

Министерство образования Республики Беларусь
Учреждение образования
«Белорусский государственный университет
информатики и радиоэлектроники»

Факультет компьютерных систем и сетей

Кафедра электронных вычислительных средств

А. А. Петровский, М.И. Вашкевич, И.С. Азаров

**ТЕОРИЯ И ПРИМЕНЕНИЕ ЦИФРОВОЙ
ОБРАБОТКИ СИГНАЛОВ.
ЛАБОРАТОРНЫЙ ПРАКТИКУМ**

*Рекомендовано УМО по образованию
в области информатики и радиоэлектроники
в качестве пособия для специальности 1-40 02 02
«Электронные вычислительные средства»*

Минск БГУИР 2016

УДК 621.391(076.5)
ББК 32.811я73
ПЗ0

Рецензенты:
кафедра информационных систем и технологий
Белорусского национального технического университета
(протокол №9 от 16.05.2015);

доцент кафедры информатики филиала «Минский радиотехнический колледж» учреждения образования «Белорусский государственный университет информатики и радиоэлектроники»,
кандидат технических наук, доцент В.Г. Лукьянец

Петровский, А. А.

ПЗ0 Теория и применение цифровой обработки сигналов. Лабораторный практикум: пособие / А. А. Петровский, М. И. Вашкевич, И.С. Азаров. – Минск : БГУИР, 2016. –87 с.: ил.
ISBN 978-985-543-223-5.

Пособие содержит описание шести лабораторных работ, посвященных теории цифровой обработки сигналов (ЦОС): базовым сигналам ЦОС, разностным уравнениям, дискретному преобразованию Фурье, фильтрам с бесконечной и конечной импульсными характеристиками. Лабораторные работы ориентированы на использование системы инженерных и научных расчетов Matlab. Пособие также содержит сведения об основных конструкциях языка Matlab.

УДК 621.391(076.5)
ББК 32.811я73

ISBN 978-985-543-223-5

© Петровский А. А., Вашкевич М.И., Азаров И.С. 2016
© УО «Белорусский государственный университет информатики и радиоэлектроники», 2016

Содержание

ЛАБОРАТОРНАЯ РАБОТА №1. Базовые сигналы в ЦОС	4
ЛАБОРАТОРНАЯ РАБОТА №2. Разностные уравнения	20
ЛАБОРАТОРНАЯ РАБОТА №3. Дискретное преобразование Фурье.....	34
ЛАБОРАТОРНАЯ РАБОТА №4. Спектральный анализ	50
ЛАБОРАТОРНАЯ РАБОТА №5. Расчет цифровых фильтров с бесконечными импульсными характеристиками	62
ЛАБОРАТОРНАЯ РАБОТА №6. Расчет цифровых КИХ-фильтров с линейной фазовой характеристикой методом взвешивания	70

ЛАБОРАТОРНАЯ РАБОТА №1. Базовые сигналы в ЦОС

ЦЕЛЬ РАБОТЫ – изучение пакета Matlab, программирование базовых сигналов цифровой обработки (ЦОС) в пакете Matlab.

1.1 Теоретические сведения

Пакет Matlab

Для выполнения лабораторных работ используется система инженерных и научных расчетов Matlab (сокращение от MATrixLABoratory – матричная лаборатория). Основным объектом системы Matlab является матрица. Все вычисления система осуществляет в арифметике с плавающей точкой.

Система Matlab работает в режиме интерпретации команд и операторов, которые вводятся в ходе сеанса в командной строке, а Matlab выполняет их немедленную обработку и выдает вычисленный результат. Однако в Matlab можно запустить на выполнение заранее подготовленную последовательность команд и операторов, записанную в виде файла с расширением .m (M-файл).

M-файлы разделяются на два вида: файлы-сценарии и процедуры-функции.

Все M-файлы должны располагаться в рабочем каталоге или каталоге, зарегистрированном в списке путей системы Matlab. Для изменения рабочего каталога необходимо выбрать в меню File пункт SetPath.... При проведении лабораторной работы система разрешает изменять только рабочий каталог и не разрешает добавлять и удалять каталоги из списка путей. При выходе из системы сведения о рабочем каталоге не сохраняются, поэтому при новом запуске Matlab его необходимо ввести заново.

Файлы-сценарии

Файл-сценарий – это текстовый файл, содержащий последовательности команд и операторов. В языке нет специальных операндов для обозначения начала и конца файла-сценария, а также точки входа. Началом сценария является начало файла, а концом соответственно конец файла.

Именем сценария является имя его файла, отсюда следуют ограничения, накладываемые на имена файлов:

1) имя файла сценария может содержать только символы латинского алфавита от A до Z и от a до z, цифры от 0 до 9 и символ подчеркивания;

2) имя файла должно начинаться с символа A...Z или a...z либо с символа подчеркивания.

Результат выполнения каждого оператора отображается в окне командной строки. Для того чтобы система не выводила результат, необходимо в конце оператора ставить точку с запятой.

Файлы процедуры-функции

Файл процедуры-функции также является текстовым файлом, содержащим последовательности команд и операторов. Отличия файла процедуры-функции от файла-сценария заключаются в следующем:

1. Первой строкой должен быть заголовок функции, имеющий следующий формат:

```
function [<список выходных параметров>] = <имя функции> (<список входных параметров>)
```

Имя функции должно совпадать с именем файла.

2. При выполнении функции она сначала компилируется во внутренний формат, после чего исполняется. В связи с этим функция выполняется на порядок быстрее файла-сценария, который работает в режиме интерпретации команд.

По соглашению, принятому в Matlab, начиная со второй строки функции может располагаться несколько строк комментария. Этот комментарий считается справкой по использованию функции и может быть вызван с помощью команды `help` и имени функции. Такую справку содержат абсолютно все функции пакета Matlab. Например:

```
help filter % справка по функции filter
```

Рассмотрим функцию, вычисляющую минимальное и максимальное значения массива (файл `minmax.m`):

```
function [minim, maxim] = minmax(a)
minim = min(a)
maxim = max(a)
```

Как видно из данного примера, функция может иметь несколько возвращаемых значений. Представленную в примере функцию можно использовать следующим образом:

```
[a,b] = minmax(x) % В a будет минимум, а в b - максимум
a = minmax(x) % В a будет минимум, а максимум теряется
```

Данные в системе Matlab

В системе Matlab присутствует только один тип данных – это прямоугольный массив комплексных чисел. Для создания переменной ей просто необходимо присвоить некоторое значение. Существует несколько вариантов записи массива:

1. *Объявление пустого массива.* Объявление пустого массива, т.е. массива нулевого размера, имеет следующий вид:

```
a = []
```

2. *Объявление скаляра.* Скаляр является частным случаем матрицы, т.е. это матрица размером 1×1 . Пример:

```
a = 5  
b = -4.8  
c = 0.25 + 18i % Комплексное число
```

3. *Объявление вектора.* Для объявления вектора необходимо записать его элементы, разделенные пробелами, в квадратных скобках. Пример:

```
a = [0 1]  
b = [3.1 -6]  
c = [a 4 b] % Результат c = [0 1 4 3.1 -6]
```

4. *Объявление монотонно возрастающего или убывающего вектора.* Формат объявления монотонно возрастающего или убывающего вектора следующий: *первый_элемент* : [шаг :] *конечный_элемент*. Если шаг равен 1, то его разрешается опустить. Пример:

```
a = 1:10 % Результат b = [1 2 3 4 5 6 7 8 9 10]  
b = 4:-2:-4 % Результат b = [4 2 0 -2 -4]  
c = -1:3:10 % Результат b = [-1 2 5 8]
```

5. *Объявление массива.* Для объявления вектора необходимо записать его элементы в квадратных скобках, притом элементы строки разделяются пробелами, а строки между собой – точкой с запятой. Пример:

```
a = [1 2 3; 4 5 6; 7 8 9] % Результат  $a = \begin{Bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{Bmatrix}$ 
```

Для объявления массивов также используются специальные функции. Вот некоторые наиболее часто используемые:

`zeros(m, n)` – возвращает массив размером $m \times n$, заполненный нулями;
`ones(m, n)` – возвращает массив размером $m \times n$, заполненный единицами;
`rand(m, n)` – возвращает массив размером $m \times n$, заполненный псевдослучайными величинами, распределенными по равномерному закону;
`randn(m, n)` – возвращает массив размером $m \times n$, заполненный псевдослучайными величинами, распределенными по нормальному закону.

В данных функциях может опускаться первый операнд. В этом случае возвращается квадратная матрица $n \times n$. Пример:

```
a = zeros(1, 100) % Вектор из 100 нулей
b = ones(20)      % Матрица 20x20 единиц
c = rand(2, 50)  % Матрица 2x50 псевдослучайных чисел
```

Доступ к массивам и его элементам

Доступ к массивам осуществляется по их именам. Все операции с массивами осуществляются по значению, т.е. например, при выполнении операции присваивания будет копироваться все содержимое массива. Пример:

```
a = [2 4 5; 4 6 1]
b = a
```

В результате выполнения данной программы переменная *b* будет иметь размер 2×3 и содержать значение $[2\ 4\ 5; 4\ 6\ 1]$, притом *a* и *b* будут размещаться в разных участках памяти.

Для доступа к элементу массива необходимо после имени задать в круглых скобках индекс элемента, состоящий из двух чисел или векторов, разделенных запятой. Нумерация начинается с единицы. Первое число или вектор определяет номер строки (номера строк), второе – соответственно столбца. Если номер строки пропущен (обычно при обращении к элементам вектора), то он считается равным единице. Пример:

```
a = [11 12 13 14 15;          % Исходный массив
     21 22 23 24 25;
     31 32 33 34 35;
     41 42 43 44 45]
b = a(2, 3)                   % Результат: b = 23
a(3) = 6                       % Результат: a(1,3) = 0
c = a(3, 2:4)                 % Результат: c = [32 33 34]
d = a([1 3], 1:3)            % Результат: d = [11 12 13;
                                           %          31 32 33]
a([1 4], [1 5]) = zeros(2,2); % Результат:
                               % a = [0 12 13 14 0;
                               %       21 22 23 24 25;
                               %       31 32 33 34 35;
                               %       0 42 43 44 0]
```

Арифметические операторы

В системе реализовано два типа арифметических операций. Операции над матрицами определены в соответствии с правилами линейной алгебры, а операции над массивами выполняются поэлементно. Для обозначения операций выполняемыми над элементами массива используется знак точки «.» (табл. 1.1).

Операции Matlab

Знак	Операция
+	<p>Сложение: $C=A+B$ Если A – скаляр, B – массив, то $C_{ij} = A + B_{ij}$. Если A – массив, B – скаляр, то $C_{ij} = A_{ij} + B$. Если A – массив, B – массив, то $C_{ij} = A_{ij} + B_{ij}$. Для выполнения операции сложения массивов они должны иметь одинаковый размер.</p>
-	<p>Вычитание: $C=A-B$ Если A – скаляр, B – массив, то $C_{ij} = A - B_{ij}$. Если A – массив, B – скаляр, то $C_{ij} = A_{ij} - B$. Если A – массив, B – массив, то $C_{ij} = A_{ij} - B_{ij}$. Для выполнения операции вычитания массивов они должны иметь одинаковый размер.</p>
*	<p>Умножение матриц: $C=A*B$ Если A – скаляр, B – массив, то $C_{ij} = A \cdot B_{ij}$. Если A – массив, B – скаляр, то $C_{ij} = A_{ij} \cdot B$. Если A – массив, B – массив, то $C_{ik} = \sum_{j=1}^n A_{ij} \cdot B_{jk}$. Для выполнения операции число столбцов первого массива должно быть равно числу строк второго массива.</p>
.*	<p>Поэлементное умножение: $C=A.*B$ Если A – скаляр, B – массив, то $C_{ij} = A \cdot B_{ij}$. Если A – массив, B – скаляр, то $C_{ij} = A_{ij} \cdot B$. Если A – массив, B – массив, то $C_{ij} = A_{ij} \cdot B_{ij}$. Для выполнения операции поэлементного умножения массивов они должны иметь одинаковый размер.</p>
`	<p>Транспонирование матрицы: $A`$ Для действительных массивов результатом является транспонированная матрица. Для комплексных массивов транспонирование дополняется комплексным сопряжением.</p>
.'`	<p>Транспонирование массива: $A.`$ Для действительных и комплексных массивов строки просто заменяются столбцами. Комплексное сопряжение не выполняется.</p>

Логические операторы. Операции отношения.

В языке Matlab используются следующие логические операторы (табл.1.2):

Операции отношения в Matlab

Операция	Отношение
>	Больше
<	Меньше
>=	Больше или равно
<=	Меньше или равно
==	Равно
~=	Не равно

Все логические операторы осуществляют операцию поэлементного сравнения двух массивов. Если один из операндов является скаляром, то он поэлементно сравнивается со всеми элементами другого операнда. Логические операторы возвращают в качестве результата массив того же размера, элементы которого равны единице, если результат сравнения соответствующих элементов равен ИСТИНА, и нулю – в противоположном случае.

Операторы <, >, <=, >= используются для сравнения только действительных частей комплексных элементов, а операции == и ~= осуществляют сравнение как действительных, так и мнимых частей. Пример:

```
X = [2 4 2.5;
      12i 6 3]
Y = [3 4 12;
      2 3 6]
Z = X >= Y

% Результат

Z = [0 1 0;
      0 0 1]
```

Логические операции

В языке Matlab есть три логические операции (табл. 1.3):

Таблица 1.3

Логические операции в Matlab

Операция	И	ИЛИ	НЕ
Обозначение	&		~

При выполнении логических операций массив рассматривается как совокупность булевых переменных, так что значение 0 соответствует булевой значению FALSE, а любое другое значение – булевой значению TRUE. Функция ИСКЛЮЧАЮЩЕЕ ИЛИ реализована в виде функции xor(A, B).

Логические операции имеют низший приоритет по отношению к операциям отношения и арифметическим операциям.

Оператор цикла с определенным числом операций

Оператор цикла с определенным числом операций имеет следующий вид:

```
for v = <выражение-массив>
<операторы>
end
```

В отличие от универсальных языков программирования переменная цикла в языке Matlab является массивом. Поэтому исполнение цикла состоит в том, что переменной цикла присваиваются значения столбцов массива и затем выполняются операторы, которые должны зависеть от переменной v . Тем не менее на практике чаще всего применяют линейные конструкции вида $m : n$ или $a : \langle \text{шаг} \rangle : b$.

Для прерывания выполнения цикла используется оператор `break`.
Пример:

```
for i = 1:10
    x(i) = i .^ 2
end
% Результат x = [1 4 9 16 ... 100]

y = []
for k = [0 3 1 2]
    y = [y 2.^k]
end
% Результат y = [1 8 2 4]
```

Оператор цикла с неопределенным числом операций

Оператор цикла с неопределенным числом операций имеет следующий вид:

```
while <логическое выражение>
<операторы>
end
```

Цикл `while ... end` выполняется до тех пор, пока массив логического выражения не станет нулевым.

Логическое выражение имеет форму

выражение <оператор отношения> выражение,

где допустимы следующие операторы отношений : `==`, `~=`, `<=`, `>=`, `<`, `>`. Пример:

```
i = 1
s = 0
while x(i) ~= 0 & i < 4
    s = s + x(i)
    i = i + 1
end
```

Условное выражение

Как и во всех языках программирования, в языке Matlab есть конструкции для организации условного выполнения операторов. Одна из конструкций имеет следующий формат:

```
if <логическое выражение>
<операторы>
elseif <логическое выражение>
<операторы>
else
<операторы>
end
```

В условном выражении может присутствовать несколько блоков `elseif` или данный блок может отсутствовать. Также может отсутствовать блок `else`.

Логическое выражение имеет форму

выражение <оператор отношения> выражение,

где допустимы следующие операторы отношений : `==`, `~=`, `<=`, `>=`, `<`, `>`. Пример:

```
if a > 0
    b = 2
elseif a < 0
    b = 0
else
    b = 1
end
```

В Matlab имеется еще одна конструкция для организации условного выполнения операторов:

```
switch <анализируемое выражение>
    case <выражение 1>
<операторы>
    case <выражение 2>
<операторы>
    ...
    otherwise
<операторы>
end
```

Оператор `switch` последовательно сравнивает анализируемое выражение с выражениями, записанными после `case`, и если выражения равны, то выполняются соответствующие операторы. Если не было найдено ни одного равенства, то выполняются операторы, записанные после `otherwise`.

Графические возможности языка Matlab

Среда обладает богатыми возможностями для графического представления массивов как в двухмерном, так и в трехмерном виде. В пакет Matlab включены демонстрационные программы для показа возможностей системы, в том числе и графических. Для вызова демонстрационных примеров необходимо в командном окне ввести команду `demo`.

Далее будут рассмотрены графические функции, которые необходимы для выполнения лабораторных работ.

Функция `plot`

Функция `plot` имеет следующий синтаксис:

```
plot(y)
plot(x,y)
plot(x,y,s)
plot(x1,y1,s1,x1,y1,s1)
```

Функция `plot(y)` строит график элементов одномерного массива `y` в зависимости от номера элемента. Если элементы массива `y` комплексные, то строится график, каждая точка которого определяется соответствующей действительной и мнимой частями числа.

Функция `plot(x,y)` соответствует построению обычной функции, когда одномерный массив `x` соответствует значениям аргумента, а одномерный массив `y` – значениям функции (табл. 1.4).

Таблица 1.4

Управление функцией `plot`

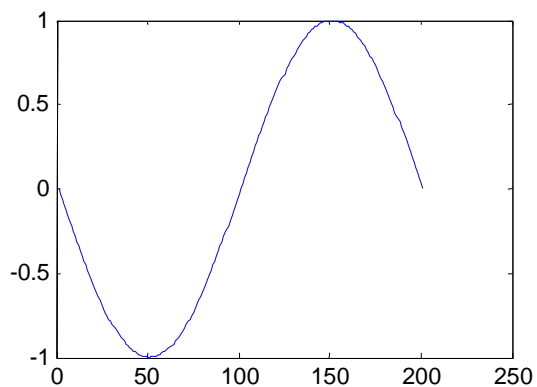
Тип линии	
Непрерывная	-
Штриховая	--
Двойной пунктир	:
Штрихпунктирная	-.
Тип точки	
Точка	.
Плюс	+
Звездочка	*
Кружок	o
Цвет	
Фиолетовый	m
Голубой	c
Красный	r
Зеленый	g
Синий	b
Черный	k

Функция `plot(x, y, s)` аналогична `plot(x, y)`, с той разницей, что текстовая строка `s` определяет цвет и стиль линии, а также вид точек графика. Строка `s` может содержать до трех символов из табл. 1.4.

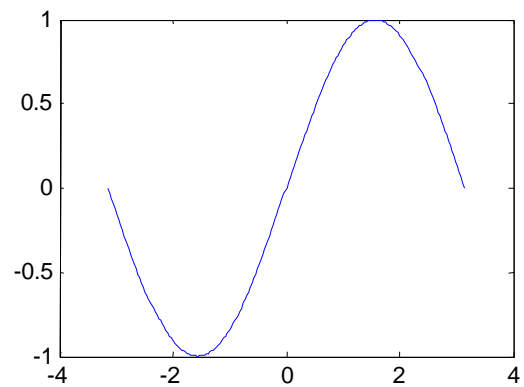
Если цвет линии не указан, он выбирается по умолчанию из шести первых цветов, с желтого до синего, повторяясь циклически.

