

Ответ:

$$\begin{aligned}\text{cov}(x, y) &= E(x - \mu_x)(y - \mu_y) = E(xy - x\mu_y - y\mu_x + \mu_x\mu_y) = \\ &= E(xy) - \mu_x\mu_y - \mu_y\mu_x + \mu_x\mu_y = \\ &= E(xy) - \mu_x\mu_y.\end{aligned}$$

$$\begin{aligned}\text{cov}^2(x, y) &= E^2(xy) - 2E(xy)\mu_x\mu_y + \mu_x^2\mu_y^2 \geq \\ &\geq E(x^2)E(y^2) - E(x^2)\mu_y^2 - E(y^2)\mu_x^2 + \mu_x^2\mu_y^2.\end{aligned}$$

2.5. Вычислите ковариационную функцию синусоиды.

2.6. Приведите формулу для \sin -члена в формуле (2.3).

Задачи для решения на ЭВМ.

2.7. Смоделируйте равномерный шум и повторите рис. 2.5.

2.8. Смоделируйте гауссовский шум и повторите рис. 2.6.

2.9. Смоделируйте синусоиду и повторите рис. 2.7.

Глава 3

СБОР И ПРЕДВАРИТЕЛЬНАЯ ОБРАБОТКА ДАННЫХ

3.1. ВВЕДЕНИЕ

В этой главе мы остановимся на некоторых основных задачах, возникающих в начальной стадии анализа временных рядов. Данные, которые будут дальше рассматриваться, распадаются по типу на два разных класса — *непрерывные* (или *аналоговые*) данные и данные, имеющие по существу *цифровой* характер.

Для работы более удобны, пожалуй, данные второго типа. К ним можно отнести сведения об уровне цен на бирже, статистику занятости, годовые уровни осадков и многие другие. Даже если поиск и регистрация таких данных могут оказаться утомительным занятием, то со статистической точки зрения или с точки зрения приложений никаких сложностей не возникает.

Исследованию же непрерывных данных сопутствует ряд проблем. Среди них измерение данных, передача и (или) их запись, преобразование из аналоговой формы в цифровую.

Вопросам измерений посвящена книга Мэгреба и Бломквиста (1971). Задача здесь сводится, по сути дела, к конструированию или выбору *датчика* — устройства, преобразующего измеряемую физическую величину в электрический потенциал. Как правило, хотя и не всегда, полученный в результате потенциал пропорционален измеряемой физической величине. К таким приборам относятся микрофоны, гидрофоны, акселерометры и устройства, измеряющие давление.

Обычно датчик выбирается так, чтобы его передаточная функция была линейной в пределах предусмотренного промежутка. Идеальное линейное устройство создать невозможно. Интервал линейности ограничен, и большинству датчиков в большей или меньшей степени присуще наличие *гистерезиса*. Последнее означает, что при изменении элемента входа от начальной точки до конечной с возвращением в начальную точку две соответствующие траектории выхода, которые должны быть одинаковыми, в точности не совпадут. Такой график называется *гистерезисной кривой* инструмента. Для ряда хорошо сконструированных аппаратов получающаяся ошибка пренебрежимо мала.

Даже и совершенно линейный датчик может иметь передаточную функцию, заглушающую вход на некоторых промежутках частот. Например, передаточные функции для некоторых микро-

Фонов выглядят почти как полоснопропускающие фильтры. Иначе говоря, они пропускают широкую полосу информации, но заглушают частоты, близкие к нулю или превосходящие определенную точку отсечки. Внутри полосы пропускания передаточные характеристики могут изменяться крайне незначительно. На этом мы еще остановимся, поскольку иногда такое глушение желательно компенсировать.

3.2. СБОР ДАННЫХ

На рис. 3.1 изображена типичная система для сбора данных в цифровой форме.

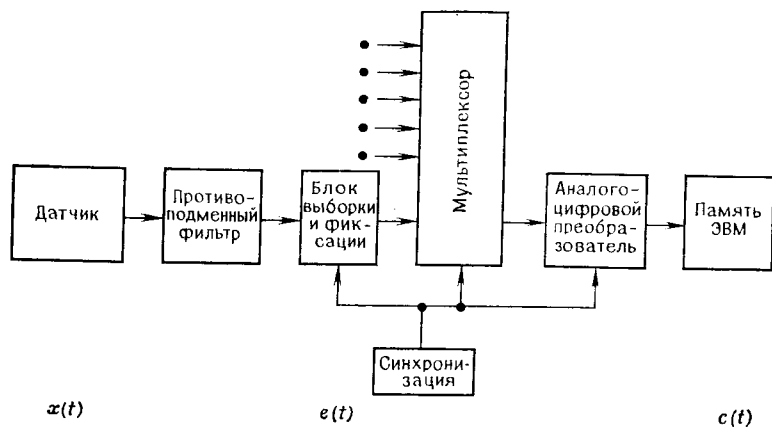


Рис. 3.1. Система для обработки цифровых данных. Все операции управляются синхронизирующей схемой.

Выход датчика поступает сначала в противоположенный фильтр. Если эффективная скорость отсчетов для этого выхода примерно равна S , то, чтобы предотвратить явление подмены частот для высокочастотной информации, фильтр должен гасить все частоты выше $S/2 = F$ Гц. Другими словами, передаточная функция идеального противоположенного фильтра в этом случае имела бы вид

$$H(f) = \begin{cases} 1, & -F < f < F, \\ 0 & \text{в остальных случаях.} \end{cases}$$

Однако построить фильтр, имеющий в точности такую передаточную функцию, невозможно. Здесь возникают два рода осложнений. Во-первых, передаточная функция приведенного вида физически неосуществима. Чтобы это устранить, следует ввести задержку в виде ненулевого фазового угла. Во-вторых, идеальный блок, устраняющего все частоты вне промежутка

$|f| > F$, не существует. Ребра или углы графика должны быть сглажены, и насколько бы незначительной ни была информация с частотами, превосходящими $|F|$, полностью избавиться от нее невозможно.

Другая проблема заключается в том, что при сравнении двух временных функций (что бывает часто необходимо) два фильтра, которые при этом используются, должны быть по возможности одинаковыми. В противном случае различие амплитудных или фазовых характеристик может послужить причиной ошибок, которые возникнут при вычислениях передаточной функции, корреляционной функции и относительной задержки между двумя входами.

Типичную форму низкочастотного фильтра, применяемого для устранения подмены частот, имеет фильтр Баттеруорта. Квадрат модуля передаточной функции этого фильтра имеет вид

$$|H(f)|^2 = \left[1 + \left(\frac{2\pi f}{2\pi B} \right)^{2M} \right]^{-1}.$$

Изобразив это соотношение графически, нетрудно установить, что $|H(0)|^2 = 1$ для $f = 0$ и $|H(B)|^2 = 1/2$ для $f = B$, величина $|H(f)|^2$ остается примерно постоянной в промежутке $0 < f < B$ вплоть до малой окрестности B , а при $f > B$ быстро убывает с ростом f . Нетрудно видеть также, что при $f \gg B$

$$|H(f)|^2 \approx \left(\frac{B}{f} \right)^{2M}.$$

Например, для $f = 8B$ и $M = 6$

$$|H(8B)|^2 \approx (1/8)^{12} = 1.46 \cdot 10^{-11} = -108.4 \text{ дБ.}$$

Фазовая характеристика фильтра Баттеруорта приближенно линейна вдоль полосы частот. Это означает, что фаза соответствует простой задержке.

Более детально фильтрация будет рассмотрена в следующих двух главах. Та терминология, которая связана главным образом с цифровыми фильтрами, в большинстве случаев применима и к аналоговым.

За противоположенным фильтром на рис. 3.1 следует устройство выборки и фиксации. Хорошим теоретическим описанием такого устройства служит формула взятия замера

$$x(i) = \int_{-\infty}^{\infty} x(t) \delta(iT - t) dt. \quad (3.1)$$

Осуществить операцию, идеально схожую с дельта-функцией, невозможно. Поэтому дельта-функцию заменяют «щелевой» функцией с конечным носителем. Если щель имеет прямоугольную

форму, то формула (3.1) преобразуется к виду

$$x(i) = \frac{1}{a} \int_{iT-a/2}^{iT+a/2} x(t) dt,$$

где параметр a — ширина щели. Вычисление по этой формуле есть, по сути дела, определение среднего значения функции $x(t)$ в промежутке от $iT-a/2$ до $iT+a/2$. Естественно, что ширина a должна быть меньше T , и чем она будет меньше, тем лучше.

