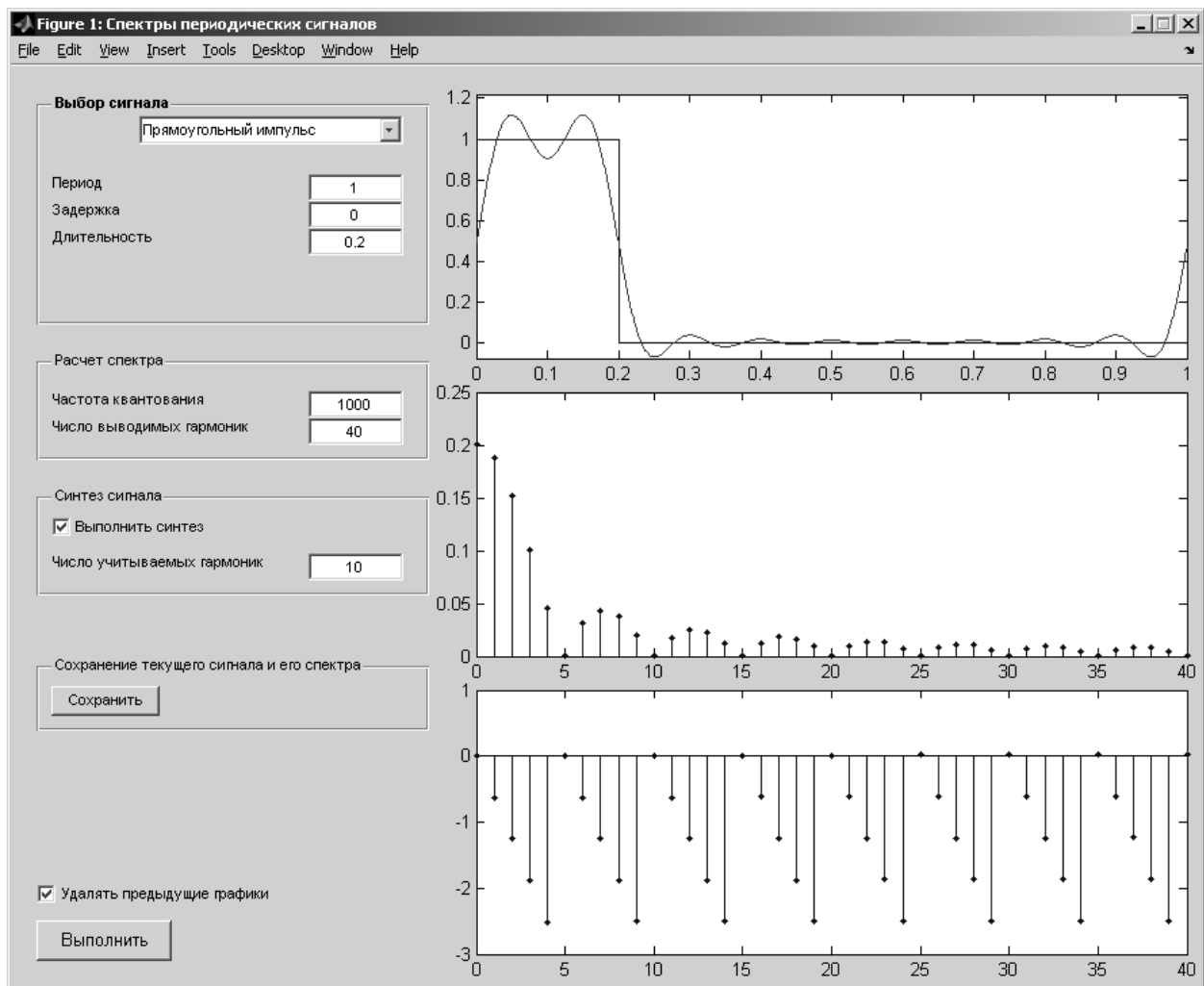


Рис. 37. Графический интерфейс программы specters.

5.3. Увеличение разрешения по частоте

Среди перечисленных выше вариантов вызова функции `fft` был вариант, в котором вторым аргументом можно задать количество отсчётов, используемых для быстрого преобразования Фурье. Если заданное количество отсчётов больше, чем количество отсчётов сигнала, то вектор с отсчётами будет дополнен нулевыми отсчётами до указанного количества. При этом получается, что период сигнала T увеличивается, за счёт увеличения периода сигнала уменьшается расстояние по частоте между гармониками $\omega_k = \frac{2\pi k}{T}$ спектра, а форма огибающей спектра остаётся прежней, так как форма самого сигнала не изменилась. То есть за счёт увеличения количества отсчётов можно увеличить разрешение по частоте.

Пример увеличения разрешения по частоте (fft_length.m)

```
clear, clc, close all

Fs = 1000; % частота квантования
x(1:5)=1; x(6:25)=0; % прямоугольный импульс

X1 = fft(x); % спектр исходного сигнала (30 отсчётов)
% спектр сигнала, дополненного нулями (100 отсчётов)
X2 = fft(x,100);

% X1 и X2 имеют разную длину
N1 = length(X1);
N2 = length(X2);
% частоты, соответствующие элементам X1 и X2
f1 = Fs * (ceil(N1/2)-N1:ceil(N1/2)-1) / N1;
f2 = Fs * (ceil(N2/2)-N2:ceil(N2/2)-1) / N2;

% вывод амплитудных спектров на графиках
subplot(2,1,1), stem(f1,abs(fftshift(X1))/N1,'k.')
xlabel('f'), ylabel('|X_1|')
subplot(2,1,2), stem(f2,abs(fftshift(X2))/N2,'k.')
xlabel('f'), ylabel('|X_2|')
```

Программе построит два графика (рис. 38). На верхнем графике построен амплитудный спектр прямоугольных импульсов длительностью 5 отсчётов и периодом 25 отсчётов, частота квантования 1000Гц, расстояние между гармониками 40Гц. На нижнем графике приведён амплитудный спектр того же сигнала, но после того как к нему в конец добавили нулевые отсчёты так, чтобы получившееся количество отсчётов было в 4 раза больше чем у исходного сигнала, то есть 100 отсчётов. В результате получается, что у нового сигнала точно такая же огибающая спектра, но расстояние между гармониками стало в 4 раза меньше 10Гц.

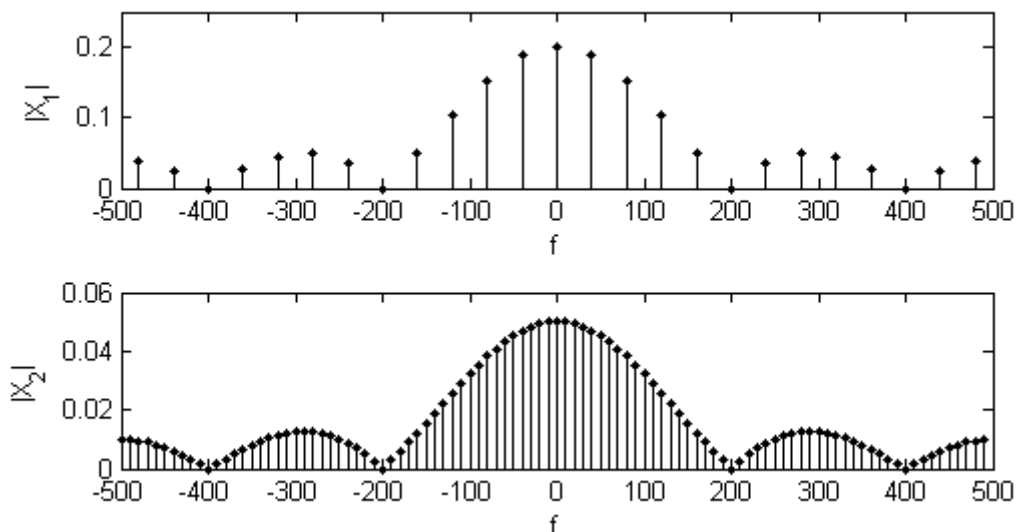


Рис. 38. Увеличение разрешения по частоте.

В пределе при увеличении количества отсчётов до бесконечности из дискретного спектра получится непрерывный спектр соответствующий дискретному неперидическому сигналу.

Пример

С помощью программы `zeros_add.m` можно поэкспериментировать с увеличением числа отсчётов и посмотреть, как при этом будут меняться спектры сигналов. Экспериментировать можно с сигналами двух типов: прямоугольным импульсом, длительностью 0,1с и периодом повторения 0,5с, и гармоническим сигналом с частотой 10Гц. В случае гармонического сигнала, добавление нулевых отсчётов превращает его в радиоимпульс длительностью 0,5с и частотой заполнения 10Гц.

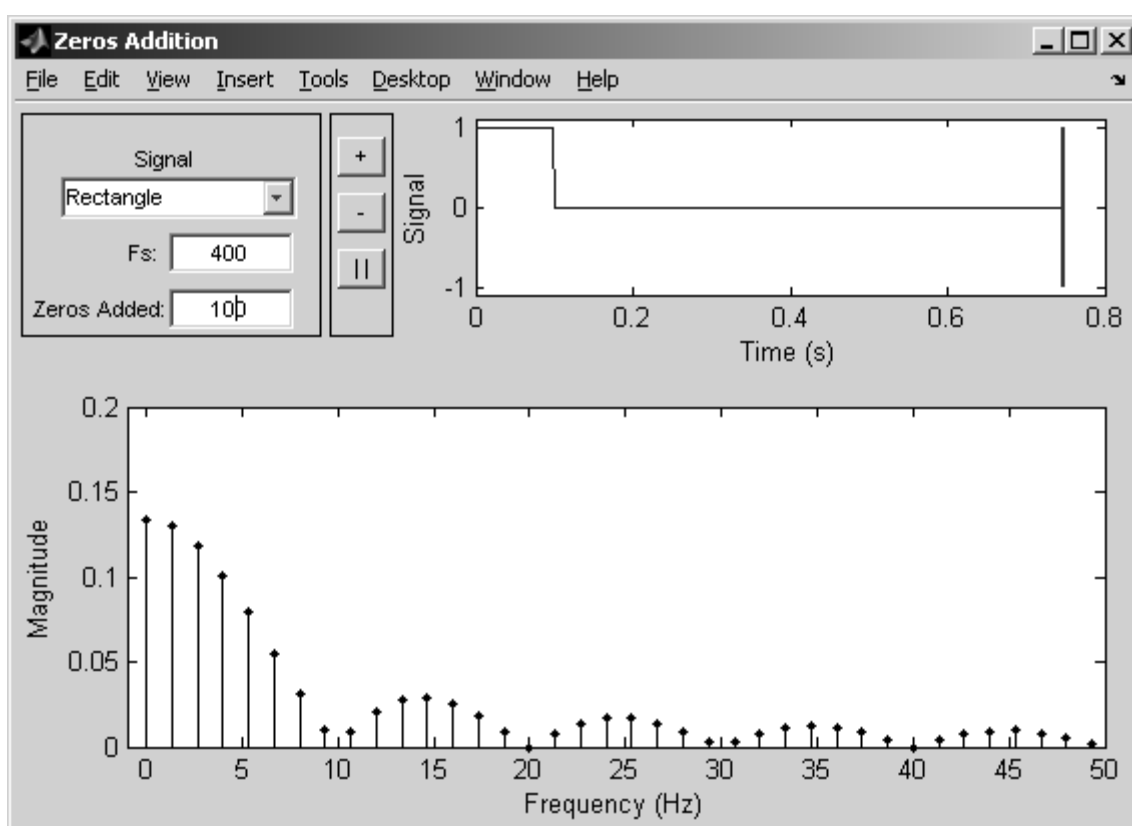


Рис. 39. Окно программы `zeros_add.m`.

5.4. Растекание спектра

Предположим, что с помощью АЦП регистрируется периодический сигнал. Обычно длительность записанного сигнала не равна периоду сигнала или целому количеству периодов сигнала. Какие изменения произойдут в спектре сигнала, если выполнить дискретное преобразование Фурье записанного сигнала, длительность которого не равна целому числу периодов исходного сигнала?

Для упрощения вывода формул будем рассматривать не дискретный периодический сигнал, а непрерывный периодический. Это упрощение не приводит к качественным изменениям, так как спектр дискретного сигнала получается из спектра непрерывного сигнала согласно с (78). Если выполнено условие теоремы Котельникова, перекрывания спектра нет, то вместо спектра дискретного сигнала достаточно рассмотреть спектр непрерывного сигнала.

Периодический сигнал имеет дискретный спектр, то есть он может быть представлен как линейная комбинация гармонических функций. Поэтому достаточно рассмотреть, какие изменения в спектре произойдут в спектре одной гармонической функции. В начале главы про дискретное преобразование Фурье говорилось, что сигнал x_n – это ровно один период сигнала. То есть в данном случае зарегистрированный отрезок периодического сигнала должен быть периодически продолжен, при этом периодом повторения должна быть длительность всей записи сигнала. Если длительность записи отлична от периода сигнала, который записывали, то при периодическом повторении записи сигнала произойдёт искажение формы сигнала, соответственно и его спектра.

Например, регистрировался синусоидальный сигнал $\sin\left(\frac{2\pi t}{T}\right)$ с периодом T , а длительность записи равна τ , причём $\tau \neq mT$, где m – целое число. Тогда при периодическом повторении записи сигнала (рис. 40) появятся разрывы первого рода, так как значения сигнала в начале и конце записи разные.

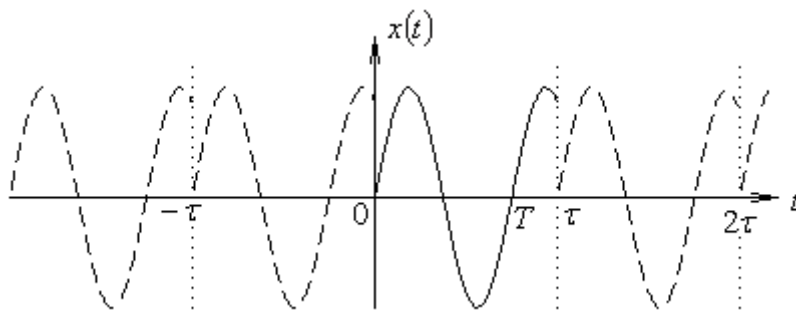


Рис. 40. Периодически повторяющийся отрезок сигнала.

На рис. 40 сплошной линией показан отрезок записанного сигнала, а штрихованной линией его периодическое повторение. Период повторения равен τ . Найдём коэффициенты разложения в тригонометрический ряд сигнала $x(t)$, показанного на рис. 40.

$$c_k = \frac{1}{\tau} \int_0^{\tau} \sin\left(\frac{2\pi t}{T}\right) e^{-i\frac{2\pi k t}{\tau}} dt \quad (116)$$

Вычисление этого интеграла даёт следующее выражение для коэффициентов:

$$c_k = \frac{T}{2\pi} \left[\frac{\tau}{\tau^2 - k^2 T^2} \left(1 - \cos\left(\frac{2\pi\tau}{T}\right) \right) - i \frac{kT}{\tau^2 - k^2 T^2} \sin\left(\frac{2\pi\tau}{T}\right) \right] \quad (117)$$

Для произвольного τ эти коэффициенты отличны от нуля при любом номере k , то есть сигнал на рис. 40 гармоническим не является. Изменения в спектре сигнала, вызванные нецелым количеством периодов в записи сигнала, показаны на рис 43. Теперь посмотрим, как изменятся коэффициенты при $\tau = mT$, где m – целое число. Если $k \neq \pm m$, то $c_k = 0$, так как $\cos(2\pi m) = 1$ и $\sin(2\pi m) = 0$. А если $k = \pm m$, то в ноль обращается и знаменатель, поэтому возникает неопределённость типа $0/0$, которую можно раскрыть по правилу Лопиталья. В результате при $\tau = mT$ для коэффициентов разложения получим:

$$c_k = \begin{cases} 0, & k \neq \pm m \\ -\frac{i}{2}, & k = m \\ \frac{i}{2}, & k = -m \end{cases} \quad (118)$$

Эти коэффициенты будут соответствовать гармоническому сигналу. Если периодом считать длину записи τ , то частоты гармоник в спектре $\omega_k = \frac{2\pi}{\tau} k$. Разложение сигнала $x(t)$ в тригонометрический ряд в случае $\tau = mT$:

$$x(t) = -\frac{i}{2} e^{i\frac{2\pi}{\tau} mt} + \frac{i}{2} e^{-i\frac{2\pi}{\tau} mt} = \frac{e^{i\frac{2\pi t}{T}} - e^{-i\frac{2\pi t}{T}}}{2i} = \sin\left(\frac{2\pi t}{T}\right) \quad (119)$$

Также можно дать другой вариант интерпретации растекания спектра. Ограниченную по времени запись сигнала можно представить как исходный сигнал, умноженный на одиночный прямоугольный импульс с амплитудой, равной единице, и длительностью, равной длительности записанного сигнала. Исходный сигнал периодический, поэтому его можно разложить в тригонометрический ряд. Рассмотрим, как изменится спектр одной гармоники в зависимости от соотношения периода гармоники и длительности прямоугольного импульса. Также для упрощения вывода

формулу воспользуемся тем, что спектр дискретного сигнала выражается через спектр соответствующего непрерывного сигнала (78). При вычислении дискретного преобразования Фурье считается, что сигнал периодический, а в данном случае рассматриваются спектр сигнала, умноженного на одиночный прямоугольный импульс, длительность которого равна длине записи, и является одновременно периодом повторения при вычислении дискретного преобразования Фурье. Поэтому для связи спектра непериодического сигнала и периодического сигнала используются соотношения (35). Теперь рассмотрим, как изменяется спектр на конкретном примере. Предположим, регистрировался синусоидальный сигнал с частотой f_0 .

$$x(t) = \sin(\omega_0 t) \quad \omega_0 = 2\pi f_0 \quad T = \frac{1}{f_0} \quad (120)$$

Длительность записи равна τ , поэтому записанный сигнал можно интерпретировать как сигнал $x(t)$, умноженный на прямоугольный импульс единичной амплитуды длительностью τ .

$$\begin{aligned} x_1(t) &= w_1(t)x(t) = w_1(t)\sin(\omega_0 t) \\ w_1(t) &= \begin{cases} 1, & t \in [0, \tau] \\ 0, & t \notin [0, \tau] \end{cases} \end{aligned} \quad (121)$$

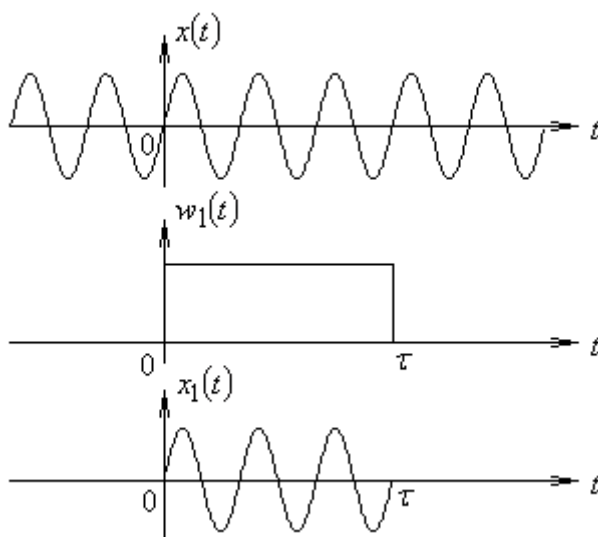


Рис. 41. Интерпретация записи отрывка сигнала.

Функцию $w_1(t)$ называют оконной функцией, в данном случае она является прямоугольным импульсом, а сигнал $x_1(t)$ может быть рассмотрен как радиоимпульс. Более сложный случай, когда $x(t)$ произвольная функция, будет рассмотрен в следующей главе. Спектр прямоугольного импульса:

$$W_1(\omega) = \frac{2}{\omega} \sin\left(\frac{\omega\tau}{2}\right) \cdot e^{-i\frac{\omega\tau}{2}} \quad (122)$$

С помощью свойств преобразования Фурье получаем спектр сигнала $x_1(t)$ как спектр радиоимпульса (49).

$$X_1(\omega) = \frac{1}{2i} [W_1(\omega - \omega_0) - W_1(\omega + \omega_0)] \quad (123)$$

Связь спектра одиночного импульса $x_1(t)$ с дискретным спектром периодически повторяющегося с периодом повторения τ определяется формулой (35), где вместо T надо поставить τ . Если проделать эти преобразования, то для $c_k = X_1(\omega_k)$, $\omega_k = \frac{2\pi k}{\tau}$ получатся выражение (117).

Чтобы уменьшить эффект растекания спектра, в качестве оконной функции надо выбрать функцию, которая на концах отрезка $[0, \tau]$ равна нулю. Тогда при периодическом повторении записанного отрезка сигнала, умноженного на оконную функцию, разрывов возникать не будет. Но в отличие от прямоугольного окна, будет искажена форма сигнала, что приведёт к искажению в спектре.

Пример

Вычислим с помощью дискретного преобразования Фурье спектр отрезков синусоиды. Рассмотрим два случая, когда запись состоит из целого числа периодов синуса в первом случае и нецелого – во втором. Сравним спектры, полученные с использованием прямоугольного (рис. 41) и треугольного окна (рис. 42).

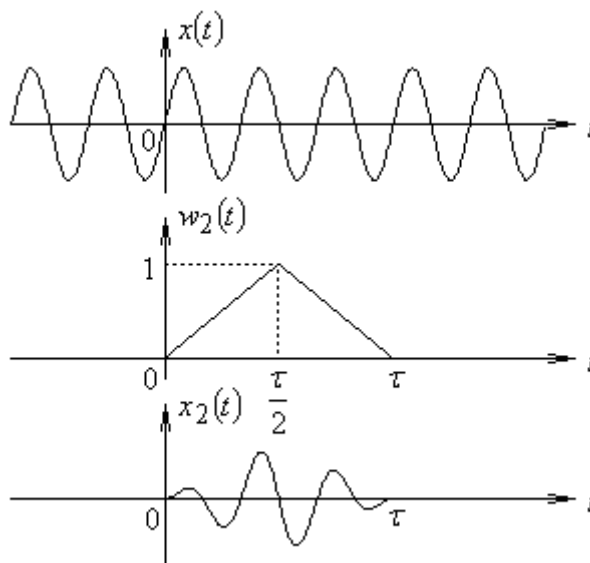


Рис. 42. Применение треугольного окна.

В данном примере выбор треугольного окна объясняется тем, что для него легко получить формулы с помощью свойств преобразования Фурье. В примере в качестве сигнала используется синусоидальный сигнал с частотой 10Гц. В первом случае длительность записи равна трём периодам синуса, во втором случае трём и одной четверти периода. Сначала спектры вычисляются с использованием прямоугольного окна, а потом для сравнения производятся вычисления с треугольным окном. Спектр отрезка синусоиды, умноженной на прямоугольное окно, уже был получен (123), чтобы его можно было наложить на график спектра полученного с помощью `fft`, надо спектр непрерывного одиночного импульса разделить на период повторения, а спектр дискретного периодического сигнала на количество отсчётов. Спектр отрезка синусоиды, умноженной на треугольное окно, может быть получен из спектра прямоугольного импульса с помощью свойств преобразования Фурье. Как было показано выше, треугольный импульс может быть представлен как свёртка прямоугольного импульса с самим собой. Треугольный импульс также надо будет разделить на τ , чтобы амплитуда была равна 1, и изменить масштаб по времени в два раза, так как длительность треугольного импульса, полученного с помощью свёртки, будет в два раза больше длительности прямоугольного импульса.

$$w_2(t) = \frac{1}{\tau} \int_{-\infty}^{+\infty} w_1(\xi) w_1(2t - \xi) d\xi \quad (124)$$

$$x_2(t) = w_2(t) x(t) = w_2(t) \sin(\omega_0 t)$$

Спектр гармонического сигнала, умноженного на треугольное окно:

$$X_2(\omega) = \frac{\left[W_1\left(\frac{\omega - \omega_0}{2}\right) \right]^2 - \left[W_1\left(\frac{\omega + \omega_0}{2}\right) \right]^2}{4i\tau} \quad (125)$$

В примере спектры непрерывных сигналов задаются с помощью безымянных функций. X_v – спектр прямоугольного импульса. X_r – спектр отрезка синусоиды. X_t – спектр синуса, умноженного на треугольное окно.

Пример растекания спектра (`width_fft.m`)

```
clear, clc, close all
```

```
% спектр прямоугольного сигнала (видеоимпульса)
Xv = @(w, tau) 2*sin(tau*w/2)./w.*exp(-i*tau*w/2);
% используя Xv и свойства преобразования Фурье задаём спектры
% прямоугольного окна, умноженного на sin(w0*t)
```

```

Xr = @(w, w0, tau) (Xv(w-w0,tau)-Xv(w+w0,tau))/2/i;
% треугольного окна, умноженного на sin(w0*t)
Xt = @(w, w0, tau) ...
      (Xv((w-w0)/2,tau).^2-Xv((w+w0)/2,tau).^2)/4/tau/i;

f0 = 10; % частота гармонического сигнала
w0 = 2*pi*f0;
Fs = 80; % частота квантования
tau1 = 3/f0; % длительность записи в первом случае
tau2 = 3.25/f0; % длительность записи во втором случае
t1 = 0:1/Fs:tau1-1/Fs; % время
t2 = 0:1/Fs:tau2-1/Fs;
x1 = sin(2*pi*f0*t1); % отрезок сигнала ровно 3 периода
x2 = sin(2*pi*f0*t2); % отрезок сигнала ровно 3.25 периода
X1 = fft(x1); % дискретное преобразование Фурье
X2 = fft(x2);
N1 = length(X1); % количество отсчётов в первом случае
N2 = length(X2); % количество отсчётов во втором случае
% частоты, соответствующие гармоникам в первом и втором случае
f1 = (ceil(N1/2)-N1:ceil(N1/2)-1)*Fs/N1;
f2 = (ceil(N2/2)-N2:ceil(N2/2)-1)*Fs/N2;
fa = -Fs/2:0.1:Fs/2; % частоты, чтобы строить огибающую спектра

% построение графиков
% верхний график - спектр отрезка из 3 периодов синуса
% нижний график - спектр отрезка из 3.25 периодов синуса
figure
subplot(2,1,1)
stem(f1,abs(fftshift(X1))/N1,'k')
hold on
plot(fa, abs(Xr(2*pi*fa,w0,tau1))/tau1, 'k:')
hold off
axis([-20 20 0 0.55]), xlabel('f'), ylabel('|X_1|')
subplot(2,1,2)
stem(f2,abs(fftshift(X2))/N2,'k')
hold on
plot(fa, abs(Xr(2*pi*fa,w0,tau2))/tau2, 'k:')
hold off
axis([-20 20 0 0.55]), xlabel('f'), ylabel('|X_2|')

% Для сравнения вычисляем спектр, используя
% треугольные оконные функции
w1 = triang(N1); % вектор столбец из N1 элементов
w2 = triang(N2);
% перед вычислением fft умножаем сигнал на оконную функцию
Xw1 = fft(x1.*w1');
Xw2 = fft(x2.*w2');

% построение спектров сигналов, умноженных на оконную функцию
% верхний график - спектр отрезка из 3 периодов синуса
% нижний график - спектр отрезка из 3.25 периодов синуса

```

```

figure
subplot(2,1,1)
stem(f1,abs(fftshift(Xw1))/N1,'k')
hold on
plot(fa, abs(Xt(2*pi*fa,w0,tau1))/tau1, 'k:')
hold off
axis([-40 40 0 0.3]), xlabel('f'), ylabel('|Xw_1|')
subplot(2,1,2)
stem(f2,abs(fftshift(Xw2))/N2,'k')
hold on
plot(fa, abs(Xt(2*pi*fa,w0,tau2))/tau2, 'k:')
hold off
axis([-40 40 0 0.3]), xlabel('f'), ylabel('|Xw_2|')

```

В результате программа построит две пары графиков (рис. 43) и (рис. 44), на которых изображены амплитудные спектры. Из графиков видно, что в случае целого количества периодов «лишние» гармоники располагаются в точках, где амплитудный спектр соответствующего непрерывного сигнала равен нулю. Когда нецелое количество периодов, то эти гармоники могут быть там, где амплитудный спектр не равен нулю. Когда сигнал умножается на оконную функцию не прямоугольной формы, то размывание спектральных линий, которых должно быть только две, как на самом первом графике, объясняется искажением формы сигнала.

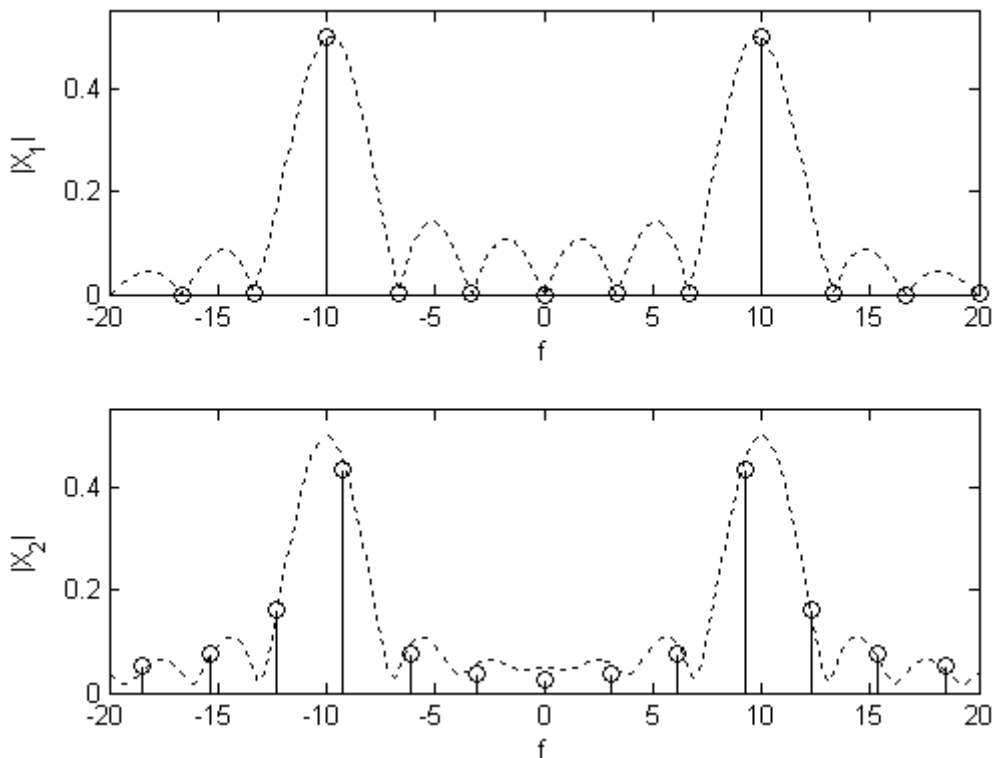


Рис. 43. Амплитудные спектры отрезков синусоиды, целое число периодов (сверху), нецелое (снизу).