Функция `plot(x1, y1, s1, x2, y2, s2)` позволяет объединить несколько функций `plot(x, y, s)`. Пример:

```
x = -pi:pi/100:pi
y = sin(x)
plot(y)           % Результат показан на рис. 1.1,а
plot(x, y)       % Результат показан на рис.1.1,б
```



а



б

Рис. 1.1. Результат работы примера, демонстрирующего работу функции `plot`

Функция `stem`

Функция `stem` выводит график элементов одномерного массива в виде вертикальных линий, которые заканчиваются в точках графика, помечаемых кружочком.

Функция `stem` имеет следующий синтаксис:

```
stem(y)
stem(x, y)
stem(x, y, s)
```

Первый вариант функции строит зависимость значений элементов массива от номера элемента, вторая зависимость — $y(x)$, а третья — аналогична второй, за исключением того, что позволяет задавать цвет и стиль линий с помощью строки `s`. Правила задания стиля линий аналогичны функции `plot`. Пример:

```
x = -pi:pi/25:pi
y = sin(x)
stem(y)           % Результат показан на рис. 1.2,а
stem(x, y)       % Результат показан на рис.1.2,б
```

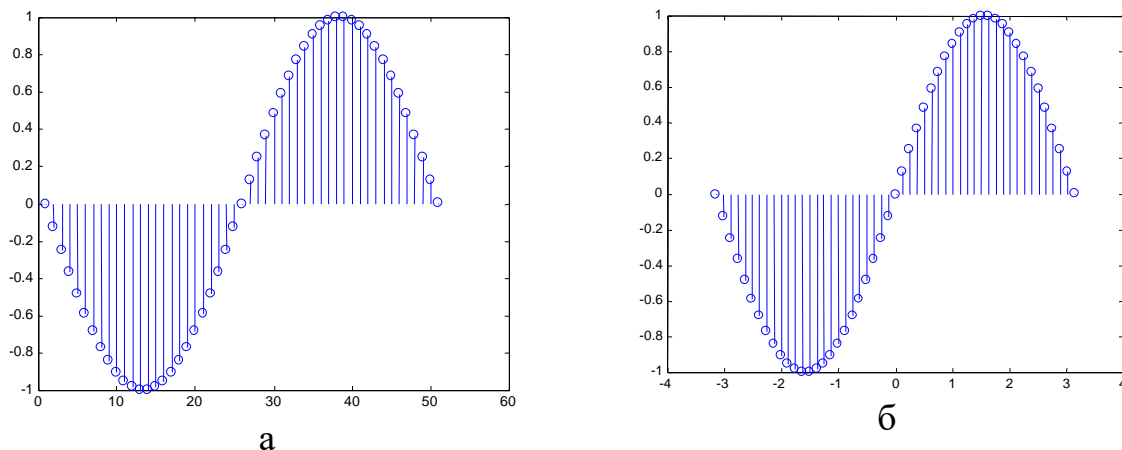


Рис. 1.2. Результат работы примера, демонстрирующего работу функции `stem`

Функция `figure`

Данная функция создает окно для вывода графика, т.е. дальнейший вывод графической информации будет осуществляться в данном окне, пока не встретится функция `figure`.

Функция `subplot`

Функция `subplot` предназначена для разбиения активного окна на области и выбора активной области для вывода графических данных. Функция `subplot` имеет следующий формат:

```
subplot (mnp) ;
subplot (m, n, p) ;
```

где `m` указывает, на сколько частей разбивается окно по вертикали, `n` – по горизонтали, `p` определяет номер области, в которую будут выводиться графические данные. Если числа `n`, `m` и `p` находятся в диапазоне от 1 до 9, то они обычно пишутся слитно (первый вариант записи функции), в противном случае они разделяются запятыми (второй вариант). Пример:

```
n = 0:99
x = sin(0.3*n)
y = cos(0.4*n)
subplot(211)
plot(x)
subplot(212)
plot(y)
```

Данная программа разбивает окно вывода на две части по вертикали и выводит график `x` в верхней части, а график `y` – в нижней. Результат работы программы показан на рис.1..

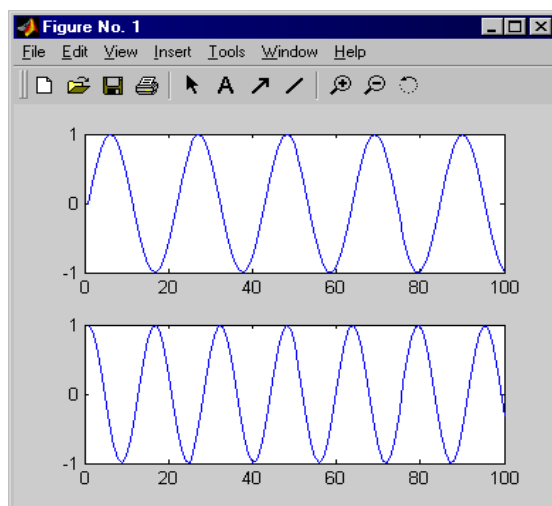


Рис. 1.3. Результат работы программы, демонстрирующей работу функции subplot

1.2 Базовые сигналы ЦОС

Наиболее важными последовательностями, которые часто используются в цифровой обработке сигналов, являются:

а) **единичный импульс** (рис. 1.2):

$$\delta(n) = \begin{cases} 1, & n = 0, \\ 0 & \text{иначе;} \end{cases}$$

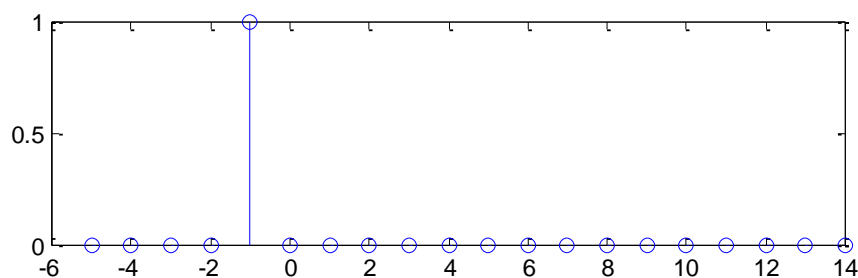


Рис. 1.2. Единичный импульс

б) **единичный скачок** (рис. 1.3):

$$u(n) = \begin{cases} 1, & n > 0, \\ 0 & \text{иначе;} \end{cases}$$

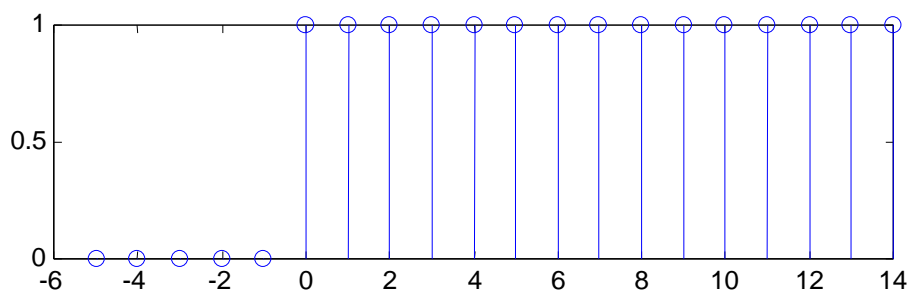


Рис. 1.3. Единичный скачок

в) **убывающая экспонента** (рис. 1.4):

$$g(n) = a^n u(n), \quad |a| < 1;$$

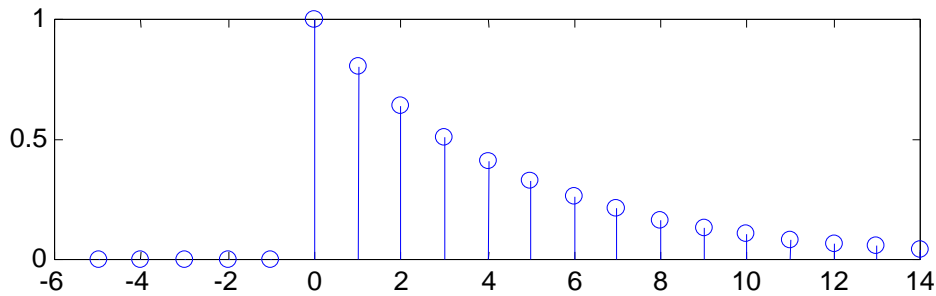


Рис. 1.4. Убывающая экспонента ($a = 0,8$)

г) **косинусоида** (рис.1.5):

$$h(n) = A \cdot \cos\left(\frac{2\pi n}{n_0} + \varphi\right) = A \cdot \cos(\omega n + \varphi);$$

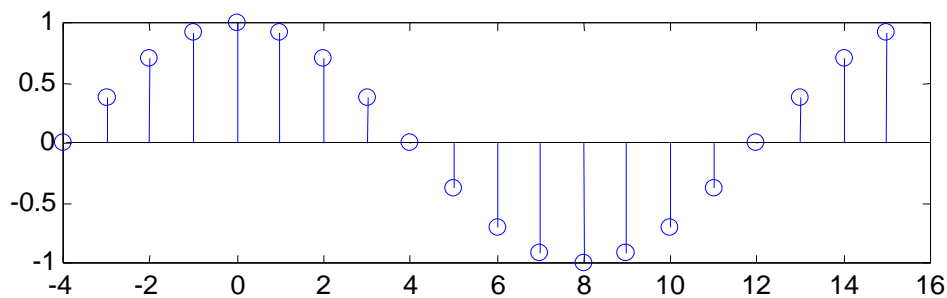


Рис. 1.5. Косинусоида $\cos\left(\frac{2\pi n}{16}\right)$

д) **комплексная экспонента**:

$$e^{j\omega n} = \cos(\omega n) + j \sin(\omega n).$$

1.3 Порядок выполнения работы

1.3.1 Сформируйте и постройте графики следующих последовательностей. Используйте для этого возможности вычисления синуса или косинуса от векторного аргумента:

$$\begin{aligned} x_1(n) &= \sin(\omega_1 n), & n &= 0 \dots 25, \\ x_2(n) &= \sin(\omega_1 n), & n &= -15 \dots 25, \\ x_3(n) &= \sin\left(\omega_2 n + \frac{\pi}{2}\right), & n &= -10 \dots 10, \\ x_4(n) &= \cos(\omega_3 n), & n &= 0 \dots 50. \end{aligned}$$

Значения $\omega_1 \dots \omega_3$ выберите из табл. 1.5.

Таблица 1.5

Значения $\omega_1 \dots \omega_3$						
Номер варианта	1	2	3	4	5	6
ω_1	$\pi/13$	$\pi/14$	$\pi/15$	$\pi/16$	$\pi/17$	$\pi/18$
ω_2	π	2π	3π	π	2π	3π
ω_3	$\pi/\sqrt{19}$	$\pi/\sqrt{21}$	$\pi/\sqrt{23}$	$\pi/\sqrt{20}$	$\pi/\sqrt{22}$	$\pi/\sqrt{18}$

Упростите $x_3(n)$, чтобы не использовать тригонометрические функции.

1.3.2. Сформируйте и постройте график следующих последовательностей:

$$\begin{aligned}
 x_1(n) &= a_1 \delta(n - b_1), & n &= 1 \dots 20, \\
 x_2(n) &= a_2 \delta(n), & n &= -15 \dots 15, \\
 x_3(n) &= a_3 u(n - b_3), & n &= 300 \dots 350, \\
 x_4(n) &= a_4 u(n + b_4), & n &= -10 \dots 0.
 \end{aligned}$$

Для выполнения задания напишите в Matlab функции `delta(n)` и `unit_step(n)` для расчета дельта-функции и функции единичного скачка соответственно. Значения $a_1 \dots a_4$ и $b_1 \dots b_4$ выберите из табл. 1.6.

Таблица 1.6

Числовые значения $a_1 \dots a_4$ и $b_1 \dots b_4$						
Номер варианта	1	2	3	4	5	6
a_1	1,2	1,3	1,4	1,5	1,6	1,7
b_1	4	5	6	7	8	9
a_2	0,6	0,7	0,8	0,9	1,1	1,2
a_3	1,8	1,7	1,6	1,5	1,4	1,3
b_3	310	312	318	321	328	333
a_4	3,8	4,0	4,2	4,4	4,6	4,8
b_4	3	4	5	6	7	8

1.3.3. Напишите функцию для формирования синусоиды, получаемой в результате дискретизации с частотой F_s непрерывной синусоиды:

$$s(t) = A \sin(2\pi f t + \varphi_0),$$

где $t = \dots -3\Delta T, -2\Delta T, -\Delta T, 0, \Delta T, 2\Delta T, \dots \Delta T = 1/F_s$; A – амплитуда; f – частота синусоиды; φ_0 – начальная фаза.

Функция должна иметь шесть входных аргументов: $A, f, \varphi_0, F_s, t_0$ – начальное время, t_1 – конечное. Выходными параметрами функции должны быть временные отметки (моменты времени в которые выполняется дискретизация синусоиды) и значения синусоиды в эти моменты.

Сформируйте синусоиду дискретного времени дискретизацией синусоиды непрерывного времени со следующими параметрами (табл. 1.7):

Таблица 1.7

Числовые значения параметров для формирования синусоид дискретного времени

Номер варианта	1	2	3	4	5	6
Частота сигнала	900 Гц	1 кГц	1,1 кГц	1,3 кГц	1,4 кГц	1,5 кГц
Начальная фаза	45°	60°	30°	45°	60°	30°
Нормализованная амплитуда	40	45	50	55	60	65
Частота дискретизации	8 кГц	8 кГц	8 кГц	10 кГц	10 кГц	10 кГц
Начальное время	0 с	0 с	0 с	0 с	0 с	0 с
Конечное время	6 мс	7 мс	8 мс	4 мс	5 мс	6 мс

1.3.4. Напишите функцию для формирования затухающей экспоненты $g(n) = a^n u(n)$. Постройте график функции, выбрав параметр a согласно варианту (табл. 1.8).

Таблица 1.8

Значения параметра a

Номер варианта	1	2	3	4	5	6
a	0,7	0,75	0,8	0,85	0,9	0,95

1.3.5. Для формирования комплексной экспоненты может быть использована формула Эйлера:

$$x(n) = z^n = r^n e^{j\theta n} = r^n (\cos(\theta n) + j \sin(\theta n)),$$

где $z = re^{j\theta}$, $j^2 = -1$.

Используйте данное выражение со следующими параметрами (табл. 1.9).

Таблица 1.9

Параметры комплексных экспонент (угол θ задан в радианах)

Номер варианта	1	2	3	4	5	6
r	0,89	0,9	0,94	0,95	0,97	0,99
θ	30°	45°	60°	30°	45°	60°

Постройте график действительной и мнимой части для $n = 0 \dots 100$. Как сказывается изменение θ ? Постройте график, откладывая по оси ординат действительную часть, а по оси абсцисс – мнимую (должна получиться спираль). Поэкспериментируйте с углом θ для получения спиралей различного вида.

1.3.6. При помощи формулы Эйлера из комплексных экспонент можно получать функции косинуса и синуса:

$$\sin(\varphi) = \frac{1}{2j}(e^{j\varphi} - e^{-j\varphi}),$$

$$\cos(\varphi) = \frac{1}{2}(e^{j\varphi} + e^{-j\varphi}).$$

Сформируйте и постройте графики следующих последовательностей используя приведенные формулы (комплексная единица в Matlab задается как $1j$ или $1i$):

$$\begin{aligned} x_1(n) &= \sin(\omega_1 n), & n &= 0 \dots 25, \\ x_2(n) &= \cos(\omega_1 n), & n &= -15 \dots 25, \\ x_3(n) &= a^n \sin(\omega_2 n), & n &= -20 \dots 0, \\ x_4(n) &= b^n \cos(\omega_2 n), & n &= 0 \dots 50. \end{aligned}$$

Значения параметров ω_1, ω_2, a, b выберите из табл. 1.10.

Таблица 1.10

Значения параметров ω_1, ω_2, a, b

Номер варианта	1	2	3	4	5	6
ω_1	$\pi/3$	$\pi/6$	$\pi/5$	$\pi/\sqrt{7}$	$\pi/8$	$\pi/13$
ω_2	$\pi/4$	$\pi/9$	$\pi/16$	$\pi/6$	$\pi/\sqrt{11}$	$\pi/3$
a	1,1	1,15	1,4	1,3	1,2	1,05
b	0,89	0,9	0,94	0,95	0,97	0,99

ЛАБОРАТОРНАЯ РАБОТА №2. Разностные уравнения

ЦЕЛЬ РАБОТЫ – изучение разностных уравнений, программирование разностных уравнений в пакете Matlab.

2.1. Теоретические сведения

Разностное уравнение

В практическом применении ЦОС особую роль играет класс систем, которые описываются разностными уравнениями с фиксированными коэффициентами:

$$a_0 y(n) = \sum_{i=0}^M b_i x(n-i) - \sum_{i=1}^N a_i y(n-i), \quad (2.1)$$

где $\{b_i\}$ – коэффициенты прямой связи, которые применяются к поступающему в систему сигналу; $x(n)$ и $\{a_i\}$ – коэффициенты обратной связи, которые применяются к выходному сигналу $y(n)$.

В зависимости от значения коэффициентов $\{b_i\}$ и $\{a_i\}$ выражение (2.1) может описывать работу различных устройств (например интегратора, дифференциатора, полосового фильтра, фильтра нижних частот и т.д.). В Matlab разностные уравнения задаются двумя векторами коэффициентов: $\{b_i\}$ и $\{a_i\}$.