Значение $x(i)$ обычно фиксируется блоком выборки и фиксации в виде разности потенциалов заряженного конденсатора и хранится до тех пор, пока мультиплексор не выберет его для пересылки в аналого-цифровой преобразователь. Мультиплексор

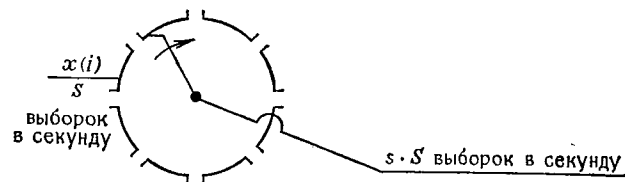


Рис. 3.2. Идеализированный мультиплексор, изображенный здесь в виде поворотного переключателя, имеет восемь сегментов — по одному на каждую обрабатываемую функцию.

можно представлять себе как поворотный переключатель, изображенный на рис. 3.2, со скоростью отсчетов S для каждой функции. Для мультиплексора с s сегментами на выходе вырабатывается sS значений в секунду. Раньше многие мультиплексоры представляли собой электромеханические переключатели. Современные мультиплексоры целиком изготавливаются на базе электроники.

Иногда в целях экономии различные для каждой функции устройства выборки и фиксации, расположенные перед мультиплексором, заменяют одним блоком выборки и фиксации, располагая его за мультиплексором. В этом случае отсчеты функций производятся не одновременно, как это происходит в случае распределения устройств выборки и фиксации по одному на каждый вход, а последовательно. Это приводит к небольшому относительному смещению времен взятия отсчетов для разных функций; чем плотнее отсчеты подаются на мультиплексор, тем меньше величина такого смещения.

Кадр данных, содержащий по одному отсчету каждой функции, получается во время одного полного поворота мультиплексора. С каждым кадром связан момент времени, в который этот кадр преобразуется в цифровую форму. Позднее, во время обработки, обычно генерируется «время» и добавляется в каждый кадр.

Поступающие с мультиплексора отсчеты в аналоговой форме подаются на цифровой преобразователь для перевода их в цифровой формат. Прежде чем обсуждать этот формат, имеет смысл остановиться на методах цифрового представления данных.

3.3. ЦИФРОВОЕ ПРЕДСТАВЛЕНИЕ ИНФОРМАЦИИ

Термином *бит*, предложенным Дж. В. Тьюки, называют одно-разрядное двоичное целое число. Таким образом, бит — это величина, принимающая только два значения — 0 и 1. В виде набора битов могут быть представлены числа и параметры. Имеются два основных способа представления — двоичное представление и представление с помощью двоично-десятичного кодирования (ДДК).

В двоичном представлении n битов b_0, b_1, \dots, b_{n-1} интерпретируют¹⁾ как число

$$(-1)^{b_0} \sum_{p=1}^{n-1} b_p 2^{n-1-p}.$$

Приведем несколько типичных примеров для $n=10$.

Двоичная запись	Цифровая запись	
	$b=0$	$b=9$
0000000000	0	0
0000000001	1	0.001953125
1000000001	-1	0.001953125
0111111111	511	0.998046875
1111111111	-511	-0.998046875

Это определение можно модифицировать, введя понятие *двоичной точки*. Двоичной точке отвечает, по сути дела, умножение числа на 2^{-b} , где b — положительное (или отрицательное) целое число. Например, для $b=9$ те же двоичные комбинации можно интерпретировать совершенно другим способом (см. табл. выше).

В ДДК-представлении для десятичных цифр используются группы по 4 бита:

0000	0	0101	5
0001	1	0110	6
0010	2	0111	7
0011	3	1000	8
0100	4	1001	9

Например, если на передней панели цифрового вольтметра указано значение +9.75, то в это же время с обратной стороны

¹⁾ Существуют и другие способы определения.

посредством электрических соединений выход реализуется ДДК-представлением 0 1001 0111 0101.

Двоично-десятичное кодирование не столь эффективно, как двоичное представление. Чтобы получать числа в промежутке ± 999 , при ДДК потребуется тринадцать битов, в то время как для числового промежутка ± 1023 в двоичном представлении достаточно лишь одиннадцати. Недостаток двоичного представления связан с необходимостью перехода на некотором этапе к более привычной десятичной записи.

Память ЭВМ содержит большие массивы битов, записанных различными способами, — на ферритовых сердечниках, магнитных дисках, лентах или на магнитном барабане. Эти биты группируются в комплекты стандартной длины, называемые *словами*. Как правило, слово содержит от 4 до 60 битов. Длина слова колеблется в зависимости от типа ЭВМ и составляет 4—16 битов для микро-ЭВМ, 12—32 для мини-ЭВМ и 32—60 битов для стандартных ЭВМ.

При использовании памяти на магнитной ленте принято оперировать терминами *символ* или *байт*. Ленту с семью дорожками можно представлять себе как прямоугольный массив битов, в котором поперек ленты расположены семерки битов, а вдоль — сотни битов в одном дюйме. Каждая семерка битов подразделяется на один *символ* и его *контрольную сумму*. Последняя является младшим значащим битом суммы остальных шести¹⁾.

Если ЭВМ имеет 36-битовые слова, то для их запоминания требуется по шесть смежных символов на магнитной ленте.

Для ленты с девятью дорожками по ширине ленты расположено девять битов; восемь из них образуют один байт, а девятый, как и в случае с семью дорожками, предназначен для контрольной суммы. Поэтому для ЭВМ с 32-битовыми словами одно слово запоминается на ленте в виде четырех последовательных байтов.

Непосредственно в ЭВМ слова допускают две интерпретации в зависимости от того, используется ли представление чисел с *фиксированной точкой* или же с *плавающей точкой*. Представление чисел с фиксированной точкой только что описано — последовательность битов понимается как число, которое следует умножить на масштабный множитель; здесь для представления используется все слово целиком, а масштабный множитель выбирается заранее (в машине его нет).

Представление с плавающей точкой, наоборот, уже в самом себе содержит двоичную точку. Остановимся на двух стандартных форматах слов для представления информации с плавающей точкой.

¹⁾ Существуют два разных способа вычисления контрольной суммы, известные как проверки на *четность* или *нечетность*.

На некоторых ЭВМ, таких, как Univac 1108, 36-битовые слова состоят из трех частей: *знака*, *характеристики* и *мантиссы*. Как определяется знак, понятно. Характеристика несет масштабную информацию, а мантисса — непосредственно числовую. Соответствующие биты располагаются в слове следующим образом: бит s с номером 0 отвечает знаку, биты (c_1, \dots, c_8) с первого по восьмой — характеристике, а биты (m_1, \dots, m_{27}) с девятого по тридцать пятый — мантиссе. Такому представлению соответствует число

$$(-1)^s \left(2^{\left(\sum_{p=1}^8 c_p 2^{(8-p)-128} \right)} \right) \sum_{q=1}^{27} m_q 2^{-q}.$$

Знак Характеристика Мантисса

Таким образом, характеристика может изменяться от $2^{-128} = 2.9387 \dots \cdot 10^{-38}$ до $2^{128} = 1.7014 \dots \cdot 10^{38}$. Мантисса меньше единицы и принимает значения на отрезке $[0, 1 - 2^{-27}]$. В такой записи единица выглядит следующим образом:

```
010 000 001 100 000 000 000 000 000 000 000 000 000
      2014000000008
```

Нижняя строчка есть запись верхней в *восьмеричной форме*. При такой записи двоичные цифры группируются по три и каждая тройка переписывается по правилу

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

Переход к восьмеричной форме сокращает запись числа, поскольку вместо каждых трех цифр используется одна (в данном случае запись сократилась с 36 до 12 цифр).