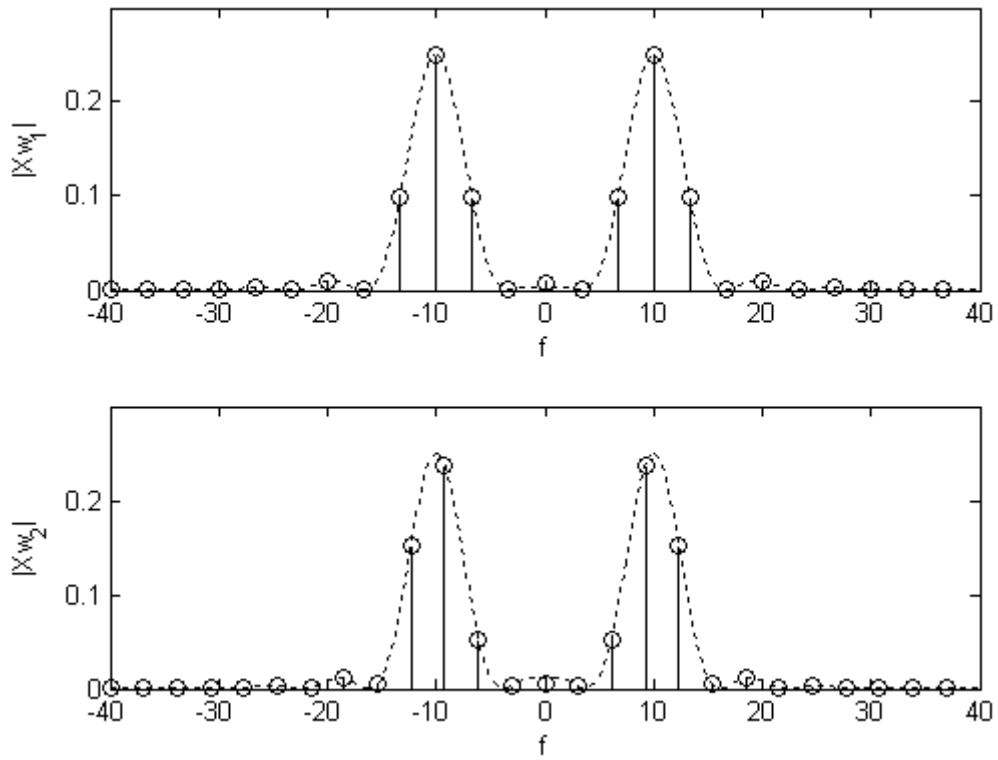


Рис. 44. Амплитудные спектры отрезков синусоиды, умноженных на треугольное окно, целое число периодов (сверху), нецелое (снизу).

6. Оконное преобразование Фурье

6.1. Вычисление оконного преобразования Фурье в Matlab

Преобразование Фурье даёт представление о сигнале в частотной области, то есть представляет в виде суммы гармонических колебаний. С помощью преобразования Фурье можно получить амплитуда и начальную фазу гармонического колебания в зависимости от частоты. Но частота гармонического колебания не меняется во времени, и сами колебания периодические и не имеют ни начала, ни конца. Поэтому с помощью преобразования Фурье нельзя получить информацию о каких-либо особенностях сигнала, локализованных во времени. Чтобы получить такую информацию, можно выделить с помощью оконной функции $w(t-\tau)$ интересующий отрезок сигнала, а потом вычислить преобразование Фурье для этого отрезка. Таким образом, получается два параметра, один – τ задаёт положение окна, а другой – ω определяет частоту гармоник, ширина окна остаётся постоянной. Оконное преобразование Фурье определяется следующим выражением:

$$X(\tau, \omega) = \int_{-\infty}^{+\infty} w(t-\tau)x(t)e^{-i\omega t} dt \quad (126)$$

Дискретное оконное преобразование Фурье выполняется похожим образом. Из вектора с отсчётами сигнала выбираются сегменты, которые могут быть перекрывающимися, каждый сегмент сигнала умножается на оконную функцию, которая также задана отсчётами, и выполняется дискретное преобразование Фурье каждого сегмента. В библиотеке **Signal Processing** есть специальная функция **spectrogram** для вычисления дискретного оконного преобразования Фурье. У этой функции существует много различных вариантов набора параметров. Рассмотрим наиболее общий вариант.

```
[S,F,T] = spectrogram(x,window,noverlap,nfft,fs)
```

x – вектор с отсчётами сигнала,

$window$ – либо целое число, указывающее количество отсчётов из которых состоит окно, либо вектор с отсчётами окна; в первом случае будет использоваться окно Хемминга,

$noverlap$ – количество отсчётов, на которое будут перекрываться сегменты сигнала; по умолчанию сегменты перекрываются на 50%,

$nfft$ – количество отсчётов, используемых для дискретного преобразования Фурье (по аналогии с функцией fft),
 f_s – частота квантования,
 S – матрица с результатом вычисления оконного преобразования Фурье,
 F – вектор частот, количество элементов вектора равно количеству строк матрицы S ,
 T – вектор со значениями времени, каждое из которых соответствует середине сегмента сигнала.

Из всех параметров обязательным является только первый, также могут отсутствовать возвращаемые параметры. Если отсутствуют возвращаемые параметры, будет построен контурный график спектра мощности в децибелах.

По аналогии с дискретным преобразованием Фурье, для оконного преобразования аналогичным образом может быть выражена граничная частота спектра, равная половине частоты квантования $\frac{f_s}{2}$, и расстояние между гармониками (шаг по частоте) равно $\frac{f_s}{N}$.

6.2. Оконные функции в Matlab

В Matlab в библиотеке Signal Processing находятся функции для создания окон различного типа. Функция **window** обладает наиболее широкими возможностями. Если вызвать эту функцию без параметров, то откроется графический интерфейс для создания и анализа оконных функций (рис. 45). С помощью кнопки **Add a new window** в список добавляется новая оконная функция. Параметры выделенной в списке оконной функции задаются с помощью элементов в **Current Window Information** (имя, тип окна, количество отсчётов, для оконных функций некоторых типов может быть параметр или функция может быть задана с помощью команд Matlab). Чтобы применить заданные параметры надо нажать кнопку **Apply**. На графиках отображаются оконные функции, которые выделены в списке. Для примера на графиках построены прямоугольное окно, треугольное окно и окно Хеннинга.

В библиотеке Signal Processing находится много функций для создания окон: `barthannwin`, `bartlett`, `blackman`, `blackmanharris`, `bohmanwin`, `chebwin`, `flattopwin`, `gausswin`, `hamming`, `hann`, `kaiser`, `nuttallwin`, `parzenwin`, `rectwin`, `triang`, `tukeywin`. Рассмотрим некоторые из этих функций. Чтобы построить график оконной функции и её спектра в библиотеке есть функция `wvtool`, которая создаёт графический интерфейс, показанный на рис. 46.

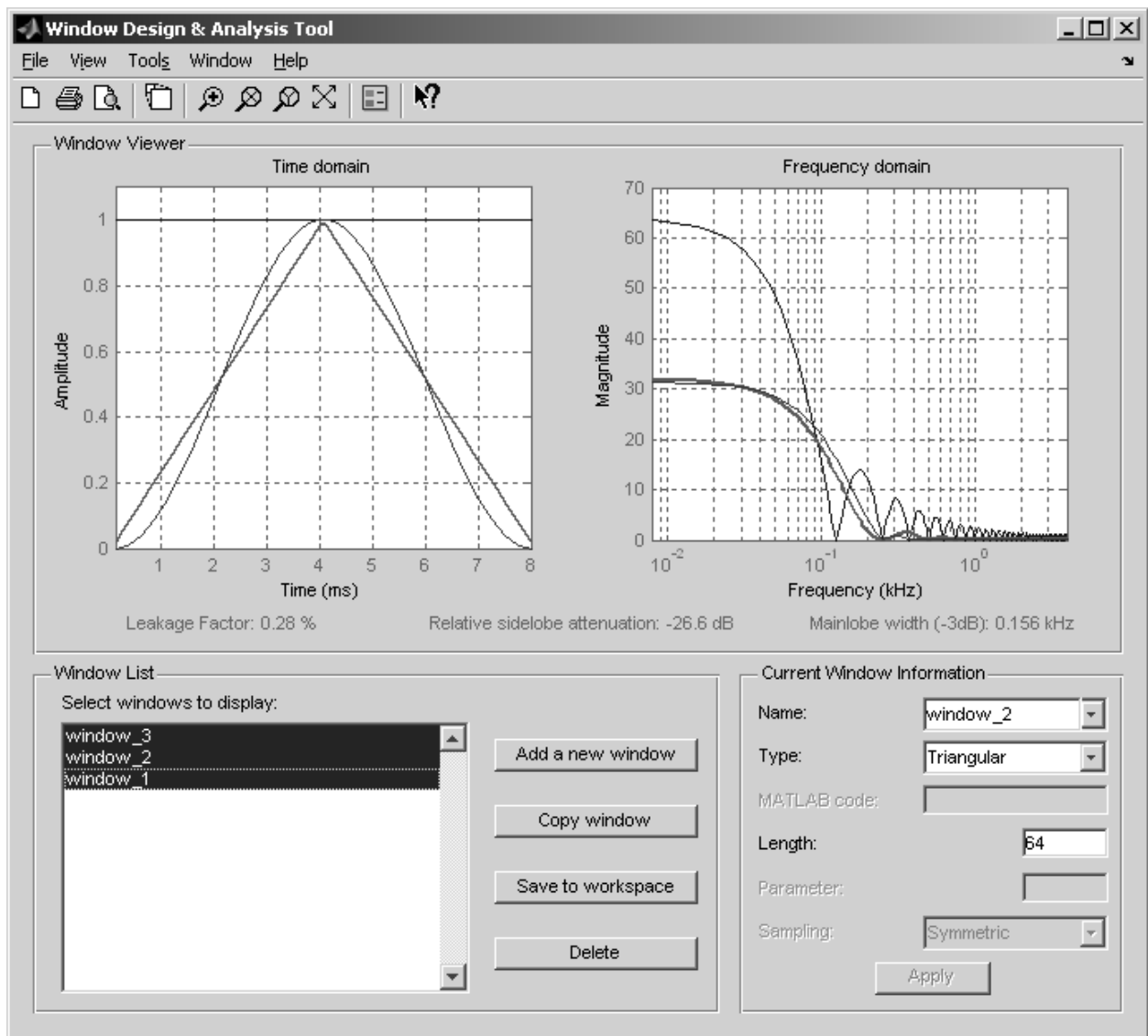


Рис. 45. Графический интерфейс window.

Как уже было отмечено, оконная функция должна обеспечивать плавное убывание сигнала на концах выделяемого окном отрывка записи, это необходимо, чтобы не происходило эффекта растекания спектра при вычислении преобразования Фурье. С другой стороны окно должно быть по возможности достаточно широким, чтобы не вносить сильного искажения в форму и спектр сигнала. Рассмотрим некоторые из этих окон. Каждая функция имеет обязательный параметр, задающий размер окна в отсчётах, у некоторых могут быть дополнительные параметры. Функция возвращает вектор-столбец с отсчётами окна, размер вектора равен количеству отсчётов (в Matlab и Signal Processing по умолчанию считается, что отсчёты сигнала хранятся в столбцах матрицы).

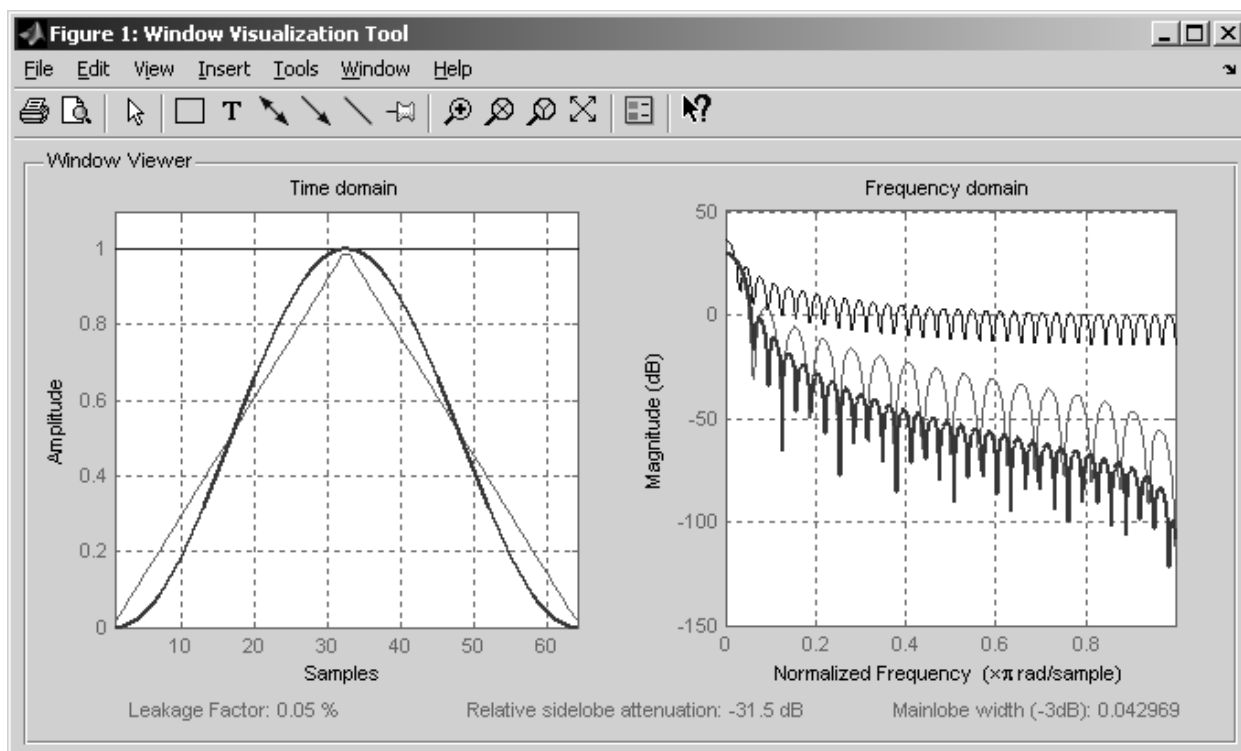


Рис. 46. Графический интерфейс wvtool.

6.2.1. Прямоугольное окно

Все отсчёты равны единице. Такое окно просто вырезает отрезок сигнала, не меняя его формы. На концах отрезка значения сигнала могут различаться, соответственно возникает эффект растекания спектра. Выше уже была дана интерпретация этого эффекта связанная со спектром прямоугольного окна.

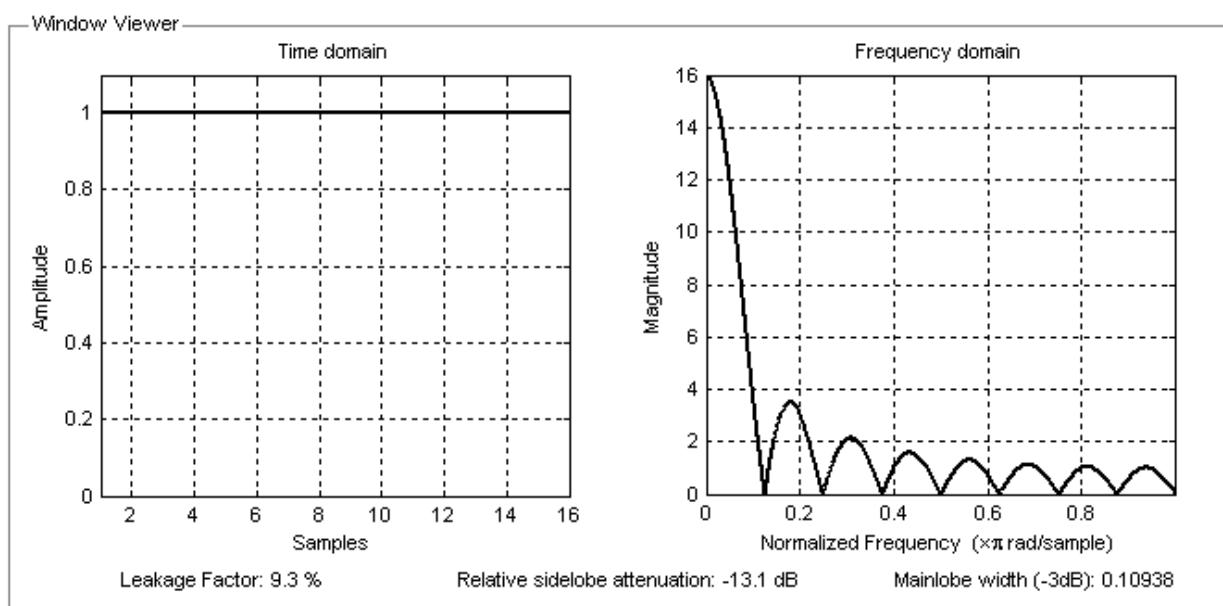


Рис. 47. Прямоугольное окно.

Медленный характер убывания амплитудного спектра прямоугольного окна приводит к растеканию спектра, если значения сигналов начале и конце интервала попавшего в окно различаются. В библиотеке Signal Processing для создания прямоугольного окна предназначена функция `rectwin`, которая заменила функцию `boxcar` из предыдущих версий Matlab. Прямоугольное окно и его спектр показаны на рис. 47.

6.2.2. Треугольное окно

Треугольное окно вычисляется по формулам:
если количество отсчётов L – нечётное:

$$w_n = \begin{cases} \frac{2n}{L+1}, & 1 \leq n \leq \frac{L+1}{2} \\ \frac{2(L-n+1)}{L+1}, & \frac{L+1}{2} < n \leq L \end{cases} \quad (127)$$

если количество отсчётов L – чётное:

$$w_n = \begin{cases} \frac{2n}{L}, & 1 \leq n \leq \frac{L}{2} \\ \frac{2(L-n)+1}{L}, & \frac{L}{2} + 1 \leq n \leq L \end{cases} \quad (128)$$

На графиках хорошо заметно, что амплитудный спектр треугольного окна (рис. 48) убывает с ростом частоты быстрее, чем амплитудный спектр прямоугольного окна. В библиотеке Signal Processing для создания треугольного окна предназначена функция `triang`.

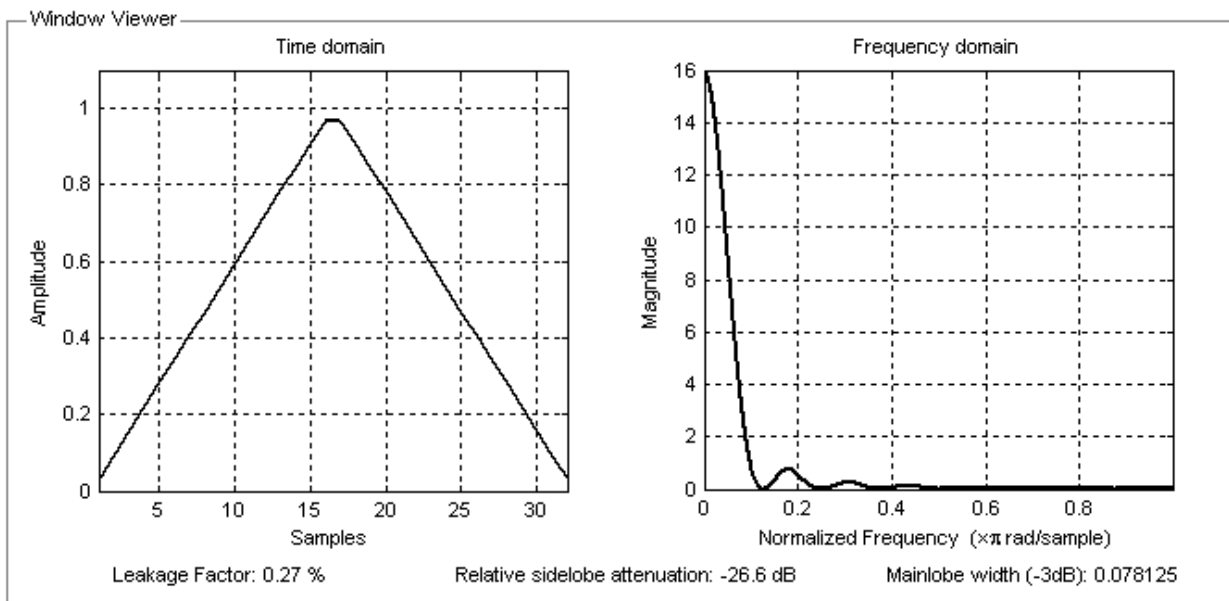


Рис. 48. Треугольное окно.

6.2.3.Окно Хенинга

Окно Хенинга вычисляется по формуле:

$$w_n = \frac{1}{2} \left(1 - \cos \left(2\pi \frac{n}{N} \right) \right), \quad 0 \leq n \leq N, \quad L = N + 1 \quad (129)$$

где L – размер окна в отсчётах. Окно Хенинга создаётся с помощью функции `hann`. По сравнению с треугольным окном амплитудный спектр окна Хенинга убывает ещё быстрее.

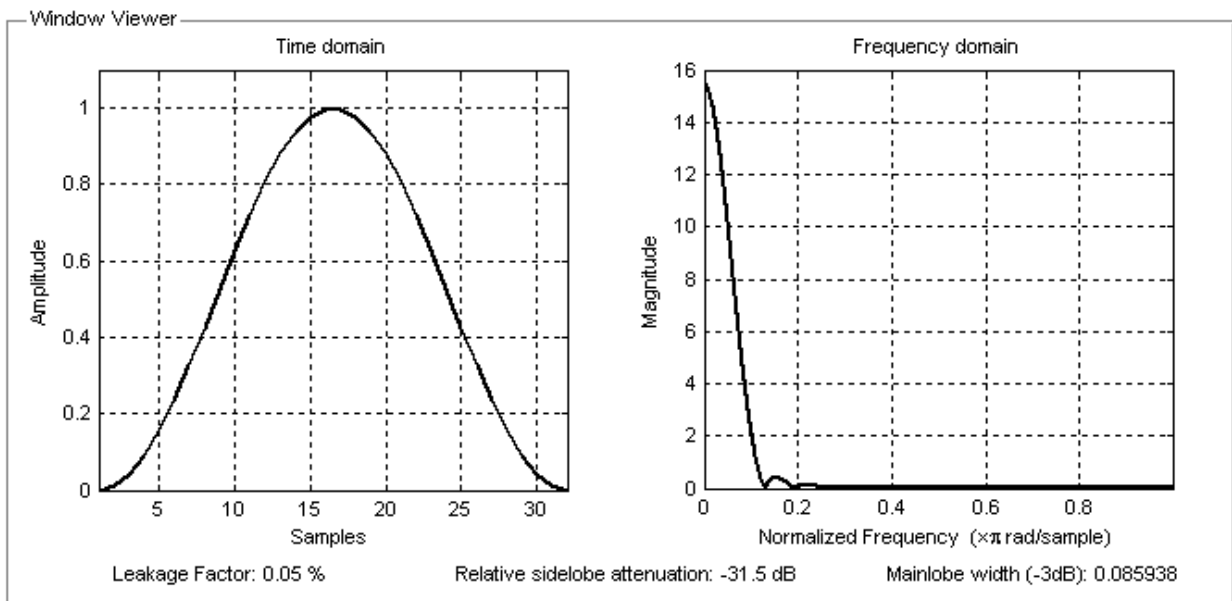


Рис. 49. Окно Хенинга.

6.2.4.Окно Хэмминга

Окно Хэмминга вычисляется по формуле:

$$w_n = 0.54 - 0.46 \cos \left(2\pi \frac{n}{N} \right), \quad 0 \leq n \leq N, \quad L = N + 1 \quad (130)$$

Амплитудный спектр убывает также быстро, как и у окна Хенинга, но за счёт небольшого отличия в коэффициентах лепестковая структура амплитудного спектра проявляется в меньшей степени. Окно Хэмминга создаётся функцией `hamming`.

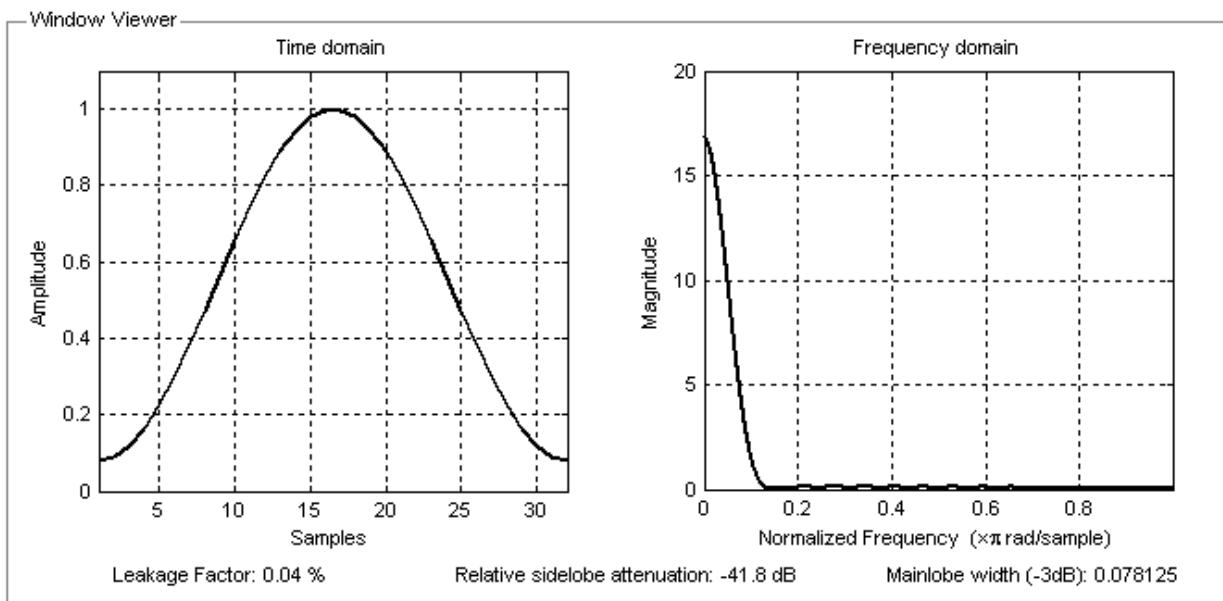


Рис. 50. Окно Хэмминга.

6.2.5. Окно Блэкмена.

Окно вычисляется по формуле:

$$w_n = 0.42 - 0.5 \cos\left(2\pi \frac{n}{N}\right) + 0.08 \cos\left(4\pi \frac{n}{N}\right), \quad 0 \leq n \leq N, \quad L = N + 1 \quad (131)$$

За счёт ещё одного слагаемого с косинусом амплитудный спектр окна убывает быстрее, чем у окон Хэмминга и Хенинга.

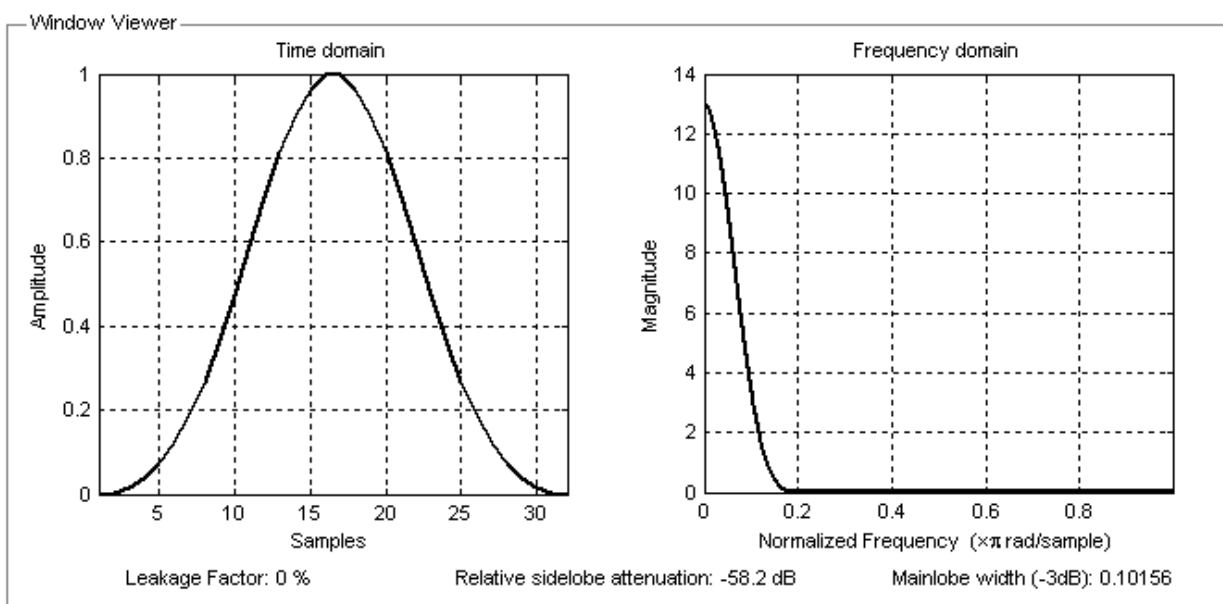


Рис. 51. Окно Блэкмена.

Для сравнения построим на одном графике окна Блэкмена и Хэмминга (рис. 52)

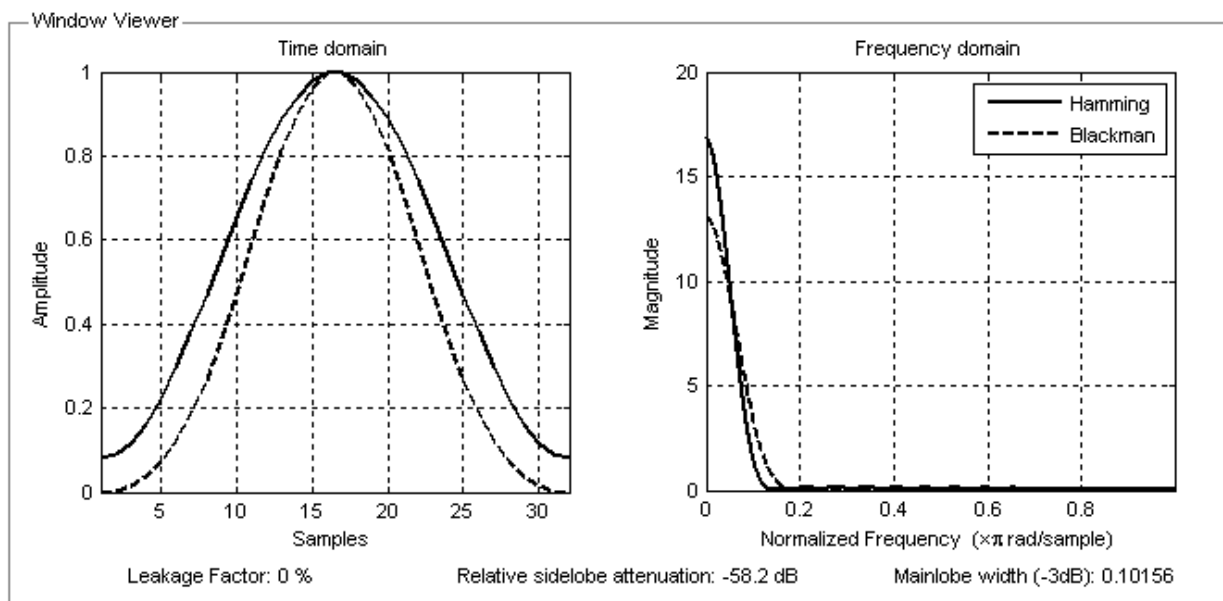


Рис. 52. Сравнение окон Блэкмена и Хэмминга.

Окно Хэмминга на графиках показано сплошной линией, окно Блэкмена штрихованной. Окно Хэмминга более широкое и на концах в ноль не обращается. Окно Блэкмена более узкое и соответственно его амплитудный спектр немного шире амплитудного спектра окна Хэмминга.