На рис. 2.1 показана схема реализации разностного уравнения (2.1). Заметим, что чаще всего коэффициент a_0 имеет единичное значение и поэтому не показан на схеме. Через z^{-1} на рисунке обозначен блок задержки (регистр), который хранит соответствующий задержанный отсчет. В Matlab линейную систему реализует функция $y = \text{filter}(b, a, x)$, где b – коэффициенты прямой связи, a – коэффициенты обратной связи, x – входной сигнал. Ниже приводится пример использования функции `filter` для реализации линейной системы:

$$y(n) = 0,3x(n) + 0,6x(n-1) + 0,3x(n-2) - 0,9y(n-2). \quad (2.2)$$

```
N = 64;  
x = [1 zeros(1,N-1)]; % дельта-импульс  
b = [0.3 0.6 0.3];  
a = [1 0 0.9];  
y = filter(b,a,x);
```

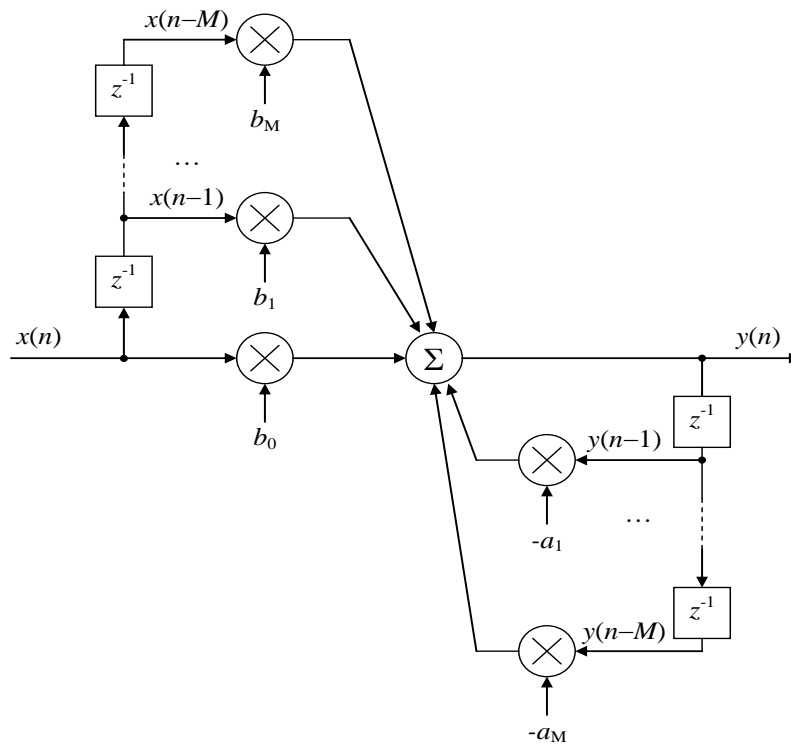


Рис. 2.1.Схема реализации разностного уравнения

На рис. 2.2показаны графики входного и выходного сигналов линейной системы из приведенного примера.

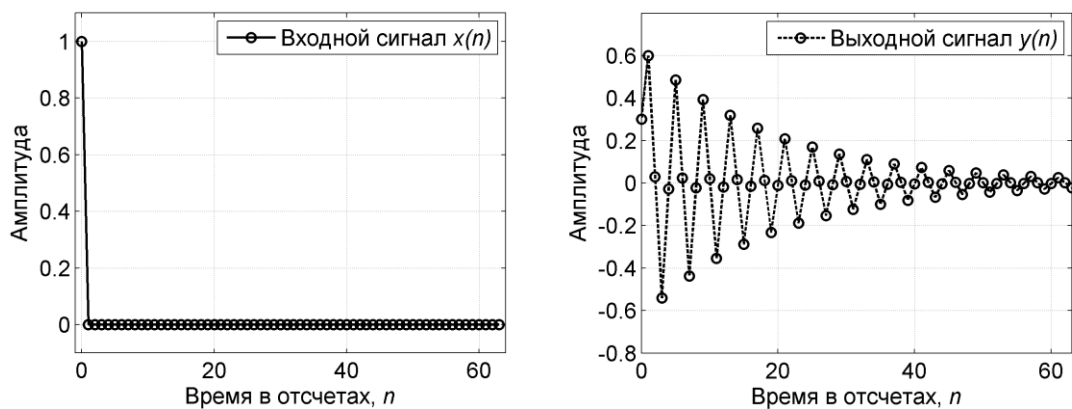


Рис. 2.2. Работа линейной системы

Импульсная характеристика

Системы, описываемые уравнением(2.1), делят на два класса:

- а) рекурсивные;
- б) нерекурсивные.

В *рекурсивных* системах выход $y[n]$ зависит как от входного сигнала $x[n]$ и его предыстории $x[n - i]$, так и от предыдущих выходных значений $y[n - i]$. В *нерекурсивных* системах выходной сигнал зависит только от входного сигнала и его предыстории. Другими словами, у нерекурсивных систем все коэффициенты a_i (кроме a_0)равны нулю.

Помимо задания коэффициентов $\{b_i\}$ и $\{a_i\}$ в выражении (2.1) линейную систему можно описать посредством ее *импульсной характеристики*. **Импульсная характеристика** – это временной сигнал, который генерирует система при подаче на ее вход дельта-импульса $\delta(n)$. Пример получения импульсной характеристики показан на рис. 2.2. Очевидно, что при подаче дельта-импульса на вход нерекурсивной системы ее выходной сигнал будет иметь конечную длительность, отчего такие системы называются системами с конечной импульсной характеристикой (КИХ). Для нерекурсивных систем вследствие наличия обратной связи характерна бесконечная импульсная характеристика (БИХ).

Собственная частота

Известно, что импульсная характеристика линейной системы может содержать колебания нескольких *собственных частот*. Эти собственные частоты определяются коэффициентами обратной связи $\{a_i\}$. Каждый корень характеристического полинома (p_i)

$$A(z) = 1 + \sum_{i=1}^N a_i z^{-i} \quad (2.3)$$

дает свой вклад в импульсную характеристику вида $p_i^n u(n)$.

В Matlab для определения корней полинома имеется функция `root`. Для получения полной информации о работе данной функции `root` наберите в консольном окне системы Matlab `help root`.

Для разностных уравнений второго порядка (как например в (2.2)) характерно наличие двух различных собственных частот. Вследствие чего их импульсная характеристика описывается выражением

$$h(n) = (\alpha p_1^n + \beta p_2^n) u(n). \quad (2.4)$$

Линейная система

Линейная система, описываемая уравнением (2.1), входит в более общий класс *систем дискретного времени*:

$$y[n] = L[x[n]].$$

В этом выражении оператор $L[\cdot]$ задает алгоритм, согласно которому из последовательности $x[n]$ получается $y[n]$. Поскольку большинство систем, рассматриваемых в курсе ТиПЦОС, являются *линейными* (КИХ, БИХ-фильтры и дискретное преобразование Фурье), то необходимо знать их отличительные свойства.

Как же отличить линейную систему от любой другой? Существенным признаком линейной системы является выполнение **принципа суперпозиции**:

$$L[a_1x_1[n] + a_2x_2[n]] = a_1L[x_1[n]] + a_2L[x_2[n]].$$

Другое важное свойство линейных систем – это их **инвариантность** к сдвигу. Инвариантность означает, что если сигнал $x[n]$ вызывает отклик $y[n]$, то задержанный сигнал $x[n - k]$ будет вызывать задержанный отклик $y[n - k]$. Рассмотрим пример, поясняющий введенные понятия.

Пример 2.1. Нелинейная система

Пусть дискретная система описывается уравнением

$$y[n] = [x[n]]^2. \quad (2.5)$$

Чтобы определить, является ли система **линейной**, проверим выполнение принципа суперпозиции для двух тестовых сигналов $x_1[n]$ и $x_2[n]$ (рис. 2.3).

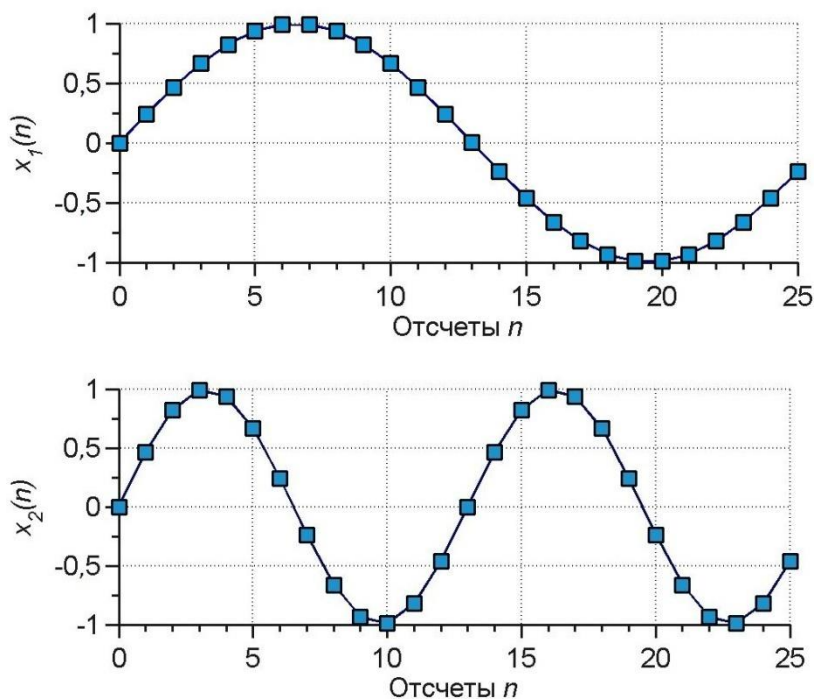


Рис. 2.3. Входные сигналы

Сложим эти два сигнала и пропустим через дискретную систему (2.5). Результат показан на рис.2.4.

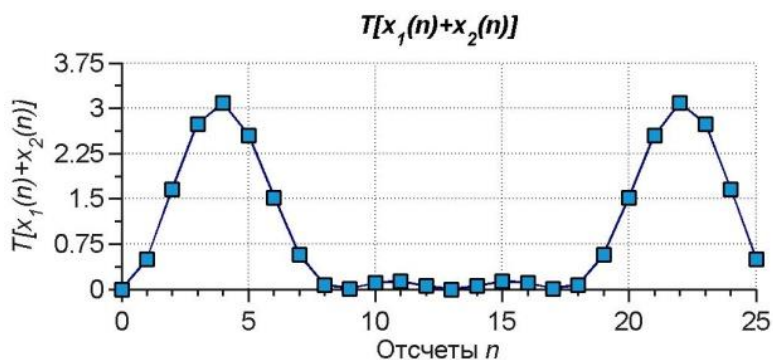


Рис. 2.4. Выходной сигнал после подачи на вход суммы сигналов $x_1[n]$ и $x_2[n]$

Теперь пропустим по отдельности сигналы $x_1[n]$ и $x_2[n]$ через рассматриваемую дискретную систему и затем сложим их, в результате чего получим сигнал, изображенный на рис. 2.5.

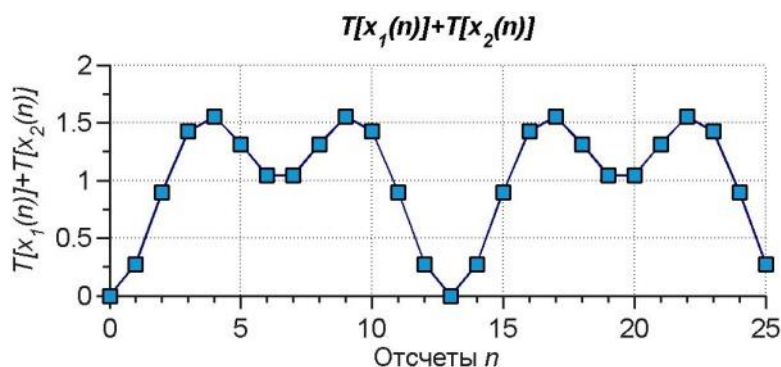


Рис. 2.5. Суммавыходных сигналов системы после подачи на вход сигналов $x_1[n]$ и $x_2[n]$

Легко видеть, что графики на рис. 2.4 и 2.5 различны. Из этого можно заключить, что (2.5) описывает **нелинейную** систему.

Z-преобразование

Z-преобразование дискретного сигнала $x(n)$ имеет вид

$$X(z) = \sum_{n=0}^{\infty} x(n)z^{-n}.$$

Одной из особенностей z-преобразования является свойство **задержки и сдвига**, которое имеет следующий смысл: если z-образ последовательности $x(n)$ равен $X(z)$, то z-образ задержанной последовательности $x(n - m)$ равен $z^{-m}X(z)$:

$$\begin{aligned} x(n) &\rightarrow X(z), \\ x(n - m) &\rightarrow z^{-m}X(z). \end{aligned}$$

Z-преобразование бывает весьма полезно, когда необходимо определить стабильность линейной системы (везде далее под термином «линейная система» мы будем иметь в виду линейную систему, инвариантную к сдвигу). Система называется *стабильной*, если на любое ограниченное воздействие $x[n]$ ее отклик также ограничен, т.е.

$$|x(nT)| \leq M_x < \infty$$

для всех $n = 0, 1, 2, \dots$, и

$$|y(n)| \leq M_y < \infty$$

для всех $n = 0, 1, 2, \dots$.

Пример 2.2. Нестабильность линейной системы

В качестве тестового сигнала будем использовать единичный скачок длительностью 5 отсчетов (рис.2.6).

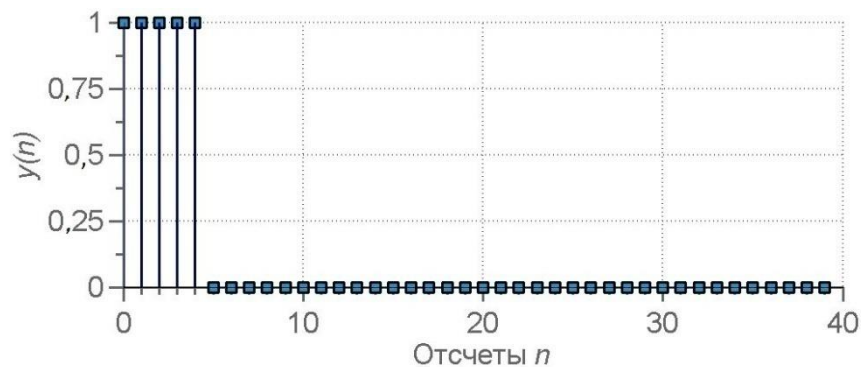


Рис. 2.6. Единичный отсчет длительностью пять отсчетов

Приведем отклик на единичный скачок нестабильной линейной системы (рис. 2.7).

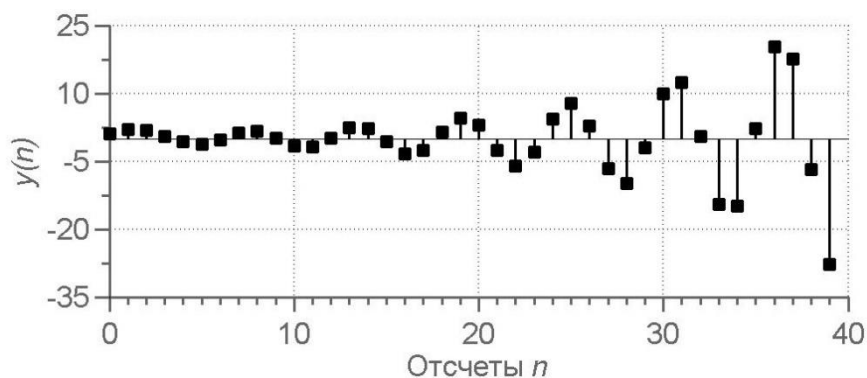


Рис. 2.7. Нестабильная система

Система нестабильна, поскольку отклик на сигнал ограниченной длительности не сходится к нулю и лишь увеличивается с увеличением n .

Приведем отклик на тот же единичный скачок стабильной системы (рис.2.8).

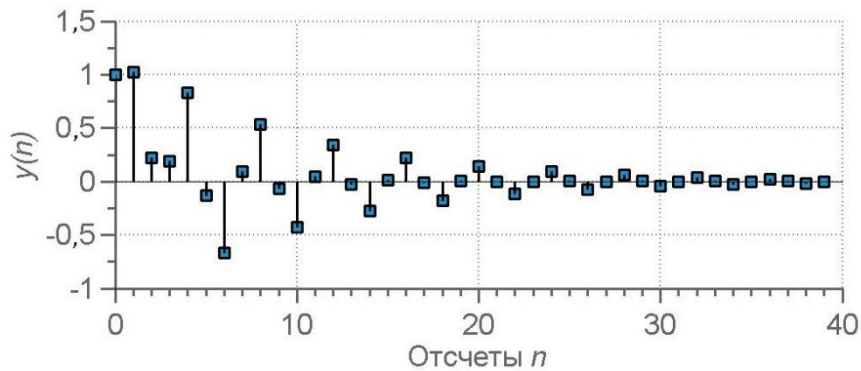


Рис. 2.8. Стабильная система

Легко увидеть, что отклик системы сходится к нулю – значит, система стабильна.

Часто необходимо определить стабильность линейной системы, зная лишь ее коэффициенты $\{b_i\}$ и $\{a_i\}$ из уравнения (2.1). Для этого необходимо обратиться к математическому аппарату z -преобразования.

Вначале рассмотрим уравнение свертки во временной области:

$$y(n) = \sum_{i=0}^M h_i x(n - i), \quad (2.6)$$

где $x(n)$ – это входной сигнал; $y(n)$ – выходной сигнал; $\{h_i\}$ – фиксированные коэффициенты.

Заметим, что сверткой описывается работа линейных нерекурсивных систем. Учитывая свойство задержки

$$\begin{aligned} h_0 x(n) &\rightarrow h_0 X(z), \\ h_i x(n - i) &\rightarrow h_i z^{-i} X(z), \end{aligned}$$

z -преобразование для (2.6) принимает вид

$$Y(z) = \sum_{i=0}^M h_i z^{-i} X(z). \quad (2.7)$$

Вынося $X(z)$ за знак суммы, получаем

$$Y(z) = X(z) \sum_{i=0}^M h_i z^{-i} = X(z) H(z). \quad (2.8)$$

Таким образом, получено еще одно свойство z -преобразования, которое показывает, что свертка двух сигналов во временной области эквивалента перемножению их образов в z -области.

Аналогичным образом можно поступить с разностным уравнением (2.1). Применяя свойство задержки z -преобразования

$$\begin{aligned} a_i x(n-i) &\rightarrow a_i z^{-i} X(z), \\ b_i y(n-i) &\rightarrow b_i z^{-i} Y(z), \end{aligned}$$

получим

$$Y(z) = \sum_{i=0}^M b_i z^{-i} X(z) - \sum_{i=1}^M a_i z^{-i} Y(z).$$

Преобразуем выражение к виду

$$Y(z) \left(1 + \sum_{i=1}^M a_i z^{-i} \right) = X(z) \sum_{i=0}^M b_i z^{-i},$$

из которого легко получить *передаточную функцию* линейной дискретной системы:

$$H(z) = \frac{Y(z)}{X(z)} = \frac{\sum_{i=0}^M b_i z^{-i}}{\left(1 + \sum_{i=1}^M a_i z^{-i} \right)}. \quad (2.9)$$

Передаточная функция $H(z)$ полностью определяет линейную систему. Зная коэффициенты $\{b_i\}$ и $\{a_i\}$, из выражения (2.9) можно записать разностное уравнение (2.1), которое описывает работу линейной системы.

Рассмотрим передаточную функцию в общем виде

$$H(z) = \frac{N(z)}{D(z)},$$

где $N(z)$ и $D(z)$ – полиномы от z^{-1} порядка M .

Поскольку полином степени M имеет ровно M корней, функцию $H(z)$ можно разложить на множители и представить в виде:

$$H(z) = \frac{K(z - z_1)(z - z_2) \dots (z - z_M)}{(z - p_1)(z - p_2) \dots (z - p_M)}, \quad (2.10)$$

где z_i – i -й нуль; p_i – i -й полюс; K – коэффициент усиления.