Другой способ представления чисел с плавающей точкой применяется в ЭВМ IBM 360/370, а также в машинах Xerox, SEL и в некоторых машинах Univac. Длина слова в этих машинах равна 32, а биты распределяются следующим образом: нулевой бит s соответствует знаку, биты (c_1, \dots, c_7) с первого по седьмой — характеристике, а биты (m_1, \dots, m_{24}) с восьмого по тридцать второй — мантиссе. Такому представлению соответствует число

$$(-1)^s 16^{\left(\sum_{p=1}^7 c_p 2^{(7-p)-64} \right)} \sum_{q=1}^{24} m_q 2^{-q}.$$

Отметим, что основанием степени здесь служит не 2, а 16. Поэтому характеристика может изменяться от $16^{-64} = 8.636 \dots \cdot 10^{-78}$ до $16^{63} = 7.237 \dots \cdot 10^{78}$, так что промежуток при *шест-*

надцатеричном масштабировании намного шире, чем в случае 36-битовых слов с двоичным масштабированием.

Вместе с тем 32-битовый формат оказывается менее точным, чем 36-битовый. Помимо основного различия на три бита в мантиссе эти форматы отличаются также *нормализованной* формой записи чисел. Числа с плавающей точкой называют нормализованными, если наибольший значащий бит мантиссы находится в крайней левой позиции. В случае 36-битового слова бит m_1 за некоторыми исключениями¹⁾ равен 1. Следовательно, в этой форме записи все 27 битов мантиссы всегда будут значащими. В 32-битовом формате с умножением на степень шестнадцати такое положение возможно не всегда. Первые 15 целых чисел в двоичной и шестнадцатеричной системах имеют вид

	Двоичная запись	Шестнадцатеричная запись	Двоичная запись	Шестнадцатеричная запись
1	0001	1	8	1000
2	0010	2	9	1001
3	0011	3	10	1010
4	0100	4	11	1011
5	0101	5	12	1100
6	0110	6	13	1101
7	0111	7	14	1110
			15	1111
				8
				9
				A
				B
				C
				D
				E
				F

Все эти числа имеют один и тот же масштабный множитель, равный 16. Если теперь представить, что слева от этих чисел стоит двоичная точка, то легко понять, что невозможно эти числа сдвинуть левее. Такое явление приводит к уменьшению среднего числа битов, в которых может запоминаться информация.

В некоторых 32-битовых машинах подобная вещь происходит и при округлении. Именно после окончания умножения с плавающей точкой округление производится также при помощи арифметической процедуры со степенью шестнадцати.

Все это приводит к тому, что 32-битовые машины не обеспечивают такой точности, как 36-битовые. Если перейти к десятичной записи, то определить число значащих десятичных цифр для 36-битового слова с плавающей точкой можно из равенства $\log_{10} 2^{27} = 8.13$.

Статистическое изучение рассмотренных представлений показывает, что при шестнадцатеричном масштабировании и округлении теряется в среднем около трех из 24 битов мантиссы 32-битового слова. Поэтому в среднем десятичные числа в этом

¹⁾ Таким исключением, например, служит число 0.

случае имеют $\log_{10} 2^{21} = 6.32$ значащих цифр. Таким образом, разница в точности для 32- и 36-битовых форматов составляет примерно две десятичные цифры.

Часто оказывается, что шести десятичных цифр недостаточно для некоторых расчетов, проводимых при обработке временных рядов. В этих случаях необходимо переходить к вычислениям с *двойной точностью*.

3.4. АНАЛОГО-ЦИФРОВОЕ ПРЕОБРАЗОВАНИЕ

Чтобы избежать в дальнейшем путаницы, при анализе процесса преобразования в цифровую форму ограничимся только одной входной функцией, хотя пользоваться будем несколькими ее представлениями. Перечислим их: $x(t)$ — исходная физическая величина; $x_e(t)$ — выход датчика в вольтах, соответствующий $x(t)$; $C(t)$ — выход $x_e(t)$ датчика, переведенный в *непрерывные* отсчеты; $c(i)$ — выход $x_e(t)$ датчика, переведенный в округленные отсчеты; $e(i)$ — погрешность, равная $C(iT) - c(i)$.

Функция $C(t)$ введена здесь только для удобства. Это должно сделать более понятными построения графиков. Как функция $C(t)$, так и функция $c(i)$ измеряются в одних единицах — *отсчетах*. Функция $C(t)$ есть просто результат линейного преобразования функции $x_e(t)$ вида

$$C(t) = Ax_e(t) + B.$$

Например, если номинальный диапазон изменения величины $x_e(t)$ лежит в пределах от -5 до $+5$ В и ему соответствует интервал изменения функции $C(t)$ от 0 до 1023, то $A = 102.3$, $B = 511.5$.

На рис. 3.3, а изображена функция $C(t)$ и соответствующие значения $c(i)$, фактически получающиеся на выходе цифрового преобразователя. Две эти функции связаны соотношением

$$c(i) = [C(iT) + 0.5^-].$$

Здесь квадратные скобки означают, что берется значение наибольшего целого числа, не превосходящего значения выражения внутри скобок. Добавление к функции $C(iT)$ величины 0.5^- вызвано необходимостью *округления*, поскольку без него происходило бы постоянное *срезание* истинных значений $C(iT)$.

Погрешность $e(i)$ определяется как разность $c(i)$ и $C(iT)$:

$$e(i) = C(iT) - c(i).$$

Если цифровой преобразователь работает правильно, то значения $e(i)$ будут ограничены интервалом $(-1/2, 1/2)$. На рис. 3.3, б показана последовательность погрешностей для функций $C(t)$ и $c(i)$, изображенных на рис. 3.3, а. Последнюю формулу можно

переписать в виде

$$c(i) = C(iT) + e(i).$$

С этой точки зрения функция $c(i)$ есть сумма функции $C(iT)$ и погрешности, или, иначе, шума.

Теперь перед нами стоит задача охарактеризовать функцию $e(i)$. Это сделать не так просто, поскольку в одних случаях функция $e(i)$ ведет себя как случайная величина, а в других она может оказаться сильно коррелированной последовательностью или четной периодической функцией.

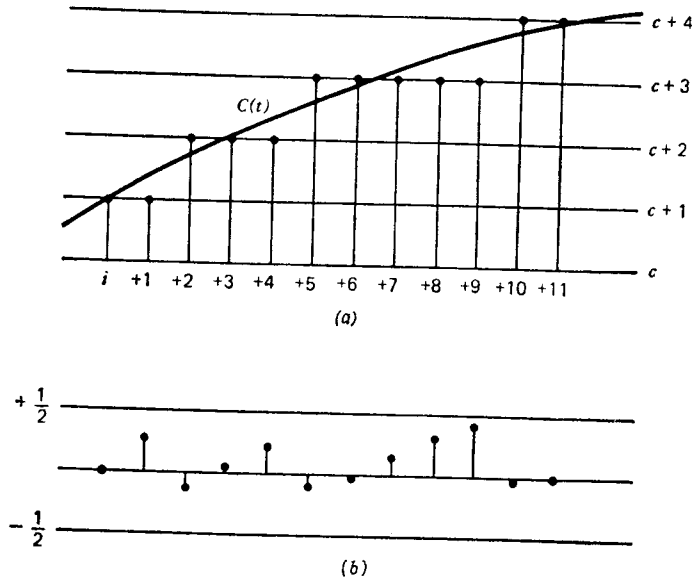


Рис. 3.3. Функция $C(t)$ пересекает четыре уровня отсчета. a — функция $C(t)$ и соответствующая ей функция $c(i)$, полученная на выходе преобразователя в цифровую форму; b — погрешность $e(i)$.

Рассмотрим пример, в котором функция $C(t)$ есть синусоида с частотой, равной частоте Найквиста:

$$C(t) = 511.25 \left[\cos\left(\frac{\pi t}{T}\right) + 1 \right] + 0.25.$$

В этом случае для $C(iT)$ (обратите внимание на замену независимой переменной) получим равенства

$$\begin{aligned} C(iT) &= 511.25 [\cos(\pi i) + 1] + 0.25 = \\ &= \begin{cases} 1022.5 + 0.25 = 1022.75 & \text{для четных } i, \\ 0.25 & \text{для нечетных } i. \end{cases} \end{aligned}$$

Следовательно,

$$c(i) = \begin{cases} 1023, & \text{если } i \text{ четное,} \\ 0, & \text{если } i \text{ нечетное,} \end{cases} \quad e(i) = \begin{cases} -0.25, & \text{если } i \text{ четное,} \\ 0.25, & \text{если } i \text{ нечетное.} \end{cases}$$

Таким образом, погрешность также есть синусоида с частотой, равной частоте Найквиста, и значит, она периодична.