6.2.6. Окно Кайзера

Окно вычисляется по формуле:

$$w_n = \frac{I_0 \left(\pi \beta \sqrt{1 - \left(\frac{2n}{N} - 1 \right)^2} \right)}{I_0(\pi \beta)}, \quad 0 \leq n \leq N, \quad L = N + 1 \quad (132)$$

где I_0 – модифицированная функция Бесселя нулевого порядка. Окно Кайзера имеет дополнительный параметр β . Для создания этого окна в библиотеке Signal Processing есть функция `kaiser`. Функция может быть вызвана с одним или двумя параметрами. Первый параметр обязательный – это количество отсчётов, второй параметр, необязательный – β , определяющий ширину окна. Если второй параметр отсутствует, то его значение считается по умолчанию равным 0,5. На рис. 53 построены графики окон Кайзера для параметров β равных 5, 10 и 20. Чтобы была заметна разница между графиками амплитудных спектров, в настройках

графического интерфейса `wvtool` задано отображать график амплитудного спектра в децибелах. На графиках видно, что чем уже окно, тем шире его спектр, и больше подавление высоких гармоник (-20дБ означают ослабление в 10 раз, -40дБ в 100 раз, -120дБ в 10^6 раз)

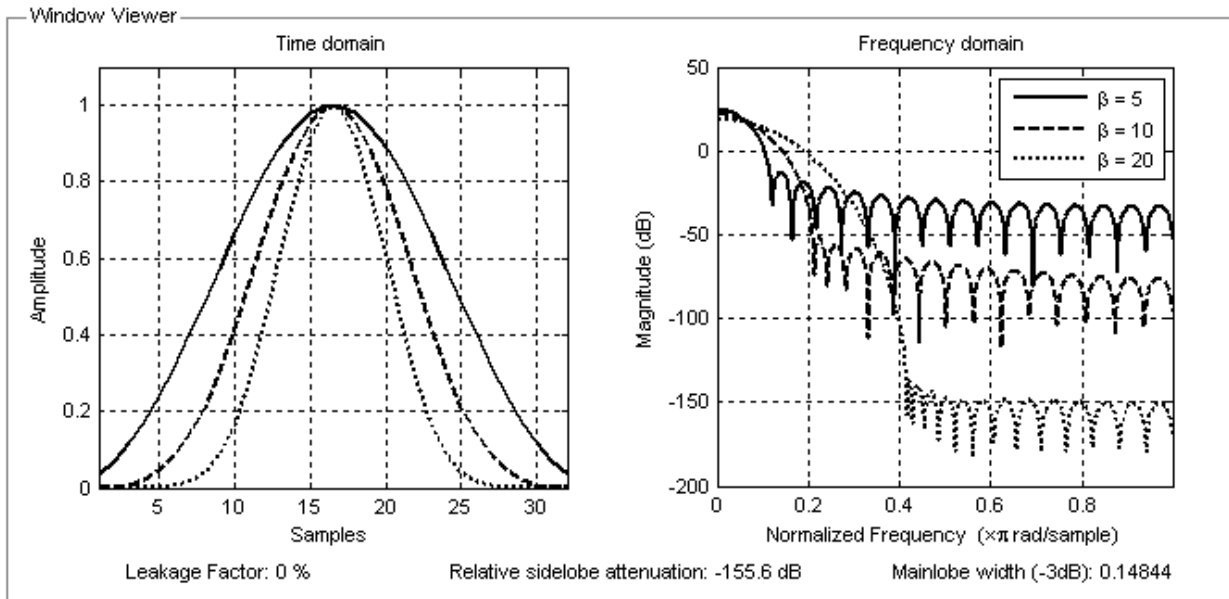


Рис. 53. Окна Кайзера.

Пример

Вычислим оконное преобразование Фурье сигнала, который образован суммой двух квазигармонических частотно модулированных сигналов. Мгновенная частота одного сигнала растёт по линейному закону от 500Гц до 3000Гц за 3 секунды, а у другого убывает по квадратичному закону от 3000Гц до 100Гц за то же время. С помощью обыкновенного преобразования Фурье нельзя определить, какие частоты были у этих сигналов в определённый момент времени. Также нельзя установить сам факт того, что сигнал состоит из двух частотно модулированных сигналов. С помощью оконного преобразования эта информация становится доступной.

Для создания частотно модулированных сигналов в библиотеке Signal Processing есть специальная функция:

```
x = chirp(t, f0, t1, f1, 'method')
```

t – вектор с значениями времени,

f_0 – начальная частота,

t_1 – время, когда сигнал будет иметь частоту f_1 ,

f_1 – мгновенная частота сигнала в момент времени t_1 ,

method – необязательный параметр, определяющий закон изменения частоты (linear, quadratic, logarithmic), по умолчанию используется linear.

Для моделирования выбрана частота квантования 8кГц, соответственно период квантования будет равен 0,125мс. Для вычисления оконного преобразования Фурье используется окно Хеннинга, которое создаётся командой hann, размер окна равен 1024 отсчётам. Такое же количество отсчётов задано для вычисления преобразования Фурье, то есть сигнал не обрезается и не дополняется нулями. Окно Хеннинга задаётся формулой:

$$w_n = \frac{1}{2} \left(1 - \cos \left(\frac{2\pi n}{N} \right) \right) \quad (133)$$

где N – это размер окна. С помощью окна в 1024 отсчёта выделяется отрезок сигнала длительностью 0,128с. Самая низкая частота сигнала 100Гц, то есть наименьшее число периодов сигнала, которое будет вырезано окном 12,8. Этого достаточно, чтобы с помощью преобразования Фурье можно было определить частоту сигнала на данном отрезке времени. Размер окна может быть выбран и меньше, например, 32 отсчёта. Тогда окно будет выделять отрезок сигнала длительностью 4мс, такое окно вырезало бы ровно два периода сигнала с частотой 500Гц. Но при уменьшении размера окна происходит растекание спектра. Для вычисления спектра используется 1024 отсчёта (во втором случае с меньшим окном сигнал будет дополнен нулями). Шаг по времени определяется количеством отсчетов, которыми перекрываются окна, и частотой квантования. В первом случае размер окна 1024, а области перекрывания 1000 отсчётов. Соответственно в процессе вычисления оконного преобразования Фурье на каждом шаге окно будет сдвигаться на 24 отсчёта, период квантования 0,125мс, соответственно окно будет сдвигаться на 3мс.

Пример оконного преобразования Фурье (test_spectrogram.m)

```
clear, clc, close all

Fs = 8000; % частота квантования
T = 3; % длительность записи
t = 0:1/Fs:T; % время
% сигнал
x = chirp(t, 500, T, 3000) + chirp(t, 3000, T, 100, 'quadratic');

% оконное преобразование Фурье
[S, fsp, tsp] = spectrogram(x, hann(1024), 1000, 1024, Fs);
```

```

% построение контурного графика
figure, imagesc(tsp, fsp, abs(S))
grid on, axis xy
c = (1:-0.05:0)'*[1 1 1];
colormap(c), colorbar
xlabel('Время, с', 'FontName', 'Arial Cyr')
ylabel('Частота, Гц', 'FontName', 'Arial Cyr')

% оконное преобразование Фурье с размером окна 32 отсчёта
[S,fsp,tsp] = spectrogram(x,hann(32),30,1024,Fs);

% построение контурного графика
figure, imagesc(tsp, fsp, abs(S))
grid on, axis xy
colormap(c), colorbar
xlabel('Время, с', 'FontName', 'Arial Cyr')
ylabel('Частота, Гц', 'FontName', 'Arial Cyr')

```

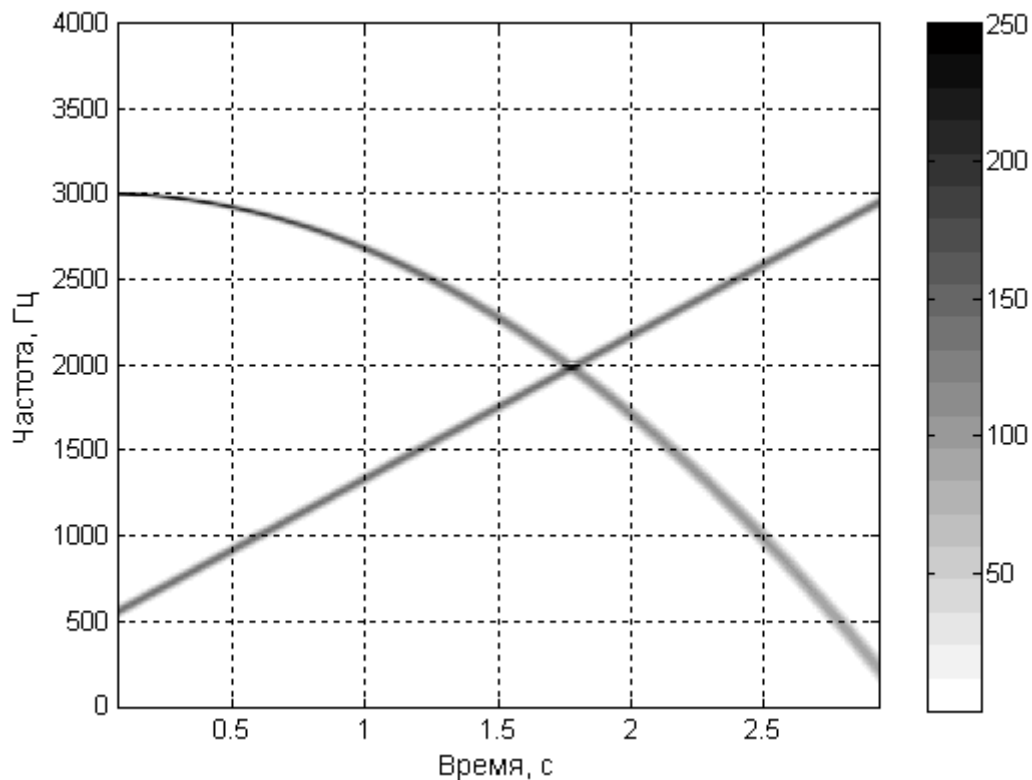


Рис. 54. Результат spectrogram с окном размером 1024 отсчёта.

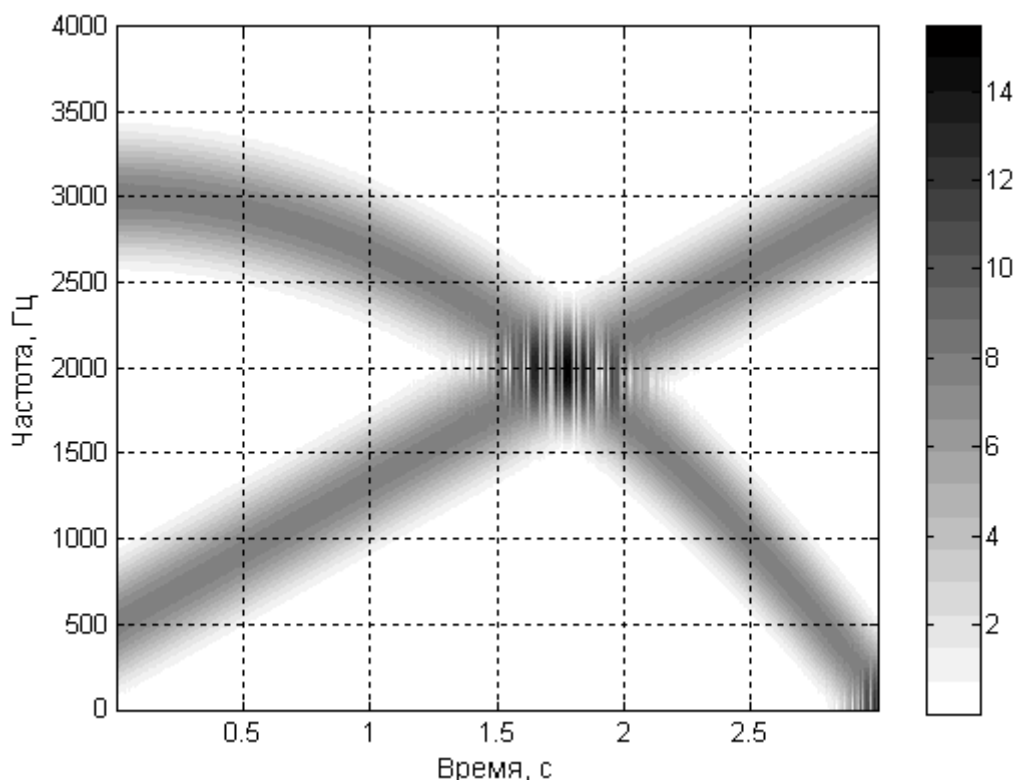


Рис. 55. Результат spectrogram с окном размером 32 отсчёта.

В результате выполнения программы будут построены два графика модуля оконного преобразования Фурье, называемые спектрограммой. Первый график (рис. 54) соответствует спектрограмме, вычисленной с использованием окна размером 1024 отсчёта. На графике отчётливо видно присутствие двух частотно модулированных сигнала. У одного частота изменяется с течением времени по линейному закону, у другого – по квадратичному. На другом графике (рис. 55) показана спектрограмма того же сигнала, но вычисленная с окном размером 32 отсчёта. По сравнению с первым графиком, на втором спектр получился размытым.

Пример

Рассмотрим два сигнала, каждый из которых состоит из двух радиоимпульсов длительностью 0,02с. Время начала первого импульса 0,02с, а второго 0,06с, частоты заполнения 400Гц и 1200Гц. Сигналы отличаются друг от друга тем, что импульсы приходят в разном порядке (рис. 57).

Имея представление сигнала во временной области, можно установить момент времени, в который приходят импульсы, но нельзя определить их частоту. И наоборот, имея представление сигнала в частотной области, полученное с помощью преобразования Фурье, можно определить их частоты, но нельзя определить время прихода импульсов. С

помощью оконного преобразования Фурье можно получить частоту и время прихода импульсов одновременно, но импульсы окажутся размытыми как по частоте, так и по времени. Если увеличивать размер окна, то на спектрограмме сигнал будет становиться менее размытым по частоте и более размытым по времени, и наоборот, если уменьшить размер окна, то на спектрограмме сигнал будет менее размытым по времени и более размытым по частоте.

В примере создаются два сигнала из двух радиоимпульсов, следующих в разном порядке, вычисляется дискретное преобразование Фурье и оконное преобразование Фурье с окнами разного размера.

Пример оконного преобразования Фурье (wnd_dft.m)

```
clear, clc, close all

Fs = 8000; % частота квантования
f1 = 400; % частота заполнения одного из импульсов
f2 = 1200; % частота заполнения другого импульса
tau = 0.02; % длительность импульсов
t1 = 0.02; % время прихода первого импульса
t2 = 0.06; % время прихода второго импульса
dt = 1/Fs; % период квантования
t = 0:dt:0.1; % время
N = length(t); % количество отсчётов
% создаём сигналы, состоящие из двух импульсов
% сигналы отличаются тем, что импульсы с разной частотой
% приходят в разном порядке
x1 = (t>=t1 & t<t1+tau).*sin(2*pi*f1*t)+ ...
      (t>=t2 & t<t2+tau).*sin(2*pi*f2*t);
x2 = (t>=t1 & t<t1+tau).*sin(2*pi*f2*t)+ ...
      (t>=t2 & t<t2+tau).*sin(2*pi*f1*t);

% сторим графики сигналов
figure
subplot(2,1,1), plot(t,x1,'k'), xlabel('t'), ylabel('x_1(t)')
subplot(2,1,2), plot(t,x2,'k'), xlabel('t'), ylabel('x_2(t)')

X1 = fft(x1); % дискретное преобразование Фурье первого
сигнала
X2 = fft(x2); % дискретное преобразование Фурье второго
сигнала
% частоты, соответствующие гармоникам
f = (ceil(N/2)-N:ceil(N/2)-1)*Fs/N;

% сторим графики амплитудных спектров сигналов
figure
subplot(2,1,1), stem(f, abs(fftshift(X1))/N, 'k.')
xlabel('f'), ylabel('|X_1|'), axis([0 1600 0 0.12])
subplot(2,1,2), stem(f, abs(fftshift(X2))/N, 'k.')
xlabel('f'), ylabel('|X_2|'), axis([0 1600 0 0.12])
```

```

% вычисляем оконное дискретное преобразование Фурье
[Xw1,F,T] = spectrogram(x1,triang(128),64,128,Fs);
Xw2 = spectrogram(x2,triang(128),64,128,Fs);

c=(1:-0.05:0)'*[1 1 1]; % палитра цветов для построения
графиков

% строим контурные графики модуля оконного преобразования
Фурье
figure
subplot(2,1,1), contourf(T,F,abs(Xw1))
colormap(c), colorbar, grid on, axis([min(T) max(T) 0 1600])
xlabel('t'), ylabel('f'), title('|Xw_1(t,f)|')
subplot(2,1,2), contourf(T,F,abs(Xw2))
colormap(c), colorbar, grid on, axis([min(T) max(T) 0 1600])
xlabel('t'), ylabel('f'), title('|Xw_2(t,f)|')

% вычисляем оконное дискретное преобразование Фурье
% уменьшили размер окна в 4 раза
[Xw1,F,T] = spectrogram(x1,triang(32),16,32,Fs);
Xw2 = spectrogram(x2,triang(32),16,32,Fs);

% строим контурные графики модуля оконного преобразования
Фурье,
% вычисленного с окном другого размера
figure
subplot(2,1,1), contourf(T,F,abs(Xw1))
colormap(c), colorbar, grid on, axis([min(T) max(T) 0 1600])
xlabel('t'), ylabel('f'), title('|Xw_1(t,f)|')
subplot(2,1,2), contourf(T,F,abs(Xw2))
colormap(c), colorbar, grid on, axis([min(T) max(T) 0 1600])
xlabel('t'), ylabel('f'), title('|Xw_2(t,f)|')

```

В результате будут построены графики сигналов (рис. 56) и их амплитудные спектры (рис. 57). Амплитудные спектры построены только на положительной полуоси. Амплитудные спектры обоих сигналов совпадают, фазовые спектры будут разными, так как на фазу влияет момент прихода импульса и начальная фаза заполнения.

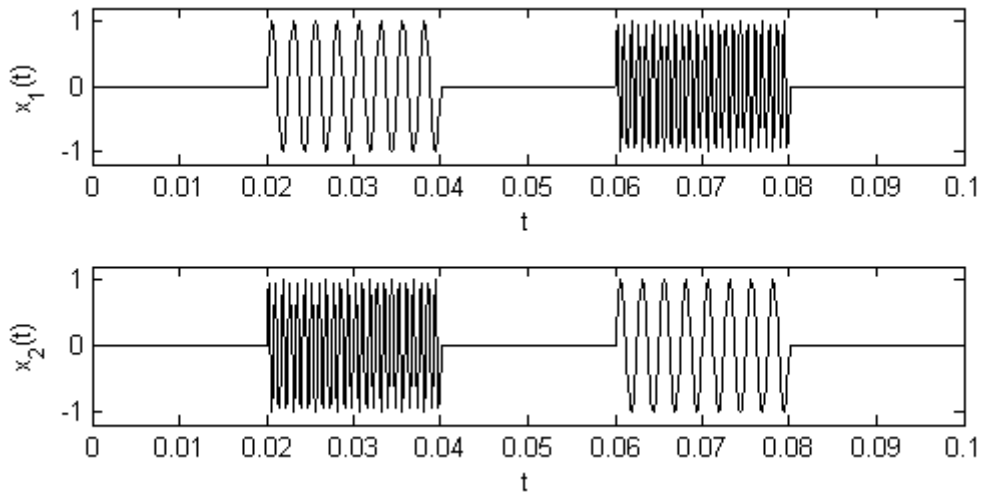


Рис. 56. Представление сигналов во временной области.

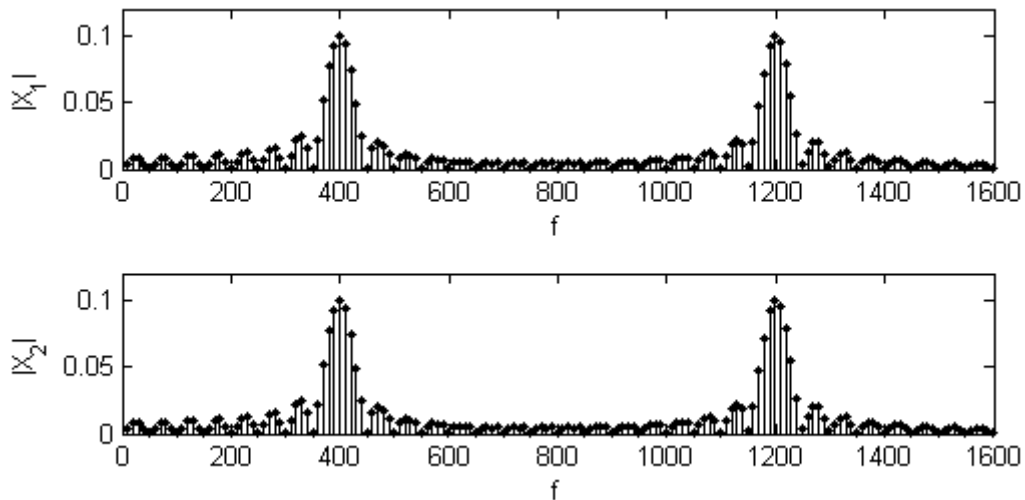


Рис. 57. Представление сигналов в частотной области.

При вычислении первой спектрограммы (рис. 58) использовалось треугольное окно размером 128 отсчётов, а при вычислении второй спектрограммы (рис. 59) – окно размером 32 отсчёта. При частоте квантования 8кГц первое окно имеет длительность 16мс, второе – 4мс. Для сравнения, период сигнала с частотой 400Гц равен 2,5мс, а длительность радиоимпульсов в примере 20мс. Из графиков спектрограмм видно, что при использовании широкого окна максимум, изображающий импульс, по времени более расплывчатый, чем по оси частот. А при использовании короткого окна – наоборот, по спектрограмме более точно можно определить момент времени начала и окончания импульсов. Но по оси частот убывание происходит медленнее, чем в первом случае.

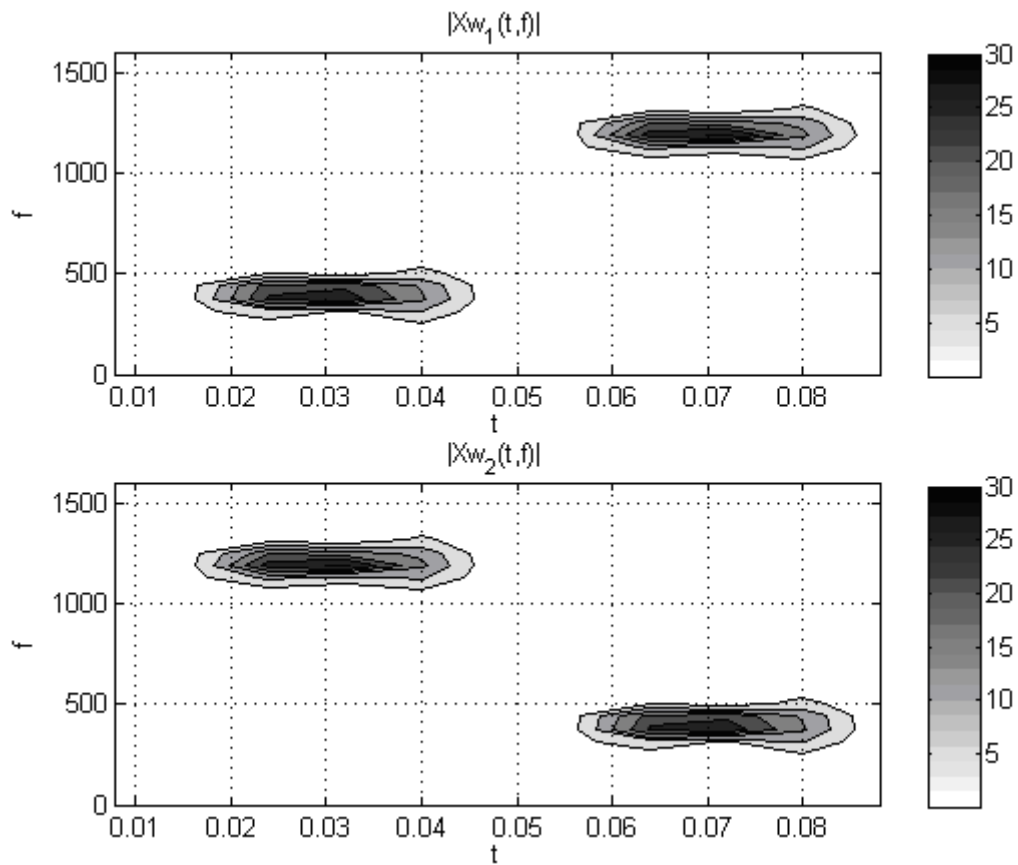


Рис. 58. Спектрограмма сигналов, вычисленная с широким окном.

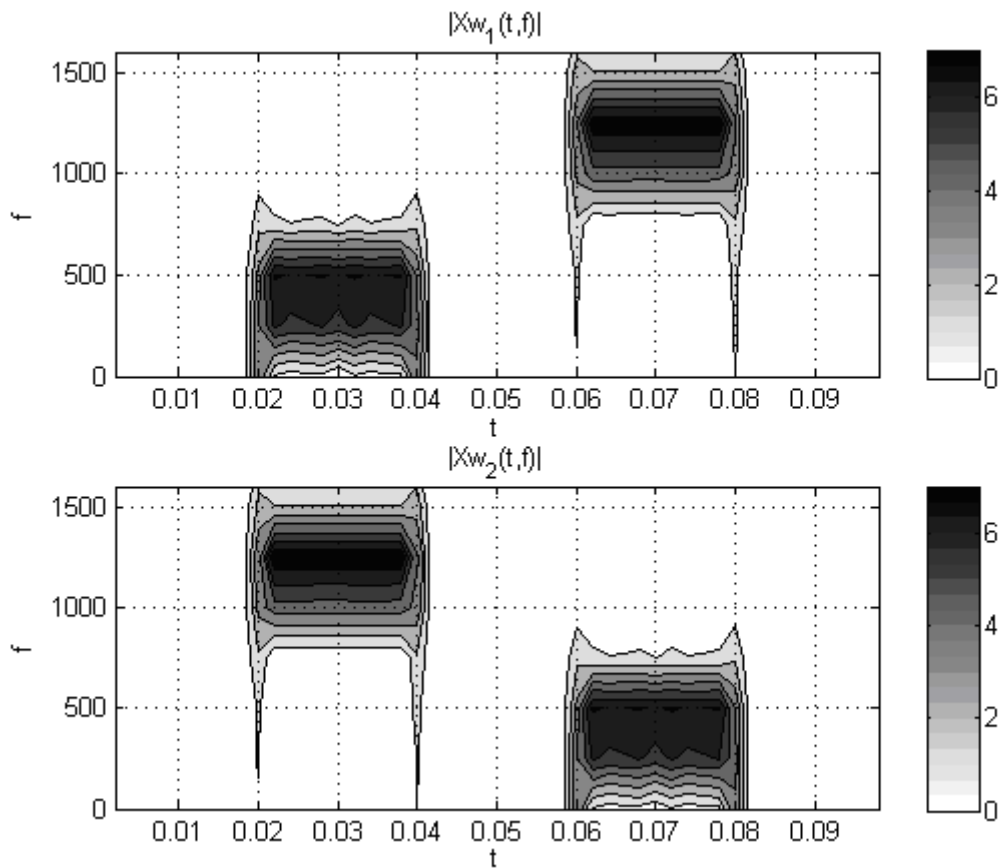


Рис. 59. Спектрограмма сигналов, вычисленная с узким окном.

В идеале, на спектрограмме должно было бы быть два горизонтальных отрезка по оси времени, соответствующих времени начала и конца импульсов, а по оси частот – мгновенной частоте заполнения импульсов. На практике это недостижимо. Чтобы приблизительно оценить, насколько расширяются спектральные линии, найдём ширину спектра треугольного импульса как расстояние между нулями. В первом случае ширина окна 16мс, соответствующая ширина спектра треугольного импульса $\frac{2}{16мс} = 125Гц$, для окна шириной 4мс соответственно $\frac{2}{4мс} = 500Гц$. Эти оценки соответствуют графикам.

7. Wavelet анализ

С помощью преобразования Фурье получается представление о спектре сигнала в целом, то есть информация о каких-либо характерных особенностях сигнала, локализованных во времени, недоступна. Такую информацию можно получить, если выполнить преобразование Фурье не всего сигнала, а только интересующего отрезка (оконное преобразование Фурье). В оконном преобразовании Фурье используются оконные функции фиксированной ширины. Раз ширина окна остаётся неизменной, то остаётся постоянным и разрешение по частоте и по времени. Но исследовать характерные особенности сигнала разного масштаба по времени целесообразно с помощью окон разного размера, чтобы соответственно иметь различное разрешение по частоте и времени.

В преобразовании Фурье происходит разложение сигнала по гармоническим функциям. Если в определённые моменты времени сигнал изменяется скачкообразно, или к нему добавляется ещё какой-нибудь сигнал, или происходит ещё какое-либо изменение, которое плохо аппроксимируется тригонометрическими функциями, то вместо них более подходящими могут быть другие функции. В основе wavelet анализа лежит разложение сигнала по функциям, локализованным как в пространстве, так и во времени, что позволяет достаточно эффективно обнаруживать различные особенности сигналов.

7.1. Непрерывное wavelet преобразование

Вейвлетом $\psi(t)$ называют волнообразную функцию, которая имеет конечную эффективную длительность, то есть достаточно быстро убывает за пределами некоторого интервала. Вейвлет также должен иметь компактный спектр, то есть быть локализованным в частотной области. Как было показано в разделе про преобразование Фурье, ширина спектра определяется, главным образом, двумя параметрами, длительностью, то есть локализацией во временной области, и гладкостью. Кроме того, вейвлет должен быть интегрируем с квадратом, а интеграл от самой функции по всей оси времени должен быть равен нулю.

$$\int_{-\infty}^{+\infty} |\psi(t)|^2 dt < \infty \quad \int_{-\infty}^{+\infty} \psi(t) dt = 0 \quad (134)$$

Функцию $\psi(t)$ также ещё называют материнским вейвлетом или оригинальным вейвлетом. По аналогии с Фурье анализом, в котором сигнал раскладывается по гармоническим функциям с различной частотой,

в вейвлет анализе сигнал раскладывается по отмасштабированным и сдвинутым по времени функциям $\psi(t)$. Соответственно, в результате вейвлет преобразования сигнал представляется в пространстве масштаб-время. По аналогии с оконным преобразованием Фурье, в котором сигнал представлялся в пространстве частота-время. Масштаб по времени в вейвлет преобразовании связан с частотой в преобразовании Фурье. На рис. 60 приведены примеры некоторых вейвлетов и их амплитудные спектры. Слева показан вейвлет, а справа – его амплитудный спектр.