Информацию, содержащуюся в уравнении (2.10), удобно и полезно изображать в виде диаграммы нулей и полюсов в z -плоскости (см. пример 2.3). На диаграмме крестиком обозначается положение полюсов, а кружком – положение нулей. Важной особенностью диаграммы нулей и полюсов является *единичная окружность*, которая задается уравнением $|z| = 1$. Для определения стабильности линейной системы существует правило: *устойчивых (стабильных) систем все полюса должны лежать внутри единичной окружности в z -плоскости*.

Пример 2.3. Определение стабильности системы

Определить стабильность системы с коэффициентами $b_0 = 1$, $b_1 = 1$ и $a_1 = -0,5$ и изобразить схему, реализующую систему.

Вначале построим передаточную функцию согласно формуле (2.9):

$$H(z) = \frac{1 + z^{-1}}{1 - 0,5z^{-1}}. \quad (2.11)$$

Затем преобразуем к виду (2.10):

$$H(z) = \frac{z + 1}{z - 0,5}. \quad (2.12)$$

Диаграмма нулей и полюсов для (2.12) показана на рис. 2.9, из которой видно, что единственный полюс рассматриваемой системы $p_1 = 0,5$ лежит внутри единичной окружности и, следовательно, линейная система стабильна.

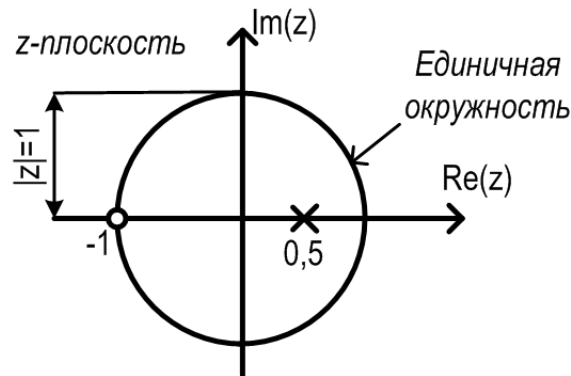


Рис. 2.9. Диаграмма нулей и полюсов

Чтобы изобразить схему, реализующую линейную систему, необходимо преобразовать передаточную функцию (2.11) к разностному уравнению:

$$y(n) = x(n) + x(n - 1) + 0,5y(n - 1). \quad (2.13)$$

На рис.2.10 приведена схема, реализующая выражение (2.13): через z^{-1} изображаются блоки задержки сигнала на один такт. При аппаратной реализации задержка реализуется в виде регистра, который хранит предыдущее значение входного или выходного отсчета.

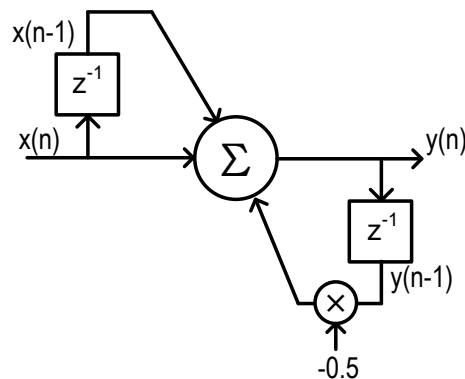


Рис. 2.10. Блок-схема фильтра

Частотная характеристика линейной системы

Для описания линейных систем в частотной области используется специальный входной сигнал:

$$x(n) = e^{j\omega n}, -\infty < n < \infty. \quad (2.14)$$

Если такая последовательность поступает на вход линейной системы с импульсной характеристикой $h(n)$, то на выходе появится последовательность

$$y(n) = \sum_{m=0}^M h(m) \cdot e^{j\omega(n-m)} = e^{j\omega n} \sum_{m=0}^M h(m) \cdot e^{-j\omega m} = x(n)H(e^{j\omega}).$$

Таким образом, при подаче на вход сигналов вида (2.14) выходной сигнал совпадает со входным с точностью до комплексного множителя $H(e^{j\omega})$. Этот комплексный коэффициент называется *частотной характеристикой системы* и выражается через ее импульсную характеристику следующим образом:

$$H(e^{j\omega}) = \sum_{m=0}^M h(m) \cdot e^{-j\omega m}.$$

Частотная характеристика является периодической функцией ω , причем ее период равен 2π . Эта периодичность связана со спецификой дискретного колебания: входная последовательность с частотой $(\omega + 2m\pi)$ ($m = \pm 1, \pm 2, \dots$) не отличается от входной последовательности с частотой ω , т.е.

$$\tilde{x}(n) = e^{j(\omega+2m\pi)n} = e^{j\omega n} = x(n).$$

Поскольку $H(e^{j\omega})$ – периодическая функция, то для полного описания достаточно задать ее на любом интервале длиной 2π . Обычно для этой цели используют интервал $0 \leq \omega \leq 2\pi$.

2.2. Порядок выполнения работы

2.2.1. Для вариантов 1–3: найдите коэффициенты $\{b_i\}$ и $\{a_i\}$ фильтра, который задан своей передаточной функцией.

$$H(z) = \frac{k(z - z_1)(z - z_2)}{(z - p_1)(z - p_2)}.$$

Таблица 2.1

Числовые значения параметров k, z_1, z_2, p_1 и p_2

Номер варианта	k	z_1	z_2	p_1	p_2
1	0,09316	-0,99868 +0,05141j	-0,99868 -0,05141j	+0,51743 +0,40197j	+0,51743 -0,40197j
2	0,44405	-0,99992 +0,01297j	-0,99992 -0,01297j	-0,28370 +0,48321j	-0,28370 -0,48321j
3	0,17534	-0,99946 +0,03280j	-0,99946 -0,03280j	+0,28602 +0,48265j	+0,28602 -0,48265j

Для вариантов 4–6: запишите передаточную функцию и найдите нули и полюса фильтра, который задан в виде блок-схемы (рис. 2.11). Значения a_1, a_2, b_0, b_1 и b_2 выберите из табл. 2.2.

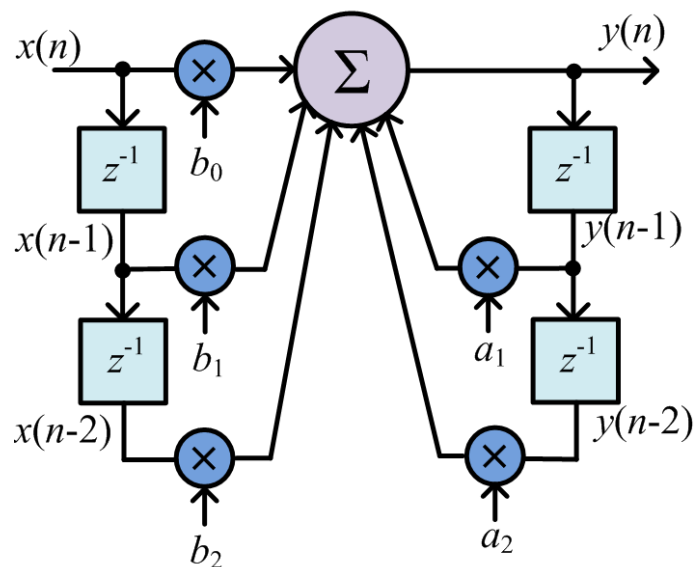


Рис. 2.11. Блок-схема линейной системы второго порядка

Числовые значения параметров a_1, a_2, b_0, b_1 и b_2

Номер варианта	a_1	a_2	b_0	b_1	b_2
4	0,22888	2,06144	0,86232	0,19501	1,73566
5	0,06797	1,53449	0,67549	0,05196	1,35097
6	0,09241	1,63060	0,70587	0,07190	1,41295

2.2.2. Напишите функцию, реализующую разностное уравнение (2.1). На вход функции поступают коэффициенты $\{b_i\}, \{a_i\}$ и входной сигнал $x[n]$. С помощью написанной функции постройте импульсную характеристику для фильтра из задания 2.2.1. Сравните результат с работой функции `filter`.

2.2.3. Определите собственные частоты импульсной характеристики фильтра из задания 2.2.1 (см. формулу (2.3) и пояснение к ней). Найдите коэффициенты α и β из уравнения (2.4) с которыми колебания собственных частот входят в импульсную характеристику. Для этого вычислите $h(n)$ для любых двух значений n (например для $n = 0$ и 1) и составьте систему линейных уравнений относительно неизвестных α и β . Решите эту систему с использованием Matlab-оператора «\» (backslash). Используя найденные значения α и β , постройте график $h(n)$ согласно уравнению (2.4) и сравните его с результатом из задания 2.2.2.

2.2.4. Найдите отклик фильтра из задания 2.2.1 при воздействии на вход единичного скачка $x(n) = u(n - n_0)$, $n = 1 \dots N$. Выбрав достаточное значение N , определите, к какому значению сходится выходной сигнал фильтра. Это значение называют устоявшимся режимом фильтра, а переменную часть характеристики – переходной характеристикой. В качестве n_0 возьмите номер своего варианта (т.е. если выполняете вариант №2, то $n_0 = 2$).

2.2.5. Вычислите отклик фильтра на следующие сигналы (табл.ица 2.3)*.

Таблица 2.3

Сигналы для воздействия на линейную систему

Номер варианта	Задание
1–2	Периодический прямоугольный сигнал с частотой 10 Гц и сигнал такой же формы с частотой 20 Гц. Частота дискретизации сигналов $f_s = 80$ Гц
3–4	Периодический треугольный сигнал с частотой 15 Гц и сигнал такой же формы с частотой 80 Гц. Частота дискретизации сигналов $f_s = 320$ Гц
5	Периодический пилообразный сигнал с частотой 20 Гц и сигнал такой же формы с частотой 110 Гц. Частота дискретизации сигналов $f_s = 440$ Гц
6	Периодический трапециевидный сигнал с частотой 18 Гц и сигнал такой же формы с частотой 130 Гц. Частота дискретизации сигналов $f_s = 520$ Гц

* Функции для генерирования сигналов заданной формы можно найти в сетевом каталоге: "\\1-311-kafevs\Docs\ГиПЦОС\Функции для ЛРН№2\".

2.2.6. Вычислите частотную характеристику линейной системы из задания 2.2.1 по формуле

$$H(e^{j\omega}) = \frac{\sum_{k=0}^M b_k e^{j\omega k}}{(1 + \sum_{k=1}^M a_k e^{j\omega k})}, \quad \omega \in [0 \ \pi].$$

Постройте график $A(\omega) = |H(e^{j\omega})|$ и $\varphi(\omega) = \arg H(e^{j\omega})$. Функция $A(\omega)$ – это амплитудно-частотная характеристика (АЧХ), а $\varphi(\omega)$ – фазо-частотная характеристика (ФЧХ).

2.2.7. Постройте АЧХ и ФЧХ линейной системы из задания 2.2.1 при помощи функции `freqz`.

2.2.8*. Алгоритм Карплуса–Стронга для имитации звука гитарной струны. Рассмотрим разностное уравнение:

$$y[n] = \alpha y[n - M] + x[n], \quad (2.15)$$

где $x[n]$ – входной сигнал; M – задержка; α – коэффициент затухания.

Предполагается, что значение задержки M равно длине входного сигнала. Число генерируемых выходных значений $y[n]$ должно быть кратно M . Согласно алгоритму на вход необходимо подать случайную последовательность длиной M . Такую последовательность можно получить следующим образом:

```
x = (1/3) * randn(M, 1);
```

Напишите в Matlab функцию `y=ks_synthesis(x, alpha, P)`, которая генерирует выходную последовательность согласно разностному уравнению (2.15), длина выходной последовательности равна $M \times P$, где M – длина последовательности x .

Для получения «реальных аккордов» значение M должно быть выбрано исходя из частоты дискретизации сигнала F_s и «опорной» частоты F_0 . Например А4 имеет «опорную» частоту 440 Гц, другие ноты могут быть получены из нее по формуле $F_0 = 440 \times 2^{n/12}$, где n – число полутонов между А4 и необходимой нотой. Например, открытый аккорд из песни Битлз «Hardday'snight» состоит из следующих нот D3, F3, G3, F4, A4, C5, G5. Ниже приведена заготовка Matlab-кода, необходимая для генерирования данного аккорда:

* Выполнение данного задания не является обязательным.


```

clear all; close all; clc;

% Параметры:
%
% - Fs      : частота дискретизации
% - F0      : частоты ноты, формирующей аккорд
% - gain    : усиление отдельных нот в аккорде
% - duration : длительность аккорда в секундах
% - alpha   : ослабление в алгоритме Карплуса-Стронга

Fs = 48000;

% D2, A2, D3, F3, G3, F4, A4, C5, G5
F0 = 440 * [2^-(31/12); 2^-(19/12); 2^-(16/12); 2^(-14/12); 2^-(4/12); 1; 2^(3/12); 2^(10/12)];
gain = [1.2 3.0 1.0 2.2 1.0 1.0 1.0 3.5];
duration = 4;
alpha = 0.9785;

% Количество отсчетов в аккорде
nbsample_chord = Fs * duration;

first_duration = ceil(nbsample_chord / round(Fs/F0(1)));

% Инициализация
chord = zeros(nbsample_chord, 1);

for i = 1:length(F0)

    current_M = round(Fs/F0(i));
    current_duration = ceil(nbsample_chord/current_M);

    current_alpha = alpha^(first_duration/current_duration);

% Формирование входного и выходного сигнала алгоритма Карплуса-
% Стронга
x = (1/3) * rand(current_M, 1);
y = ks_synthesis(x, current_alpha, current_duration);
y = y(1:nbsample_chord);

    % Добавление ноты к аккорду
    chord = chord + gain(i) * y;
end

% Проигрывание аккорда
sound(chord, Fs);

```

ЛАБОРАТОРНАЯ РАБОТА №3. Дискретное преобразование Фурье

ЦЕЛЬ РАБОТЫ – Изучение дискретного преобразования Фурье (ДПФ) и его свойств, получение практических навыков использования ДПФ в пакете Matlab.

3.1. Теоретические сведения

Частотно-временной анализ. Преобразование Фурье

Преобразование Фурье является одним из мощнейших и эффективных средств для решения многих связанных с обработкой сигналов. *Жан Батист Жозеф Фурье* (1768–1830) предложил концепцию представления заданного сигнала в виде тригонометрического ряда из косинусов и синусов:

$$x(t) = \frac{a_0}{2} + \sum_{k=1}^{\infty} (a_k \cos \omega_k t + b_k \sin \omega_k t), \quad (3.1)$$

где $\omega_k = \frac{2\pi k}{T_p}$; T_p – период сигнала.

Особенность разложения (3.1) в том, что оно применимо только для периодического сигнала (рис. 3.1).

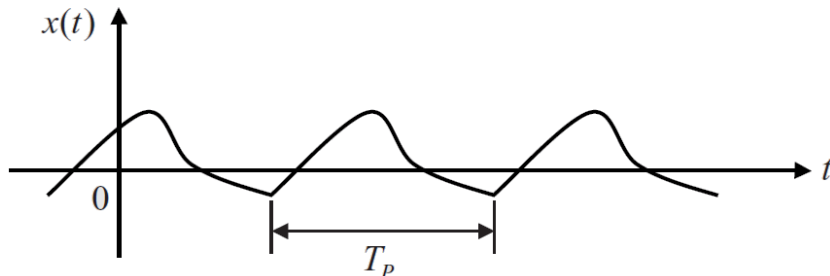


Рис. 3.1. Периодический сигнал

Самая низкочастотная составляющая сигнала, которая входит в (3.1), называется *основной частотой*, или *частотой основного тона*, $f_1 = 1/T_p$, поскольку все остальные частоты кратны ей. Частотные компоненты сигнала называются *гармониками*. Например, частотная компонента $(a_2 \cos \omega_2 t + b_2 \sin \omega_2 t)$ является второй гармоникой. Коэффициент $a_0/2$ является постоянной составляющей и представляет собой *среднее значение* сигнала за период. Причина, по которой стремятся представить сигнал в форме (3.1), заключается в том, что часто полезно разложить «сложный» сигнал в сумму «простых» – в данном случае косинусов и синусов. Не следует считать, что периодические сигналы редкость в природе. Например, на рис. 3.2 показан участок речевого сигнала (звук И), форма которого близка к периодической. Похожей периодичностью обладают почти все гласные (вокализованные) звуки.

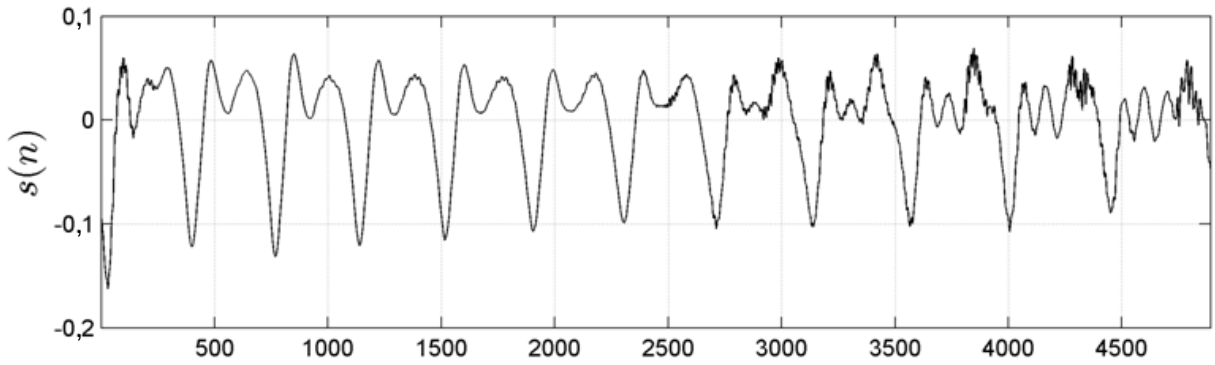


Рис. 3.2. Пример речевого сигнала

Ортогональность функций \sin и \cos

Часто возникает вопрос, почему для разложения функции в ряд (3.1) выбраны именно функции \sin и \cos ? Так произошло благодаря тому, что эти функции обладают особым свойством *ортогональности*, известным из курса геометрии, где оно относилось к векторам. Оказывается, между функциями и векторами существует аналогия: как произвольный вектор можно представить в виде суммы ортогональных векторов (составляющих базис), так и произвольную функцию можно представить в виде суммы ортогональных функций (рис. 3.3).