Другая крайняя возможность — полностью случайная функция $e(i)$. Такая функция встречается во многих ситуациях. Этот случай представляет большой интерес, и мы остановимся на нем подробнее.

Сделаем следующие предположения:

1) Функция $e(i)$ случайна, и значит, $E[e(i)e(i+p)] = 0$ для $p \neq 0$.

2) Значения $e(i)$ имеют равномерное распределение.

Второе предположение означает, что все значения между $-1/2$ и $1/2$ равновероятны. Поэтому вероятностная функция плотности $p(e)$ имеет вид

$$p(e) = \begin{cases} 1, & -1/2 \leq e < 1/2, \\ 0 & \text{в остальных случаях.} \end{cases}$$

Вычислим μ_e и σ_e^2 , т. е. среднее и дисперсию величины $e(i)$:

$$\mu_e = E[e] = \int_{-\infty}^{\infty} ep(e) de = 0,$$

$$\sigma_e^2 = E[(e - \mu_e)^2] = \int_{-\infty}^{\infty} (e - \mu_e)^2 p(e) de = \int_{-1/2}^{1/2} e^2 de = \frac{e^3}{3} \Big|_{-1/2}^{1/2} = 1/12.$$

Размерностями среднего, равного 0, и дисперсии, равной $1/12$, будут линейные и квадратные единицы соответственно. Для того чтобы перевести дисперсию в физические единицы, нужно определить соответствующий коэффициент. Если один отсчет соответствует p физическим единицам, то дисперсия станет равной $p^2/12$. Например, если промежуток отсчетов $0-1023$ соответствует интервалу $0-5$ В, то каждому отсчету соответствует $5/1023 = 4.8876 \cdot 10^{-3}$ В/отсчет, а дисперсия будет равна $1.9907 \cdot 10^{-6}$ В². Как правило, полный диапазон цифрового преобразователя является степенью 2. Если всему диапазону цифрового преобразователя соответствует E физических единиц, то

$$\sigma_x^2 = \left(\frac{E}{2^n}\right)^2 \frac{1}{12} = \frac{E^2 2^{-2n}}{12},$$

где n — число цифр в слове цифрового преобразователя. Отметим, что эта ошибка экспоненциально убывает с ростом n . С каждым дополнительным битом в слове цифрового преобразователя эта

ошибка убывает в четыре раза (или на 6 децибел). Таким образом, теоретически шум можно сделать произвольно малым, увеличивая n . Этот результат детально рассмотрен Оливером, Пирсом и Шенноном (1948).

На практике стоимость цифрового преобразователя зависит среди прочего от числа битов. Крайние значения этого числа суть 6 и 18, но в большинстве работ по временным рядам применяются цифровые преобразователи с 10 и 12 битами. Шестибитовые преобразователи используются в устройствах, работающих с частотами порядка мегагерц, например в устройствах обработки радарных сигналов, где 18-битовые преобразователи оказались бы, пожалуй, слишком медленными. Восемнадцать битов соответствуют точности примерно в $5^{1/2}$ десятичных цифр. Устройства с таким уровнем очень трудно изготавливать и эксплуатировать. Довольно часто то же говорят и о 15-битовых цифровых преобразователях.

Таблица 3.1.

Диапазон отсчетов и число десятичных цифр для нескольких двоичных цифровых преобразователей

Число битов	Диапазон отсчетов	Число десятичных цифр
6	63	1.8
8	255	2.4
10	1 023	3.0
12	4 095	3.6
15	32 767	4.5
18	262 143	5.4

В табл. 3.1 приводятся диапазоны отсчетов и числа десятичных цифр для шести цифровых преобразователей, относящихся к рассмотренному типу со степенью двух.

Довольно просто можно вычислить плотность спектра мощности шума (погрешности). Поскольку предполагается, что шум некоррелирован, он будет иметь плоский спектр. В частности, произведение плотности спектра мощности на ширину интервала от 0 до частоты Найквиста должно быть равно дисперсии. Это следует из формулы

$$\sigma_x^2 = 2 \int_0^F S_x(f) df = 2S_x(f)F,$$

если предположить, что функция $S_x(f)$ постоянна. Значит,

$$S_x(f) = \frac{\sigma_x^2}{2F} = T\sigma_x^2.$$

Для диапазона цифрового преобразователя, соответствующего E физическим единицам, плотность спектра мощности шума, преобразованного в цифровую форму, имеет вид

$$S_e(f) = \frac{E^{22-2nT}}{12}.$$

Интересно сравнить этот случай с теоретической плотностью спектра мощности синусоиды. Можно показать, что синусоида максимальной амплитуды имеет пик в точке $A^2P/4$. В данном случае $A = E/2$, поэтому амплитуда пика станет равной $E^2P/16$. Отношение пика ПСМ синусоиды к среднему значению ПСМ шума равно

$$\left(\frac{E^2P}{16}\right) / \left(\frac{E^{22-2nT}}{12}\right) = \frac{3}{4} \frac{P2^{2n}}{T}.$$

Поскольку $P = NT$, получаем величину $3/4 N2^{2n}$.

После подстановки сюда часто встречающихся значений $N = 1024$ и $n = 10$ и перехода к децибелам получается примерно 90 дБ. Но это приближенное значение. Вычислительный шум, обусловленный округлениями, и другие факторы приводят к снижению этого значения до 60 дБ, что является типичным результатом, получаемым цифровыми спектроанализаторами с приведенными выше параметрами. Это один из методов определения динамического диапазона анализаторов плотности спектра мощности.

3.5. ДРУГИЕ ПОГРЕШНОСТИ

Перечислим наиболее важные проблемы, связанные с применением систем сбора данных. Среди них назовем в первую очередь проблемы посторонних электрических шумов, вызванных внешними условиями, а также шумов, обусловленных применением усилителей; проблемы, связанные с искажениями, имеющими причиной слишком высокие уровни сигналов, и чрезмерный уровень шума, возникающий вследствие слишком низкого уровня сигналов. Отметим также наложение высокочастотной информации в тех случаях, когда или не применяются противоподменные фильтры, или применяются низкокачественные фильтры. Кроме того, в устройствах выборки возникают щелевые ошибки и дрожание. Наконец, укажем на нелинейность и пропадание знаков в цифровом преобразователе, шум в цифровом преобразователе и ошибки операций.

Чтобы избежать этих проблем или уменьшить их значение, следует обратить внимание на три важных вопроса.

1. Прежде чем заказывать оборудование, все нужно взвесить с точки зрения конечного результата: какова природа решаемой задачи? Каковы параметры, которые нужно вычислять? Насколько точными должны быть их значения, чтобы получились исчерпы-

вающие ответы? Только после решения этих вопросов можно уточнить состав оборудования. На практике не всегда можно ответить на все эти вопросы, но экспериментатор должен задавать их себе, иначе он обнаружит, что характер решения определяется не поиском решения, а оборудованием.

2. Система должна быть как можно более автоматизированной. Как правило, система сбора данных управляется ЭВМ. Последняя должна комплектоваться и программироваться так, чтобы быть подчиненной эксперименту. Например, ЭВМ может выдавать такие ключевые параметры, как скорость выборки и калибровочные множители. Если их не регистрировать независимо, то станет весьма вероятным появление ошибки или при задании системе параметров, или при записи их значений.

3. Систему необходимо сделать самоконтролирующей и периодически проводить контрольные проверки. В решающих экспериментах они должны проводиться как до, так и после прогона теста. Сравнительно просто к системе можно присоединить цифро-аналоговый преобразователь (ЦАП). Имея один или несколько таких преобразователей, можно прогнать такой круговой тест: получить с помощью ЭВМ системы некоторые цифровые данные и преобразовать их в аналоговую форму посредством цифро-аналоговых преобразователей; затем выход преобразователей ввести в систему как данные, выбрать их и преобразовать в цифровую форму; после этого сравнить начальную функцию и конечный результат. Такого типа круговые проверки могут выявить нелинейности, некоторые виды исчезновения знаков, нарушения в наборе коэффициентов цифрового преобразования и некоторые другие особенности. Такую процедуру настоятельно рекомендуется проводить всегда, когда для этого есть возможность.