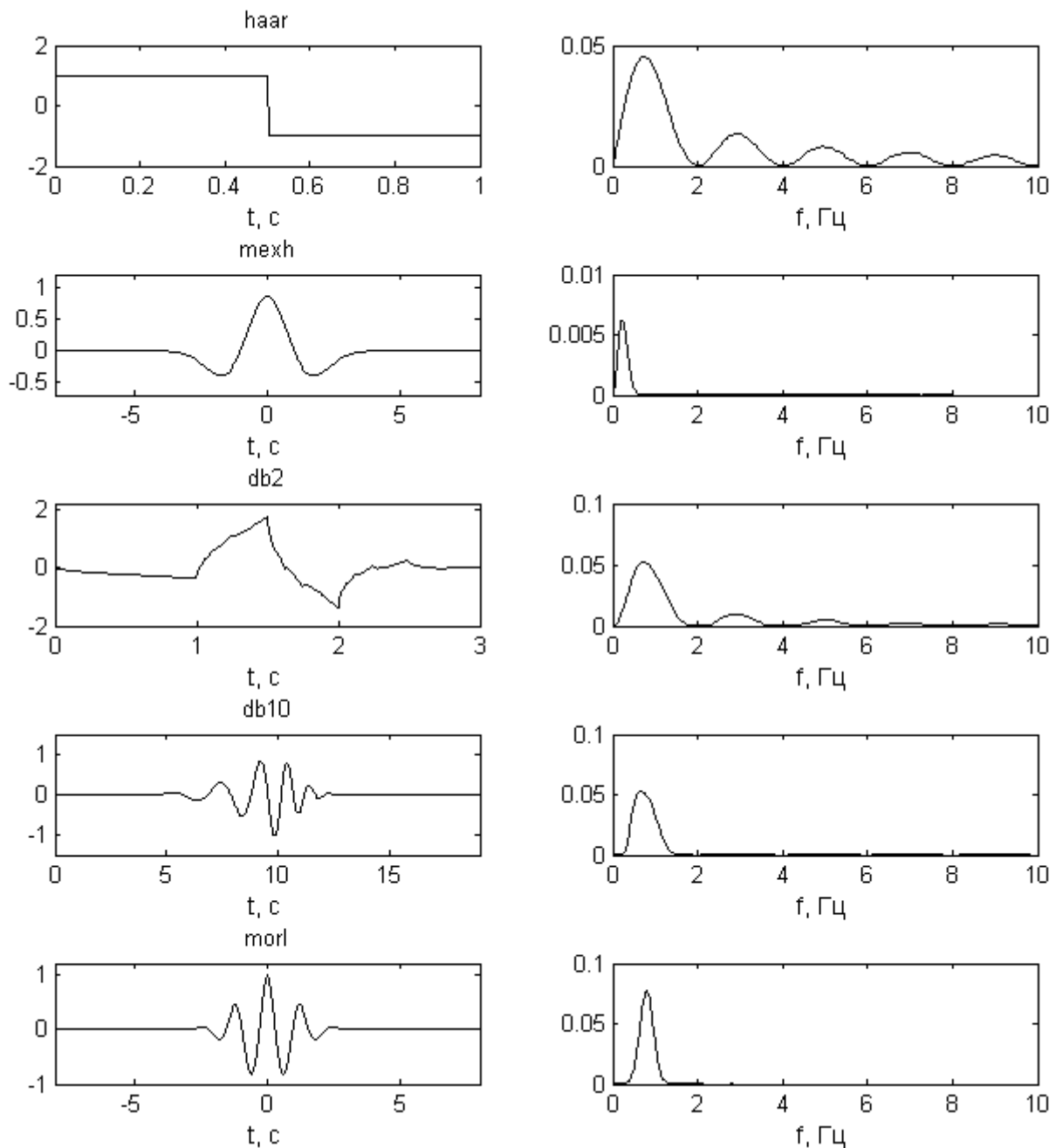


Рис. 60. Амплитудные спектры вейвлетов.

На первой паре графиков изображён вейвлет Хаара в частотной и временной области. Вейвлет Хаара – это прямоугольный импульс, поэтому его спектр имеет лепестковую структуру и, по сравнению со спектрами других вейвлетов, плохо локализован в частотной области. Амплитудный спектр вейвлета мексиканская шляпа и вейвлета Морле наиболее локализованы в частотной области. В зависимости от того, в каких целях используется вейвлет, предъявляются различные требования к локализации во временной и частотной области.

Непрерывное вейвлет преобразование вычисляется по формуле:

$$C(a,b) = \frac{1}{\sqrt{a}} \int_{-\infty}^{+\infty} x(t) \psi\left(\frac{t-b}{a}\right) dt \quad (135)$$

Переменная a определяет изменение масштаба по оси времени, а переменная b сдвигает вейвлет вдоль оси времени. Множитель $\frac{1}{\sqrt{a}}$ перед интегралом необходим, чтобы сохранялась нормировка вейвлета при изменении масштаба. По аналогии с преобразованием Фурье, вейвлет преобразование представляет сигнал в виде линейной комбинации сдвинутых, растянутых или сжатых во времени материнских вейвлетов.

Операции сдвига и масштабирования показаны на рис. 61 на примере вейвлета "мексиканская шляпа". На левой верхней картинке изображён оригинальный вейвлет. В левом столбике изображён результат операции сдвига, в правом – масштабирования.

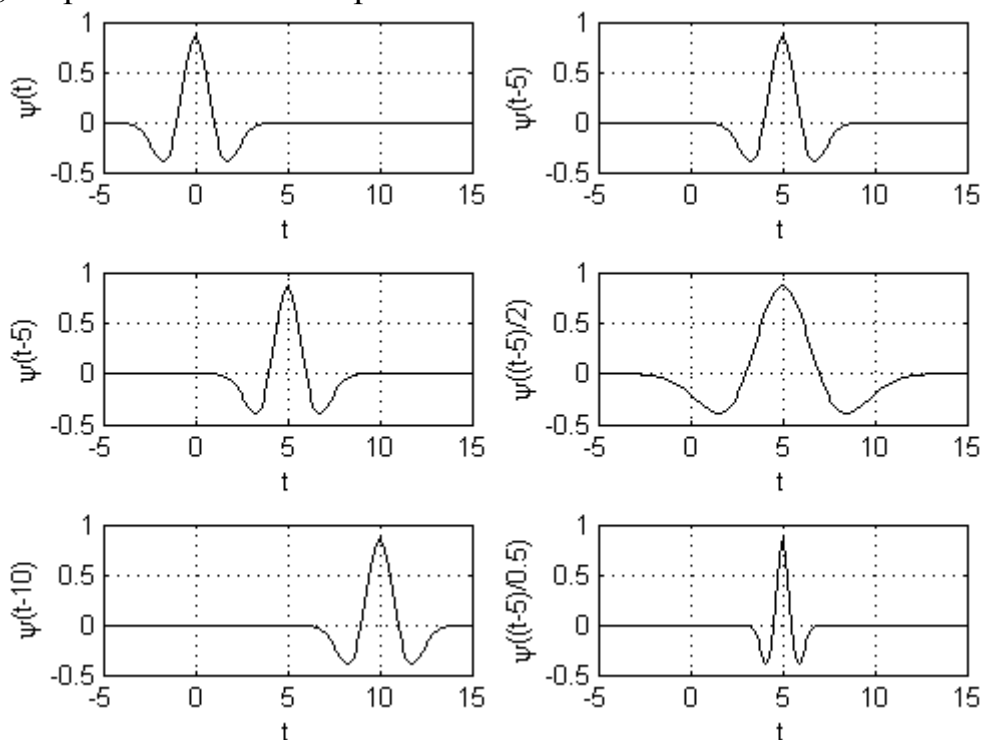


Рис. 61 Операция сдвига и масштабирования.

Сам же интеграл (135) является также функцией взаимной корреляции между сигналом и отмасштабированным вейвлетом. В теории сигналов функция корреляции показывает, насколько взаимосвязаны между собой два случайных сигнала. В данном случае она служит мерой того, насколько вейвлет похож на отрезок сигнала.

В литературе [2] доказывается, что для некоторых вейвлетов существует обратное вейвлет преобразование, вычисляемое по формуле:

$$x(t) = \frac{1}{C_\psi} \int_{-\infty}^{+\infty} \int_{-\infty}^{+\infty} C(a,b) \frac{1}{\sqrt{a}} \psi\left(\frac{t-b}{a}\right) \frac{dad b}{a^2} \quad (136)$$

$$C_\psi = \int_{-\infty}^{+\infty} \frac{|\Psi(\omega)|^2}{|\omega|} d\omega \quad \Psi(\omega) = \int_{-\infty}^{+\infty} \psi(t) e^{-i\omega t} dt$$

Для того чтобы существовало обратное вейвлет преобразование необходимо, чтобы $C_\psi < \infty$.

В Matlab для вычисления непрерывного вейвлет преобразования в библиотеке Wavelet Toolbox предназначена функция **cwt** (Continuous Wavelet Transform). Синтаксис вызова этой функции:

```
COEFS = cwt(S, SCALES, 'wname')
COEFS = cwt(S, SCALES, 'wname', 'plot')
COEFS = cwt(S, SCALES, 'wname', PLOTMODE)
COEFS = cwt(S, SCALES, 'wname', PLOTMODE, XLIM)
```

COEFS – коэффициенты $C(a,b)$

S – вектор с отсчётами сигнала

SCALES – вектор со значениями масштабов (значения a)

'wname' – краткое название вейвлета

PLOTMODE и XLIM – определяет вид графика

Значения параметров PLOTMODE описаны в справке. Если в качестве PLOTMODE указать 'plot', то при вызове команды cwt будет построен график с коэффициентами $C(a,b)$.

Подробную информацию о вейвлетах и их свойствах можно получить с помощью команды waveinfo('wname'), где 'wname' – название вейвлета. Сами названия перечислены в документации этой функции.

Краткое название	Полное название
haar	Haar wavelet
db	Daubechies wavelets
sym	Symlets

coif	Coiflets
bior	Biorthogonal wavelets
rbio	Reverse biorthogonal wavelets
meyr	Meyer wavelet
dmey	Discrete approximation of Meyer wavelet
gaus	Gaussian wavelets
mexh	Mexican hat wavelet
morl	Morlet wavelet
cgau	Complex Gaussian wavelets
shan	Shannon wavelets
fbsp	Frequency B-Spline wavelets
cmor	Complex Morlet wavelets

Некоторые из кратких названий, когда они используются в функции, должны быть дополнены числом, например, 'db10' или 'gaus4'. Например, для гауссовского вейвлета, который определён как производная функции Гаусса, номер означает порядок этой производной.

Несмотря на то, что преобразование называется непрерывным, в Matlab сигналы всё равно хранятся и обрабатываются в дискретном виде. Кроме непрерывного преобразования, существует также и дискретное преобразование. Принципиальная разница между дискретным и непрерывным преобразованием заключается в том, что дискретное вычисляется для строго определённых значений масштаба и сдвига, а непрерывное может быть вычислено для любых значений.

Предположим, что есть две записи одинаковой длительности одного и того же сигнала, выполненные с разной частотой квантования. Записи, соответственно, будут иметь разное количество отсчётов. Поскольку сигнал один и тот же, то и результат непрерывного вейвлет преобразования также должен быть одинаковым. Но в функцию `cwt` передаются только значения масштабов и отсчёты сигнала, а период или частота квантования, необходимые для однозначного определения масштаба по времени, не передаются. В функции `cwt` вместо времени используются номера отсчётов сигнала, что эквивалентно периоду квантования, равному 1с. Соответственно, для того чтобы для обоих сигналов получить одинаковые результаты преобразования, необходимо задавать различные значения масштабов. Если считать, что отсчёты соответствуют моментам времени $t_n = nT_s$, где T_s – период квантования, то операции сдвига и масштабирования выражаются через номер отсчёта n как $\frac{t_n - b_m}{a} = \frac{nT_s - mT_s}{a} = \frac{n - m}{a/T_s}$, где $b_m = mT_s$ – это смещение оригинального

вейвлета, выражаемое также через количество отсчётов m . В функции `cwt` необходимо задавать масштаб, рассчитанный для единичного периода

квантования $\tilde{a} = \frac{a}{T_s}$. Например, если с помощью `cwt` вычисляется вейвлет преобразование для масштаба равного 3, то есть оригинальный вейвлет растягивается во времени в три раза, а частота квантования сигнала была равна 10Гц, то в качестве параметра `SCALES` функции `cwt` необходимо указать число 30.

По аналогии с преобразованием Фурье, вейвлету можно сопоставить некоторую частоту, называемую центральной частотой. В Wavelet Toolbox для определения центральной частоты предназначена команда `centfrq`. Первый аргумент обязательный, это название вейвлета, второй аргумент определяет количество отсчётов, используемых для вычисления частоты и построения графиков, третий параметр необходим, чтобы построить график. Матлаб работает только с дискретными сигналами, поэтому для задания сигнала необходимо указать сколько отсчётов использовать для того, чтобы задать вейвлет. Вейвлет задаётся на отрезке, где он считается отличным от нуля, для каждого вейвлета этот отрезок времени разный. Количество отсчётов определяется как 2^k в некоторой степени, степень двойки указывается вторым аргументом функции `centfrq`. Без третьего параметра функция возвращает значение центральной частоты в герцах. На графиках (рис. 62) приведены примеры значений центральной частоты и сравнение формы оригинального вейвлета с гармоническим сигналом с частотой, равной центральной частоте.

Команды чтобы построить графики (рис. 62)

```
subplot(3,1,1), centfrq('haar',8,'plot')
subplot(3,1,2), centfrq('mexh',8,'plot')
subplot(3,1,3), centfrq('db10',8,'plot')
```

На графиках построены оригинальные вейвлеты и гармонические функции с центральной частотой для этого вейвлета. Над графиком написано название вейвлета, под графиком – центральная частота и период. Графики построены для вейвлетов Хаара, "мексиканской шляпы" и Добеши 10-го порядка.

В Wavelet Toolbox есть специальная функция для преобразования масштаба в соответствующую частоту для данного вейвлета.

```
F = scal2frq(A, 'wname', DELTA)
```

F – частота F_a , соответствующая масштабу a

A – масштаб \tilde{a} при периоде квантования 1с

wname – краткое название вейвлета

DELTA – период квантования T_s

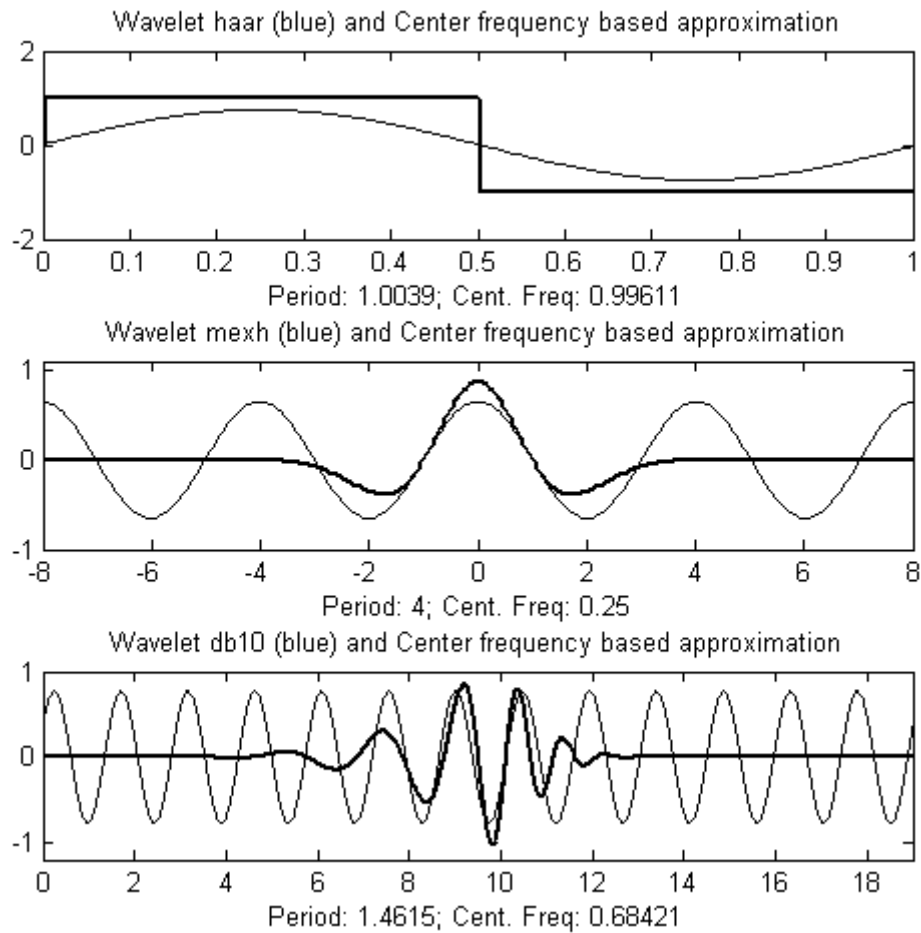


Рис. 62 Пример центральных частот вейвлетов.

В разделе про свойства преобразования Фурье показывалось (40), что при увеличении масштаба по времени в a раз, масштаб по частоте изменяется в $\frac{1}{a}$ раз. В формуле вейвлет преобразования (135) масштаб вейвлета по времени изменяется в $\frac{1}{a}$, значит, масштаб по частоте в спектре вейвлета изменился в a раз. Центральная частота вейвлета определялась как частота, на которой спектральная функция оригинального (немасштабированного) вейвлета по модулю принимает максимальное значение. После изменения масштаба по времени в $\frac{1}{a}$, и масштаба по частоте, соответственно, в a раз, положение максимума модуля спектральной функции вейвлета с масштабом a определяется по формуле.

$$F_a = \frac{F_c}{a} = \frac{F_c}{\tilde{a} \cdot T_s} \quad (137)$$

В функции `scal2frq`, также как и в функции `cwt` используется масштаб $\tilde{a} = \frac{a}{T_s}$, рассчитанный для единичного периода квантования.

Пример

Для демонстрации выполним вейвлет преобразование гармонического сигнала. Пусть частота сигнала будет 20 Гц, частота квантования 1024Гц, и мы хотим выполнить анализ сигнала в диапазоне частот от 4Гц до 256Гц. Диапазон частот выбран таким, чтобы внутри него оказалась частота сигнала. Для вейвлет анализа используем вейвлет «мексиканская шляпа», его центральная частота равна 0,25Гц. Для данного примера выбор конкретного вейвлета большого значения не имеет.

Преобразование гармонического сигнала (`wavelet_example_1.m`)

```
clear, clc, close all

Fs = 1024; % частота квантования
T = 0.25; % длительность сигнала
dt = 1/Fs; % период квантования
t = 0:dt:T; % время
f = 20; % частота сигнала
x = sin(2*pi*f*t); % сигнал
wname = 'mexh'; % используемый вейвлет
Fc = centfrq(wname); % центральная частота
% задаём границы диапазона частот,
% в котором хотим выполнить анализ сигнала
F_min = 4;
F_max = 256;
% так как вейвлет преобразование имеет дело с масштабом,
% а не с частотой, переводим частоту в масштаб
a_max = Fc/F_min/dt;
a_min = Fc/F_max/dt;
% значения масштабов, для которых будет вычислено
% вейвлет преобразование
a = a_min:1:a_max;

C = cwt(x,a,wname); % выполняем преобразование

% выводим результат на графике
figure,
subplot(2,1,1), plot(t,x,'k')
xlabel('time'), ylabel('x(t)')
subplot(2,1,2), imagesc(t,a,abs(C))
axis xy, grid on
xlabel('b (time)'), ylabel('a (scale)')
```

В результате строятся два графика один под другим (рис. 63). На верхнем графике показан сигнал, а на нижнем построен результат вейвлет

преобразования. Результат вейвлет преобразования является функцией двух переменных сдвига по времени b и масштаба a . График строится с использованием значения масштаба для периода квантования, равного единице. На нижнем графике по горизонтальной оси – сдвиг во времени, по вертикальной – масштаб, цветом показана величина $|C(a,b)|$. На иллюстрации в книге цветовая палитра изменена так, что максимуму соответствует чёрный минимуму – белый.

Установим соответствие между графиками. Максимумы на нижнем графике расположены в тех местах, где вейвлет по форме лучше всего совпадал с формой сигнала, так, как это показано на графиках (рис. 62). В данном случае максимумы на нижнем графике совпали с максимумами и минимумами на верхнем графике. Для гармонического сигнала ряд максимумов расположен вдоль горизонтальной линии соответствующий масштабу $\tilde{a} = \frac{F_c F_s}{F_a} = \frac{0.25 \cdot 1024}{20} = 12.8$. Крайний левый и крайний правый максимумы в отличие от остальных выглядят искажёнными. Это отличие объясняется тем, что мы имеем дело только с отрезком сигнала, а не с периодическим.

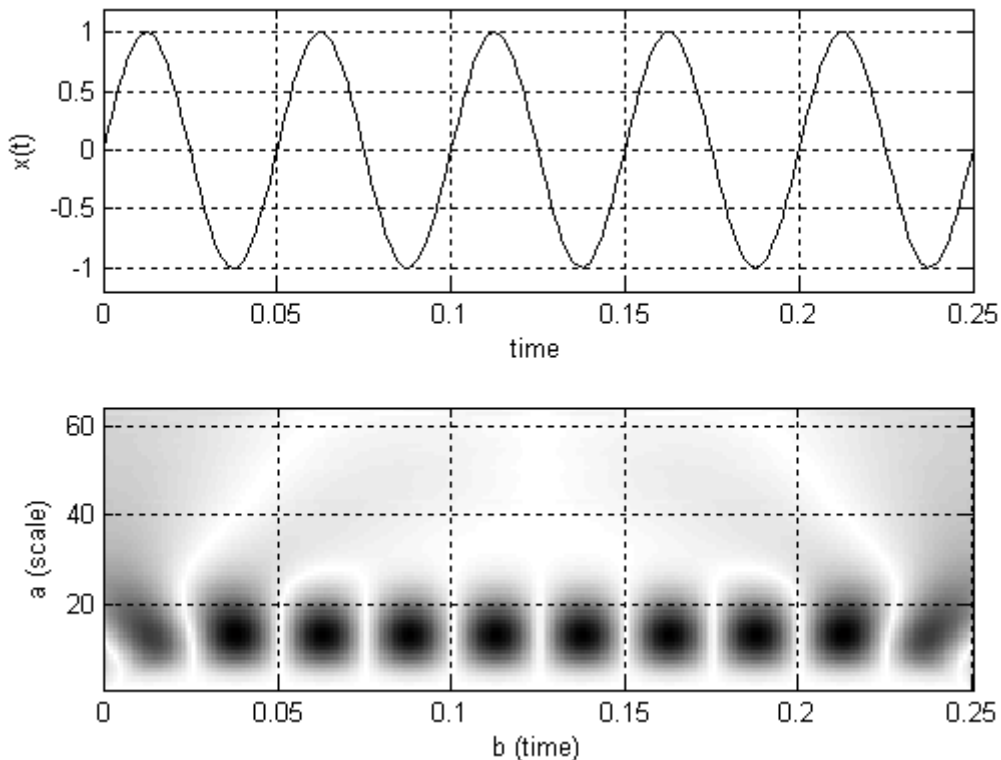


Рис 63. Преобразование гармонического сигнала.

За пределами отрезка сигнал условно можно считать равным нулю. Вообще давать интерпретацию результата вейвлет преобразования надо исходя из выбранного вейвлета, так как разные вейлеты обладают разными свойствами.

Пример

Посмотрим, что получится в результате непрерывного вейвлет преобразования сигнала, состоящего из суммы двух гармонических сигналов с разными частотами. Пусть частота одной гармоники 10Гц, другой 2,5Гц, частота квантования 300Гц, длительность записи 3с. Для анализа гармонических сигналов из библиотеки Wavelet Toolbox выбираем Morlet wavelet. Оригинальный вейвлет вычисляется по формуле:

$$\psi(t) = e^{-\frac{t^2}{2}} \cos(5t) \quad (138)$$

Выбор этого вейвлета объясняется тем, что он является произведением гармонической функции на функцию Гаусса. То есть вейвлет преобразование с таким вейвлетом сходно с оконным преобразованием Фурье с функцией Гаусса в качестве окна. Изменение масштаба по времени эквивалентно изменению частоты, одновременно с этим изменяется и ширина окна, что является характерным отличием от оконного преобразования Фурье. Масштаб по времени равен отношению центральной частоты оригинального вейвлета к частоте соответствующей

данному масштабу $a = \frac{F_c}{F_a}$. В примере для построения графиков

оригинальный вейвлет задан явной формулой. Центральная частота вейвлета определяется с помощью функции `centfrq`. Сначала выводятся графики (рис. 64) сигнала и двух вейвлетов, для которых подобран такой масштаб по времени, чтобы соответствующая им частота была равна одной из частот, присутствующих в сигнале. Затем на отдельном графике (рис. 65) выводится результат вейвлет преобразования, чтобы получить значения сдвига во времени числа вдоль горизонтальной оси надо разделить на частоту квантования.

Вейвлет преобразование гармонического сигнала (test_centfrq.m)

```
clear, clc, close all

% оригинальный вейвлет "Morlet"
mw = @(t) exp(-t.^2/2).*cos(5*t);

Fs = 300; Ts = 1/Fs; % частота и период квантования
f1 = 10; f2 = 2.5; % частоты гармоник
t = 0:Ts:3; % время
```

```

x = sin(2*pi*f1*t)+sin(2*pi*f2*t); % сигнал

Fc = centfrq('morl'); % центральная частота
a1 = Fc/f1; % масштаб для первой гармоники
a2 = Fc/f2; % масштаб для второй гармоники

% строим графики сигнала и вейвлетов
figure
subplot(3,1,1), plot(t,x,'k'), xlabel('t'), ylabel('x(t)')
subplot(3,1,2), plot(t,mw((t-
0.625)/a1),'k',t,sin(2*pi*f1*t),'k:')
xlabel('t'), title('\psi((t-0.625)/a_1)) and sin(2\pif_1t)')
subplot(3,1,3), plot(t,mw((t-
1.3)/a2),'k',t,sin(2*pi*f2*t),'k:')
xlabel('t'), title('\psi((t-1.3)/a_2)) and sin(2\pif_2t)')

% выполняем вейвлет преобразование и выводим результат на
график
figure, cwt(x,1:200,'morl','plot');

```

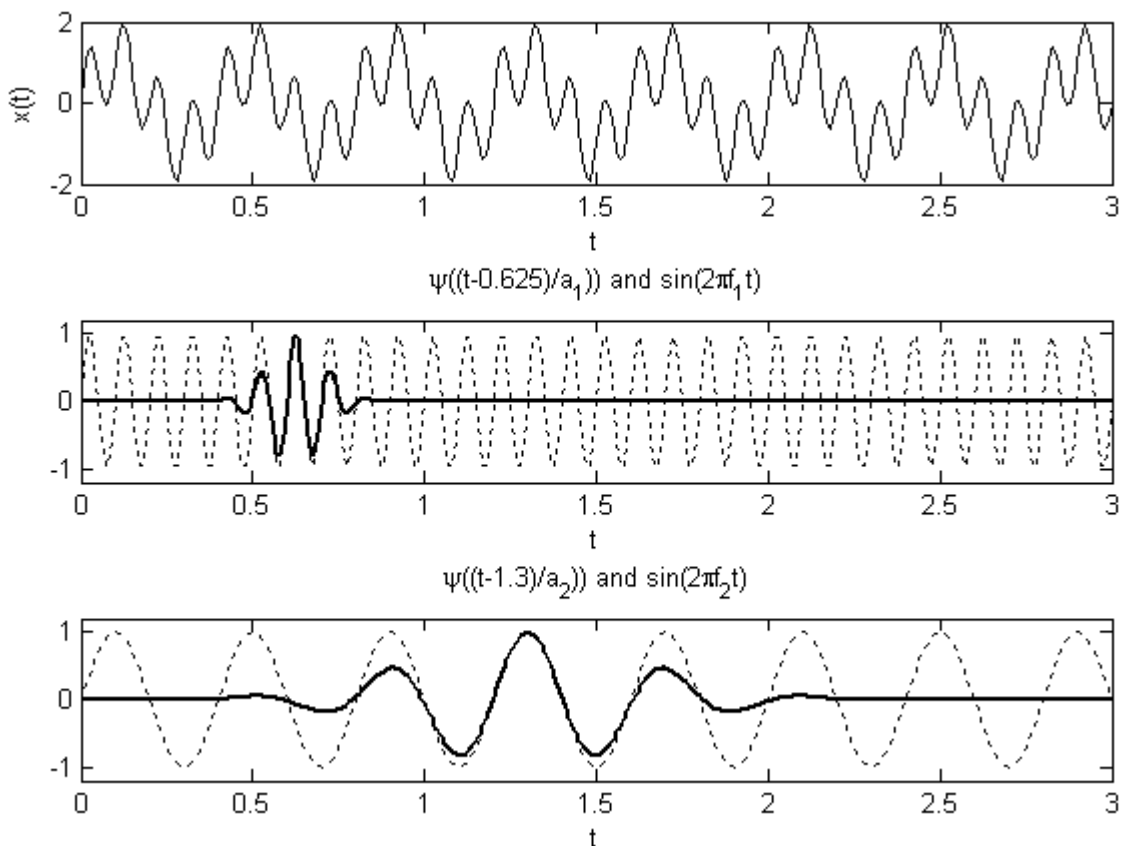


Рис. 64. График сигнала и вейвлетов.

На рис. 65 отчётливо видно два ряда максимумов (на иллюстрации максимумы обозначены тёмным цветом, на самом деле по умолчанию используется цветовая схема pink, в которой максимумам соответствует светлый цвет). Результат вычисления интеграла (135) в данном примере

принимает значения, близкие к нулю, когда фаза колебаний вейвлета Morlet отличается от фазы гармоники на четверть периода, максимальное по модулю значение получается в случае когда фазы совпадают или отличаются на на пол периода. В данном случае под фазой вейвлета понимается фаза сдвинутого и отмасштабированного косинуса, входящего в формулу (138).

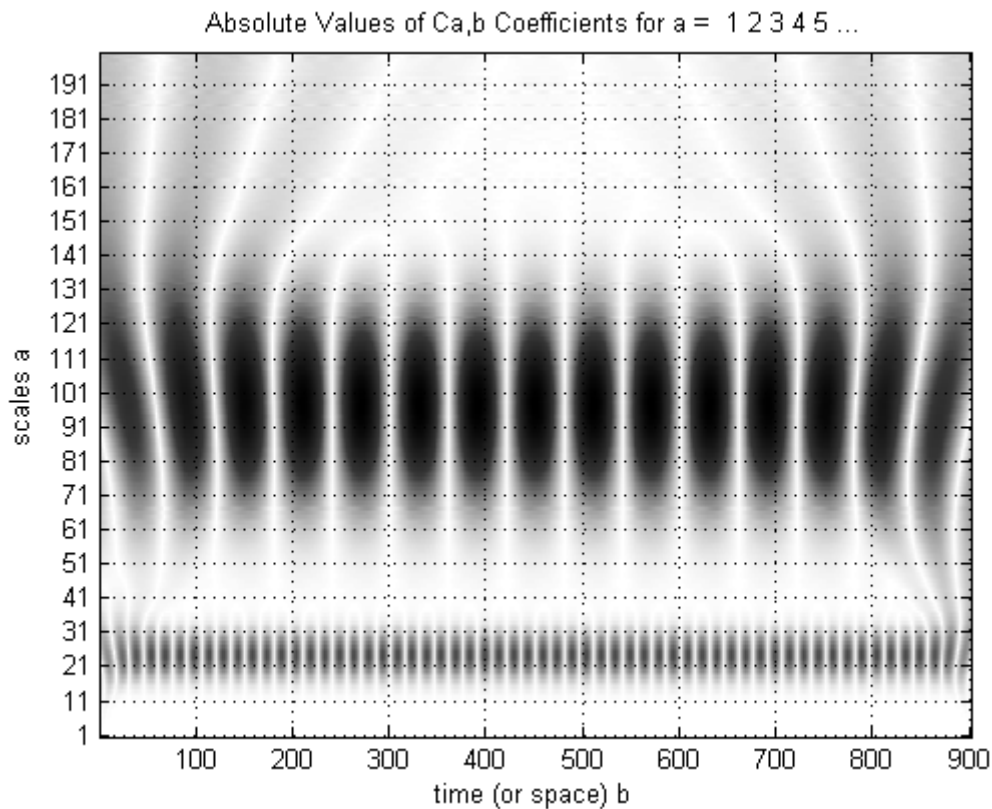


Рис. 65. Результат вейвлет преобразования.