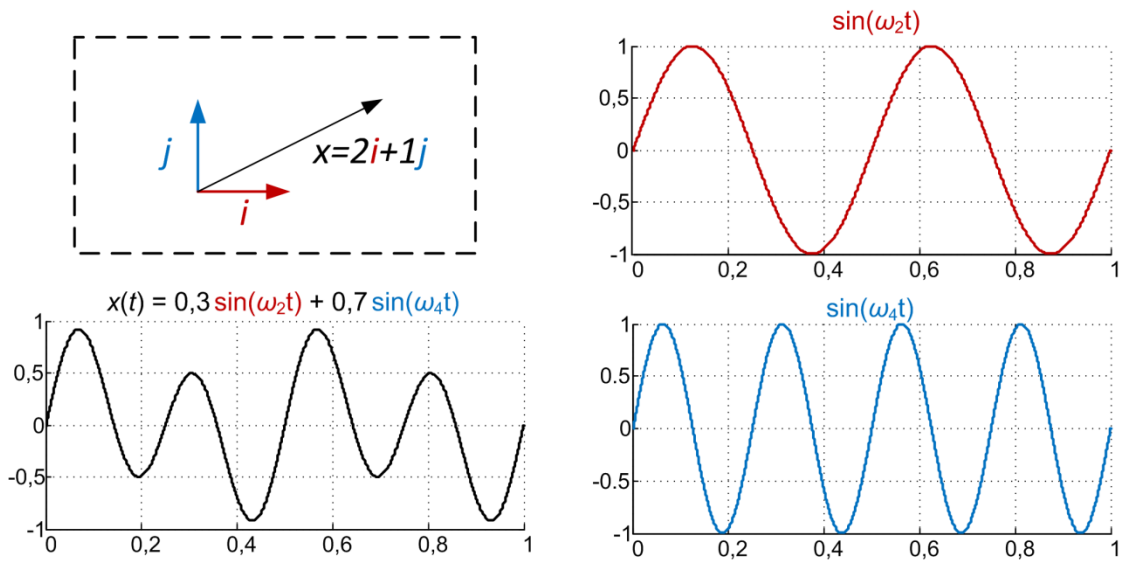


Рис. 3.3. Разложение в ортогональный базис вектора и функции

Различие при разложении вектора и функции в том, что количество базисных векторов конечно, а число базисных функций бесконечно*.

* Иногда говорят, что функция – это *бесконечномерный* вектор.

Как определить, является ли одна функция ортогональной по отношению к другой? Известно, что ортогональные векторы имеют нулевую проекцию друг на друга. С понятием проекции связано понятие скалярного произведения векторов

$$\mathbf{a} \cdot \mathbf{b} = a_1 b_1 + a_2 b_2 + \dots + a_n b_n = \sum_{i=1}^n a_i b_i,$$

которое равно нулю для ортогональных векторов. Для функций также есть аналог скалярного произведения:

$$\int_a^b f_1(t) \cdot f_2(t) dt.$$

Это выражение равно нулю, если функции $f_1(t)$ и $f_2(t)$ ортогональны на интервале $[a, b]$, иначе получаемое число показывает *проекцию* одной функции на другую. Исходя из того, что геометрический смысл интеграла – площадь под кривой, на рис. 3.4 иллюстрируется понятие скалярного произведения функций.

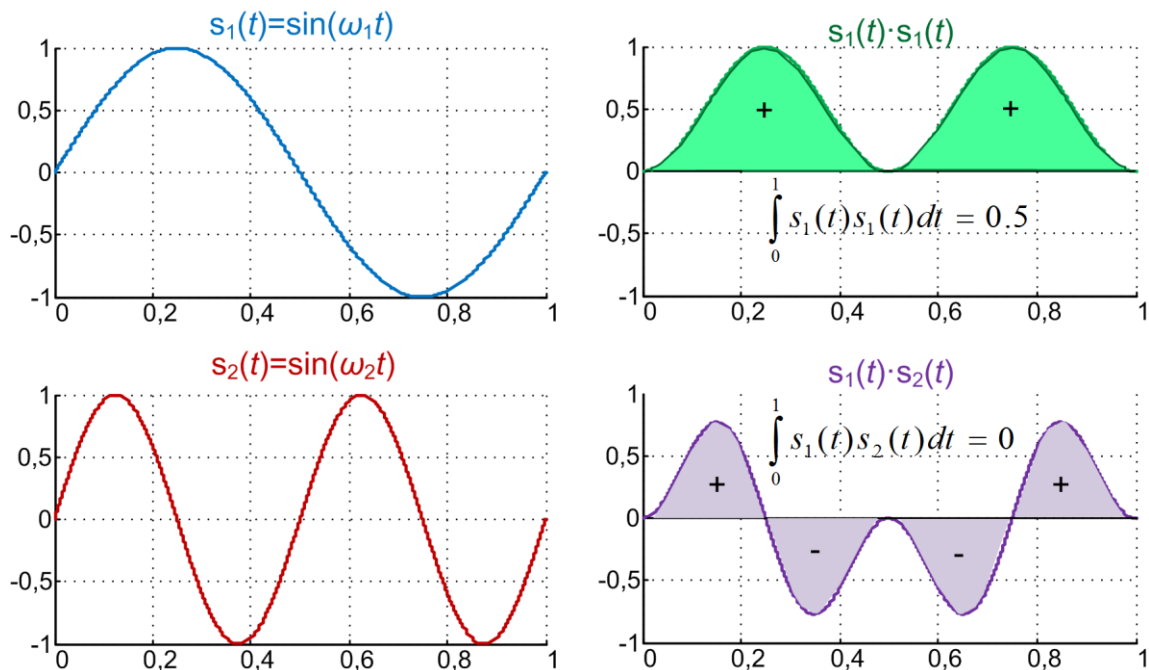


Рис. 3.4. Скалярное произведение функций

Учитывая введенные вначале обозначения, свойство ортогональности функций \sin и \cos записываются следующим образом:

$$\int_{-T/2}^{T/2} \cos \omega_k t \sin \omega_m t dt = 0, \quad \forall k, m, \quad (3.2)$$

$$\int_{-T/2}^{T/2} \cos \omega_k t \cos \omega_m t dt = \begin{cases} 0, & k \neq m \\ T/2, & k = m, \end{cases}$$

$$\int_{-T/2}^{T/2} \sin \omega_k t \sin \omega_m t dt = \begin{cases} 0, & k \neq m \\ T/2, & k = m. \end{cases}$$

Фурье-анализ

Основная задача Фурье-анализа найти коэффициенты a_k и b_k в выражении (3.1). Для этого необходимо умножить левую и правую части (3.1) на $\cos \omega_m t$:

$$x(t) \cos \omega_m t = \frac{a_0}{2} + \sum_{k=1}^{\infty} (a_k \cos \omega_k t \cos \omega_m t + b_k \sin \omega_k t \cos \omega_m t).$$

Интегрируя обе части по переменной $t \in [-T/2, T/2]$, получаем

$$\int_{-T/2}^{T/2} x(t) \cos \omega_m t dt = \frac{a_0}{2} \int_{-T/2}^{T/2} \cos \omega_m t dt +$$

$$+ \sum_{k=1}^{\infty} \left(a_k \int_{-T/2}^{T/2} \cos \omega_k t \cos \omega_m t dt + b_k \int_{-T/2}^{T/2} \sin \omega_k t \cos \omega_m t dt \right).$$

Используя свойство ортогональности (3.2), легко можно найти выражения для коэффициентов a_k и b_k :

$$a_k = \frac{2}{T} \int_{-T/2}^{T/2} x(t) \cos \omega_k t dt,$$

$$b_k = \frac{2}{T} \int_{-T/2}^{T/2} x(t) \sin \omega_k t dt.$$

Часто бывает полезным следующее разложение функции $x(t)$, которое может быть получено из (3.1):

$$x(t) = \frac{a_0}{2} + \sum_{k=1}^{\infty} M_k \cos(\omega_k t + \varphi_k),$$

где $\omega_1 = 2\pi/T$ – основная частота (частота основного тона); $M_k = \sqrt{a_k^2 + b_k^2}$ – амплитуда компоненты сигнала на частоте $k\omega_1$; $\varphi_k = \text{arctg}(-b_k/a_k)$ – фаза компоненты сигнала на частоте $k\omega_1$.

Пример 3.1. Сигнал-меандр

В качестве примера рассмотрим ряд Фурье для периодического сигнала $x(t+nT) = x(t)$, который определен следующим образом (рис. 3.5):

$$x(t) = \begin{cases} -1, & -T/2 < t < 0, \\ 1, & 0 \leq t < T/2. \end{cases} \quad (3.3)$$

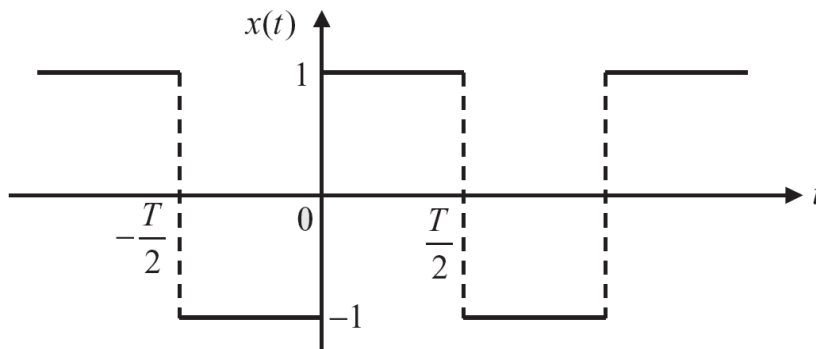


Рис. 3.5. Меандр

Легко заметить, что среднее значение сигнала равно нулю:

$$\frac{a_0}{2} = \frac{1}{T} \int_{-T/2}^{T/2} x(t) dt = 0,$$

а коэффициенты a_k и b_k равны

$$a_k = \frac{2}{T} \int_{-T/2}^{T/2} x(t) \cos(\omega_k t) dt = \frac{2}{T} \left[\int_{-T/2}^0 -\cos(\omega_k t) dt + \int_0^{T/2} \cos(\omega_k t) dt \right] = 0,$$

$$\begin{aligned} b_k &= \frac{2}{T} \int_{-T/2}^{T/2} x(t) \sin(\omega_k t) dt = \frac{2}{T} \left[\int_{-T/2}^0 -\sin(\omega_k t) dt + \int_0^{T/2} \sin(\omega_k t) dt \right] = \\ &= \frac{2}{k\pi} (1 - \cos k\pi). \end{aligned}$$

Подставляя полученные значения в (3.1), получим выражение для сигнала:

$$x(t) = \frac{4}{\pi} \left[\sin(\omega_1 t) + \frac{1}{3} \sin(3\omega_1 t) + \frac{1}{5} \sin(5\omega_1 t) + \dots \right]. \quad (3.4)$$

Необходимо отметить, что в разложении участвуют только функции \sin . Это происходит из-за того, что функция меандра является нечетной и для ее представления не нужны четные функции (\cos).

Полезно рассмотреть, как частичные суммы (3.4) аппроксимируют исходный сигнал $x(t)$. Обозначим через $S_n(t)$ сумму первых n членов в (3.4). Графики первых трех частичных сумм приведены на рис. 3.6.

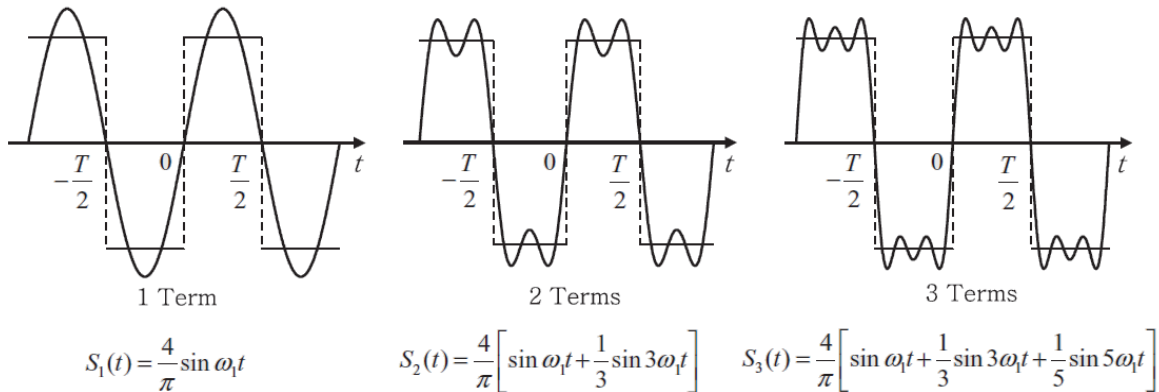


Рис. 3.6. Частичные суммы, аппроксимирующие сигнал меандр

Эффект Гиббса

Поведение частичных сумм ряда Фурье в точке разрыва функции называют *эффектом Гиббса*. Создается впечатление, что колебания в точках разрыва исчезнут, если просуммировать больше членов ряда, однако этого не происходит (рис. 3.7). Заметьте, что от количества слагаемых в ряде Фурье амплитуда выброса не изменяется.

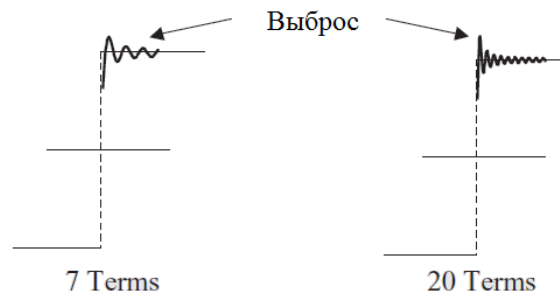


Рис. 3.7. Эффект Гиббса

Комплексная запись ряда Фурье

Наиболее часто ряд Фурье записывается в терминах комплексных экспонент $e^{j\omega_k t}$, где $\omega_k = 2\pi k/T_p$. Переход к комплексной форме записи ряда Фурье можно выполнить, если учесть, что

$$\cos \theta = \frac{1}{2}(e^{j\theta} + e^{-j\theta}) \quad \text{и} \quad \sin \theta = \frac{1}{2j}(e^{j\theta} - e^{-j\theta}). \quad (3.5)$$

Используя эти соотношения, выражение (3.1) можно преобразовать к виду

$$x(t) = \sum_{k=-\infty}^{\infty} c_k e^{-j\omega_k t}, \quad (3.6)$$

$$c_k = \frac{1}{T_P} \int_0^{T_P} x(t) e^{-j\omega_k t} dt,$$

где комплексные коэффициенты c_k связаны с a_k и b_k следующим образом:

$$c_k = (a_k - jb_k)/2.$$

Коэффициент c_k несет в себе информацию об амплитуде синусной и косинусной волны k -й компоненты разложения Фурье (или k -й гармоники). Форма (3.6) связана с разложением

$$x(t) = \frac{a_0}{2} + \sum_{k=1}^{\infty} M_k \cos(\omega_k t + \varphi_k) \quad (3.7)$$

следующим образом*:

$$|c_k| = \sqrt{a_k^2 + b_k^2} = M_k, \quad \arg(c_k) = \operatorname{arctg}\left(\frac{-b_k}{a_k}\right) = \varphi_k.$$

Заметим, что в (3.6) «кирпичиками», из которых складывается сигнал, являются *комплексные экспоненты* $e^{-j\omega_k t}$. Коэффициенты c_k также имеют комплексные значения и несут в себе информацию не только об амплитуде, но и о фазе k -й гармоники. Кроме того, в (3.6) возникают компоненты «отрицательной частотой». Понятие отрицательной частоты имеет чисто математическую природу и берет свои истоки в (3.5).

Спектр сигнала

Для пояснения понятия спектра сигнала обратимся к комплексной форме ряда Фурье (3.6). График зависимости $|c_k|$ от частоты ω_k называется *амплитудным спектром* сигнала $x(t)$. График зависимости фаз частотных компонент $\arg c_k$ от частоты ω_k называется *фазовым спектром* сигнала $x(t)$. Примеры амплитудного и фазового спектров показаны на рис. 3.8.

* В Matlab для вычисления модуля комплексного числа используется функция `abs`, а для вычисления аргумента комплексного числа – функция `angle`.

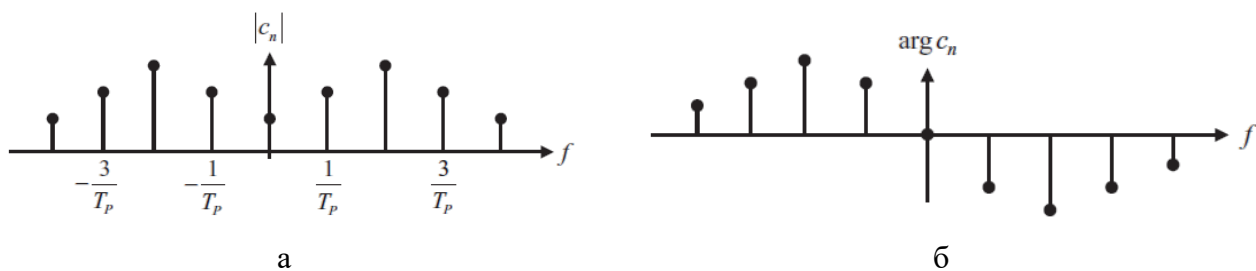


Рис.3.8. Амплитудный (а) и фазовый (б) спектры

Альтернативным способом изобразить спектр сигнала является построение графиков для M_k и φ_k из выражения (3.7).

Дискретное преобразование Фурье

ДПФ является аналогом ряда Фурье для сигнала, определенного в дискретные моменты времени. В ДПФ «предполагается», что вне заданного интервала сигнал имеет периодическое продолжение, как и в случае ряда Фурье (сравните рис. 3.9 и 3.1).

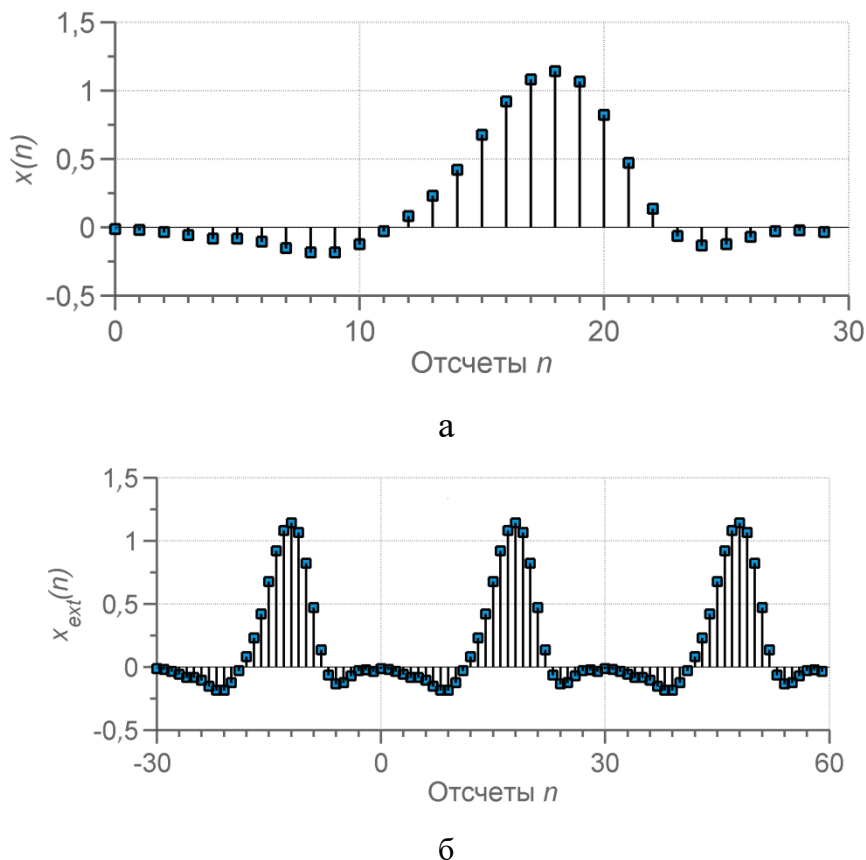


Рис. 3.9. Дискретный сигнал (а) и его ДПФ-расширение (б)

По этой причине полный период функции \sin с точки зрения ДПФ представляется, как показано на рис. 3.10 (без нулевого отсчета в конце). Так проис-

ходит потому, что именно в этом случае ДПФ-расширение сигнала тоже является синусом (рис. 3.11).