Щелевая ошибка. Упомянутая выше *щелевая ошибка* возникает вследствие того, что взятие замеров непрерывной функции производится не с помощью дельта-функции, а при помощи функции с конечным носителем, подобной той, которая встречалась в § 3.2. С этого момента будем предполагать, что в результате операции с прямоугольной щелью, примененной к $x(t)$, получается функция $y(t)$:

$$y(t) = \frac{1}{a} \int_{t-a/2}^{t+a/2} x(\tau) d\tau.$$

Эта функция есть свертка $x(t)$ с прямоугольным импульсом

$$u^*(t) = \begin{cases} a^{-1}, & -\frac{a}{2} \leq t < \frac{a}{2}, \\ 0 & \text{при остальных значениях } t. \end{cases}$$

Звездочка здесь поставлена для того, чтобы подчеркнуть отличие этой функции от стандартного прямоугольного импульса, приведенного в гл. 1. В частотной области преобразование Фурье $Y(f)$ равно произведению $X(f)$ и $U^*(f)$:

$$Y(f) = X(f) U^*(f),$$

где функция $U^*(f)$ имеет вид

$$U^*(f) = \frac{\sin(\omega a/2)}{\omega a/2}.$$

Предположим, что $a = T$, т. е. ширина щели в точности совпадает с интервалом выборки, используемым при цифровом преобразовании. Получающаяся в результате функция изображена на рис. 3.4.

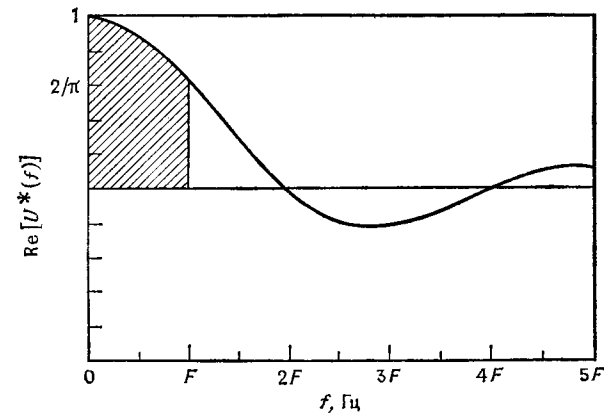


Рис. 3.4. Щелевой прямоугольный импульс при $a = T$.

Как видно из рисунка, на основном интервале $0 - F$ Гц происходит некоторое затухание. Значение $U^*(f)$ уменьшается от единицы в нуле до $2/\pi$ в частоте Найквиста. Поскольку $U^*(f)$ есть по существу передаточная функция щелевой операции, понятно, что в этом случае происходит дальнейшее затухание информации более высоких частот.

Такое положение может смягчаться уменьшением ширины щели a . Следует также помнить, что прямоугольная щелевая функция есть только идеализация настоящей функции. Углы такой функции, вероятно, закруглены, и ее вершина приподнимается.

Дрожание — это процесс, в силу которого выборки происходят не в точно заданные моменты, как это должно быть. Иначе говоря, выборки происходят не в моменты $t = iT$, а в моменты $iT + J(t)$, где $J(t)$ — случайный процесс. В качестве примера можно было

бы рассмотреть случайную функцию с распределением

$$p[J(t)] = \begin{cases} 1/T, & -T/2 \leq J(t) < T/2, \\ 0 & \text{в остальных случаях.} \end{cases}$$

Другими словами, предполагается, что истинное время выборки распределено равномерно в интервале длины T с центром в точке iT .

Анализ дрожания представляет собой довольно сложную задачу. Здесь обычно ссылаются на работу Балакришнана (1962), который характеризовал дрожание следующим образом. Определим $y(i)$ как $x(i)$ с дрожанием:

$$y(i) = x[i + J(i)].$$

Балакришнан показал, что односторонняя плотность спектра мощности $y(i)$, обозначенная $G_y(f)$, равна

$$G_y(f) = |C(f)|^2 G_x(f) + \frac{R_x(0) - a^2}{F},$$

где $G_x(f)$ — односторонняя ПСМ функции $x(i)$ ($= 2S_x(f)$), $C(f)$ — характеристическая функция распределения величины $J(i)$, $R_x(0)$ — значение в нуле корреляционной функции $x(i)$,

$$a^2 = \int_0^F |C(f)|^2 G_x(f) df.$$

При этом предполагается, что для $n \neq m$

$$E[J(n)J(m)] = 0,$$

т. е. дрожание не коррелирует с самим собой. Член a^2 предназначен для того, чтобы полная мощность оставалась той же самой. Полная мощность $[R_y(0)]$, получаемая интегрированием $G_y(f)$, окажется равной $R_x(0)$, и, стало быть, дрожание на нее не повлияет.

Функция $C(f)$, определенная как характеристическая функция вероятностной функции распределения, имеет вид

$$C(f) = \int_{-\infty}^{\infty} p[J] \exp(j2\pi fJ) dJ.$$

Балакришнан рассмотрел три вида распределений $J(i)$:

1. Дрожание с гауссовским распределением, имеющим дисперсию σ^2 . В этом случае

$$|C(f)|^2 = \exp(4\pi^2 f^2 \sigma^2).$$

2. Дрожание с равномерным распределением

$$p[J(t)] = \begin{cases} (\gamma T)^{-1}, & \text{если } |J(t)| < \gamma T/2, \\ 0, & \text{если } |J(t)| \geq \gamma T/2. \end{cases}$$

В этом случае

$$|C(f)|^2 = \left(\frac{\sin(\pi\gamma f T)}{\pi\gamma f T} \right)^2.$$

3. Дрожание с распределением

$$p[J(i)] = \begin{cases} P & \text{при } J(i) = -\gamma T, \\ 1-P & \text{при } J(i) = \gamma T. \end{cases}$$

В этом случае

$$C(f) = (1-P)^2 + P^2 + 2P(1-P) \cos(4\pi\gamma f T).$$

Наиболее важны два первых случая. Для каждого из них характерно следующее. Во-первых, поведение $|C(f)|^2$ при возрастании f приводит к затуханию $G_x(f)$. Значит, ошибка увеличивается с ростом значений частот. Во-вторых, к процессу добавляется ровный белый шум, которому отвечает слагаемое $(R_x(0) - a^2)/F$.

Балакришнан отмечал, что сделать какой-то вывод о затухании с ростом частот в третьем случае невозможно. Однако в большинстве практических ситуаций хорошими моделями оказываются случаи 1 и 2.

3.6. ПЕРЕХОД К ФИЗИЧЕСКИМ ЕДИНИЦАМ

В этом параграфе мы остановимся на *шаговой* и *синусоидальной калибровке* — широко распространенных процедурах перевода в технические единицы.

В обоих случаях предлагается перед записью данных исключить из системы датчик. Это связано с тем, что датчик, как правило, не может обеспечить эталонной точности.

При шаговой калибровке используется набор уровней напряжений, называемых шагами. Например, для датчика, напряжения которого изменяются в диапазоне от 0 до 5 В, такими уровнями могут служить 0.0, 2.5 и 5.0 В¹⁾. Напряжение каждого уровня подается в течение установленного промежутка времени, равного, например, одной секунде, и преобразуется в цифровую форму. Для удобства последующих вычислений результаты следует располагать так, чтобы их нельзя было спутать. Если используется магнитный носитель, то этого можно достичь, например, записью

¹⁾ В этой процедуре датчик нередко заменяют набором калибровочных сопротивлений, по одному на каждый шаг.

вая результаты на магнитную ленту отдельными файлами. Кроме того, можно начинать и прекращать запись результатов для каждого шага при помощи прерываний работы устройства, подающих соответствующие напряжения.

Сначала в ЭВМ вычисляются средние значения результатов, полученных для каждого уровня. Получающиеся по одному для каждого уровня средние числа обозначим a_1, \dots, a_N (предполагается, что всего используется N шагов).