Пример

В этом примере демонстрируется, как отразятся на результате вейвлет преобразования изменения частоты сигнала со временем. Частота сигнала меняется по линейному закону от 20Гц до 100Гц, частота квантования 1024Гц. Будем рассматривать диапазон частот от 16Гц до 256Гц, этому диапазону соответствуют диапазон масштабов от 1 до 64. Причём частоты обратно пропорциональны масштабу, это означает, что большим частотам будут соответствовать меньшие масштабы. Для того, чтобы результаты примера можно было сравнить с результатами на рис. 63, используется тот же вейвлет 'mexh'.

Анализ частотномодулированного сигнала (wavelet_example_2.m)

```
clear, clc, close all

Fs = 1024; % частота квантования
T = 0.25; % длительность сигнала
dt = 1/Fs; % период квантования
t = 0:dt:T; % время
f1 = 20; % частота сигнала
f2 = 100;
x = chirp(t, f1, T, f2); % сигнал
wname = 'mexh'; % используемый вейвлет
Fc = centfrq(wname); % центральная частота
% задаём границы диапазона частот,
% в котором хотим выполнить анализ сигнала
F_min = 16;
F_max = 256;
% так как вейвлет преобразование имеет дело
% с масштабом, а не с частотой, переводим
% частоту в масштаб
a_max = Fc/F_min/dt;
a_min = Fc/F_max/dt;
% значения масштабов, для которых будет вычислено
% вейвлет преобразование
a = a_min:1:a_max;

C = cwt(x, a, wname); % выполняем преобразование

% выводим результат на графике
figure,
subplot(2,1,1), plot(t, x, 'k')
xlabel('time'), ylabel('x(t)')
subplot(2,1,2), imagesc(t, a, abs(C))
axis xy, grid on
xlabel('b (time)'), ylabel('a (scale)')
```

Плавное увеличение частоты привело к смещению максимумов в сторону уменьшения масштабов, сами максимумы соответствуют максимумам и минимумам сигнала, и с ростом частоты они соответственно становятся меньшего размера и ближе друг к другу.

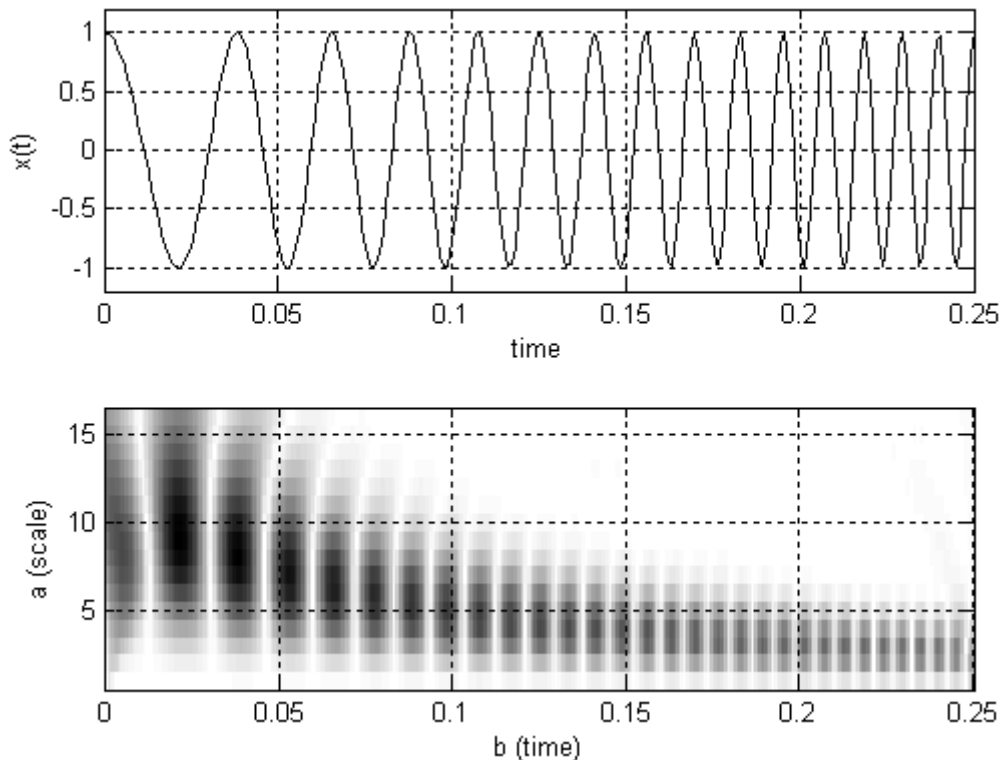


Рис. 66. Вейвлет преобразование частотномодулированного сигнала.

Пример

В отличие от предыдущего примера, частота меняется скачком таким образом, чтобы сигнал оставался непрерывным.

Скачок частоты сигнала (wavelet_example_3.m)

```
clear, clc, close all

Fs = 1024; % частота квантования
T = 0.25; % длительность сигнала
dt = 1/Fs; % период квантования
t = 0:dt:T; % время
f1 = 20; % частота сигнала
f2 = 80;
x = (t<0.125).*sin(2*pi*f1*t) - (t>=0.125).*sin(2*pi*f2*t); %
сигнал
wname = 'mexh'; % используемый вейвлет
Fc = centfrq(wname); % центральная частота
% задаём границы диапазона частот,
% в котором хотим выполнить анализ сигнала
F_min = 8;
F_max = 256;
% так как вейвлет преобразование имеет дело
% с масштабом, а не с частотой, переводим
% частоту в масштаб
a_max = Fc/F_min/dt;
a_min = Fc/F_max/dt;
```

```

% значения масштабов, для которых будет вычислено
% вейвлет преобразование
a = a_min:1:a_max;

C = cwt(x,a,wname); % выполняем преобразование

% выводим результат на графике
figure,
subplot(2,1,1), plot(t,x,'k')
xlabel('time'), ylabel('x(t)')
subplot(2,1,2), imagesc(t,a,abs(C))
axis xy, grid on
xlabel('b (time)'), ylabel('a (scale)')

```

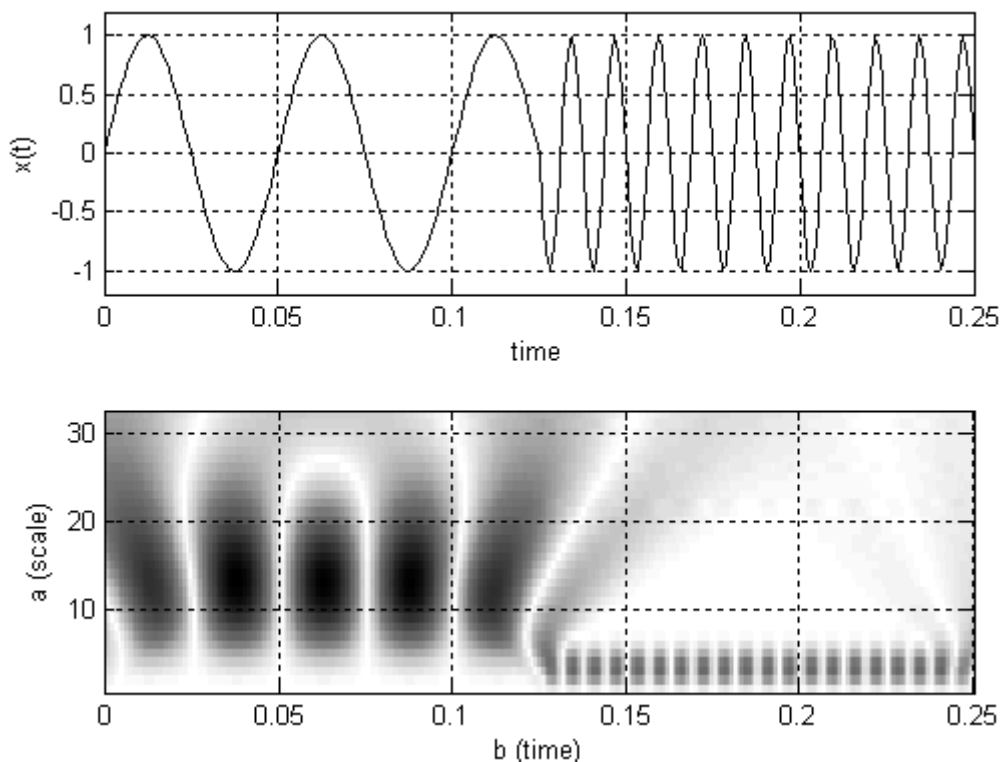


Рис. 67. Вейвлет преобразование сигнала, частота которого меняется скачком.

Как и следовало ожидать, в результате получилось две группы максимумов, на разных масштабах, каждый соответствует своей частоте и интервалу времени.

Пример

В этом примере разбирается случай, когда в гармоническом сигнале происходит скачкообразное изменение фазы.

$$x(t) = \begin{cases} \sin(\omega_0 t), & t < \tau \\ \sin(\omega_0 t + \varphi_0), & t \geq \tau \end{cases} \quad (139)$$

Для анализа будем использовать вейвлет Морле (138), он уже использовался в одном из примеров. С помощью этого вейвлета удаётся выделить непрерывную составляющую сигнала и отдельно локализованную во времени особенность. Стоит отметить, что похожий результат может быть получен и с использованием другого вейвлета, при этом существенных качественных отличий не будет.

Анализ сигнала со скачком фазы (wavelet_example_4.m)

```
clear, clc, close all

Fs = 1024; % частота квантования
T = 0.25; % длительность сигнала
dt = 1/Fs; % период квантования
t = 0:dt:T; % время
f = 20; % частота сигнала
% сигнал
x = (t<0.125).*sin(2*pi*f*t)+(t>=0.125).*sin(2*pi*f*(t-
0.012));

wname = 'morl'; % используемый вейвлет
Fc = centfrq(wname); % центральная частота
% задаём границы диапазона частот,
% в котором хотим выполнить анализ сигнала
F_min = 8;
F_max = 256;
% так как вейвлет преобразование имеет дело
% с масштабом, а не с частотой, переводим
% частоту в масштаб
a_max = Fc/F_min/dt;
a_min = Fc/F_max/dt;
% значения масштабов, для которых будет вычислено
% вейвлет преобразование
a = a_min:1:a_max;

C1 = cwt(x,a,wname); % выполняем преобразование
% отдельно выполняем преобразование на масштабе,
% соответствующем частоте сигнала
C2 = cwt(x,Fc/f/dt,wname);

% выводим результат на графике
figure,
subplot(3,1,1), plot(t,x,'k')
xlabel('time'), ylabel('x(t)')
subplot(3,1,2), imagesc(t,a,abs(C1))
axis xy, grid on
xlabel('b (time)'), ylabel('a (scale)')
subplot(3,1,3), plot(t,C2,'k')
xlabel('time'), ylabel('C_2(t)')
```

```

% отдельно вычисляем вейвлет преобразование
% на мелких масштабах, чтобы выделить скачок
% значения масштабов выбирались исходя из графика
a = 1:0.1:5;
C1 = cwt(x,a,wname);
% также построим график преобразования для
% одного выбранного значения масштаба
C2 = cwt(x,2,wname);
% выводим результат на графике
figure,
subplot(3,1,1), plot(t,x,'k')
xlabel('time'), ylabel('x(t)')
subplot(3,1,2), imagesc(t,a,abs(C1))
axis xy, grid on
xlabel('b (time)'), ylabel('a (scale)')
subplot(3,1,3), plot(t,C2,'k')
xlabel('b (time)'), ylabel('C(a=2,b)')

```

Вейвлет преобразование выделяет особенности сигналов, имеющие разный масштаб по времени. Соответственно, вычисляя интеграл (135) при фиксированном значении масштаба, получим функцию времени, поведение которой отражает детали сигнала, имеющие этот масштаб.

Рассмотрим результат непрерывного вейвлет преобразования сигнала, который используется в этом примере. Результат вейвлет преобразования – это вещественная функция от двух аргументов. Она может быть представлена в виде поверхности в трёхмерной системе координат (рис. 68). На графике чётко видно пять больших максимумов, идущих друг за другом вдоль одной линии. Кроме того, на этом графике заметны два небольших «горба» перед средним максимумом.

Теперь проанализируем полученный результат более детально. Для этого вместо трёхмерной картинки построим двумерный график вейвлет преобразования, на котором цветом показана величина $|C(a,b)|$. На иллюстрациях максимумам соответствует тёмный цвет. Это всё равно, что посмотреть на поверхность сверху.

Кроме того, для отдельных выбранных масштабов будем строить график зависимости результата преобразования от времени. Такой график зависимости от времени мы бы увидели, если посмотрели, что лежит в сечении поверхности вертикальной плоскостью, параллельной оси времени (рис. 69). Подобным образом исследуем поведение $C(a,b)$ на двух масштабах. В одном случае выберем масштаб, соответствующий частоте синусоидального сигнала, в программе он вычисляется по формуле (137) $f_c/f/dt$. На этом масштабе видно, что в сигнале присутствовала периодическая составляющая. На рис. 70 на верхнем графике показан исходный сигнал, на среднем графике – результат вейвлет преобразования для масштабов, где присутствуют следы периодического сигнала,

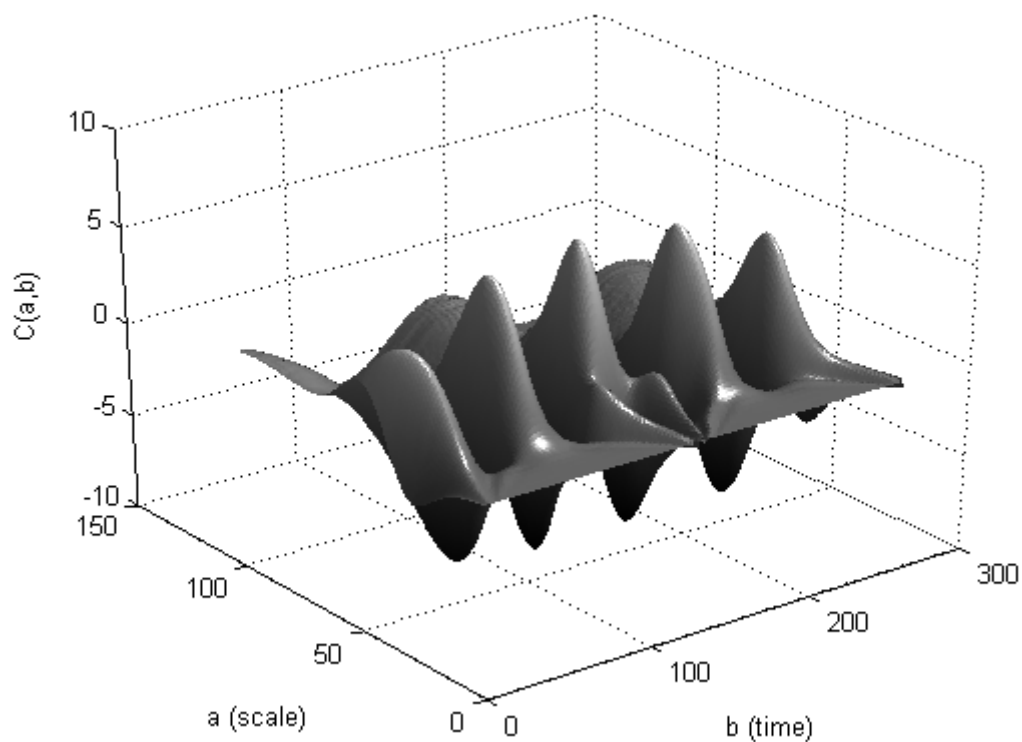


Рис. 68. Результат вейвлет преобразования синусоидального сигнала со скачком.

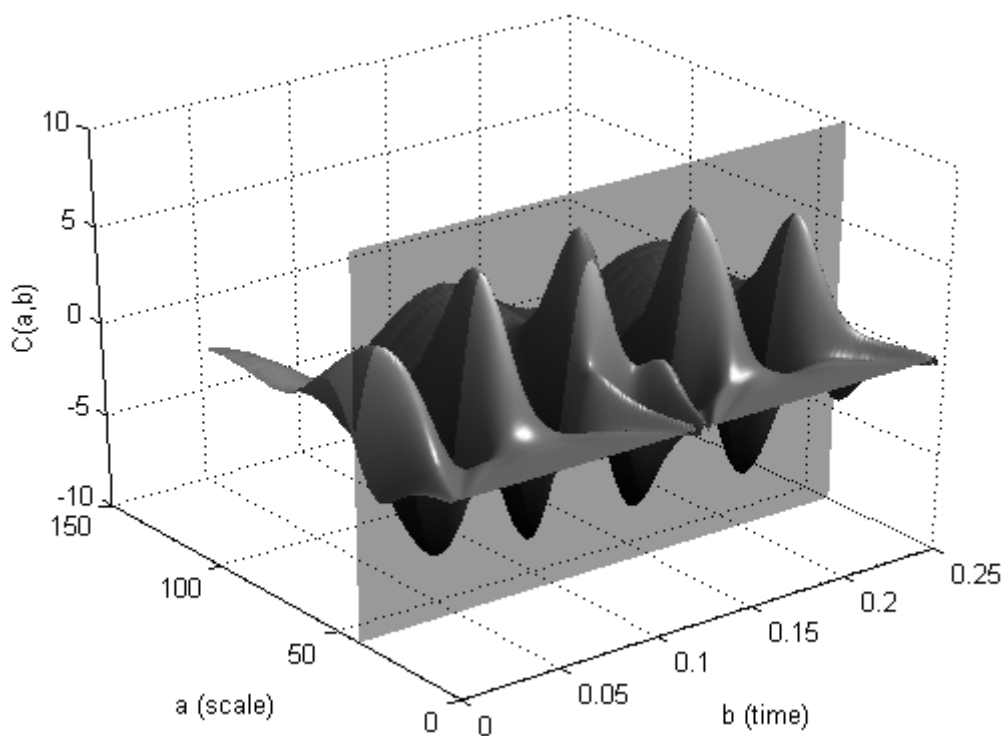


Рис. 69. Сечение поверхности плоскостью.

на нижнем графике построен тот график вейвлет преобразования при фиксированном масштабе, это как раз то, что находится в сечении поверхности плоскостью. Отметим, что на нижнем графике нет следов скачка, так как скачок имеет значительно меньший масштаб по сравнению с периодом синуса.

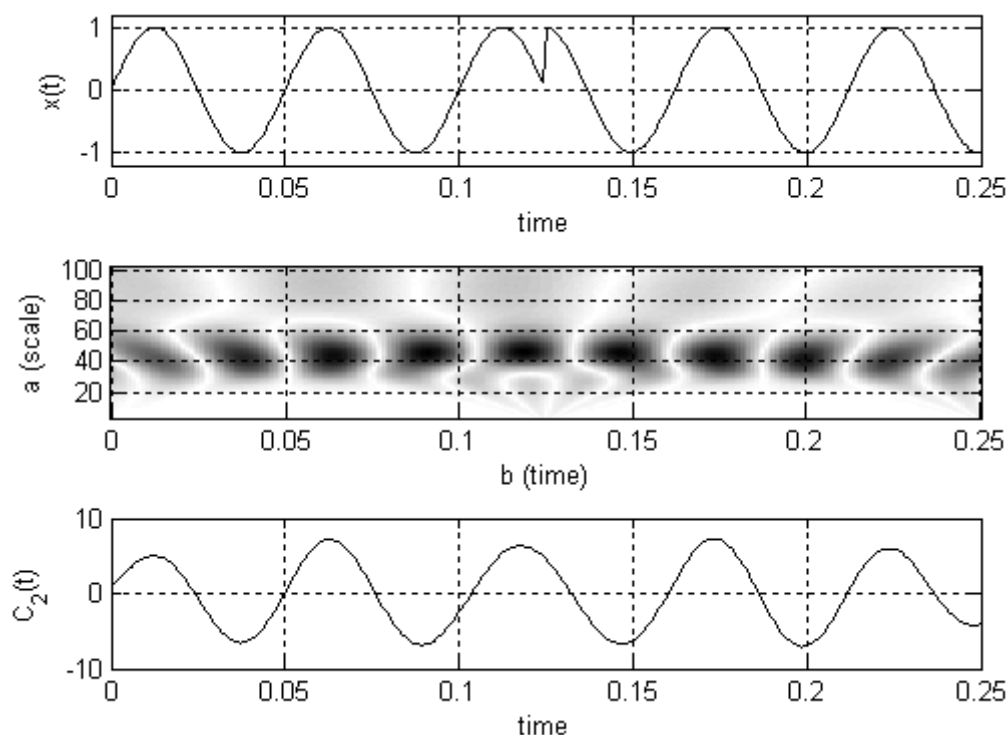


Рис. 70. Выделение крупномасштабных деталей.

Теперь проанализируем результат вейвлет преобразования в области меньших масштабов, там должен присутствовать след от резкого скачка. Рассмотрим масштабы порядка 1. Если частота квантования 1024Гц, и центральная частота вейвлета Морле 0.8125Гц, то масштабы порядка 1, согласно формуле (137), соответствуют частотам порядка сотен герц или деталям сигнала, локализованным в интервале времени порядка 10^{-2} - 10^{-3} с. На рис. 71 на результате вейвлет преобразования виден след от скачка, а также похожий след по краям, причём слева он значительно меньше, чем справа, а тот, что справа, по величине такой же, как и средний. Наличие этих следов по краям объясняется тем, что сигнал за пределами отрезка времени, на котором была сделана запись, считается равным нулю. Справа, если продолжить сигнал нулевыми отсчётами, получится ещё один скачок сигнала, причём той же величины, что и посередине, поэтому след от него такой же заметный. А слева, при продолжении сигнала нулевыми отсчётами, сигнал остаётся непрерывным, но первая производная рвётся. Разрыв первой производной при использовании для анализа данного

вейвлета, оставляет слабый след по сравнению со следом от разрывов самого сигнала.

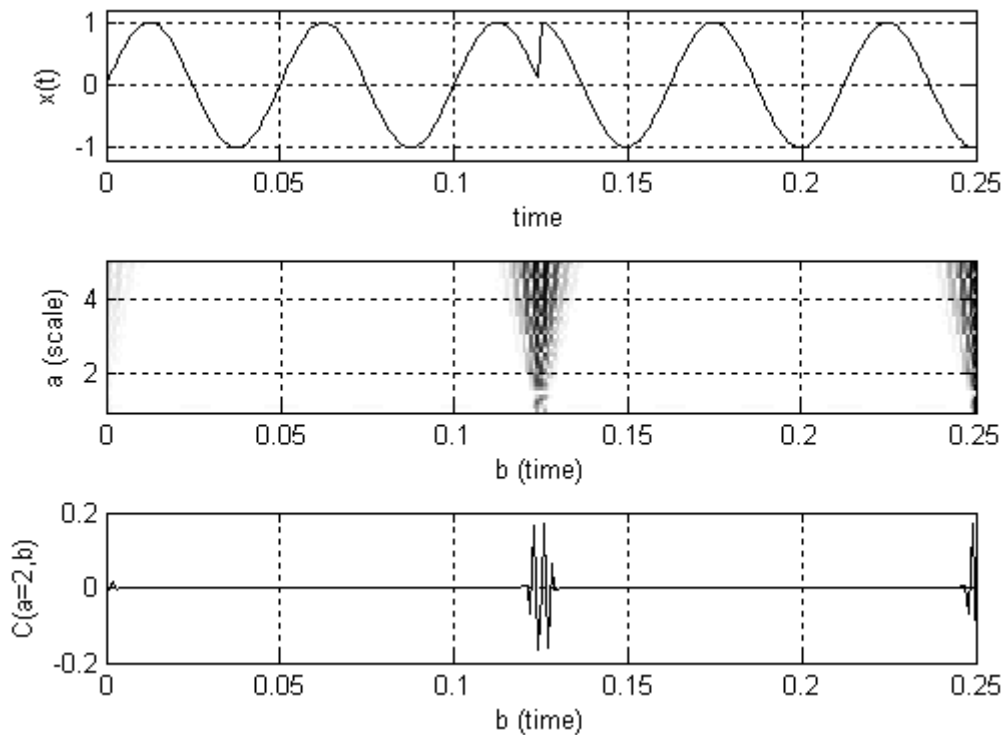


Рис. 71. Выделение мелкомасштабных деталей.

Пример

Составим сигнал, который состоит из двух функций "мексиканская шляпа", имеющих разный масштаб по времени. Вейвлет "мексиканская шляпа" задан формулой:

$$\psi(t) = \frac{2}{\sqrt{3}} \pi^{-\frac{1}{4}} (1-t^2) e^{-\frac{t^2}{2}} \quad (140)$$

Сигнал, который будет подвергнут вейвлет анализу, задан как сумма двух этих функций:

$$x(t) = \psi(t-5) + \psi\left(\frac{t-25}{3}\right) \quad (141)$$

В примере функция (140) задана в явном виде и хранится в переменной mh, с помощью этой функции создаётся вектор с отсчётами сигнала. Сначала с помощью функции swt выполняется непрерывное вейвлет преобразование. Вторым аргументом указан вектор 1:64, чтобы получить значения масштабов, значения вектора надо умножить на период квантования $dt = 0.1$. Третьим аргументом указано название вейвлета, используемого

для анализа, в данном случае это 'mexh'. Чтобы построить график, последним аргументом указан 'plot'. Сигнал состоит из суммы двух функций (140), одна из которых сдвинута на $5c$, а другая сдвинута на $25c$ и растянута во времени в 3 раза. Второй раз вейвлет преобразование вычисляется для масштабов равных 1 и 3. В этом случае результат не выводится на график, а сохраняется в переменной c . Переменная c – это матрица, состоящая из двух строк, в первой строке результат преобразования для масштаба, равного 1, вторая строчка – для 3.

Пример вейвлет преобразования (test_cwt.m)

```
clear, clc, close all

% функция mexican hat
mh = @(t) (2/sqrt(3)*pi^(-1/4))*(1-t.^2).*exp(-t.^2/2);

dt = 0.1;      % шаг по времени
t = 0:dt:40;  % время
x = mh(t-5) + mh((t-25)/3); % сигнал

% вычисляем непрерывное вейвлет преобразование
cwt(x, 1:64, 'mexh', 'plot');

% вычисляем непрерывное вейвлет преобразование для двух
масштабов
C = cwt(x, [1 3]/dt, 'mexh');

figure
% вейвлет 'mexh', сдвинутый на 5с вправо
subplot(5,1,1), plot(t,mh(t-5),'k')
xlabel('t'), title('\psi(t-5)')
% вейвлет 'mexh', сдвинутый на 25с вправо и растянутый в 3
раза
subplot(5,1,2), plot(t,mh((t-25)/3),'k')
xlabel('t'), title('\psi((t-25)/3)')
% сигнал из суммы двух импульсов
subplot(5,1,3), plot(t,x,'k')
xlabel('t'), title('x(t)')
% преобразование вейвлет на масштабе a = 1
subplot(5,1,4), plot(t,C(1,:), 'k')
xlabel('b'), title('C(a=1, b)')
% преобразование вейвлет на масштабе a = 3
subplot(5,1,5), plot(t,C(2,:), 'k')
xlabel('b'), title('C(a=3, b)')
```

После выполнения первой команды `cwt` будет построен график (рис. 72) на котором цветом обозначен модуль величины $C(a,b)$. Для графика используется цветовая схема `Pink`, в которой максимальному значению соответствует белый цвет, минимальному – чёрный, а промежуточным –

оттенки розового. Для создания иллюстрации цветовая схема была изменена так, что максимальному значению соответствует чёрный цвет, минимальному – белый, а промежуточным – оттенки серого. Во втором окне строится пять графиков (рис. 73). На первом графике изображён вейвлет «мексиканская шляпа», сдвинутый на 5с. На втором графике вейвлет, сдвинутый на 25с и растянутый в 3 раза. На третьем графике сигнал,

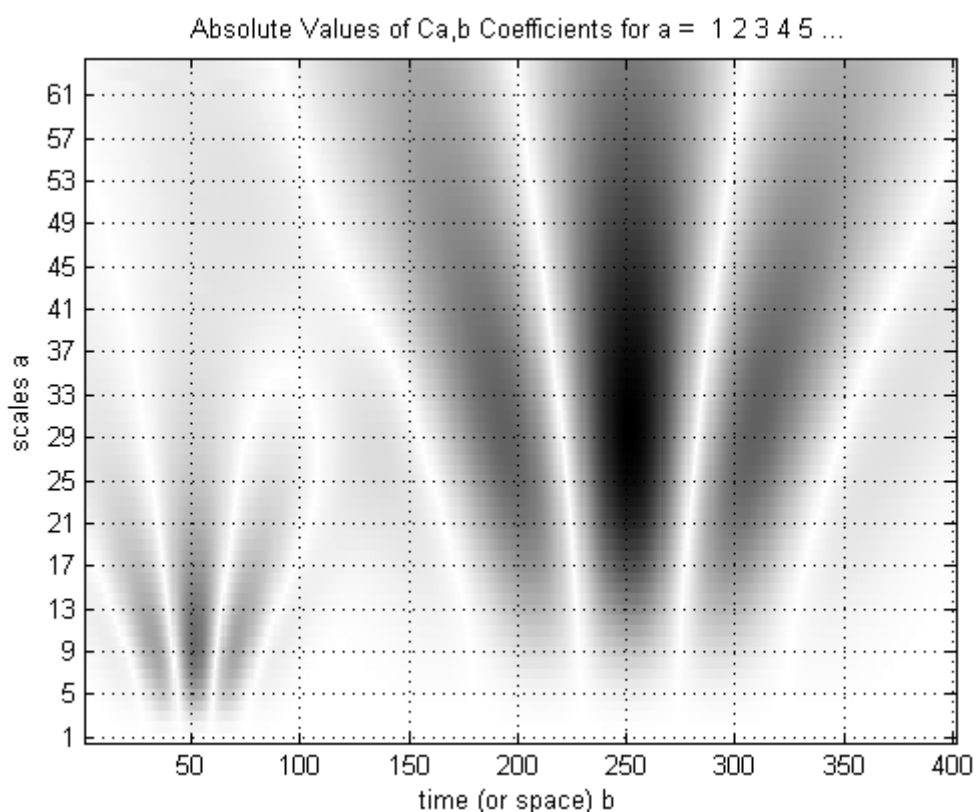


Рис. 72 Результат вейвлет преобразования.

образованный их суммой. Четвёртый и пятый график это результат вейвлет преобразования для масштабов 1 и 3 соответственно. Из графиков видно, что максимальные значения $C(a,b)$ достигается, когда значение сдвига для вейвлета соответствующего масштаба совпадает с моментом времени, когда в сигнале встречается отрезок наиболее похожей формы.