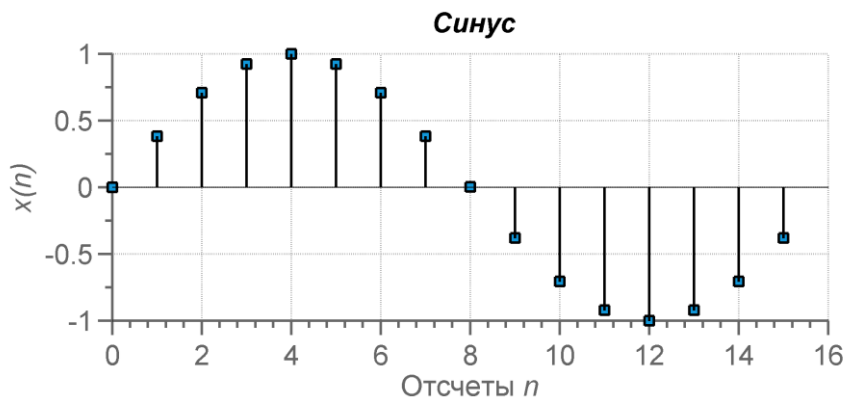


Рис. 3.10. Один период дискретного синуса

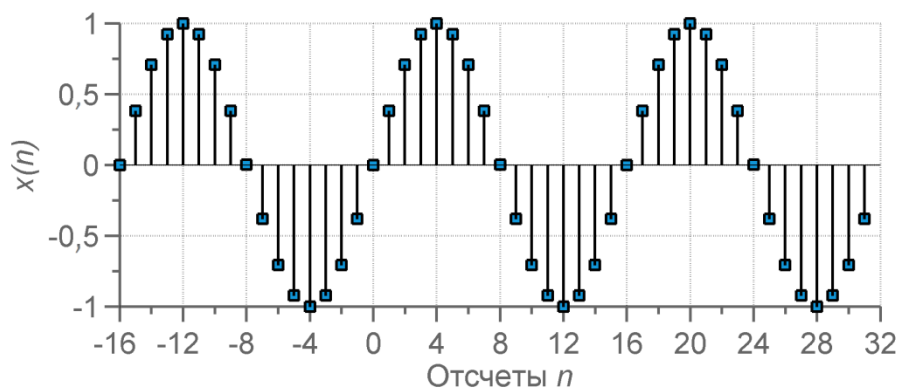


Рис.3.11. Дискретный синус: ДПФ-расширение

Что делает ДПФ?

Как показано на рис.3.12, ДПФ переводит N точек входного сигнала в два выходных сигнала из $N/2 + 1$ точек. Входной сигнал – это сигнал, который подвергается разложению, а два выходных сигнала содержат *амплитуды* синусов и косинусов. Как правило, входной сигнал определен во *временной области*. Это вызвано тем, что большинство сигналов, встречающихся в ЦОС, состоят из отсчетов, полученных через равные интервалы *времени*. Термин *частотная область* используется для описания амплитуд синусов и косинусов (включая специальное масштабирование, которое будет объяснено позже).

Частотная область содержит точно такую же информацию, как и временная, но в другой форме. Если известно представление сигнала в одной из областей, всегда можно представить в другой. Если известен сигнал во временной области, то процесс вычисления сигнала в частотной области называется *разложением, анализом, прямым ДПФ* или просто *ДПФ*. Если известен сигнал в частотной области, то вычисление сигнала во временной области называется *синтезом, или обратным ДПФ*. Как синтез, так и анализ могут быть представлены в виде формул и компьютерных алгоритмов.

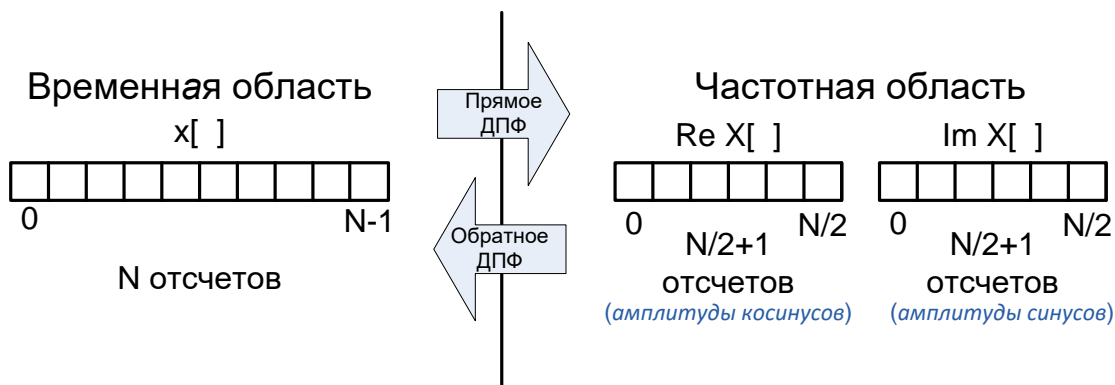


Рис.3.12.ДПФ:представление сигнала во временной и частотной областях

Число отсчетов во временной области обычно обозначается *переменной* N . Значение N может быть любой положительной величиной, но чаще всего его выбирают кратным степени числа 2, т. е. 128, 256, 512 и т. д. Это происходит по следующим двум причинам: 1) при хранении цифровых данных адресное пространство памяти кратно степени двойки; 2) большинство эффективных алгоритмов для вычисления ДПФ используют N , кратное степени двойки.

На рис. 3.13 показан пример ДПФ для $N = 32$. Сигнал во временной области представляется в виде массива $x[0], \dots, x[31]$, сигнал в частотной области – в виде двух массивов $Re X[0] \dots Re X[16]$ и $Im X[0] \dots Im X[16]$. Заметим, что 32 точки временной области соответствуют 17 точкам в каждом массиве частотной области с номерами частот от 0 до 16, т. е. N точек временной области соответствуют $N/2 + 1$ точек частотной области (не $N/2$ точек). Опускание этой дополнительной точки является типичной ошибкой при программировании ДПФ.

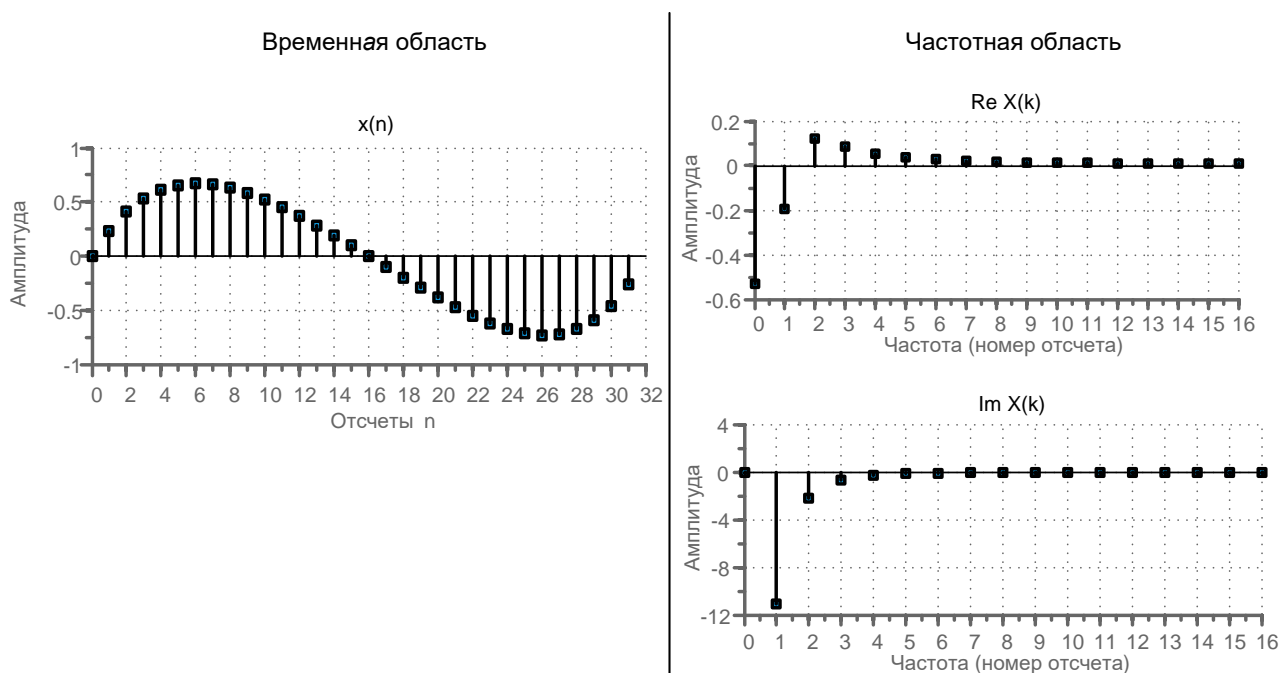


Рис. 3.13.Пример ДПФ для 32 отсчетов: представление сигнала во временной и частотной областях

Масштаб частотной оси

Горизонтальная ось в частотной области может быть обозначена четырьмя различными способами, все они общеприняты в ЦОС. При *первом способе* горизонтальная ось размечена номерами отсчетов от 0 до $N/2$. В этом случае индексы в частотной области есть целые числа, например, $Re X[k]$ и $Im X[k]$, где k меняется от 0 до $N/2$ с шагом единица. Этот способ часто применяют программисты, потому что он соответствует коду, который используется для доступа к массиву (см. рис. 3.13).

Во втором способе горизонтальная ось размечена в долях частоты дискретизации. Это означает, что величина вдоль горизонтальной оси всегда изменяется от 0 до 0,5, т. к. дискретные данные могут содержать частоты только от постоянной составляющей до половины частоты дискретизации. Для этого обозначения используется переменная f . Реальная и мнимая части записываются как $Re X[f]$ и $Im X[f]$, где f принимает значения $N/2 + 1$ различных значений, равномерно расположенных в интервале от 0 до 0,5. Связь между первым и вторым способом масштабирования частотной оси выражается следующим образом:

$$f = \frac{k}{N}.$$

Третий способ по сути совпадает со вторым, за исключением того, что горизонтальная ось умножена на 2π . Индекс, используемый для разметки, — ω . При этом обозначении реальная и мнимая части записываются как $Re X[\omega]$ и $Im X[\omega]$, где ω принимает значения $N/2 + 1$ величин, равномерно расположенных между 0 и π . Параметр ω называется *нормированной круговой частотой*. Этот способ обозначений удобен тем, что позволяет сократить математические выкладки. Например, рассмотрим запись косинуса, используя три различных способа задания частотной оси:

$$k: c(n) = \cos(2\pi kn/N),$$

$$f: c(n) = \cos(2\pi fn),$$

$$\omega: c(n) = \cos(\omega n).$$

Четвертый способ заключается в разметке горизонтальной оси в величинах аналоговых частот, используемых в *практических* приложениях. Например, если в системе применяется частота дискретизации 10 кГц (т.е. 10 000 отсчетов в секунду), то на графике частота будет меняться от 0 до 5 кГц. Этот способ хорош для представления частотных данных в обозначениях, используемых на практике. Недостаток заключается в том, что он привязан к конкретному значению частоты дискретизации входного сигнала, и поэтому неприменим, когда ведется речь об абстрактном сигнале.

Базисные функции ДПФ

Синусы и косинусы, используемые в ДПФ, обычно называются *базисными функциями*. Эти базисные функции имеют *единичную* амплитуду. Выходом ДПФ является набор чисел, которые представляют амплитуды базисных функций. Если определить каждую амплитуду (в частотной области) соответствующего синуса или косинуса, то в результате будут получены значения, которые отличаются от реальных *намасштабирующий коэффициент*.

Базисные функции ДПФ определяются следующими выражениями:

$$c_k[n] = \cos\left(\frac{2\pi kn}{N}\right), \quad s_k[n] = \sin\left(\frac{2\pi kn}{N}\right),$$

где $n = 0, 1 \dots N - 1$; $k = 0, 1 \dots N/2$; $c_k[n]$ – косинусная волна с амплитудой, содержащейся в $Re X[k]$; $s_k[n]$ – синусная волна с амплитудой, содержащейся в $Im X[k]$. Например, рис. 3.14 показывает некоторые из 17 синусов и 17 косинусов, используемых в 32-точечном ДПФ.

Рассмотрим некоторые из этих базисных функций. Косинусная волна $c_0[n]$ имеет нулевую частоту, т.е. это постоянный сигнал с единичной амплитудой. Это означает, что $Re X[0]$ содержит величину, усредненную по всем точкам сигнала во временной области. В электронике можно было бы сказать, что $Re X[0]$ содержит *постоянную составляющую*. Синусная волна с нулевой частотой $s_0[n]$ состоит из одних нулей. Она не воздействует на формирование сигнала.

Синусоиды $c_2[n]$ и $s_2[n]$ совершают 2 полных колебания на отрезке из N точек. Им соответствуют $Re X[2]$ и $Im X[2]$. Подобным образом $c_{10}[n]$ и $s_{10}[n]$ совершают 10 полных колебаний на N точках. Этим синусоидам соответствуют амплитуды $Re X[10]$ и $Im X[10]$. Проблема заключается в том, что отсчеты $c_{10}[n]$ и $s_{10}[n]$ не отражают *наглядно* характер синусных и косинусных волн. Если бы не было непрерывной кривой на графике, было бы трудно определить вид сигнала.

Самые высокие частоты в базисных функциях – это $c_{N/2}$ и $s_{N/2}$ (для рассматриваемого примера $c_{16}[n]$ и $s_{16}[n]$). Дискретная косинусная волна чередуется между 1 и -1 , что может быть интерпретировано как отсчеты по пикам непрерывной синусоиды. В противоположность этому дискретная синусная волна состоит из всех нулей в результате попадания отсчетов на *нулевые пересечения*. Это делает величину $Im X[N/2]$ такой же, как и $Im X[0]$: всегда равной нулю и не влияющей на синтез сигнала во временной области, что дает ответ на вопрос, почему из N отсчетов на входе ДПФ образуется $N + 2$ отсчетов на выходе. Ответ в том, что два выходных отсчета не содержат информацию, позволяя другим N быть полностью независимыми.

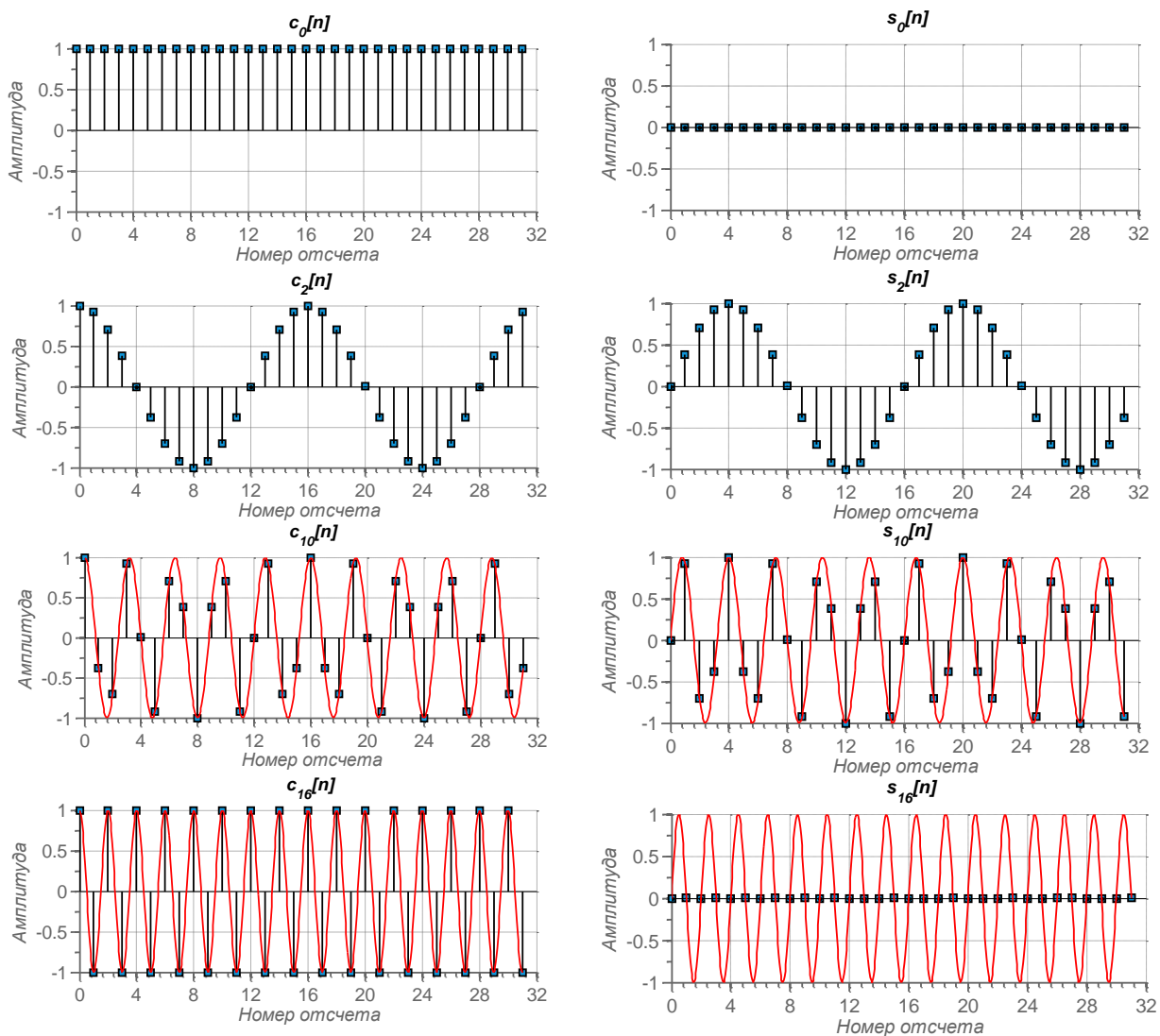


Рис.3.14.Базовые функции 32-точечного ДПФ

Вычисление ДПФ

Выражения для вычисления прямого ДПФ:

$$ReX[k] = \sum_{n=0}^{N-1} x[n] \cos\left(\frac{2\pi kn}{N}\right), ImX(k) = - \sum_{n=0}^{N-1} x[n] \sin\left(\frac{2\pi kn}{N}\right),$$

где $x[n]$ – анализируемый сигнал во временной области; $Re X(k)$ и $Im X(k)$ – вычисляемые сигналы в частотной области; индекс k изменяется от 0 до $N/2$.