После этого в ЭВМ вводятся величины в физических единицах для каждого уровня, по одной для каждого числа a_k . Эти величины обозначим p_1, \dots, p_N .

Допустим теперь, что в систему вводятся данные для обработки, которые поступают с цифрового преобразователя в отсчетах. Они составляют последовательность $c(i)$, рассмотренную ранее. С учетом всей полученной выше информации эта последовательность преобразуется в последовательность калиброванных данных $x(i)$.

Для этой цели обычно используется линейная интерполяция, сводящаяся к следующим действиям. Сначала определяется такое K , что $a_K \leq c(i) < a_{K+1}$, а затем по формуле

$$x(i) = p_K + (p_{K+1} - p_K) \frac{c(i) - a_K}{a_{K+1} - a_K}$$

получают соответствующее $c(i)$ калиброванное значение. При этом для крайних точек вычисления производят по таким же формулам, полагая $K=1$, если $c(i) < a_1$, и $K=N-1$, если $c(i) > a_N$. В последних случаях интерполяционная формула будет выступать в качестве экстраполяционной.

При синусоидальной калибровке в систему вводится синусоидальный сигнал с известной амплитудой и на большом числе периодов производится запись и преобразование в цифровую форму. При этом определяется среднее значение

$$\bar{d} = \frac{1}{N} \sum_{i=1}^N d(i)$$

последовательности $d(i)$ чисел, полученных при преобразовании синусоиды в цифровую форму. Помимо этого, вычисляется выборочная дисперсия по приводившейся ранее формуле

$$s_d^2 = \frac{1}{N-1} \sum_{i=1}^N (d(i) - \bar{d})^2.$$

По этим величинам определяют три точки:

$$a_3 = \bar{d} + \sqrt{2}s_d, \quad a_2 = \bar{d}, \quad a_1 = \bar{d} - \sqrt{2}s_d,$$

которые отвечают положительному пику, нулю и отрицательному пику синусоиды соответственно (все эти величины измерены в отсчетах). Затем вводятся соответствующие физические единицы и применяются приведенные выше формулы для интерполяции и экстраполяции.

Если бы не было шума и использовалось целое число периодов синусоиды, то эта процедура была бы точной. Однако избежать шума невозможно. Нельзя также начать и остановиться в одной и той же точке периода синусоиды. Поэтому для того, чтобы не получить ошибок или по крайней мере свести их к минимуму, берется большое число периодов.

3.7. РЕДАКТИРОВАНИЕ ДАННЫХ С НЕПРАВДОПОДОБНЫМИ ЗНАЧЕНИЯМИ

Большинство систем сбора информации вносит в данные ложные значения. Это может происходить по многим причинам — из-за потери сигналов в линиях связи, сбоев в цифровом преобразователе, сбоев в устройстве задания форматов. Неправдоподобные значения, возникающие в результате этих сбоев, могут вызвать значительные трудности при последующем анализе. Одно неправдоподобное значение, равное максимально представимой в цифровом преобразователе величине, может повысить предельный уровень шума и, значит, привести к значительному изменению плотности спектра мощности. Близко расположенная пара таких значений может дать несколько ложных частот в ПСМ.

По этим причинам лучше всего с помощью схемы предварительного преобразования предельных данных организовать поиск и удаление неправдоподобных значений. К сожалению, трудно определить точно, какие данные неправдоподобны. Общих процедур для автоматического удаления таких данных нет.

Поэтому разумно запастись программами редактирования или, если это необходимо, средствами ручной коррекции. Все машинные программы, предназначенные для этих целей, действуют следующим образом. Сначала для поиска неправдоподобных значений просматриваются все данные. Если такие значения обнаружены, то они выводятся на печать в виде таблицы или графика. После просмотра этих распечаток исследователь вводит очередную программу, чтобы удалить плохие значения, заменить их или вовсе не использовать. Последнее лучше всего, если, правда, это можно сделать.

Для устранения неправдоподобных значений предложено несколько схем автоматического редактирования, но ни одна из них не является полностью удовлетворительной. Приведем две такие схемы, чтобы пояснить, какого рода операции в этом случае производятся.

В первой из этих схем, представленной на рис. 3.5, используются два цифровых RC-фильтра для сглаживания. Эти фильтры подробно рассмотрены в гл. 4. Они производят низкочастотную фильтрацию исходной функции и вырабатывают сглаженные оценки. При рассмотрении обеих схем неявно предполагается, что нужные данные имеют «плавный» характер, а неправдоподобные значения — «резкий». Определим две последовательности $[\bar{x}(i)]^2$ и $\overline{x^2(i)}$ как последовательности сглаженных данных с последующим

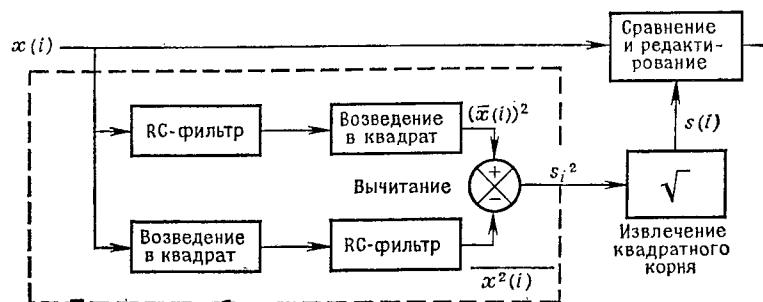


Рис. 3.5. Схема редактирования данных с неправдоподобными значениями.

возведением их в квадраты и сглаженных данных, предварительно возведенных в квадраты, соответственно.

Задача той части схемы, которая на рисунке ограничена пунктирной линией, состоит в выработке постоянно обновляемого значения выборочной дисперсии $s^2(i)$. Это значение определяется по формуле

$$s^2(i) = \overline{x^2(i)} - [\bar{x}(i)]^2.$$

Извлекая из него квадратный корень, получают стандартное отклонение.

На следующем шаге проверяют очередное значение ряда $x(i+1)$. Это значение считается хорошим, если

$$\bar{x}(i) - ks(i) < x(i+1) < \bar{x}(i) + ks(i). \quad (3.2)$$

Подходящий для данных параметр k выбирается самим исследователем. Как правило, он колеблется в пределах от 3 до 9, но для начала лучше выбирать его равным 6.

Плохие значения $x(i+1)$ могут заменяться значениями $\hat{x}(i+1)$, вычисленными по формуле

$$\hat{x}(i+1) = 2x(i) - x(i-1),$$

т. е. полученными, по сути дела, линейной экстраполяцией.

Эта процедура требует некоторых дополнительных уточнений. Для числа последовательных экстраполяций следует с самого начала установить предел, чтобы избежать постоянной экстраполяции. Дело в том, что последовательная экстраполяция для ряда плохих точек может достаточно далеко увести полосу (3.2), и в результате в нее не попадут те хорошие данные, которые в конце концов появятся.

Вторую схему редактирования данных с неправдоподобными значениями, частично принадлежащую Дж. В. Тьюки, называют процедурой «Тьюки 53X». Основная идея этой процедуры состоит в том, чтобы получить оценку гладкой кривой, которую затем вычитают из данных; после удаления гладкой составляющей или тренда выделить неправдоподобные значения гораздо легче.

В этой процедуре используется тот факт, что *медиана* есть *робастная* оценка среднего. Если все данные упорядочить в порядке возрастания, то медианой будет просто то значение, которое в этом ряде данных займет место посередине. Например, оценку среднего роста 101 студента можно получить следующим образом: отперфорировать рост каждого студента на 80-колоночных перфокартах, ввести эти карты в ЭВМ и определить среднее и медиану; среднее вычисляется обычным способом, а медиана — при помощи упорядочения результатов в порядке возрастания и выбора значения, оказавшегося посередине. Если при перфорации не было серьезных ошибок, то эти значения скорее всего окажутся довольно близкими. Посмотрим теперь, что произойдет, если один из студентов отперфорировует на карте свой рост равным 100 футам. В этом случае среднее возрастет примерно на 1 фут, в то время как медиана останется практически неизменной.