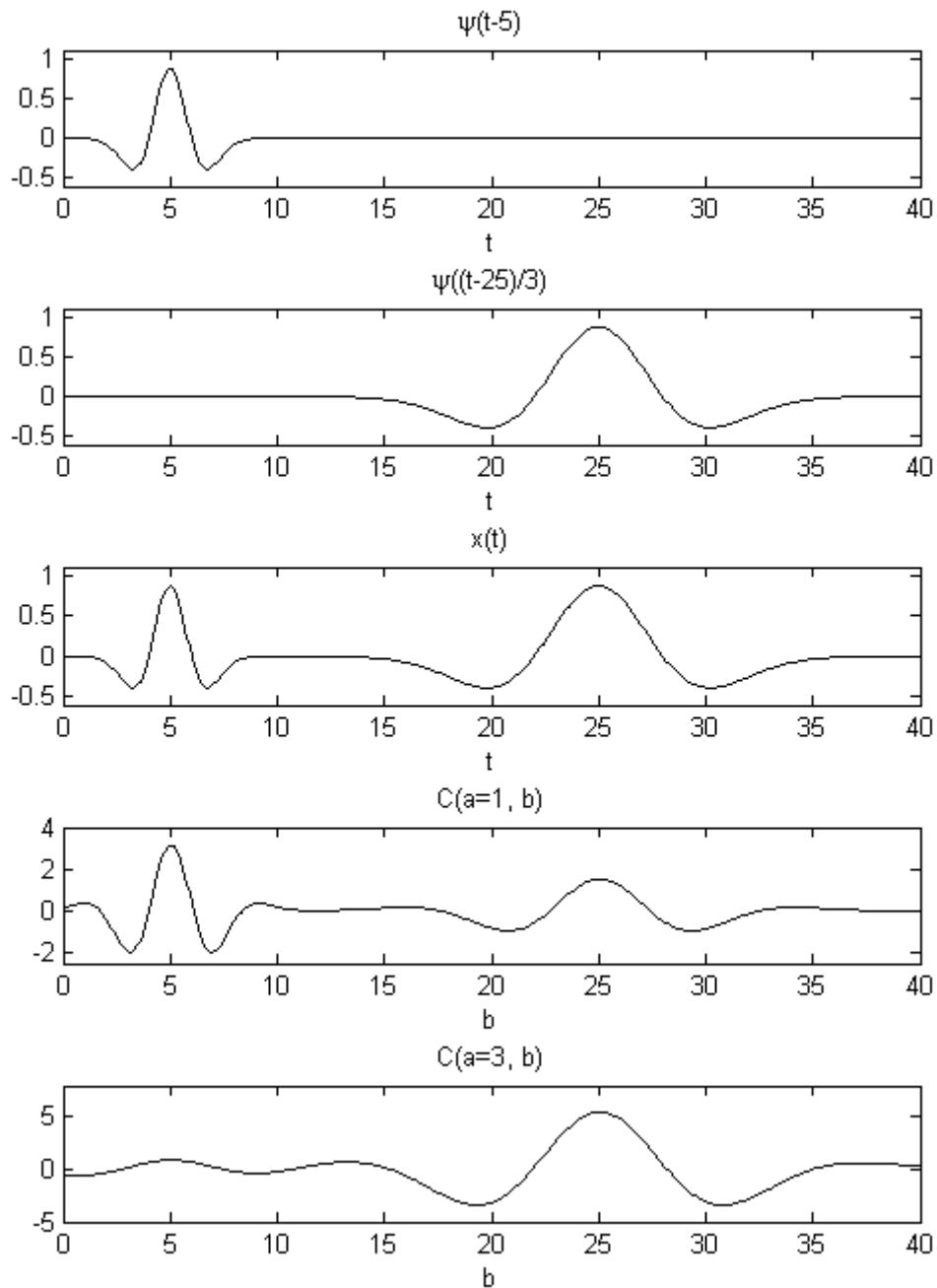


Рис. 73 Графики вейвлетов, сигнала и результата преобразования.

Пример

Сигнал состоит из суммы гармонического сигнала с частотой 100Гц, к которому прибавлены прямоугольные импульсы, длительностью 0,1 и 0,3с, середина первого импульса 0,5с, второго 2с. Вычислим вейвлет преобразование такого сигнала, используя вейвлет Хаара и вейвлет Морле, и сравним полученные результаты. Вейвлет Хаара имеет форму двух полярного прямоугольного импульса (рис. 60, слева вверху). Вейвлет такой формы хорошо выделяет разрывы функции (рис. 74), но вместо одного гармонического сигнала получается его многократное повторение по оси масштабов. Противоположный результат получается при

использовании вейвлета Морле (рис. 75). Гармонический сигнал в виде чередующихся максимумов и минимумов, а фронты прямоугольных импульсов получились размытыми по сравнению с результатом, полученным при использовании вейвлета Хаара.

Сравнение вейвлетов Хаара и Морле (`haar_and_morlet.m`)

```
clear, clc, close all

Ts = 0.001; % Период квантования
t = 0:Ts:3; % время
% сигнал из двух импульсов и синуса
x = rectpuls(t-0.5,0.1)+rectpuls(t-2,0.3)+sin(2*pi*100*t);

% строим графики результатов вейвлет преобразований
figure, cwt(x, 1:128, 'haar', 'plot'); % с вейвлетом Хаара
figure, cwt(x, 1:128, 'morl', 'plot'); % с вейвлетом Морле
```

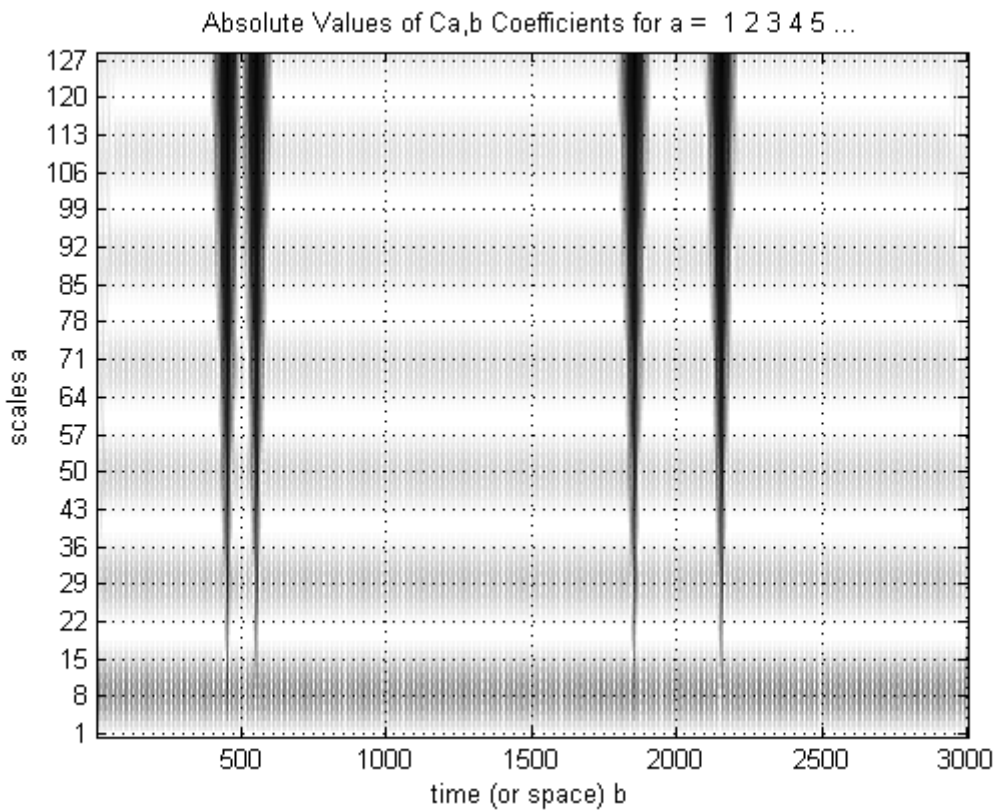


Рис. 74 Результат преобразования с вейвлетом Хаара.

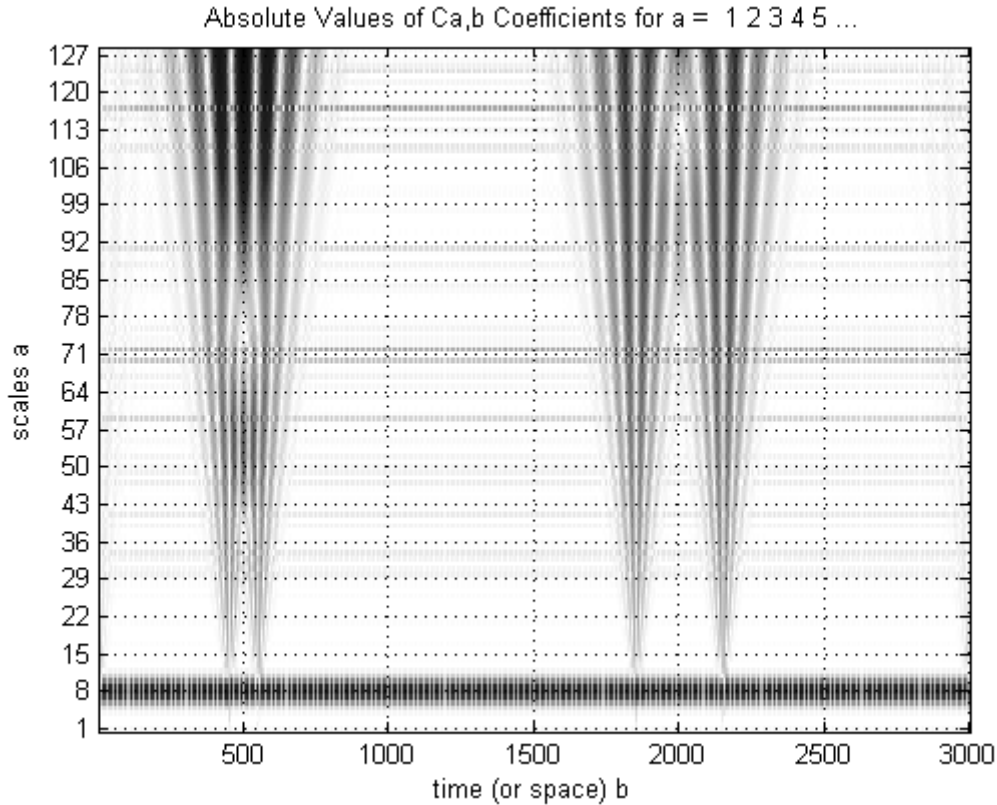


Рис. 75 Результат преобразования с вейвлетом Морле.

7.2. Дискретное вейвлет преобразование

Когда вейвлет преобразование применяется для фильтрации, сжатия информации или аппроксимации, непрерывное преобразование для этих целей даёт избыточную информацию. При выполнении определённых условий, о которых будет сказано ниже, вейвлет преобразование необходимо вычислить только для некоторых значений масштабов и сдвигов. По аналогии с разложением в тригонометрический ряд, где сигнал раскладывался в ортонормированном базисе, состоящим из гармонических функций с разной частотой, в дискретном вейвлет преобразовании сигнал раскладывается по ортонормированному базису, состоящего из семейства функций, образованных сдвигом и изменением масштаба оригинального вейвлета $\psi(t)$. Семейство функций, образующих базис:

$$\psi_{j,k}(t) = 2^{-\frac{j}{2}} \psi(2^{-j}t - k), \quad j, k \in Z \quad (142)$$

То есть масштаб равен степени двойки $a = 2^j$, а сдвиг пропорционален масштабу $b = ka$. Масштабный множитель равный степени 2 используется

в алгоритме быстрого вейвлет преобразования. Тогда сигнал $x(t)$ представляется в виде линейной комбинации функций $\psi_{j,k}(t)$

$$x(t) = \sum_{j=-\infty}^{+\infty} \sum_{k=-\infty}^{+\infty} c_{j,k} \psi_{j,k}(t) \quad (143)$$

Функции $\psi_{j,k}(t)$ образуют ортонормированный базис, то есть скалярное произведение двух таких функций равно единице, если значки попарно совпадают, и нулю во всех остальных случаях.

$$\int_{-\infty}^{+\infty} \psi_{j,k}(t) \psi_{l,m}(t) dt = \delta_{j,l} \delta_{k,m} \quad (144)$$

Тогда, чтобы получить коэффициенты разложения, надо умножить скалярно (143) на $\psi_{l,m}(t)$.

$$c_{j,k} = \int_{-\infty}^{+\infty} x(t) \psi_{j,k}(t) dt \quad (145)$$

Вейвлеты, используемые в дискретном преобразовании, имеют так называемую масштабирующую функцию $\varphi(t)$, и соответственно также семейство сдвинутых с разными масштабами функций:

$$\varphi_{j,k}(t) = 2^{-\frac{j}{2}} \varphi(2^{-j}t - k), \quad j, k \in Z \quad (146)$$

В отличие от вейвлетов, масштабирующая функция должна быть ортогональна только сдвинутым масштабирующим функциям на том же масштабе, и интеграл от функции по всей оси равен единице.

$$\begin{aligned} \int_{-\infty}^{+\infty} \varphi_{j,k}(t) \varphi_{j,l}(t) dt &= \delta_{k,l} & \int_{-\infty}^{+\infty} \varphi_{j,k}(t) dt &= 1 \\ \int_{-\infty}^{+\infty} \varphi_{j,k}(t) \psi_{l,m}(t) dt &= 0, \quad l \leq j \end{aligned} \quad (147)$$

Кратномасштабный анализ сигнала заключается в разложении сигнала по разным ортонормированным базисам, состоящим из функций $\varphi_{j,k}(t)$ и $\psi_{j,k}(t)$. Базисы должны обладать полнотой в пространстве интегрируемых

с квадратом функций L^2 . Свойство полноты гарантирует, что любая функция из пространства L^2 может быть разложена по такому базису. Предположим, что существует наименьший масштаб $j = J_{\min}$, на котором сигнал представляется в виде линейной комбинации масштабирующих функций $\varphi_{j,k}(t)$.

$$x(t) = \sum_{k=-\infty}^{+\infty} s_{J_{\min},k} \varphi_{J_{\min},k}(t) \quad (148)$$

На практике может быть выбран такой масштаб J_{\min} , при котором сигнал будет с наперёд заданной точностью аппроксимироваться таким разложением по масштабирующим функциям. Интеграл от масштабирующей функции при любом значении масштаба равен единице, а носитель, отрезок на котором функция отлична от нуля, определяется выбранным масштабом, поэтому в пределе при $J_{\min} \rightarrow -\infty$ разложение по масштабирующим функциям превращается в разложение по дельта-функциям.

Характерной особенностью кратномасштабного анализа является разложение одного и того же сигнала в разных базисах, соответствующих разным масштабам, причём переход от одного базиса к другому происходит определённым образом и не зависит от масштаба. Чтобы перейти к новому базису, соответствующему масштабу с номером на 1 больше, надо в текущем базисе разложить масштабирующие функции через масштабирующие функции и вейвлеты с масштабом на 1 больше. На рис. 76 показана последовательность преобразования базиса, по которому раскладывается сигнал.

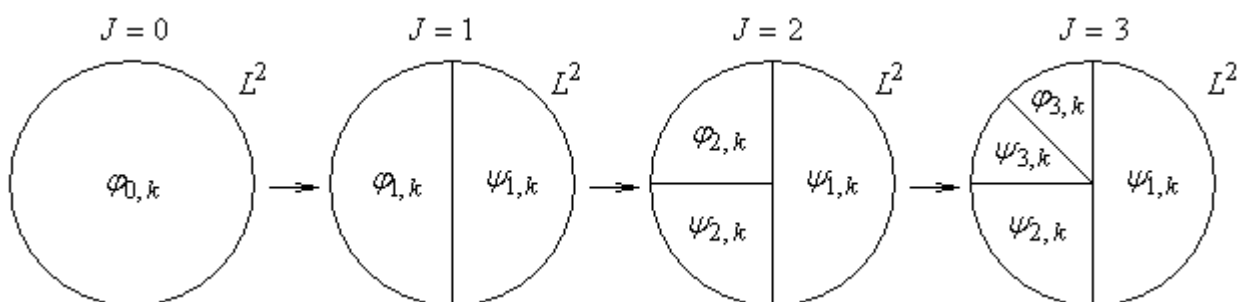


Рис. 76. Последовательное преобразование базиса.

На самом наименьшем масштабе базис состоит из одних только масштабирующих функций, по этому базису может быть разложена любая функция, интегрируемая с квадратом. Эти масштабирующие функции с первым индексом, равным 0, раскладываются по базису из масштабирующих функций и вейвлетов с первым индексом 1. Новый базис

также обладает полнотой в пространстве L^2 . Далее масштабирующие функции с индексом 1 раскладываются через масштабирующие функции и вейвлеты с первым индексом 2. Итак, на каждом следующем шаге получается базис, состоящий из вейвлетов с различными масштабами и масштабирующих функций с одним большим масштабом. Для произвольного масштаба J справедливо разложение:

$$x(t) = \sum_{k=-\infty}^{+\infty} s_{J,k} \varphi_{J,k}(t) + \sum_{j=-\infty}^J \sum_{k=-\infty}^{+\infty} d_{j,k} \psi_{j,k}(t) \quad (149)$$

Коэффициенты разложения находятся также из условия ортогональности:

$$s_{j,k} = \int_{-\infty}^{+\infty} x(t) \varphi_{j,k}(t) dt \quad d_{j,k} = \int_{-\infty}^{+\infty} x(t) \psi_{j,k}(t) dt \quad (150)$$

В основе кратномасштабного анализа лежит ещё одно условие, которое предъявляется к вейвлету и масштабирующей функцией. Это условие позволяет перейти от разложения сигнала на уровне J к разложению сигнала на уровне $J-1$ и наоборот. Для этого масштабирующие функции и вейвлеты с масштабом $j=J$ должны быть представимы в виде линейных комбинаций сдвинутых масштабирующих функций с масштабом $j=J-1$. Это условие означает, что базис с масштабом J должен раскладываться по базису с масштабом $J-1$, причём разложение одного базиса по другому не должно зависеть от номера J .

$$\begin{aligned} \varphi_{J,0}(t) &= \sum_{k=0}^{N-1} h_k \varphi_{J-1,k}(t) \\ \psi_{J,0}(t) &= \sum_{k=0}^{N-1} g_k \varphi_{J-1,k}(t) \end{aligned} \quad (151)$$

Количество коэффициентов N зависит от выбранного типа вейвлетов. Например, для вейвлетов Добеши $N = 2M$, где M – порядок вейвлета.

Так как коэффициенты h_k и g_k не зависят от масштаба, то эти выражения обычно записывают для оригинального вейвлета и масштабирующей функции.

$$\begin{aligned} \varphi(t) &= \sqrt{2} \sum_{k=0}^{N-1} h_k \varphi(2t-k) \\ \psi(t) &= \sqrt{2} \sum_{k=0}^{N-1} g_k \varphi(2t-k) \end{aligned} \quad (152)$$

Коэффициенты h_k и g_k находятся из условия ортогональности масштабирующих функций одного масштаба друг другу и вейвлету.

$$h_k = \int_{-\infty}^{+\infty} \varphi(t)\varphi(2t-k)dt \quad g_k = \int_{-\infty}^{+\infty} \psi(t)\varphi(2t-k)dt \quad (153)$$

Коэффициенты h_k и g_k должны подчиняться условиям, аналогичным условиям для функций $\varphi(t)$ и $\psi(t)$.

Из условия ортогональности сдвинутых масштабирующих функций следует:

$$\int_{-\infty}^{+\infty} \varphi(t)\varphi(t-m)dt = \delta_{0,m} \Rightarrow \sum_{k=0}^{N-1} h_k h_{k+2m} = \delta_{0,m} \quad (154)$$

$$\int_{-\infty}^{+\infty} \varphi(t)\varphi(t-m)dt = \sum_{k=0}^{N-1} h_k \int_{-\infty}^{+\infty} \varphi(t)\varphi(2t-(k+2m))dt = \sum_{k=0}^{N-1} h_k h_{k+2m}$$

Из условия нормировки масштабирующей функции следует:

$$\int_{-\infty}^{+\infty} \varphi(t)dt = 1 \Rightarrow \sum_{k=0}^{N-1} h_k = \sqrt{2} \quad (155)$$

Из условия ортогональности вейвлета масштабирующим функциям следует:

$$\int_{-\infty}^{+\infty} \psi(t)\varphi(t-m)dt = 0 \Rightarrow \sum_{k=0}^{N-1} h_k g_{k+2m} = 0 \quad (156)$$

$$\int_{-\infty}^{+\infty} \psi(t)\varphi(t-m)dt = \sum_{k=0}^{N-1} h_k \int_{-\infty}^{+\infty} \psi(t)\varphi(2t-(k+2m))dt = \sum_{k=0}^{N-1} h_k g_{k+2m}$$

Непосредственно из этого условия получается взаимосвязь между коэффициентами.

$$g_k = (-1)^k h_{N-1-k} \quad (157)$$

Это соотношение играет важную роль в алгоритме быстрого вейвлет преобразования, о котором будет сказано ниже.

Точность и масштабная разрешающая способность аппроксимации анализируемых функций вейвлетами зависит от их гладкости, то есть от их порядка дифференцируемости. При использовании вейвлет

преобразования для сжатия информации, путём отбрасывания малозначимых коэффициентов разложения, вейвлеты с более высокой гладкостью обеспечивают более качественное восстановление сигнала. Для обеспечения знакопеременности и гладкости до степени $M-1$ вейвлеты должны быть ортогональны полиномам соответствующих степеней.

$$\int_{-\infty}^{+\infty} t^m \psi(t) dt = 0, \quad m = 0, 1, \dots, M-1 \Rightarrow \sum_{k=0}^{N-1} k^m g_k = 0 \quad (158)$$

Для представления принципа кратномасштабного вейвлет преобразования запишем выражение (143) в более простом виде. Для этого введём новые обозначения. Обозначим сумму всех сдвинутых вейвлетов с одним масштабом:

$$D_j(t) = \sum_{k=-\infty}^{+\infty} c_{j,k} \psi_{j,k}(t) \quad (159)$$

Функция $D_j(t)$ описывает характерные детали поведения сигнала, имеющие масштаб, соответствующий номеру j . Чем больше номер j , тем более протяжённый масштаб изменений сигнала описывает функция $D_j(t)$. Соответственно, если сложить такие функции со всевозможными масштабами, то получим исходный сигнал. Другими словами, сигнал представляется как сумма функций, каждая из которых сама по себе описывает изменение сигнала с определённым масштабом.

$$x(t) = \sum_{j=-\infty}^{+\infty} D_j(t) \quad (160)$$

Выберем значение масштаба $j = J$. Из суммы по всем значениям масштабов (160) часть слагаемых с номерами $j > J$ описывает приближенное поведение сигнала и имеет соответственно масштаб изменения не меньше, чем масштаб, соответствующий номеру J . То есть чем больше J , тем более грубое приближение сигнала получается. Введём для этой суммы обозначение:

$$A_J(t) = \sum_{j=J+1}^{+\infty} D_j(t) \quad (161)$$

Тогда сигнал представляется в виде суммы приближения сигнала $A_J(t)$ и всех остальных слагаемых в сумме по j , образующих мелкомасштабные изменения сигнала.

$$x(t) = A_J(t) + \sum_{j=-\infty}^J D_j(t) \quad (162)$$

$A_J(t)$ называется аппроксимацией сигнала на масштабе J , а $D_j(t)$ – детализацией. Теперь выделим из суммы в (161) одно слагаемое с номером $J+1$, сумма из оставшихся слагаемых равна $A_{J+1}(t)$.

$$A_J(t) = A_{J+1}(t) + D_{J+1}(t) \quad (163)$$

Это рекуррентное соотношение описывает взаимосвязь приближений сигналов, полученных на разных масштабах, и роль функции $D_j(t)$. То есть более точное приближение сигнала получается как сумма грубого приближения и мелкомасштабных изменений.

Пусть приближение с масштабом изменений, соответствующих номеру $J=0$ – это и есть в точности сам сигнал. Тогда, используя соотношение (163), сигнал представляется как сумма более грубого приближения и мелкомасштабных изменений, причём этот процесс разложения можно продолжить и дальше до интересующего масштаба.

$$\begin{aligned} x(t) &= A_0(t) = \\ &= A_1(t) + D_1(t) = \\ &= A_2(t) + D_2(t) + D_1(t) = \\ &= A_3(t) + D_3(t) + D_2(t) + D_1(t) = \dots \end{aligned} \quad (164)$$

Сравнивая формулу (162) с (149), становится очевидным ещё одно выражение для $A_J(t)$, записанное через масштабирующие функции, аналогично тому, как $D_J(t)$ выражается через вейвлеты (159).

$$A_J(t) = \sum_{k=-\infty}^{+\infty} s_{J,k} \varphi_{J,k}(t) \quad (165)$$

Функции $A_J(t)$ и $D_j(t)$ обладают свойством ортогональности, это объясняется тем, что они выражаются как линейная комбинация функций $\varphi_{j,k}(t)$ и $\psi_{j,k}(t)$, которые в свою очередь обладают свойствами ортогональности (144) и (147). Из свойств ортогональности следует:

$$\int_{-\infty}^{+\infty} D_j(t)D_k(t)dt = 0, \quad j \neq k$$

$$\int_{-\infty}^{+\infty} A_j(t)D_k(t)dt = 0, \quad k \leq J$$
(166)

На примере вейвлета Хаара продемонстрируем основные принципы кратномасштабного анализа. Оригинальный вейвлет Хаара $\psi(t)$ и соответствующая ему масштабирующая функция $\varphi(t)$ определены следующем образом:

$$\psi(t) = \begin{cases} 1, & t \in \left[0, \frac{1}{2}\right) \\ -1, & t \in \left[\frac{1}{2}, 1\right) \\ 0, & t \notin [0, 1) \end{cases} \quad \varphi(t) = \begin{cases} 1, & t \in [0, 1) \\ 0, & t \notin [0, 1) \end{cases}$$
(167)

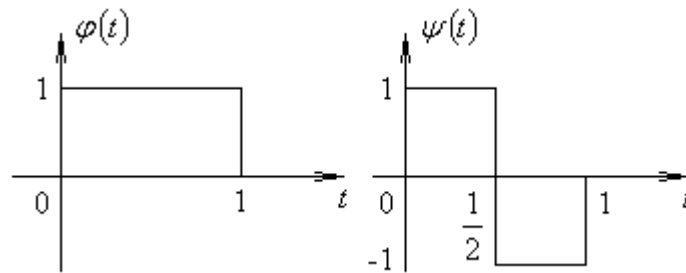


Рис. 77. Масштабирующая функция и вейвлет Хаара.

Из рис. 77 видно, что интеграл от $\psi(t)$ равен нулю, а интеграл от $\varphi(t)$ равен единице. Семейства сдвинутых и масштабированных функций $\psi_{j,k}(t)$ и $\varphi_{j,k}(t)$ определяется формулами (142) и (146). Множитель $2^{-\frac{j}{2}}$ обеспечивает нормировку этих функций на 1 при любом масштабе. Проверим свойства ортогональности (144) и (147). Например, $\psi_{0,0}(t)$ отличен от нуля на отрезке от 0 до 1, а вейвлет, у которого второй значок отличается хотя бы на 1, например, $\psi_{0,1}(t)$ отличен от нуля на отрезке от 1 до 2, то есть там, где $\psi_{0,0}(t)$ равен нулю, поэтому скалярное произведение $\psi_{0,0}(t)$ и $\psi_{0,1}(t)$ равно нулю.

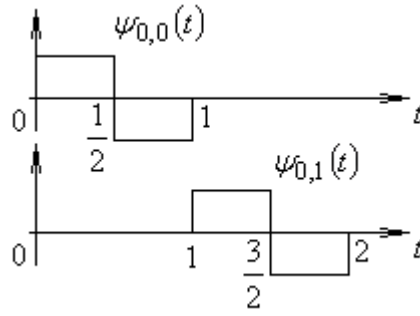


Рис. 78. Ортогональность $\psi_{0,0}(t)$ и $\psi_{0,1}(t)$.

Ортогональность вейвлетов Хаара с разным масштабом объясняется тем, что если масштаб будет отличаться хотя бы на 1, то вейвлет с меньшим масштабом будет всегда отличен от нуля только на том отрезке, где вейвлет с большим масштабом сохраняет своё значение.

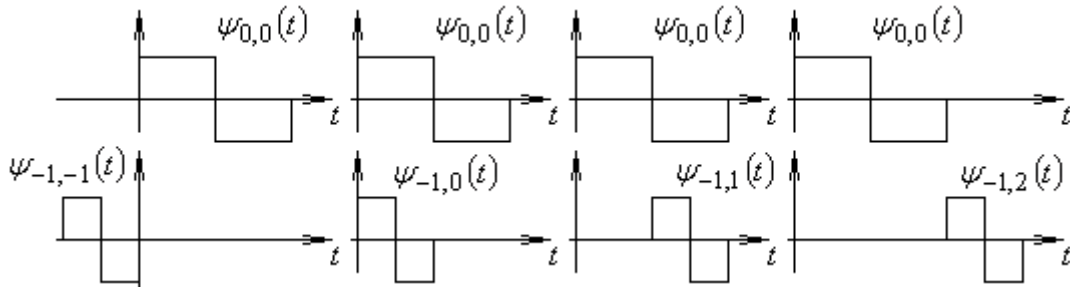


Рис. 79. Ортогональность вейвлетов с разным масштабом.

Ортогональность масштабирующих функций одного масштаба и масштабирующих функций и вейвлетов объясняется аналогичным образом.

Теперь разберёмся с формулами (152). Необходимо определить коэффициенты h_k и g_k в этих формулах. Выше было сказано, что функции $\psi(t)$ и $\varphi(t)$ должны быть такими, чтобы эти коэффициенты не зависели от масштаба функций. Поэтому соотношения (152) позволяют выполнить разложения масштабирующих функций $\varphi_{J,k}(t)$ с масштабом J , по базису, образованному семейством функций $\psi_{J+1,k}(t)$ и $\varphi_{J+1,k}(t)$ с масштабом на 1 больше. Это разложение позволяет записать приближение сигнала $A_J(t)$ как сумму более грубого приближения $A_{J+1}(t)$ и той составляющей сигнала $D_{J+1}(t)$, которую убрали, чтобы получить более грубое приближение. То есть формулы (152) описывают преобразование базиса, за счёт которого и выполняется разложение сигнала (163).