Выражение для ДПФ можно записать и в комплексной форме, используя формулу Эйлера $e^{j\varphi} = \cos \varphi + j \sin \varphi$:

$$X(k) = \sum_{n=0}^{N-1} x[n] e^{-j\frac{2\pi kn}{N}}, \quad k = 0, 1 \dots N/2.$$

Уравнение синтеза или обратного ДПФ можно записать как

$$x[n] = \sum_{k=0}^{N/2} \operatorname{Re} \hat{X}(k) \cos\left(\frac{2\pi nk}{N}\right) + \sum_{k=0}^{N/2} \operatorname{Im} \hat{X}(k) \sin\left(\frac{2\pi nk}{N}\right). \quad (3.8)$$

Иными словами, любой сигнал из N точек $x[n]$ может быть создан суммированием $N/2 + 1$ косинусных волн и $N/2 + 1$ синусных волн. Амплитуды косинусных и синусных волн содержатся в массивах $\operatorname{Re} \hat{X}(k)$ и $\operatorname{Im} \hat{X}(k)$ соответственно.

В выражении (3.8) массивы обозначаются $\operatorname{Re} \hat{X}(k)$ и $\operatorname{Im} \hat{X}(k)$ вместо $\operatorname{Re} X(k)$ и $\operatorname{Im} X(k)$. Это вызвано тем, что амплитуды, необходимые для синтеза, слегка отличаются от значений сигнала в частотной области. Это влияние коэффициента масштабирования. Для получения правильного результата требуется нормализация. Незнание этого факта часто приводит к ошибкам при программировании формул ДПФ. Формально масштабирование выглядит следующим образом:

$$\operatorname{Re} \hat{X}(k) = \frac{\operatorname{Re} X(k)}{N/2}, \quad \operatorname{Im} \hat{X}(k) = -\frac{\operatorname{Im} X(k)}{N/2},$$

за исключением двух специальных случаев:

$$\operatorname{Re} \hat{X}(0) = \frac{\operatorname{Re} X(0)}{N}, \quad \operatorname{Re} \hat{X}(N/2) = \frac{\operatorname{Re} X(N/2)}{N}.$$

По аналогии с выражением (3.7) формулу для обратного преобразования Фурье можно записать в следующем виде:

$$x[n] = \sum_{k=0}^{N/2} M_k \cos\left(\frac{2\pi nk}{N} + \varphi_k\right), \quad (3.9)$$

где

$$M_k = \sqrt{\left(\operatorname{Re} \hat{X}(k)\right)^2 + \left(\operatorname{Im} \hat{X}(k)\right)^2}, \quad \varphi_k = \operatorname{arctg} \frac{-\operatorname{Im} \hat{X}(k)}{\operatorname{Re} \hat{X}(k)}.$$

Выражение (3.9) показывает, что с помощью ДПФ можно представить сигнал в виде суммы гармонических колебаний. При этом каждая гармоника имеет свою амплитуду M_k и фазу φ_k . Для правильного вычисления фазы в среде Matlab предусмотрена функция `atan2(imX, reX)`, которая возвращает значение угла в диапазоне $[-\pi, \pi]$.

Существует и комплексная форма записи обратного ДПФ:

$$x[n] = \frac{1}{N} \sum_{k=0}^{N-1} X(k) e^{\frac{j2\pi kn}{N}}, \quad n = 0, 1, \dots, N-1.$$

Вычисление ДПФ в пакете Matlab

В пакете Matlab для вычисления дискретного преобразования Фурье используется функция `FFT(X)`. В случае матричного аргумента ДПФ рассчитывается для каждого столбца матрицы. Может оказаться полезным явное указание размера преобразования: `FFT(X, N)`. Если реальная длина вектора X меньше заданной размерности N , он дополняется нулями. При этом можно получить больше частотных отчетов и, следовательно, улучшить условия различения синусоидальных компонент сигнала. Если длина входного вектора больше N , лишние отсчеты отсекаются.

Обратное дискретное преобразование Фурье вычисляется с помощью функции `IFFT(X)`, которая используется аналогично предыдущей.

1.4 Порядок выполнения работы

3.2.1. Разработайте функцию `DET`, вычисляющую ДПФ от входного вектора, не используя функцию `MatlabFFT`, и рисующую графики действительной и мнимой частей результата преобразования. Сравните результаты работы своей функции с функцией `MatlabFFT`.

3.2.2. Предположим, что задан входной сигнал $x[n]$ значения ДПФ сигнала $X(k)$. Разработайте в среде Matlab функцию `[cA, sA]=SinCosAmps(X)`, которая из комплексных значений $X(k)$ вычисляет амплитуды косинусов и синусов, на которые раскладывается сигнал $x[n]$. Если входной сигнал имеет размерность N , то выходные массивы `cA` и `sA` должны иметь размерность $N/2 + 1$.

3.2.3. Напишите Matlab-функцию, которая выполняет синтез сигнала $x[n]$ из амплитуд косинусов и синусов, полученных функцией `SinCosAmps`. Проверьте работу функции.

3.2.4. Напишите Matlab-функцию которая преобразует комплексные значения ДПФ сигнала $X(k)$ в гармонические параметры M_k и φ_k (см. формулу (3.9)). Если $X(k)$ имеет размерность N , то размерность массивов M_k и φ_k должна быть $N/2 + 1$. Используя разработанную функцию произвольного сигнала $x[n]$, постройте амплитудный и фазовый спектры сигнала.

3.2.5. Напишите Matlab-функцию которая выполняет синтез сигнала из гармонических параметров M_k и φ_k (см. формулу(3.9)). Проверьте работу функции.

3.2.6. Используя функцию из задания 3.2.5, выполните Фурье-анализ ЭКГ-сигнала**. Постройте график сигнала во временной и частотной областях. По оси абсцисс в частотной области отложите аналоговые частоты.

3.2.7. Используйте функции из задания 3.2.5 и 3.2.6 для изменения тембра речевого сигнала. Для это запрограммируйте в Matlab следующий алгоритм:

- загрузите wav-файл при помощи функции `[x, fs]=wavread('путь_к_файлу');`
- входной сигнал разбейте на последовательные секции по 512 отсчетов;
- для каждой секции сигнала выполните ДПФ и найдите гармонические параметры M_k и φ_k ;
- преобразуйте амплитуды гармоник M_k при помощи функции `ChangeTimbre(M_k, alpha)`, оставляя фазы гармоник неизменными. Параметр α влияет на степень изменения тембра: $\alpha = 1$ – не изменит тембра, $\alpha > 1$ – тембр становится более высоким, $\alpha < 1$ – более низким. При $\alpha > 2$ и $\alpha < 0,5$ в речевой сигнал может быть значительно искажен;
- выполните синтез секции речевого сигнала из параметров M_k и φ_k при помощи функции из задания 3.2.6;
- запишите синтезированный речевой сигнал y в wav-файл при помощи функции `wavwrite(y, fs, 'имя_файла')`.

3.2.8. Выполните моделирование работы полупроводникового диода при прохождении через него синусоидального сигнала (рис. 3.15).

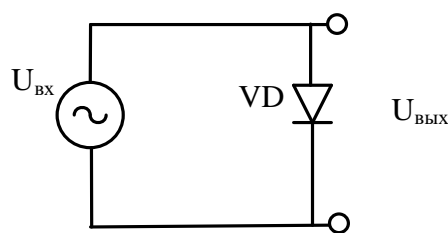


Рис.3.15. Схема для задания 3.2.8

Выходное напряжение определяется следующим образом

$$U_{\text{вых}} = \begin{cases} U_{\text{вх}} \times \text{acrtg}\left(\frac{1}{r_{\text{пр}}}\right), & U_{\text{вх}} > U_{\text{отс}} \\ U_{\text{отс}} \left(\frac{U_{\text{вх}}}{U_{\text{отс}}}\right)^2, & 0 < U_{\text{вх}} < U_{\text{отс}} \\ 0, & U_{\text{вх}} < 0 \end{cases}$$

** Файл `ecg_data` хранится в папке "\\1-311-kafevs\Docs\ТиПЦОС\Материалы для ЛР№3\". Для загрузки необходимо скопировать файл в текущий каталог Matlab и выполнить команду `load('ecg_data')`; частота дискретизации сигнала равна 360 Гц.

Постройте график выходного напряжения, если на входе действует сигнал $x(t) = 0,5 \sin(2\pi 10t)$, $0 < t < 0,5$ с. Для получения дискретного сигнала $x[n]$ выполните дискретизацию сигнала с частотой 100 Гц. Постройте спектры входного ($x[n]$) и выходного сигнала $y[n]$. Какие выводы можно сделать по внешнему виду полученных спектров?

ЛАБОРАТОРНАЯ РАБОТА №4. Спектральный анализ

ЦЕЛЬ РАБОТЫ –изучение методов определения основных свойств случайных сигналов, таких, как мощность, спектральная плотность мощности (СПМ) и автокорреляционная функция (АКФ),получение практических навыков анализа случайных сигналов в среде Matlab.

4.1. Теоретические сведения

Случайные сигналы

Наряду с детерминированными (аналитическими*) сигналами существуют стохастические, или *случайные сигналы*. Отличительная черта случайного сигнала состоит в том, что его мгновенные значения заранее непредсказуемы. Характеристики таких сигналов довольно точно можно описать в вероятностном (статистическом) смысле. Важнейшими характеристиками случайного сигнала является его *автокорреляционная функция* и *спектральная плотность мощности*, которые тесно связаны между собой.

Анализ случайных сигналов выполняется в предположении, что рассматриваемый сигнал является *стационарным*. Свойство *стационарности* означает, что статистические характеристики сигнала, такие как среднее значение μ и дисперсия (разброс) σ^2 , не зависят от времени. Так, периодический сигнал является стационарным, а кратковременный переходной** – нестационарным.

Пример транзientного сигнала показан на рис. 4.1,а. Чтобы убедиться в его нестационарности, вычислим *кратковременное* среднее значение. Для этого выберем значения сигнала для моментов времени с 0 по 99 и посчитаем среднее, которое отложим на графике в момент времени 0, затем окно наблюдения сместим на один отсчет влево и посчитаем среднее значение сигнала в промежутке от 1 до 100, а результат отложим на графике в момент времени 1 и т.д. Ниже приведен Matlab-код, выполняющий описанное действие:

```
x = wavread('letter_T_female_low_8k');
N = length(x);
Npt = 100;
y = zeros(1, N);
for n = 1:N-Npt
    y(n) = sum(x(n:n+Npt-1))/Npt;
end
plot(1:N, y)
```

* Аналитические сигналы имеют однозначное математическое описание, например \sin или x^2 .

** Часто употребляют название «транзientный» сигнал.

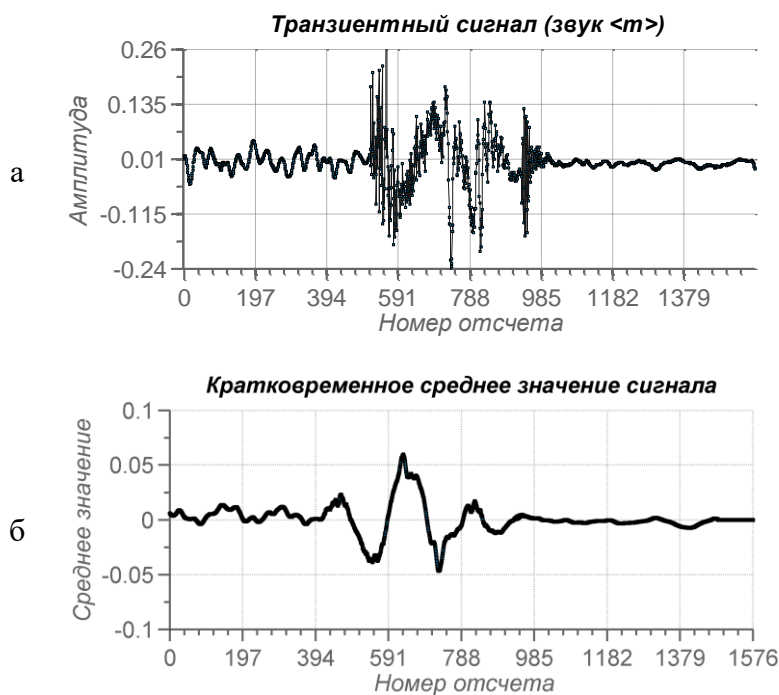


Рис. 4.1. Пример нестационарного сигнала (а) и график изменения его среднего значения во времени (б)

На рис. 4.2 показан пример стационарного сигнала и график изменения его среднего значения. Можно сделать вывод, что среднее значение сигнала практически не меняется со временем, это позволяет утверждать, что представленный сигнал отвечает условию стационарности.

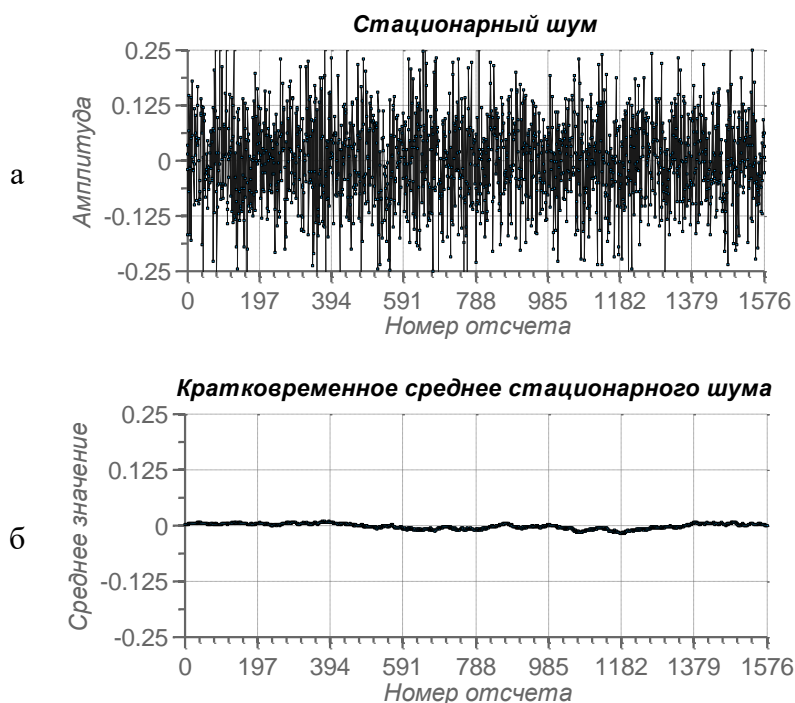


Рис. 4.2. Пример стационарного сигнала (а) и график изменения его среднего значения во времени (б)

Мощность и спектральная плотность мощности

Мгновенная мощность сигнала $x(t)$ равна квадрату его амплитуды $|x(t)|^2$. Это утверждение верно, даже когда $x(t)$ принимает комплексные значения. Часто более полезным является понятие *средней мощности*. Пусть дискретный сигнал $x(n)$ состоит из отсчетов непрерывного сигнала $x(t)$, тогда средняя мощность такого сигнала вычисляется как

$$P_a = \frac{1}{N} \sum_{n=0}^{N-1} x^2(n). \quad (4.1)$$

Также мы можем найти среднюю мощность сигнала, если известно его ДПФ:

$$P_a = \frac{1}{N^2} \sum_{k=0}^{N-1} |X(k)|^2. \quad (4.2)$$

Из выражений (4.1) и (4.2) можно сделать вывод, что

$$\sum_{n=0}^{N-1} x^2(n) = \frac{1}{N} \sum_{k=0}^{N-1} |X(k)|^2.$$

Этот результат известен как теорема *Парсеваля*, которая показывает, что энергия сигнала может быть найдена как из временных отсчетов сигнала, так и из компонент преобразования Фурье. Теорема Парсеваля позволяет ввести понятие *периодограммы*:

Периодограмма:
$$P_{xx}(k) = \frac{1}{N} |X(k)|^2, \quad k = 0, 1, \dots, N - 1. \quad (4.3)$$

Таким образом, периодограмма – это функция, принимающая действительные значения и обладающая свойством симметрии $P_{xx}(k) = P_{xx}(N - k)$ вследствие симметричности $X(k)^*$. Зная периодограмму можно записать выражение для средней мощности:

Средняя мощность:
$$P_a = \frac{1}{N} \sum_{n=0}^{N-1} x^2(n) = \frac{1}{N} \sum_{k=0}^{N-1} P_{xx}(k). \quad (4.4)$$

* ДПФ $X(k)$ действительного сигнала $x(n)$ обладает свойством симметрии $X(k) = X^*(N - k)$, где «*» знак комплексного сопряжения.

Выражение(4.4) дает нам понять, что также как $x^2(n)$ показывает мощность сигнала в отдельной момент времени n , так $P_{xx}(k)$ показывает мощность отдельной k -й частотной компоненты сигнала. Говорят, что периодограмма является *оценкой спектральной плотности мощности* сигнала $x(n)$.

Зачем знать СПМ сигнала?

Знание спектральной плотности мощности сигнала позволяет ответить на вопрос, какие частотные составляющие сигнала имеют наибольшую энергию. Часто записанный сигнал представляет собой смесь нескольких источников. В этом случае если каждый источник имеет свою «собственную» частоту, то по СПМ можно судить о количестве имеющихся источников: оно будет равно числу максимумов (пиков) функции СПМ. Ниже приведен Matlab-код, моделирующий описанную ситуацию. Имеется два источника сигнала: первый – частотой 500 Гц, второй – 1125 Гц. Записанный сигнал представляет собой смесь сигналов, поступающих от двух источников на фоне шума*.

```
Npt = 256;
Fs = 8000; % частота дискретизации 8 кГц
n = 0:Npt-1;
f1 = 500; % частота первого источника (в Гц)
f2 = 1125; % частота второго источника (в Гц)
x = 0.4*cos(2*pi*f1/Fs*n) + 0.2*cos(2*pi*f2/Fs*n) +
0.2*randn(1,Npt);
plot(n, x)
```

По временной форме сигнала (рис. 4.3) трудно судить о количестве источников. Однако это возможно сделать, если построить периодограмму сигнала (рис. 4.4) и по оси абсцисс отложить аналоговые частоты. Периодограмма показывает, что исходный сигнал состоял из двух синусоидальных компонент с разными амплитудами и частотами (в нашем случае каждая из компонент генерировалась своим источником). Также периодограмма позволяет оценить частоту синусоидальных компонент. Наличие в периодограмме компонент с малыми амплитудами объясняется присутствием шума во входном сигнале.



Рис. 4.3. Временная форма смеси сигналов

* Гауссовский шум с нулевым средним значением и СКО, равным единице, генерируется в Matlabкомандой `randn(1, N)`.