При вычислении оценки гладкой составляющей в процедуре Тьюки 53X это свойство робастности медианы используется дважды. Сама процедура выглядит следующим образом:

1. По последовательности $x(i)$ строится новая последовательность $x'(i)$. Сначала определяется медиана данных $x(1), \dots, x(5)$. Это значение становится в новой последовательности элементом $x'(3)$. Теперь из пяти первых данных убирается $x(1)$ и добавляется $x(6)$. Медиана этого нового множества есть $x'(4)$. Так продолжается до тех пор, пока все данные не будут исчерпаны. По сравнению с последовательностью $x(i)$ новая последовательность будет короче на четыре элемента. Медиана всегда выбирается из группы пяти последовательных значений.

2. Почти таким же способом по последовательности $x'(i)$ строится последовательность $x''(i)$. Единственное различие в том, что теперь медиана выбирается из *троек* последовательных значений.

3. На последнем этапе строится последовательность $x'''(i)$ по правилу

$$x'''(i) = 1/4 x''(i-1) + 1/2 x''(i) + 1/4 x''(i+1),$$

отвечающему сглаживающему фильтру Хэннинга.

После этого, как и раньше, для элементов последовательности $x(i) - x'''(i)$ проверяется условие $|x(i) - x'''(i)| > k$, где k — заранее выбранное число. Если оно выполняется, то соответствующую величину заменяют интерполированным значением.

3.8. УДАЛЕНИЕ ТРЕНДА

Иногда из некоторых временных рядов нужно удалить линейный или медленно меняющийся тренд. Такого рода тренды наблюдаются в рядах, например, при суммировании одной или нескольких компонент, приводящем к ошибкам двух типов. Во-первых, при неправильной калибровке нулевой точки в каждый момент отбора данных будет возникать небольшая ошибка. После суммирования эта постоянная величина даст прямую. Такой линейный тренд может привести к большим ошибкам при определении плотности спектра мощности и в связанных с этим вычислениях.

Ошибка второго типа возникает из-за возрастания в процессе суммирования мощности, соответствующей низкочастотному шуму. Как правило, такой шум в данных всегда есть. При суммировании он обретает форму случайного, но медленно меняющегося тренда. Насколько быстро меняется такой тренд, до некоторой степени зависит от интервала квантования.

Наилучшим способом удаления тренда служит применение высокочастотных фильтров, которые будут предметом обсуждения гл. 4. Полиномиальный тренд можно удалять методом наименьших квадратов. Это делается следующим образом. Как обычно, предположим, что $x(i)$ ($i=0, \dots, N-1$) — выборка функции на постоянном интервале T и что требуется подобрать многочлен вида

$$\hat{x}(i) = \sum_{k=0}^K (iT)^k c_k,$$

в котором $\hat{x}(i)$ — оцениваемые значения многочлена, присутствующего в ряде $x(i)$ в виде тренда. Для этой цели применяют стандартный способ, при котором определяется функция от коэффициентов многочлена

$$E(c) = \sum_{i=0}^{N-1} \left(x(i) - \sum_{k=0}^K (iT)^k c_k \right)^2, \quad (3.3)$$

называемая погрешностью. Ее можно минимизировать, применив стандартный прием, а именно приравняв все частные производные

по c_k нулю и решив относительно них систему уравнений

$$\frac{\partial E}{\partial c_j} = \sum_{i=0}^{N-1} 2 \left[x(i) - \sum_{k=0}^K (iT)^k c_k \right] [-(iT)^j] = 0, \quad j=0, \dots, K.$$

Эту систему можно переписать в виде

$$\sum_{k=0}^K c_k \sum_{i=0}^{N-1} (iT)^{k+j} = \sum_{i=0}^{N-1} (iT)^j x(i), \quad j=0, \dots, K.$$

Решить ее относительно c при больших значениях K трудно. К счастью, довольно редко приходится решать эту систему при K , большем 3 или 4. Если требуется удаление многочленов только низких порядков, то решение соответствующей системы методом обратной¹⁾ матрицы можно свести к непосредственному вычислению коэффициентов с использованием памяти ЭВМ. Например, таким решением в случае $K=0$ будет коэффициент $c_0 = N^{-1} \sum_{i=0}^{N-1} x(i)$.

В случае $K=1$ решение дают формулы

$$c_0 = \frac{2(2N-1) \sum_{i=0}^{N-1} x(i) - 6 \sum_{i=0}^{N-1} ix(i)}{N(N+1)},$$

$$c_1 = \frac{12 \sum_{i=0}^{N-1} ix(i) - ((N-1)/2) \sum_{i=0}^{N-1} x(i)}{TN(N^2-1)}.$$

Вычисления значительно упрощаются, если N — нечетное число. Разумеется, и при четном N последнюю точку можно удалить. Оказывается, что при нечетном числе точек и интервале $(-N/2, N/2)$ изменения независимой переменной i многие члены обращаются при вычислениях в нуль, в частности все те, которые имеют противоположные значения. Это сокращает число необходимых вычислений и с арифметической точки зрения улучшает их²⁾.
Далее знак Σ будет обозначать суммирование от $-N/2 + 1/2$ до $N/2 - 1/2$, т. е.

$$\Sigma = \sum_{i=-N/2+1/2}^{N/2-1/2}.$$

¹⁾ Замечание. Почти всегда вместо вычисления обратной матрицы лучше применять другие методы решения систем.

²⁾ Обращаем внимание читателя на то, что N здесь уже другое. С этого момента мы сменили обычную индексацию от 0 до $N-1$. Теперь используется $N+1$ точек.

Для первого случая ($K=1$)

$$d_0 = \frac{\sum x(i)}{N}, \quad d_1 = \frac{\sum ix(i)}{\sum i^2},$$

для $K=2$

$$d_0 = \frac{\sum i^2 \sum i^2 x(i)}{(\sum i^2)^2 - N \sum i^4}, \quad d_1 = \frac{\sum ix(i)}{\sum i^2},$$

$$d_2 = \frac{\sum i^2 \sum x(i) - N \sum i^2 x(i)}{[(\sum i^2)^2 - N \sum i^4]},$$

для $K=3$

$$d_0 = \frac{\sum x(i) \sum i^4 - \sum i^2 \sum i^2 x(i)}{N \sum i^4 - (\sum i^2)^2}, \quad d_1 = \frac{\sum i^4 \sum i^3 x(i) - \sum i^6 \sum ix(i)}{[(\sum i^4)^2 - \sum i^2 \sum i^6]},$$

$$d_2 = \frac{\sum i^2 \sum x(i) - N \sum i^2 x(i)}{[(\sum i^2)^2 - N \sum i^4]}, \quad d_3 = \frac{\sum i^4 \sum ix(i) - \sum i^2 \sum i^3 x(i)}{[(\sum i^4)^2 - \sum i^2 \sum i^6]}$$

и для $K=4$ (сначала вычисляется d_4 , затем d_3 и т. д.)

$$d_0 = (\sum x(i) - d_2 \sum i^2 - d_4 \sum i^4) / N,$$

$$d_1 = \left\{ \sum ix(i) [\sum i^2 \sum i^6 - (\sum i^4)^2] - [\sum i^4 \sum i^2 \sum i^3 x(i) - \sum i^4 \sum ix(i)] \right\} / \left\{ \sum i^2 [\sum i^2 \sum i^6 - (\sum i^4)^2] \right\},$$

$$d_2 = [\sum i^2 \sum x(i) - N \sum i^2 x(i) - d_4 (\sum i^2 \sum i^4 - N \sum i^6)] / [(\sum i^2)^2 - N \sum i^4],$$

$$d_3 = [\sum i^2 \sum i^3 x(i) - \sum i^4 \sum ix(i)] / [\sum i^2 \sum i^6 - (\sum i^4)^2],$$

$$d_4 = \left\{ [N \sum i^4 x(i) - \sum i^4 \sum x(i)] [(\sum i^2)^2 - N \sum i^4] + [\sum i^2 \sum x(i) - N \sum i^2 x(i)] (\sum i^4 \sum i^2 - N \sum i^6) \right\} / \left\{ (\sum i^2 \sum i^4 - N \sum i^6)^2 - [(\sum i^4)^2 - N \sum i^8] [(\sum i^2)^2 - N \sum i^4] \right\},$$

где $(N+1)$ — число равноотстоящих друг от друга выборок $x(i)$ (N может быть четным),

$$\sum i^2 = \frac{N(N^2-1)}{12}, \quad \sum i^4 = \frac{N(N^2-1)(2N^2-7)}{240},$$

$$\sum i^6 = \frac{N(N^2-1)(3N^4-18N^2+31)}{1344},$$

$$\sum i^8 = \frac{N(N^2-1)(5N^6-55N^4+239N^2-381)}{11520}, \quad i = -\frac{N}{2}, \dots, \frac{N}{2}.$$

Нужный полином имеет вид

$$\hat{x}(i) = \sum_{k=0}^K i^k d_k, \quad i = -\frac{N}{2}, \dots, \frac{N}{2}.$$

Отметим, что все вычисления проводятся с целыми числами; время T нигде не появляется. Работа в целочисленном формате также увеличивает точность.