Для того чтобы определить коэффициенты h_k , надо выяснить, как из двух функций $\varphi_{-1,0}(t)$ и $\varphi_{-1,1}(t)$ получить функцию $\varphi_{0,0}(t)$. Функции $\varphi_{-1,0}(t)$ и $\varphi_{-1,1}(t)$ принимают значение $\sqrt{2}$ на отрезках $\left[0, \frac{1}{2}\right)$ и $\left[\frac{1}{2}, 1\right)$

соответственно, за пределами этих отрезков они принимают значение, равное 0. Функция $\varphi_{0,0}(t)$ принимает значение равное 1 на отрезке $[0,1)$. Очевидно, что $\varphi_{-1,0}(t)$ и $\varphi_{-1,1}(t)$ надо сложить и разделить на $\sqrt{2}$. Аналогично, чтобы получить $\psi_{0,0}(t)$, вместо сложения надо выполнить вычитание.

$$\begin{aligned}\varphi_{0,0}(t) &= \frac{1}{\sqrt{2}}\varphi_{-1,0}(t) + \frac{1}{\sqrt{2}}\varphi_{-1,1}(t) \\ \psi_{0,0}(t) &= \frac{1}{\sqrt{2}}\varphi_{-1,0}(t) - \frac{1}{\sqrt{2}}\varphi_{-1,1}(t)\end{aligned}\tag{168}$$

Из этих равенств получаем выражения для масштабирующих функций с масштабом -1 через масштабирующие функции и вейвлет с масштабом 0.

$$\begin{aligned}\varphi_{-1,0}(t) &= \frac{1}{\sqrt{2}}\varphi_{0,0}(t) + \frac{1}{\sqrt{2}}\psi_{0,0}(t) \\ \varphi_{-1,1}(t) &= \frac{1}{\sqrt{2}}\varphi_{0,0}(t) - \frac{1}{\sqrt{2}}\psi_{0,0}(t)\end{aligned}\tag{169}$$

Полученные соотношения справедливы для перехода между двумя любыми соседними значениями масштаба.

Теперь продемонстрируем, как происходит разложение сигнала. В следующих формулах для краткости записи $\varphi_{j,k} = \varphi_{j,k}(t)$ и $\psi_{j,k} = \psi_{j,k}(t)$, это всегда функции t , даже в тех формулах, где это явно не указано. Пусть сигнал (рис. 80) отличен от нуля на отрезке $[0,1)$, и имеет ступенчатую форму, каждая ступенька имеет ширину $\frac{1}{8}$.

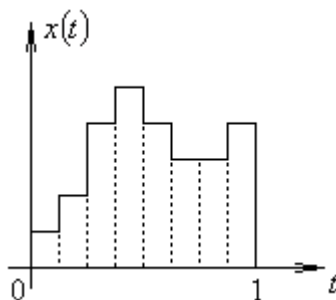


Рис. 80 Ступенчатый сигнал.

Такой сигнал может быть представлен без каких-либо искажений с помощью линейной комбинации из масштабирующих функций с масштабом $j = -3$, соответствующих вейвлету Хаара.

$$x(t) = s_{-3,0}\varphi_{-3,0} + s_{-3,1}\varphi_{-3,1} + s_{-3,2}\varphi_{-3,2} + s_{-3,3}\varphi_{-3,3} + \\ + s_{-3,4}\varphi_{-3,4} + s_{-3,5}\varphi_{-3,5} + s_{-3,6}\varphi_{-3,6} + s_{-3,7}\varphi_{-3,7} \quad (170)$$

Теперь выразим функции $\varphi_{-3,k}$ через $\varphi_{-2,k}$ и $\psi_{-2,k}$:

$$\begin{aligned} \varphi_{-3,0} &= \frac{1}{\sqrt{2}}\varphi_{-2,0} + \frac{1}{\sqrt{2}}\psi_{-2,0} \\ \varphi_{-3,1} &= \frac{1}{\sqrt{2}}\varphi_{-2,0} - \frac{1}{\sqrt{2}}\psi_{-2,0} \\ \varphi_{-3,2} &= \frac{1}{\sqrt{2}}\varphi_{-2,1} + \frac{1}{\sqrt{2}}\psi_{-2,1} \\ \varphi_{-3,3} &= \frac{1}{\sqrt{2}}\varphi_{-2,1} - \frac{1}{\sqrt{2}}\psi_{-2,1} \\ \varphi_{-3,4} &= \frac{1}{\sqrt{2}}\varphi_{-2,2} + \frac{1}{\sqrt{2}}\psi_{-2,2} \\ \varphi_{-3,5} &= \frac{1}{\sqrt{2}}\varphi_{-2,2} - \frac{1}{\sqrt{2}}\psi_{-2,2} \\ \varphi_{-3,6} &= \frac{1}{\sqrt{2}}\varphi_{-2,3} + \frac{1}{\sqrt{2}}\psi_{-2,3} \\ \varphi_{-3,7} &= \frac{1}{\sqrt{2}}\varphi_{-2,3} - \frac{1}{\sqrt{2}}\psi_{-2,3} \end{aligned} \quad (171)$$

Подставим (171) в (170) и приведём подобные слагаемые.

$$x(t) = \frac{s_{-3,0} + s_{-3,1}}{\sqrt{2}}\varphi_{-2,0} + \frac{s_{-3,2} + s_{-3,3}}{\sqrt{2}}\varphi_{-2,1} + \frac{s_{-3,4} + s_{-3,5}}{\sqrt{2}}\varphi_{-2,2} + \frac{s_{-3,6} + s_{-3,7}}{\sqrt{2}}\varphi_{-2,3} + \\ + \frac{s_{-3,0} - s_{-3,1}}{\sqrt{2}}\psi_{-2,0} + \frac{s_{-3,2} - s_{-3,3}}{\sqrt{2}}\psi_{-2,1} + \frac{s_{-3,4} - s_{-3,5}}{\sqrt{2}}\psi_{-2,2} + \frac{s_{-3,6} - s_{-3,7}}{\sqrt{2}}\psi_{-2,3} \quad (172)$$

или в новых обозначениях для коэффициентов:

$$x(t) = s_{-2,0}\varphi_{-2,0} + s_{-2,1}\varphi_{-2,1} + s_{-2,2}\varphi_{-2,2} + s_{-2,3}\varphi_{-2,3} + \\ + d_{-2,0}\psi_{-2,0} + d_{-2,1}\psi_{-2,1} + d_{-2,2}\psi_{-2,2} + d_{-2,3}\psi_{-2,3} \quad (173)$$

слагаемые с $\varphi_{-2,k}$ образуют более грубое приближение сигнала, а слагаемые с $\psi_{-2,k}$ поправку к этому приближению, необходимую, чтобы получить исходный сигнал. В результате сделанной замены получилось представление сигнала в базисе из функций со значком -2, которые отличны от нуля на отрезках в два раза шире, чем функции предыдущего базиса с первым значком -3. Старый и новый базис показан на рис. 81.

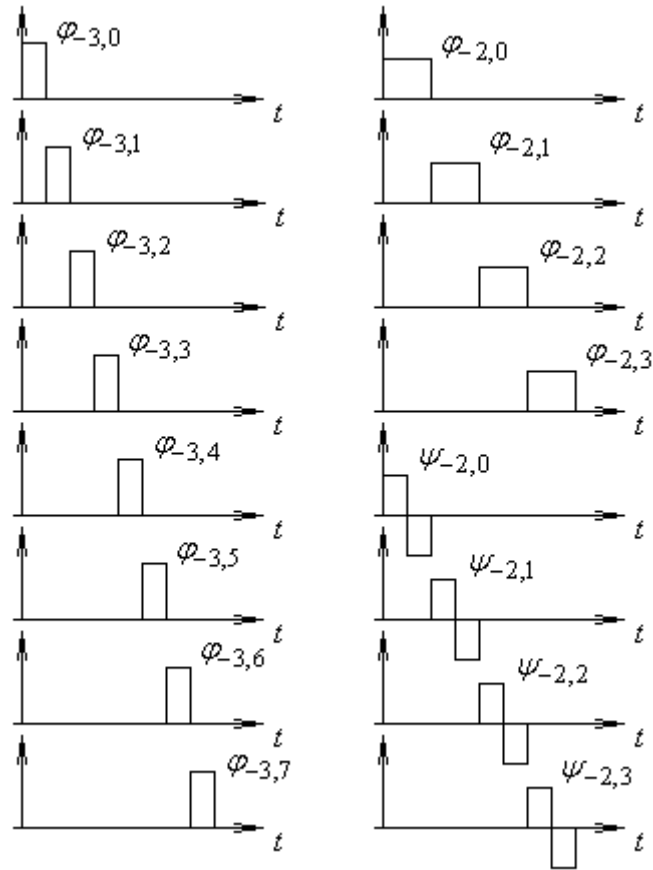


Рис. 81. Базисы из функций с масштабом -3 и -2.

В новом базисе есть четыре масштабирующие функции, с которыми выполняем те же действия. Выражаем через функции с первым значком -1, подставляем в (173), приводим подобные слагаемые и получаем представление сигнала в новом базисе, в котором присутствуют масштабирующие функции с масштабом -1. Эти масштабирующие функции образуют ещё более грубое приближение.

$$\begin{aligned}
 \varphi_{-2,0} &= \frac{1}{\sqrt{2}} \varphi_{-1,0} + \frac{1}{\sqrt{2}} \psi_{-1,0} \\
 \varphi_{-2,1} &= \frac{1}{\sqrt{2}} \varphi_{-1,0} - \frac{1}{\sqrt{2}} \psi_{-1,0} \\
 \varphi_{-2,2} &= \frac{1}{\sqrt{2}} \varphi_{-1,1} + \frac{1}{\sqrt{2}} \psi_{-1,1} \\
 \varphi_{-2,3} &= \frac{1}{\sqrt{2}} \varphi_{-1,1} - \frac{1}{\sqrt{2}} \psi_{-1,1}
 \end{aligned}
 \tag{174}$$

В новом базисе будут масштабирующие функции с масштабом -1, вейвлеты со значком -1, а также вейвлеты из предыдущего базиса со значком -2 (рис. 82). По новому базису раскладываются только масштабирующие функции.

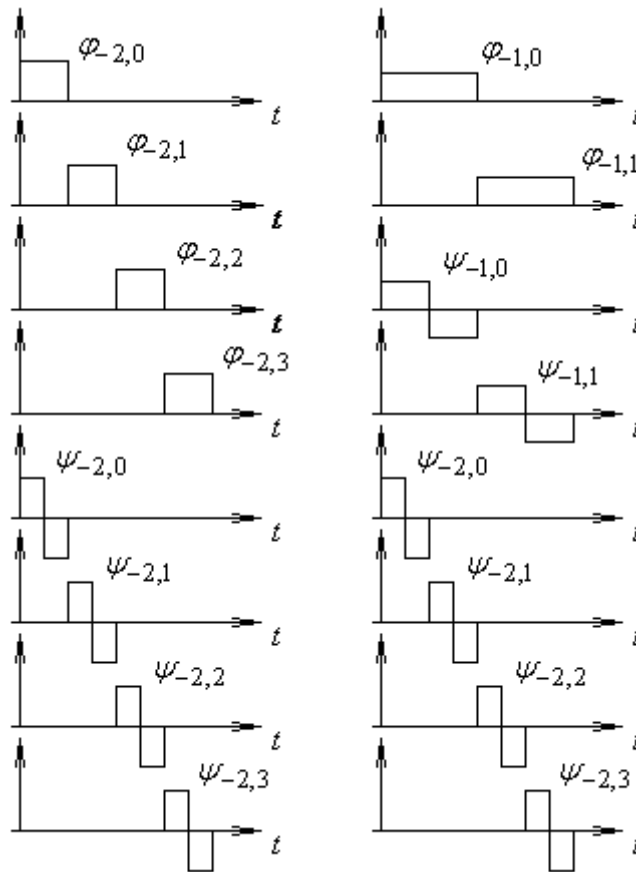


Рис. 82. Базисы из функций с масштабом -2 и -1.

Сигнал представляется как сумма приближения сигнала, в котором масштаб изменений -1 и суммы поправок к этому приближению, которые имеют масштабы -1 и -2.

$$\begin{aligned}
 x(t) = & s_{-1,0}\varphi_{-1,0} + s_{-1,1}\varphi_{-1,1} + \\
 & + d_{-1,0}\psi_{-1,0} + d_{-1,1}\psi_{-1,1} + \\
 & + d_{-2,0}\psi_{-2,0} + d_{-2,1}\psi_{-2,1} + d_{-2,2}\psi_{-2,2} + d_{-2,3}\psi_{-2,3}
 \end{aligned}
 \tag{175}$$

Выполнив замену

$$\begin{aligned}
 \varphi_{-1,0} &= \frac{1}{\sqrt{2}}\varphi_{0,0} + \frac{1}{\sqrt{2}}\psi_{0,0} \\
 \varphi_{-1,1} &= \frac{1}{\sqrt{2}}\varphi_{0,0} - \frac{1}{\sqrt{2}}\psi_{0,0}
 \end{aligned}
 \tag{176}$$

В результате перейдем к новому базису (рис. 83).

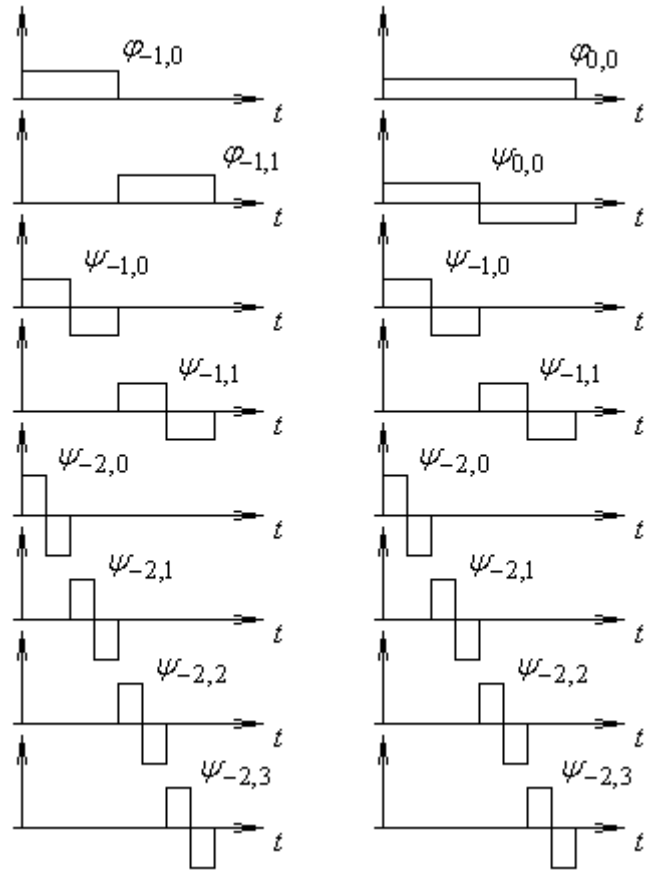


Рис. 83. Переход к окончательному базису разложения по вейвлетам.

получаем в окончательной форме разложение сигнала по вейвлетам с разными масштабами и сдвигами.

$$\begin{aligned}
 x(t) = & s_{0,0}\varphi_{0,0} + \\
 & + d_{0,0}\psi_{0,0} + \\
 & + d_{-1,0}\psi_{-1,0} + d_{-1,1}\psi_{-1,1} + \\
 & + d_{-2,0}\psi_{-2,0} + d_{-2,1}\psi_{-2,1} + d_{-2,2}\psi_{-2,2} + d_{-2,3}\psi_{-2,3}
 \end{aligned}
 \tag{177}$$

Пример разложения сигнала, приведённого на рис. 80 показан на рис. 84. На верхнем графике показан исходный сигнал, ниже слева – аппроксимация, справа – детализация. В окончательном виде сигнал представлен в виде среднего значения сигнала на отрезке $[0,1)$ – это слагаемое $s_{0,0}\varphi_{0,0}$. А также суммы вейвлетов $d_{j,k}\psi_{j,k}$ с разными масштабами, которые имеют смысл отклонения от среднего значения. Найдя коэффициенты $d_{j,k}$, мы получаем информацию о величине изменений сигнала, имеющих определённый масштаб.

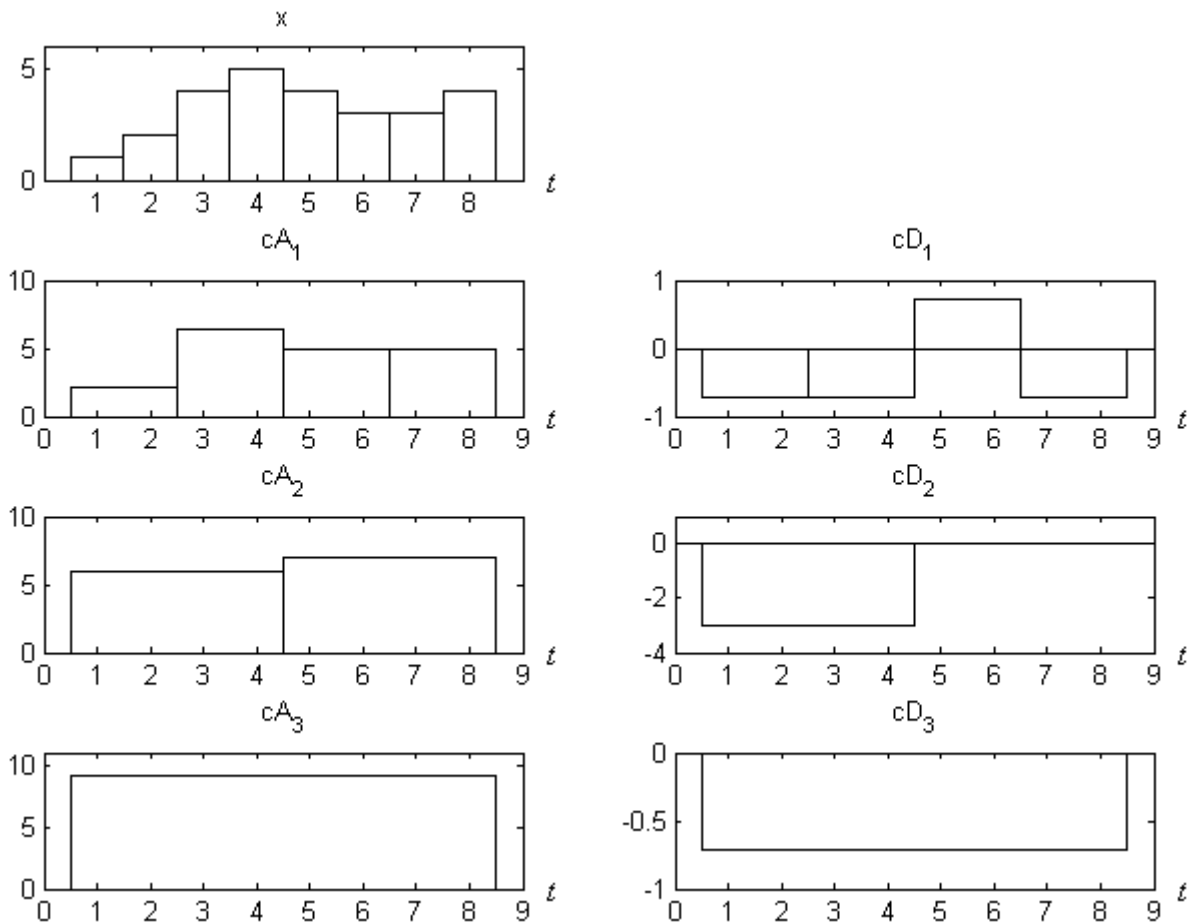


Рис. 84. Пример разложения с помощью вейвлета Хаара.

Представление сигнала в виде разложения по вейвлетам даёт возможность выполнять фильтрацию сигнала или сжатие информации. Если, например, в разложении присутствуют коэффициенты, значения которых значительно меньше, по сравнению с другими, то этими слагаемыми можно пренебречь, оставив в разложении только значимые слагаемые. Фильтрацию можно осуществить, например, убрав из разложения слагаемые с определённым масштабом, тогда в восстановленном сигнале будут отсутствовать изменения с соответствующим масштабом. Выше говорилось о том, что вейвлету может быть сопоставлена центральная частота, а масштаб обратно пропорционален частоте.

Кроме вейвлета Хаара в Matlab для дискретного вейвлет преобразования предназначены также следующие семейства вейвлетов:

Название	Короткое название в Matlab
Вейвлет Добеши	'db2', 'db3', ..., 'db45'
Койфлет	'coif1', ..., 'coif5'
Симлет	'sym2', ..., 'sym45'
Дискретный вейвлет Мейера	'dmey'

Биортогональный	'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'
Обратный биортогональный	'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'

Аналитические формулы существуют только для вейвлета Хаара, он же является одновременно вейвлетом Добеши первого порядка. Остальные вейвлеты получаются с помощью численных методов. В правом столбце в таблице перечислены короткие названия вейвлетов, используемые в Matlab в качестве параметров функций.

На практике дискретное вейвлет преобразование вычисляется по алгоритму быстрого дискретного вейвлет преобразования. Для преобразования сигнала с помощью этого алгоритма требуется количество операций пропорциональное, количеству отсчётов сигнала, также требуется знать только коэффициенты h_k и g_k из (152). А вычисление коэффициентов разложения по формулам (150) требует знания самого вейвлета и масштабирующей функции, а также приближенного вычисления интегралов с помощью численных методов.

7.3. Быстрое вейвлет преобразование

Пусть для некоторого заданного мелкого масштаба $j = 0$ коэффициенты $s_{0,k}$ уже известны. Приближение сигнала на этом масштабе:

$$A_0(t) = \sum_{k=-\infty}^{+\infty} s_{0,k} \varphi_{0,k}(t) \quad (178)$$

Тогда легко вычислить коэффициенты $s_{j,k}$ и $d_{j,k}$ для любого масштаба $j \geq 1$. Для этого используем соотношения (152):

$$\begin{aligned} \varphi(t) &= \sum_{k=0}^{N-1} h_k 2^{\frac{1}{2}} \varphi(2t-k) \\ \psi(t) &= \sum_{k=0}^{N-1} g_k 2^{\frac{1}{2}} \varphi(2t-k) \end{aligned} \quad (179)$$

Чтобы выразить функции с масштабом $j+1$ и сдвигом k через масштабирующие функции с масштабом j , заменим в этих выражениях t на $2^{-j}t - k$, а также умножим левые и правые части равенств на $2^{-\frac{j}{2}}$.

$$\begin{aligned} 2^{-\frac{j}{2}}\varphi(2^{-j}t - k) &= \sum_{n=0}^{N-1} h_n 2^{\frac{1}{2}} 2^{-\frac{j}{2}} \varphi(2(2^{-j}t - k) - n) = \sum_{n=0}^{N-1} h_n 2^{-\frac{j-1}{2}} \varphi(2^{-(j-1)} - (n+2k)) \\ 2^{-\frac{j}{2}}\psi(2^{-j}t - k) &= \sum_{n=0}^{N-1} g_n 2^{\frac{1}{2}} 2^{-\frac{j}{2}} \varphi(2(2^{-j}t - k) - n) = \sum_{n=0}^{N-1} g_n 2^{-\frac{j-1}{2}} \varphi(2^{-(j-1)} - (n+2k)) \end{aligned} \quad (180)$$

В левой части получились $\varphi_{j,k}(t)$ и $\psi_{j,k}(t)$, а в правой части под суммой – $\varphi_{j-1,n+2k}(t)$

$$\begin{aligned} \varphi_{j,k}(t) &= \sum_{n=0}^{N-1} h_n \varphi_{j-1,n+2k}(t) \\ \psi_{j,k}(t) &= \sum_{n=0}^{N-1} g_n \varphi_{j-1,n+2k}(t) \end{aligned} \quad (181)$$

Выполним замену индексов j на $j+1$ и n на $n-2k$. В окончательном виде:

$$\begin{aligned} \varphi_{j+1,k}(t) &= \sum_{n=0}^{N-1} h_{n-2k} \varphi_{j,n}(t) \\ \psi_{j+1,k}(t) &= \sum_{n=0}^{N-1} g_{n-2k} \varphi_{j,n}(t) \end{aligned} \quad (182)$$

Коэффициенты для разложения сигнала на масштабе $j+1$ вычисляются по формулам (150) с заменой j на $j+1$ соответственно. Чтобы получить выражения для $s_{j+1,k}$ и $d_{j+1,k}$ через $s_{j,n}$, подставим в формулы (150) вместо $\varphi_{j+1,k}(t)$ и $\psi_{j+1,k}(t)$ их выражения через $\varphi_{j,k}(t)$.

$$\begin{aligned} s_{j+1,k} &= \int_{-\infty}^{+\infty} x(t) \varphi_{j+1,k}(t) dt = \sum_{n=0}^{N-1} h_{n-2k} \int_{-\infty}^{+\infty} x(t) \varphi_{j,n}(t) dt \\ d_{j+1,k} &= \int_{-\infty}^{+\infty} x(t) \psi_{j+1,k}(t) dt = \sum_{n=0}^{N-1} g_{n-2k} \int_{-\infty}^{+\infty} x(t) \varphi_{j,n}(t) dt \end{aligned} \quad (183)$$

Получившиеся интегралы под знаком суммы заменяются обратно на коэффициенты разложения $s_{j,n}$.

$$\begin{aligned} s_{j+1,k} &= \sum_{n=0}^{N-1} h_{n-2k} s_{j,n} \\ d_{j+1,k} &= \sum_{n=0}^{N-1} g_{n-2k} s_{j,n} \end{aligned} \quad (184)$$

Коэффициенты разложения $s_{j,k}$ более точного приближения сигнала получают похожим образом по коэффициентам $s_{j+1,k}$ и $d_{j+1,k}$. Реконструкция сигнала заключается в получении последовательных приближений сигнала, где каждое последующее приближение $A_j(t)$ получается как сумма предыдущего приближения $A_{j+1}(t)$ и разницы между этими приближениями $D_{j+1}(t)$.

$$A_j(t) = A_{j+1}(t) + D_{j+1}(t) \quad (185)$$

Каждое из слагаемых в правой части и левая часть равенства в соответствии с (159) и (165) представимы в виде суммы.

$$\begin{aligned} A_j(t) &= \sum_{n=-\infty}^{+\infty} s_{j,n} \varphi_{j,n}(t) \\ A_{j+1}(t) &= \sum_{k=-\infty}^{+\infty} s_{j+1,k} \varphi_{j+1,k}(t) \\ D_{j+1}(t) &= \sum_{k=-\infty}^{+\infty} d_{j+1,k} \psi_{j+1,k}(t) \end{aligned} \quad (186)$$

Подставим их в (185) и умножим скалярно на $\varphi_{j,m}(t)$. Операцию суммирования и интегрирования можно поменять местами.

$$\begin{aligned} &\sum_{n=-\infty}^{+\infty} s_{j,n} \int_{-\infty}^{+\infty} \varphi_{j,n}(t) \varphi_{j,m}(t) dt = \\ &= \sum_{k=-\infty}^{+\infty} s_{j+1,k} \int_{-\infty}^{+\infty} \varphi_{j+1,k}(t) \varphi_{j,m}(t) dt + \sum_{k=-\infty}^{+\infty} d_{j+1,k} \int_{-\infty}^{+\infty} \psi_{j+1,k}(t) \varphi_{j,m}(t) dt \end{aligned} \quad (187)$$

Чтобы выяснить, чему равны эти интегралы, воспользуемся свойством ортогональности масштабирующих функций одного масштаба. После скалярного умножения формул (182) на $\varphi_{j,m}(t)$ под знаком суммы останется только слагаемое с номером $n = m$.

$$\begin{aligned} \int_{-\infty}^{+\infty} \varphi_{j+1,k}(t) \varphi_{j,m}(t) dt &= h_{m-2k} \\ \int_{-\infty}^{+\infty} \psi_{j+1,k}(t) \varphi_{j,m}(t) dt &= g_{m-2k} \end{aligned} \quad (188)$$

В левой части равенства (187) аналогично из всех слагаемых останется только слагаемое с номером $n = m$, а интеграл будет равен единице, так как в этом случае он является нормой масштабирующей функции с масштабом j , а нормировка функций одинакова для всех масштабов и равна единице. В итоге для реконструкции сигнала получается простая формула:

$$s_{j,m} = \sum_{k=-\infty}^{+\infty} s_{j+1,k} h_{m-2k} + \sum_{k=-\infty}^{+\infty} d_{j+1,k} g_{m-2k} \quad (189)$$

Формулы (184) и (189) также являются дискретным линейным фильтром с конечной импульсной характеристикой. Импульсная характеристика является конечной, так как для кратномасштабного анализа используют такие масштабирующие функции, для которых в уравнении (152) для $\varphi(t)$ должно быть конечное число ненулевых коэффициентов h_k .

В цифровой обработке сигналов (184) и (189) являются шагами анализа и синтеза в схеме субполосной фильтрации с точным восстановлением сигнала. В двухканальной схеме субполосной фильтрации берётся свёртка входящей последовательности с двумя импульсными характеристиками дискретных фильтров. Так как эти фильтры должны быть нерекурсивными, то их коэффициенты совпадают с отсчётами импульсной характеристики. Фильтр для получения аппроксимирующих коэффициентов является фильтром нижних частот, а фильтр для получения детализирующих коэффициентов – верхних частот. После фильтрации последовательности коэффициентов прореживаются, то есть в них оставляются только элементы с чётными или нечётными номерами. В формулах (184) операция прореживания в два раза заключается в наличии множителя 2 перед индексом k у коэффициента h_{n-2k} . В схеме субполосной фильтрации вместо этого, сначала вычисляется свёртка двух последовательностей коэффициентов, а потом из последовательности, полученной в результате свёртки, оставляется каждый второй элемент.

При восстановлении сигнала сначала в последовательности аппроксимирующих и детализирующих коэффициентов добавляются нулевые элементы, а потом к каждой из последовательностей применяется свой фильтр для реконструкции сигнала (операция свёртки с последовательностью коэффициентов фильтра), а потом результаты свёрток поэлементно складываются. В документации к Wavelet Toolbox в Matlab вектор с коэффициентами аппроксимации обозначается cA_j , вектор с коэффициентами детализации cD_j . Вектора с коэффициентами дискретных фильтров низких и высоких частот для разложения сигнала обозначаются Lo_D и Hi_D , а с коэффициентами фильтров для реконструкции – Lo_R и Hi_R . Схематично процесс разложения сигнала, представленного аппроксимирующими коэффициентами cA_j на уровне j , показана на рис. 85.

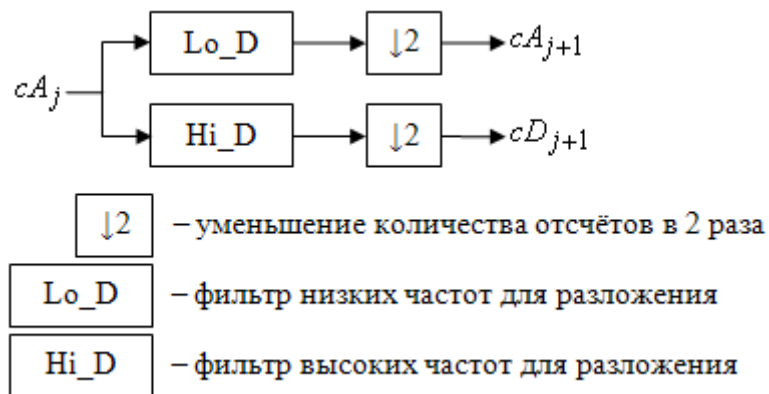


Рис. 85. Схема разложения сигнала.

Реконструкция сигнала выполняется в обратном порядке (рис. 86).



Рис. 86. Схема реконструкции сигнала.