Вычисление коэффициентов происходит в два этапа:

1. Вычисление моментов $\sum i^k x(i)$, где $k=0, 1, \dots, K$.
2. Вычисление коэффициентов d_0, \dots, d_K по приведенным выше формулам.

Для этих вычислений рекомендуется использовать двойную точность. В 32-битовых ЭВМ это следует делать обязательно.

Хотя вычисление и использование коэффициентов d_0, \dots, d_K удобно, сами по себе они играют незначительную роль. Многие исследователи предпочитают работать с коэффициентами c_0, \dots, c_K . Одной из причин этого служит тот факт, что d_0 не является средним значением, за исключением случаев $K=0$ или 1.

Переход от коэффициентов d к коэффициентам c производится следующим образом. Сначала определяется величина M : $M = -NT/2$. После этого i заменяют величиной $[(M+iT)/T]$, где снова $i=0, \dots, N$. Теперь формула для $\hat{x}(i)$ принимает вид

$$\hat{x}(i) = \sum_{k=0}^K \left(\frac{M+iT}{T} \right)^k d_k, \quad i=0, \dots, N.$$

Можно расписать ее для случая $K=4$:

$$d_0 + \left(\frac{M+iT}{T} \right) d_1 + \left(\frac{M+iT}{T} \right)^2 d_2 + \left(\frac{M+iT}{T} \right)^3 d_3 + \left(\frac{M+iT}{T} \right)^4 d_4.$$

Раскрывая скобки и собирая члены при одинаковых степенях $(iT)^k$, получаем

$$\hat{x}(i) = \left(d_0 + \frac{Md_1}{T} + \frac{M^2 d_2}{T^2} + \frac{M^3 d_3}{T^3} + \frac{M^4 d_4}{T^4} \right) +$$

$$+ iT \left(\frac{d_1}{T} + \frac{2Md_2}{T^2} + \frac{3M^2 d_3}{T^3} + \frac{4M^3 d_4}{T^4} \right) +$$

$$+ (iT)^2 \left(\frac{d_2}{T^2} + \frac{3Md_3}{T^3} + \frac{6M^2 d_4}{T^4} \right) +$$

$$+ (iT)^3 \left(\frac{d_3}{T^3} + \frac{4Md_4}{T^4} + (iT)^4 \frac{d_4}{T^4} \right).$$

Приравнивая полученные коэффициенты коэффициентам c_k , имеем

$$c_0 = d_0 + \frac{Md_1}{T} + \frac{M^2d_2}{T^2} + \frac{M^3d_3}{T^3} + \frac{M^4d_4}{T^4},$$

$$c_1 = \frac{d_1}{T} + \frac{2Md_2}{T^2} + \frac{3M^2d_3}{T^3} + \frac{4M^3d_4}{T^4}, \quad c_2 = \frac{d_2}{T^2} + \frac{3Md_3}{T^3} + \frac{6M^2d_4}{T^4},$$

$$c_3 = \frac{d_3}{T^3} + \frac{4Md_4}{T^4}, \quad c_4 = \frac{d_4}{T^4}.$$

По этим же формулам вычисляются коэффициенты и при меньших K . В этих случаях соответствующие коэффициенты d_k полагают равными 0.

Подгонка и удаление среднего. Подгонка и удаление среднего — самый частный случай полиномиальной фильтрации и удаления полиномиального тренда. Обычно эти операции объединяют с вычислениями корреляционной функции и плотностей спектра мощности. Само вычисление выглядит до смешного просто:

$$c_0 = m = \frac{1}{N} \sum_{i=0}^{N-1} x(i).$$

Довольно легко понять, что использования в некоторых машинах вычислений с обычной точностью для этой цели может оказаться недостаточным. Вспоминается один пример из приложений, в котором на 32-битовой машине с одинарной точностью в режиме плавающей точки суммировалось около 14 000 величин со средним, равным приблизительно 100. Сработала программа антипереполнения, и с некоторого момента вычисления суммы все новые данные были просто потеряны.

Поскольку по сравнению с вычислением ПСМ с одинарной точностью стоимость вычисления среднего с двойной точностью мала, для среднего рекомендуется программно задавать двойную точность. Исключения составляют только случаи с малыми значениями N равными, например 1024. Для них вполне достаточно и одинарной точности.

Упражнения

3.1. Смоделируйте квантизацию синусоиды 10-битовым цифровым преобразователем при помощи программы

```

DIMENSION E(1000)
:
:
C      ВВОД PHI И FACT
:
:
DO 10 I=1, 1000
ANG=FLOAT(I-1)*FACT+PHI
CT=511.5*(SIN(ANG)+1.)
CI=FLOAT(IFIX(CT+0.49999))
10    E(I)=CT-CI

```

- Опишите характеристики величины СТ.
- То же самое сделайте для С1.
- Выполните программу, дополнив ее вычислениями среднего и дисперсии для величины Е, а также вычислением коэффициентов корреляции для соседних величин Е при следующих значениях FACT и PHI:

FACT	PHI
0.0015708	0.0000
3.0753802	0.5218
1.5826000	0.7538

Объясните полученные ответы.

3.2. Напишите программу, моделирующую дрожание при преобразовании синусоиды в цифровой формат. Рекомендуется следующее:

- Получить последовательность $x(i) = \sin(2\pi f_0 i T)$, выбирая $T=1$ и $f_0=0.5/8$. В этом случае f_0 составит $1/8F$.
- Смоделировать дрожание, выбирая новое значение iT равным

$$(iT)_{\text{новое}} = (iT)_{\text{исходное}} + q [u(i) - 0.5],$$

где $u(i)$ — случайный шум с равномерным распределением, полученным с помощью процедуры TDRAND, а q , $0 \leq q \leq 1$, — масштабный множитель. Случаю $q=0$ соответствует отсутствие дрожания.

с) Получить последовательность длины 64 и вычислить преобразование Фурье при помощи процедуры FFTTRAN. Не забудьте перед этим действительным частям присвоить значения $x(i)$, а мнимым 0.

д) Просчитать модель при $q=0, 0.5$ и 1. Прокомментировать результаты.

3.3. При синусоидальной калибровке результаты получаются точными, если используется целое число циклов (ошибки преобразования в цифровую форму при этом не учитываются). Вычислите верхние границы для максимальной ошибки \bar{d} и \bar{s}_d^2 как функции N , T и частоты калибровочной синусоиды. Как ведут себя ошибки с ростом N ?

3.4. В этой задаче проверяется арифметика ЭВМ на антипереполнение. Напишите тестовую программу, которая должна выполнять следующую последовательность действий:

- Вычислить $x(i) = 99.5 + u(i)$, где $u(i)$ — случайное число, полученное с помощью процедуры TDRAND. Чему равно математическое ожидание отдельной реализации этой последовательности?
- Вычислить для $k=1, 2, \dots, 12$

$$\bar{x}_k = \frac{1}{N_k} \sum_{i=1}^{N_k} x(i),$$

где $N_k = \lceil 10^{(k+1)/2} \rceil$.

- Чему равны математическое ожидание и дисперсия \bar{x}_k как функции N_k ?
 - Приведите график \bar{x}_k как функции N_k и объясните результат.
 - Если позволяет машинное время, еще раз прогоните программу, но суммирование производите с двойной точностью.
- 3.5. Докажите (3.3).