Пусть приближение дискретного сигнала с самым лучшим разрешением представлено его отсчётами $s_{j,k} = x_k = x(kT_s)$. Выберем масштаб таким, чтобы этому приближению соответствовал номер $j = 0$. Считаем, что коэффициенты фильтров известны, это эквивалентно тому, что мы выбрали определённый вейвлет для анализа. Пропускаем дискретный сигнал через фильтр нижних частот и через фильтр верхних частот. В результате получаем две последовательности коэффициентов, в Matlab они считаются как два новых дискретных сигнала. Но в них теперь содержится избыточная информация, так как общее количество отсчётов стало в два раза больше. Каждый из получившихся в результате фильтрации сигналов содержит то же количество коэффициентов, что и исходный. Следующий шаг – уменьшение количества отсчётов в два раза, при этом удаляется каждый второй отсчёт. Уменьшение количества отсчётов в два раза эквивалентно уменьшению частоты квантования в два раза. В главе про теорему Котельникова говорилось, что для того, чтобы восстановить сигнал по его отсчётам необходимо, чтобы частота квантования была как минимум в два раза больше максимальной частоты в спектре сигнала, иначе в процессе оцифровки произойдёт наложение спектра. Когда выполняется вейвлет разложение сигнала с помощью алгоритма быстрого вейвлет преобразования, наложения спектров в результате уменьшения количества отсчётов не происходит, так как сигналы предварительно прошли через фильтры, которые устроены таким образом, что подавляют половину гармоник в спектре сигнала, оставляя либо низкочастотную составляющую сигнала, либо высоко частотную. В процессе уменьшения количества отсчётов спектр меняется, но искажения информации не происходит. На практике нельзя создать такую пару фильтров высоких и низких частот, чтобы при разложении с помощью них сигнала, не происходило потери информации, но существует способ найти коэффициенты фильтров так, чтобы минимизировать эти потери. Такие фильтры называются квадратурно зеркальными фильтрами (QMF) [2].

Теперь рассмотрим основные функции, используемые для дискретного вейвлет разложения.

```
[cA, cD] = dwt(X, 'wname')
[cA, cD] = dwt(X, Lo_D, Hi_D)
```

Функция `dwt` выполняет разложение сигнала на один уровень, в результате получается два сигнала `cA` – вектор с коэффициентами аппроксимации и `cD` – вектор с коэффициентами детализации. Первый аргумент функции – это вектор с отсчётами сигнала, либо с аппроксимирующими коэффициентами. Существует два варианта задания вейвлета. Один из вариантов – указать название вейвлета `'wname'`, тогда коэффициенты соответствующих ему фильтров будут вычислены автоматически. Другой

вариант – задать коэффициенты фильтров Lo_D и Hi_D самому. Вторым способом позволяет производить разложение сигнала даже с использованием фильтров, не удовлетворяющих условиям, которым должны подчиняться фильтры для кратномасштабного анализа.

Обратное преобразование выполняется с помощью команды:

```
X = idwt(cA, cD, 'wname')
X = idwt(cA, cD, Lo_R, Hi_R)
```

Входными аргументами функции являются векторы с аппроксимирующими и детализирующими коэффициентами. Вейвлет задаётся с помощью названия, либо с помощью коэффициентов фильтров для реконструкции. Функция возвращает вектор с аппроксимирующими коэффициентами, следующего по порядку более точного приближения.

Функции `dwt` и `idwt` выполняют разложение или реконструкцию только на один уровень. Для того, чтобы выполнить разложение или реконструкцию сигнала на заданное количество уровней используются команды:

```
[C, L] = wavedec(X, N, 'wname')
[C, L] = wavedec(X, N, Lo_D, Hi_D)
```

```
X = waverec(C, L, 'wname')
X = waverec(C, L, Lo_R, Hi_R)
```

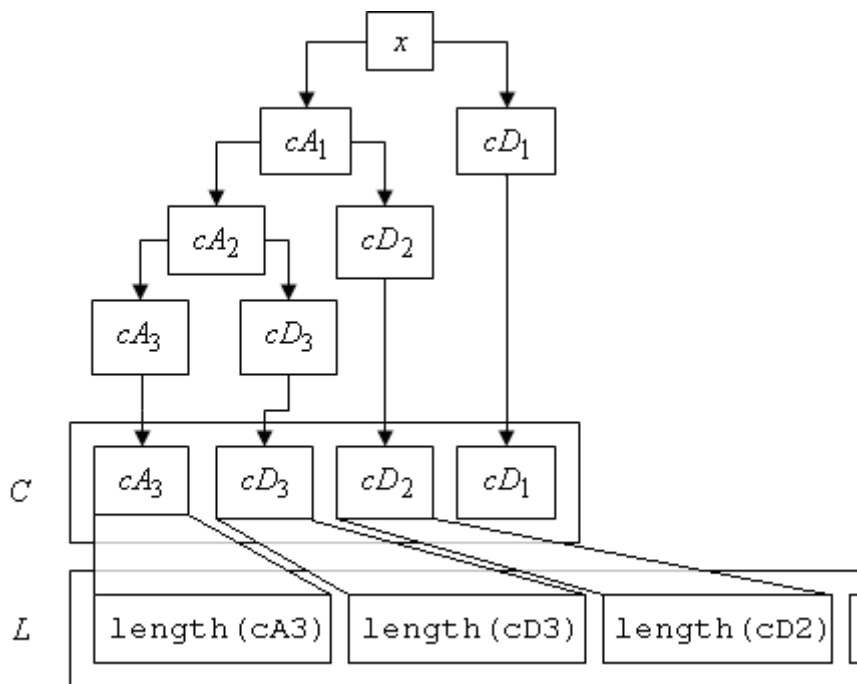


Рис. 87. Схема дискретного вейвлет преобразования.

Функция `wavedec` выполняет разложение сигнала x на N уровней и возвращает два вектора. В первом векторе C находятся аппроксимирующие коэффициенты последнего уровня разложения и детализирующие коэффициенты всех уровней разложения, а во втором векторе L сохранено количество коэффициентов на каждом уровне разложения. Значения в векторе L расположены в том же порядке, что и отрезки с коэффициентами в векторе C . Функция `waverec` выполняет полную реконструкцию сигнала.

Для того чтобы получить аппроксимирующие коэффициенты разложения на уровне N используется функция

```
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)
```

Также как и у описанных выше функций два варианта задания вейвлета: по имени и с помощью коэффициентов. Если последний параметр не указан, то возвращаются коэффициенты самого последнего уровня разложения.

Для того, чтобы получить детализирующие коэффициенты на заданном уровне разложения N используется функция:

```
D = detcoef(C, L, N)
D = detcoef(C, L)
```

В отличие от функции `appcoef`, для этой функции не требуется указывать вейвлет, так как она просто возвращает отрезок вектора C , в котором находятся детализирующие коэффициенты заданного уровня разложения. Если последний параметр не указывать, то будут возвращены коэффициенты самого последнего уровня разложения.

Для определения максимально возможного уровня разложения предназначена команда

```
L = wmaxlev(S, 'wname')
```

где S – количество отсчётов сигнала. Максимальный уровень разложения сигнала зависит от выбранного вейвлета, так как разные вейвлеты имеют различное количество коэффициентов дискретных фильтров. L

Пример

В примере демонстрируется разложение сигнала на один уровень функцией `dwt` с использованием вейвлета Добеши 2-го порядка. Сигнал (рис. 88) состоит из прямоугольного и треугольного импульса. Вейвлет Добеши второго порядка обладает свойством ортогональности полиномам 0-го и 1-го порядка. Это значит, что детализирующие коэффициенты не

будут содержать следов постоянной и линейно меняющейся составляющей сигнала.

Пример дискретного преобразования (dwt_example_1.m)

```
clear, clc, close all

Fs = 1000; % частота квантования
T = 1; % длительность записи
dt = 1/Fs; % период квантования
t = 0:dt:T; % время
% сигнал из прямоугольного и треугольного импульса
x = rectpuls(t-0.3,0.2) + tripuls(t-0.7,0.2,1);

[cA, cD] = dwt(x, 'db2');

% так как в процессе быстрого вейвлет преобразования
% происходит
% уменьшение количества отсчётов в 2 раза, необходимо изменить
% в 2 раза шаг по времени
dt = 2*dt;
t = (0:length(cA)-1)*dt;

% выводим результат на графике
subplot(2,1,1), plot(t,cA,'k') % аппроксимирующие коэффициенты
subplot(2,1,2), plot(t,cD,'k') % детализирующие коэффициенты
```

На рисунке 88 на верхнем графике изображены аппроксимирующие коэффициенты, которые практически повторяют форму самого сигнала, и отличаются лишь в окрестности 3-х точек около скачков сигнала, там происходят переходные процессы в дискретных фильтрах. На нижнем графике показаны детализирующие коэффициенты.

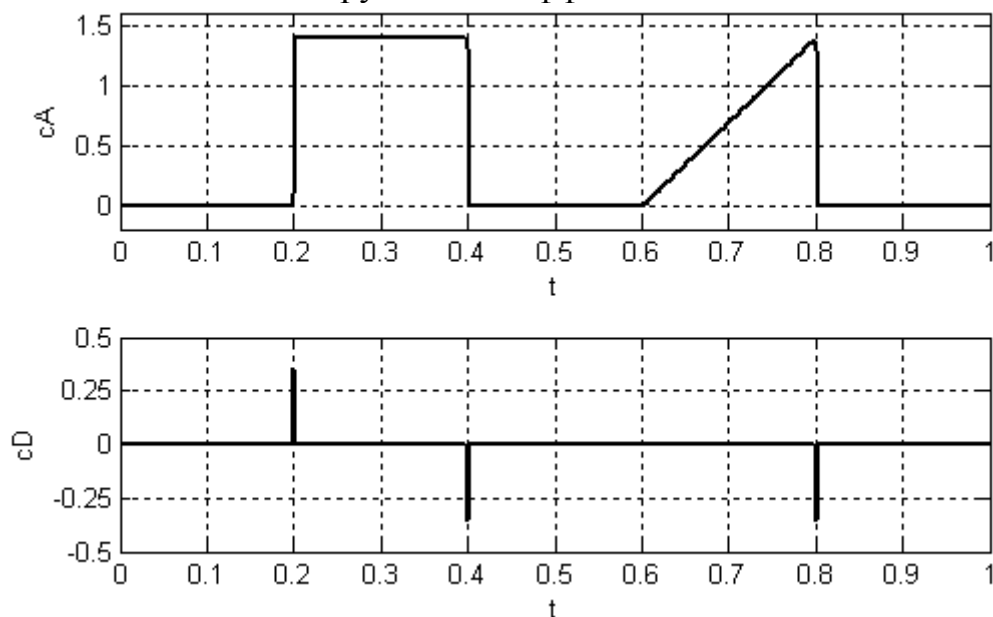


Рис. 88. Аппроксимирующие и детализирующие коэффициенты.

Из нижнего графика видно, что вейвлет Добеши 2-го порядка хорошо выделяет разрывы сигнала, и устраняет постоянную и линейную составляющую.

Пример

Продолжая идею предыдущего примера, посмотрим, как влияет на результат разложения одного и того же сигнала порядок вейвлета. В качестве исследуемого сигнала используем сигнал, показанный на рис. 89 вверху (графики построены относительно номера отсчёта):

$$x(t) = \begin{cases} t^4 - 2t^2 + 1, & t \in [-1, 1] \\ 0, & t \notin [-1, 1] \end{cases} \quad (190)$$

Сигнал является непрерывным вместе с первой производной на всей оси времени, в точках $t = \pm 1$ вторая производная меняется скачком. Найдём детализирующие коэффициенты первого уровня разложения, используя вейвлеты Добеши 1-го, 2-го, 3-го и 4-го порядка. Порядок показывает скольким полиномам ортогонален вейвлет. Например, вейвлет 4-го порядка будет ортогонален полиномам от 0-го до 3-го порядка.

Разрыв второй производной (dwt_example_2.m)

```
clear, clc, close all

dt = 0.01;
t1 = -1:dt:1;
x1 = t1.^4-2*t1.^2+1;
x(1:300) = 0; x(301:501) = x1; x(502:802) = 0;

subplot(5,1,1), plot(x)

[cA,cD] = dwt(x,'db1');
subplot(5,1,2), plot(cD)

[cA,cD] = dwt(x,'db2');
subplot(5,1,3), plot(cD)

[cA,cD] = dwt(x,'db3');
subplot(5,1,4), plot(cD)

[cA,cD] = dwt(x,'db4');
subplot(5,1,5), plot(cD)
```

Из графиков видно, что действие вейвлет преобразования, подобно операции дифференцирования, понижает порядок полинома. Например, для вейвлета 1-го порядка коэффициенты фильтра высоких частот для

разложения равны $h_1 = -\frac{1}{\sqrt{2}}$ и $h_2 = \frac{1}{\sqrt{2}}$, то есть детализирующие коэффициенты первого уровня разложения получаются как разность двух соседних отсчётов сигнала, разделенная на $\sqrt{2}$. Это с точностью до постоянного множителя совпадает с операцией численного дифференцирования. На втором графике приведены коэффициенты детализации, полученные с вейвлетом первого порядка. На графике видно, что первая производная непрерывна на всей оси, а на отрезке $[-1, 1]$ меняется как кубическая функция. По аналогии вейвлет второго порядка действует подобно второй производной. На третьем графике показаны детализирующие коэффициенты, полученные с использованием вейвлета 2-го порядка. Появились скачки, вызванные тем, что вторая производная имеет разрывы, на отрезке $[-1, 1]$ детализирующие коэффициенты ведут себя как парабола. На четвёртом графике, полученном с использованием вейвлета 3-го порядка. Там, где вторая производная меняется скачком, находятся отдельные импульсы, причём знак импульса соответствует направлению скачка, а на отрезке $[-1, 1]$ осталась линейная зависимость. С использованием вейвлета 4-го порядка детализирующие коэффициенты отражают только разрывы второй производной, а полиномиальные составляющие сигнала в процессе фильтрации уходят.

Процесс многоуровневого вейвлет разложения по схеме, представленной на рис. 87, с помощью алгоритма быстрого вейвлет преобразования может быть представлен как субполосная фильтрация. Согласно алгоритму быстрого вейвлет-преобразования после фильтрации происходит уменьшение количества отсчётов в два раза. При многоуровневом разложении происходит многократное чередование фильтрации и уменьшения отсчётов, при этом коэффициенты фильтров верхних и нижних частот остаются одни и те же, то есть форма частотных характеристик фильтра не меняется. Уменьшение количества отсчётов в два раза означает уменьшение частоты квантования в два раза, то есть изменение масштаба по частоте в два раза. Пусть оба фильтра высоких и нижних частот имеют частоту отсечки в 4 раза меньше частоты квантования исходного сигнала F_s . Это обычное требование для фильтров используемых в вейвлет разложении. Амплитудный спектр дискретного вещественного сигнала расположен на отрезке $\left[0, \frac{F_s}{2}\right)$. Фильтрация необходима для того, чтобы избежать наложения спектра при уменьшении количества отсчётов в 2 раза. Поэтому частоты отсечки фильтров должны быть на середине этого отрезка, то есть $\frac{F_s}{4}$.

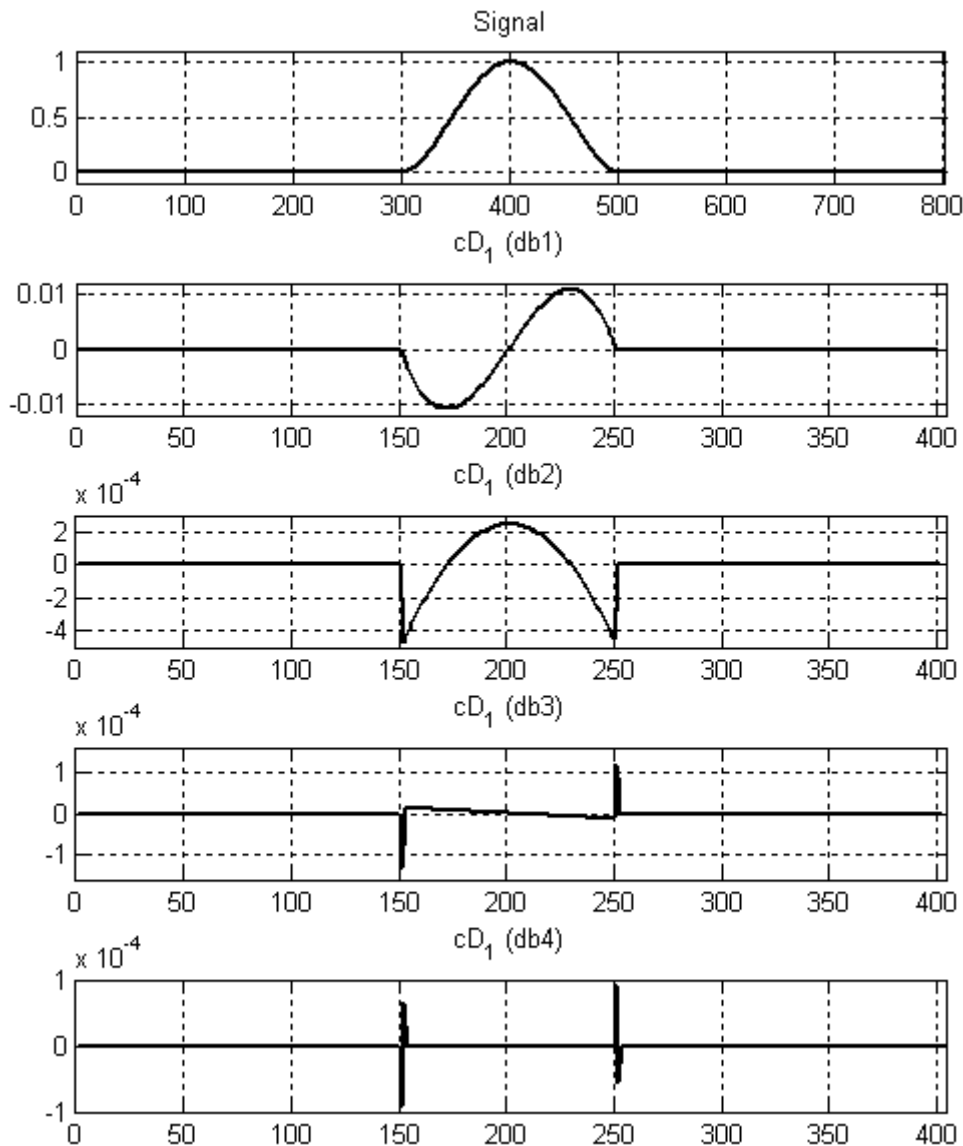


Рис. 89. Сигнал и детализирующие коэффициенты на первом уровне разложения с использованием вейвлетов Добеши разного порядка.

Для следующей операции будут применены фильтры с частотами отсечки в два раза меньше $\frac{F_s}{8}$, так как после уменьшения количества отсчётов изменился масштаб по частоте (всё равно, что частота квантования уменьшилась в два раза). В результате многократного разложения сигнал последовательно проходит через фильтры, полосы пропускания которых не перекрываются и каждая из полос отличается по ширине в 2 раза по сравнению с соседней. Далее будет продемонстрирован принцип субполосной фильтрации сигнала на примере суммы двух гармонических сигналов. На рис. 90 показаны АЧХ фильтров для разложения сигнала.

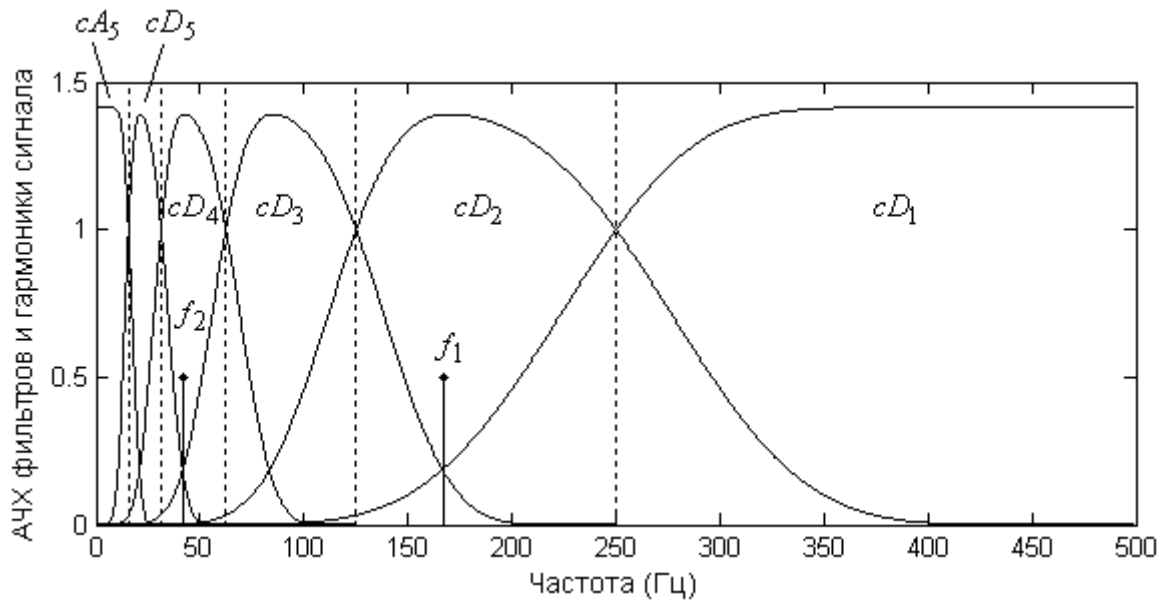


Рис. 90. АЧХ фильтров для разложения сигнала.

Графики АЧХ построены для случая $F_s = 1000 \text{ Гц}$, поэтому они построены на отрезке от 0 до 500 Гц, а частоты отсечек соответственно 250 Гц, 125 Гц, 62.5 Гц, и т. д. Полосы пропускания и частоты отсечек обозначены пунктирными линиями. Частота отсечки для данных фильтров определяется по уровню 1, так как сами коэффициенты передачи нормированы на $\sqrt{2}$. В каждой полосе пропускания подписано, какой набор коэффициентов детализации и аппроксимации получается в результате фильтрации $cD_1, cD_2, cD_3, cD_4, cD_5, cA_5$. Из рисунка видно, что АЧХ фильтра для получения детализирующих коэффициентов на первом уровне разложения в точности соответствует фильтру высоких частот. АЧХ фильтра для получения детализирующих коэффициентов второго уровня разложения соответствует полосопропускающему фильтру, так как образована произведением АЧХ фильтра нижних частот для получения аппроксимации на первом уровне и фильтра высоких частот для получения детализации на втором уровне. У фильтров на втором уровне частота отсечки в 2 раза меньше, чем у фильтров на первом уровне. Эта цепочка фильтров продолжается до последнего уровня разложения, каждый раз частота отсечки уменьшается в 2 раза, на последнем уровне разложения остаётся самая грубая аппроксимация сигнала, которая получается с помощью фильтра нижних частот полоса пропускания которого от 0 до $\frac{F_s}{2^{M+1}}$, где M – номер последнего уровня разложения.

Пример

Теперь рассмотрим пример вейвлет разложения на несколько уровней. Как отмечалось выше, цель вейвлет разложения – разделить сигнал на составляющие, каждая из которых имеет определённый масштаб

изменения (161), либо (162), если ограничимся разложением до некоторого масштаба. В качестве сигнала используем сумму двух гармонических сигналов с разным периодом колебаний $x(t) = \sin(2\pi f_1 t) + \sin(2\pi f_2 t)$. Каждое из слагаемых сигнала будет иметь свой характерный масштаб изменения во времени, поэтому колебания с разной частотой должны будут оказаться на разных уровнях разложения. В примере частоты f_1 и f_2 специально выбраны таким образом, чтобы заранее можно было качественно описать результат разложения (рис. 90). В разделе про непрерывное вейвлет преобразование описывалась связь частоты с масштабом. используем функцию `scal2frq` для того, чтобы определить частоты f_1 и f_2 . Пусть, для примера, частоты f_1 и f_2 будут соответствовать масштабам с номерами $j_1 = 2$ и $j_2 = 4$, тогда, как это было продемонстрировано во многих примерах из раздела про непрерывное вейвлет преобразование, следует ожидать, что детализирующие коэффициенты именно этих масштабов отражать поведение сигнала, причём на каждом масштабе будет своя частота колебаний.

Многоуровневое разложение (`dwt_example_3.m`)

```
clear, clc, close all

Fs = 1000; % частота квантования
T = 0.5; % длительность записи
dt = 1/Fs; % период квантования
t = 0:dt:T; % время
wname = 'db8'; % используемый вейвлет
% частоты
f1 = scal2frq(2^2, wname, dt);
f2 = scal2frq(2^4, wname, dt);
% сигнал
x = sin(2*pi*f1*t)+sin(2*pi*f2*t);

M = 5; % уровень разложения
[C,L]=wavedec(x,M,wname); % разложение сигнала

% отдельно для каждого масштаба строим график
% аппроксимации и детализации сигнала
figure('Color','w')
for m=1:M
    cA = appcoef(C,L,wname,m); % аппроксимация
    cD = detcoef(C,L,m); % детализация
    subplot(M,2,2*(m-1)+1), plot(cA,'k')
    axis([0, length(cA), -5, 5]), title(['cA_', num2str(m)])
    subplot(M,2,2*(m-1)+2), plot(cD,'k')
    axis([0, length(cA), -5, 5]), title(['cD_', num2str(m)])
end
```

На графиках (рис. 91) слева изображены аппроксимации сигнала, справа – детализации. На первом уровне разложения аппроксимация сигнала похожа на сам сигнал, в ней присутствуют колебания с обеими частотами, а детализация состоит только из высокочастотного колебания, причём, его амплитуда на два порядка меньше чем амплитуда аппроксимации. Появление высокочастотной составляющей малой амплитуды в детализации на первом уровне разложения вызвано тем, что эта частота подавляется фильтром высоких частот, используемым для вейвлет разложения, не полностью, её амплитуда уменьшается приблизительно на два порядка. На втором уровне разложения произошло разделение крупномасштабной и мелкомасштабной составляющей сигнала. Аппроксимация сигнала на втором уровне разложения не учитывает высокочастотной составляющей, которая перешла в детализацию сигнала. На третьем уровне разложения аппроксимация также состоит только из низкочастотной составляющей сигнала, которая получила искажение в результате фильтрации. Детализация сигнала на третьем уровне также как и детализация на первом уровне значительно меньше по амплитуде, чем аппроксимация или сам сигнал. На четвёртом уровне разложения низкочастотная составляющая перешла в детализацию, а аппроксимация теперь состоит только из низкочастотной составляющей с малой амплитудой, которая осталась из-за того, что фильтр нижних частот, используемый для разложения, не в состоянии подавить её полностью. На пятом уровне разложения аппроксимация и детализация содержат только оставшуюся не до конца подавленную низкочастотную составляющую. Отметим ещё один эффект, который виден на этих графиках. После каждой операции разложения возникает переходной процесс в начале и в конце как аппроксимации, так и детализации сигнала. После каждой операции фильтрации происходит уменьшение количества отсчётов сигнала, поэтому необходимо следить за тем, чтобы переходной процесс по краям сигнала не вносил существенных искажений в анализируемый сигнал.

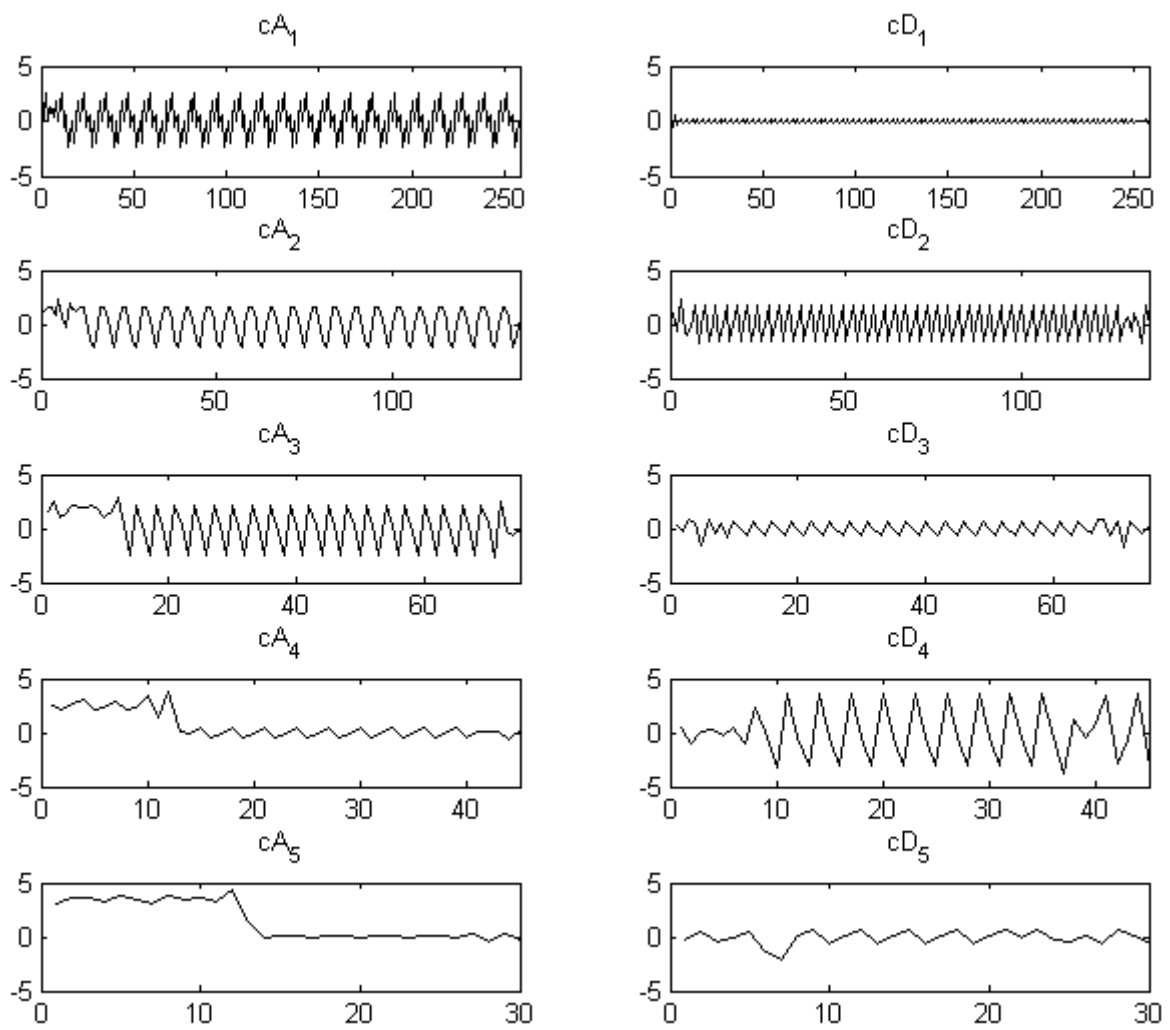


Рис. 91. Аппроксимация и детализация сигнала на разных уровнях разложения.

8. Литература

1. Будылин А.М. Ряды и интегралы Фурье. СПбГУ. 2002.
2. Добеши И. Десять лекций по вейвлетам. НИЦ «Регулярная и хаотическая динамика». 2001.
3. Дремин И. М. Иванов О. В. Нечитайло В. А. УФН 171, №5. Вейвлеты и их использование. 2001.
4. Смирнов В.И. Курс высшей математики (том 2). 1974.
5. Смирнов В.И. Курс высшей математики (том 3, часть 2). 1974.
6. MATLAB Documentation. Wavelet Toolbox. Advanced Concepts.– The MathWorks, Inc., 2007 (<http://www.mathworks.com>)
7. MATLAB Documentation. Signal Processing Toolbox.– The MathWorks, Inc., 2007 (<http://www.mathworks.com>)