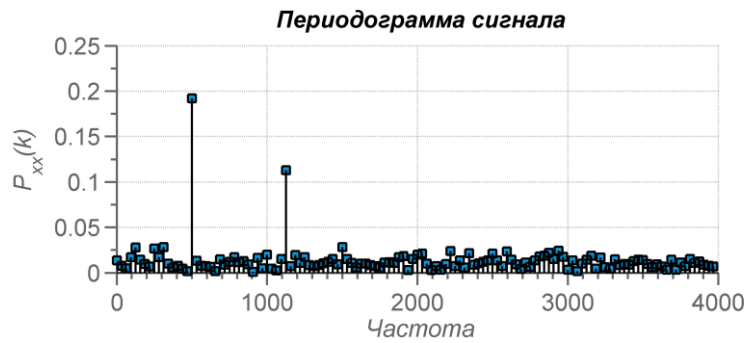


Рис. 4.4. Периодограмма смеси сигналов

Автокорреляционная функция сигнала

Объясним понятие автокорреляционной функции сигнала $x(n)$, $n = 0, 1, \dots, N - 1$. Предположим, что сигнал $x(n)$ имеет за пределами своего определения периодическое продолжение. Это предположение весьма обоснованно, особенно для стационарных сигналов. АКФ сигнала $x(n)$ определяется следующим образом:

$$\varphi_{xx}(m) = \frac{1}{N} \sum_{n=0}^{N-1} x(n)x(n+m), \quad m = 0, 1, \dots, N-1. \quad (4.5)$$

Можно заметить, что мощность сигнала сконцентрирована в первом отсчете АКФ:

$$P_a = \frac{1}{N} \sum_{n=0}^{N-1} x^2(n) = \varphi_{xx}(0).$$

Однако более важным является другой результат: *дискретное преобразование Фурье, взятое от автокорреляционной функции сигнала, равно периодограмме этого сигнала:*

$$\text{ДПФ}\{\varphi_{xx}(m)\} = \frac{1}{N} |X(k)|^2 = P_{xx}(k), \quad k = 0, 1, \dots, N-1.$$

Повторим, что последнее утверждение справедливо *только* в том случае, когда сигнал $x(n)$ имеет периодическое продолжение, т.е. когда $x(N) = x(0)$, $x(N+1) = x(1)$ и т. д.* Автокорреляционная функция φ_{xx} показывает, насколько сильна связь между отсчетами сигнала. Например $\varphi_{xx}(1)$ показывает, насколько сильно зависимы соседние отсчеты в сигнале, $\varphi_{xx}(2)$ обнаруживает связь между отсчетами $x(n)$ и $x(n \pm 2)$, т. е. между отсчетами, отстоящими друг

* В Matlab можно легко получить периодическое продолжение сигнала, используя операцию конкатенации. Например, если вектор-строка `sig` содержит отсчеты сигнала, то периодическое продолжение сигнала можно получить командой `periodic_sig = [sig sig];`

от друга на два интервала дискретизации. Таким образом, с помощью АКФ можно выявить скрытую в сигнале периодичность. Допустим, что сигнал содержит гармоническое колебание с периодом в 50 отсчетов. Можно уверенно сказать, что автокорреляционная функция $\varphi_{xx}(m)$ такого сигнала будет иметь локальный максимум в точке $m = 50$.

Отметим, что ни $\varphi_{xx}(m)$, ни $P_{xx}(k)$ не содержат никакой информации о фазе сигнала $x(n)$. По этой причине сдвиг $x(n)$ во временной области не влияет на автокорреляционную функцию и периодограмму.

В качестве характерных примеров рассмотрим АКФ и периодограмму гармонического сигнала и белого шума (рис. 4.5).

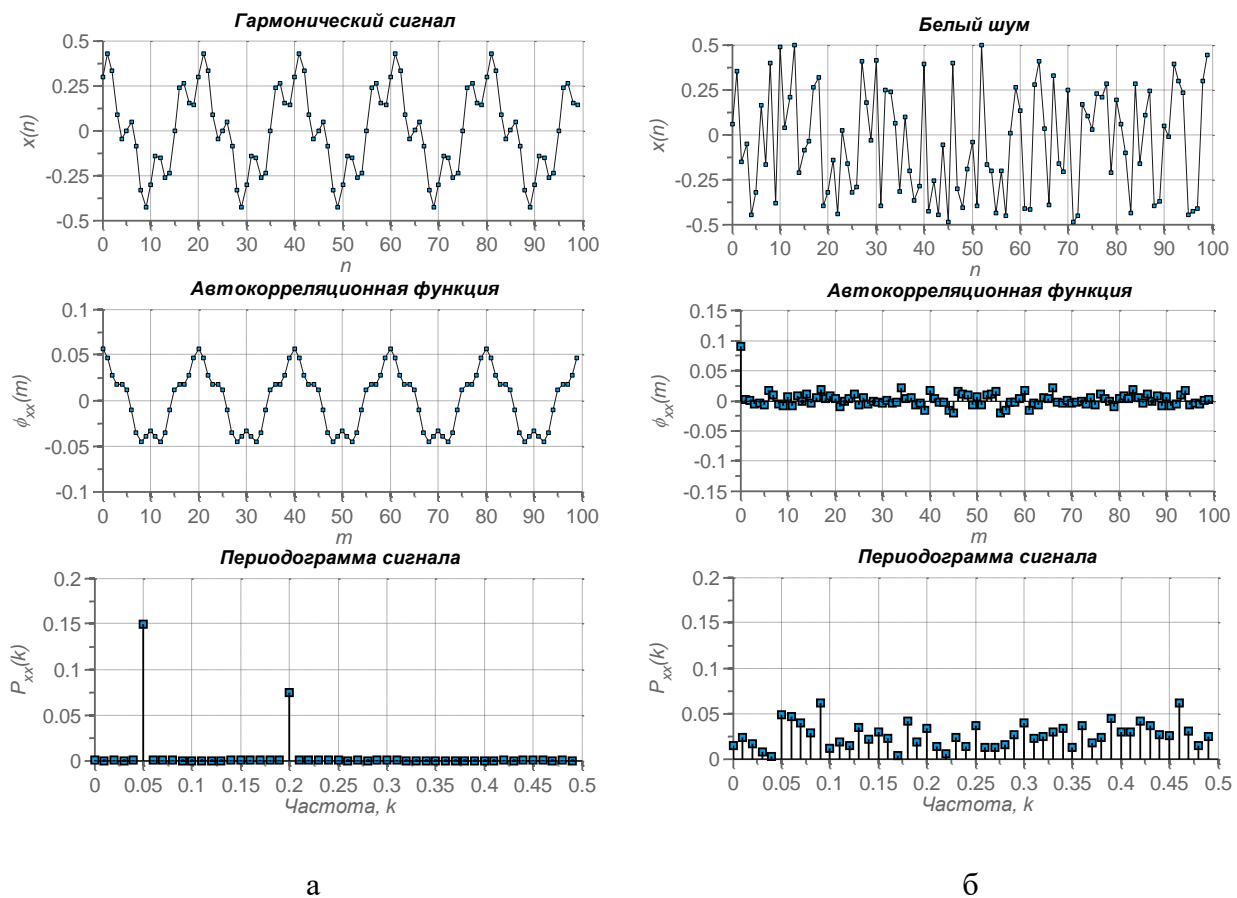


Рис. 4.5. Временная форма, АКФ и периодограмма гармонического сигнала $x(n) = 0,3 \cos(2\pi \cdot 0,05n) + 0,15 \sin(2\pi \cdot 0,2n)$, $n = 0, 1, \dots, 99$ (а) и белого шума (б)

В обоих случаях частота дискретизации сигнала равна 1 Гц. Периодический сигнал содержит две синусоидальные компоненты: одна с периодом $T_1 = 20$ отсчетов (частота $f_1 = 1/T_1 = 0,05$), другая с периодом $T_2 = 5$ отсчетов (частота $f_2 = 1/T_2 = 0,20$). Таким образом, в сигнале имеется частота основного тона f_1 и ее 4-я гармоника, поскольку $f_2 = 4f_1$. С помощью АКФ можно обнаружить период частоты основного тона, как правило, он соответствует первому максимуму $\varphi_{xx}(m)$ для $m > 0$. В нашем случае мы наблюдаем максимум при $m = 20$, что в точности соответствует периоду основного колебания. Периодо-

грамма гармонического сигнала имеет два спектральных пика, показывая тем самым, что вся энергия сигнала сосредоточена в двух гармонических компонентах. *Белый шум* (рис. 4.5, б) представляет собой случайный сигнал. Анализ его периодограммы показывает, что мощность белого шума равномерно распределена во всем частотном диапазоне. Каждый последующий отсчет белого шума не зависит от предыдущего. Именно этот факт отражает АКФ $\varphi_{xx}(m)$ белого шума, в которой имеется пик только для $m = 0$. При любом другом сдвиге m АКФ близка к нулю, что говорит о статистической независимости отсчетов белого шума.

Оценка СПМ сигнала

Представим, что дана запись длительного стационарного сигнала и необходимо определить его СПМ. Прямой метод определения СПМ основан на вычислении квадрата модуля преобразования Фурье этого сигнала (см. формулу (4.3)). Полученная таким образом периодограмма, которую иногда называют *выборочным спектром*, представляет собой *оценку* СПМ. На практике такой метод практически не применяется, поскольку полученная СПМ является *статистически несостоятельной*. Это означает, что наличие в СПМ пика на какой-либо частоте не гарантирует нам наличие гармонической компоненты с такой частотой в сигнале. Например, на рис. 4.5 периодограмма белого шума содержит максимум на частоте 0,09 Гц, однако это не значит, что в сигнале присутствует стационарный гармонический сигнал с такой частотой и амплитудой.

Для получения *устойчивой* (достоверной) оценки СПМ применяют метод усреднения периодограмм. Метод заключается в выполнении следующих действий:

- 1) разбить исходный сигнал $x(n)$ на K перекрывающихся или неперекрывающихся сегментов (рис. 4.6);
- 2) для каждого сегмента $x_i(n)$ выполнить расчет периодограммы $P_{xx}^{(i)}(k)$;
- 3) усреднить полученные периодограммы $P_{xx}(k) = \frac{1}{K} \sum_{i=1}^K P_{xx}^{(i)}(k)$.

Данный метод имеет регулируемые параметры $NSAMP$ – число отсчетов на сегмент – и $NSHIFT$ – число отсчетов, на которое необходимо сдвинуть начало следующего сегмента. Ниже приводится каркас Matlab-кода, который необходим для реализации метода усреднения периодограмм.

```
Npt = length(x);
Nsamp = 50;
Nshift = 10;
K = floor((Npt - Nsamp) / Nshift) + 1;
for i = 1:K
    x_i = x(1 + (i - 1) * Nshift : Nsamp + (i - 1) * Nshift);
    % Вычисления с сегментом x_i
    % ...
end
```

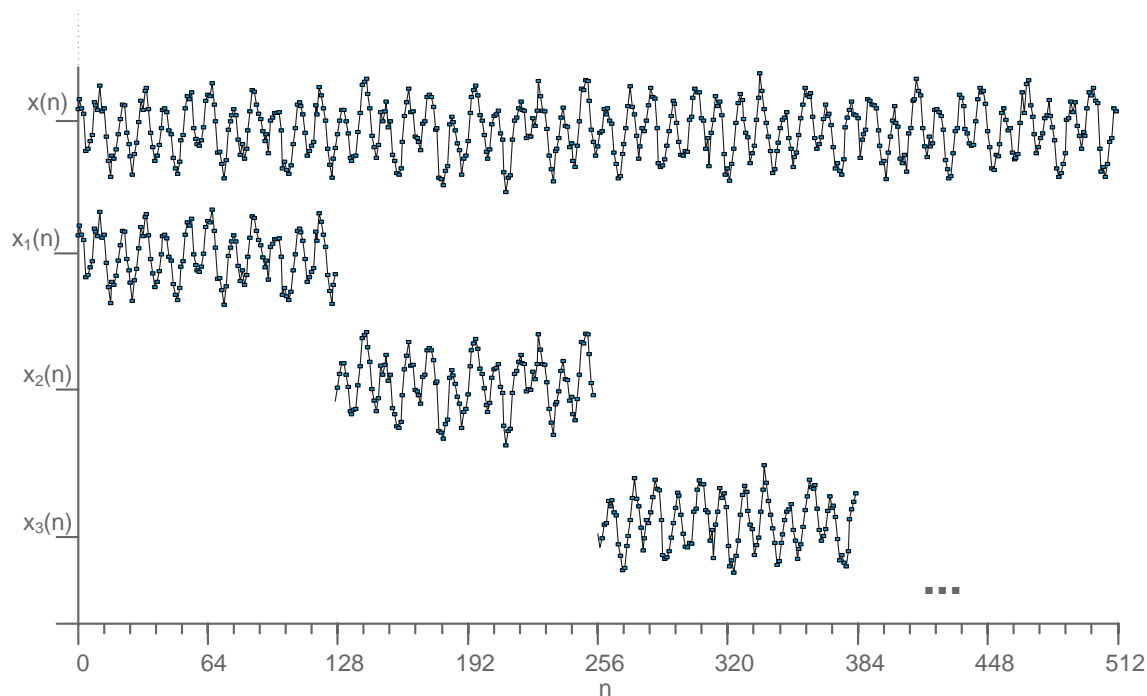


Рис. 4.6. Сегментация сигнала (без перекрытия)

Количество сегментов выбирается в зависимости от необходимой точности (надежности) спектральной оценки. При малом значении параметра $NSAMP$ получается больше сегментов, по которым производится усреднение, и, следовательно, ошибка уменьшается, но падает частотное разрешение. Увеличение параметра $NSAMP$ повышает частотное разрешение (спектр становится более детализированным), но, естественно, за счет увеличения ошибки, возникающей из-за уменьшения числа усредняемых сегментов.

Кратковременный анализ речевых сигналов

На рис.4.7 показана временная форма речевого сигнала. Из графика видно, что свойства речевого сигнала, например, характер возбуждения на вокализованных и невокализованных участках, пиковая амплитуда и др., изменяются во времени. Вокализованными называются те сегменты речи, где задействованы голосовые связки диктора. Такие звуки как 'а', 'о', 'ж', являются вокализованными, в то время как звуки 'с', 'ш', 'щ' являются невокализованными.

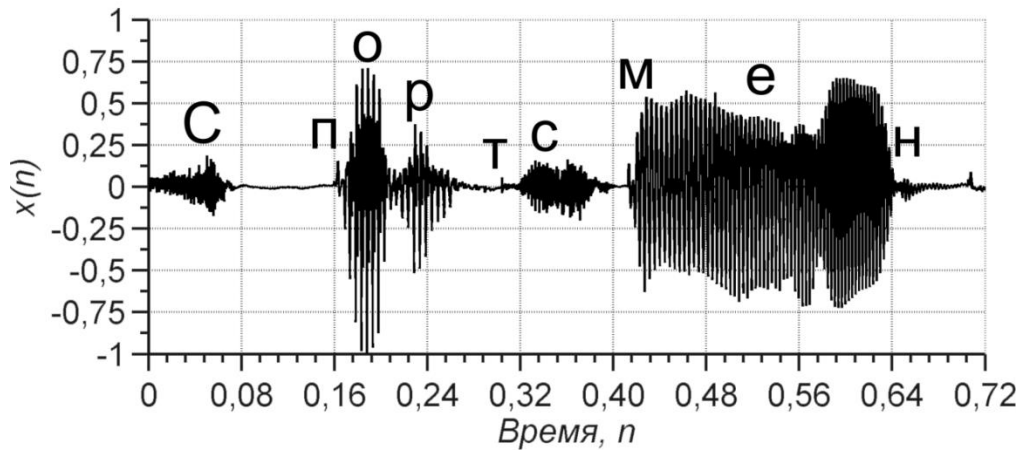


Рис. 4.7. Речевой сигнал (слово «спортсмен»)

В основе большинства методов обработки речи лежит предположение о том, что свойства речевого сигнала с течением времени медленно изменяются. Это предположение приводит к методам *кратковременного анализа*, в которых сегменты речевого сигнала выделяются и обрабатываются так, как если бы они были короткими участками отдельных звуков с отличающимися свойствами. *Сегменты*, которые иногда называют *кадрами анализа*, или *фреймами*, обычно пересекаются. Результатом обработки на каждом интервале является число или совокупность чисел. Следовательно, подобная обработка приводит к новой, зависящей от времени последовательности, которая может служить характеристикой речевого сигнала.

В общем виде большинство методов кратковременного анализа могут быть описаны выражением

$$Q(n) = \sum_{m=-\infty}^{\infty} T[x(m)]w(n-m). \quad (4.6)$$

Смысл этого выражения в следующем: сигнал $x(m)$ подвергается преобразованию $T[\cdot]$, затем результирующая последовательность умножается на последовательность значений временного окна (весовая функция) с центром на отсчете n . Результаты суммируются по всем ненулевым значениям.

Примером такого анализа может служить измерение кратковременной мощности сигнала, которая определяется как

$$P(n) = \frac{1}{N+1} \sum_{m=n-N/2}^{n+N/2} x^2(m). \quad (4.7)$$

Таким образом, кратковременная мощность в момент времени n есть просто средняя сумма квадратов отсчетов в диапазоне от $n - N/2$ до $n + N/2$. Из (4.6) видно, что в (4.7) $T[\cdot]$ представляет собой возведение в квадраты деление на $(N+1)$, а

$$w(n) = \begin{cases} 1, & -N/2 \leq n \leq N/2, \\ 0 & \text{в противном случае.} \end{cases} \quad (4.8)$$

Вычисление кратковременной мощности иллюстрирует рис. 4.8. Отметим, что окно «скользит» вдоль последовательности квадратов значений сигнала, ограничивая длительность интервала, используемого в вычислениях.

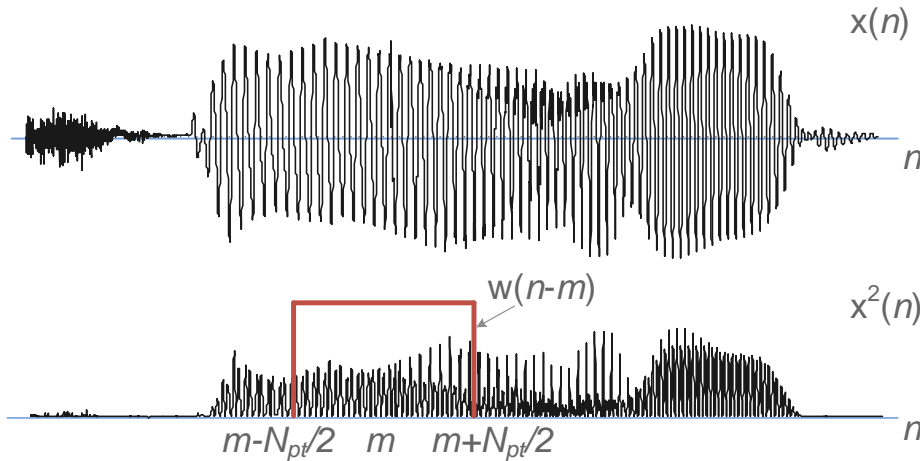


Рис. 4.8. Иллюстрация вычисления функции кратковременной мощности

Основное значение функции $P(n)$ в применении к обработке речи состоит в том, что эта величина позволяет отличить вокализованные речевые сегменты от невокализованных.

Часто в методах кратковременного анализа вместо прямоугольного окна (см. формулу (4.8) и рис. 4.8) используют окна, имеющие колоколообразный вид. Эти окна имеют значения, близкие к единице в центре сегмента и близкие к нулю по краям. Применением этих окон стараются ослабить краевые эффекты, которые возникают, когда на границах сегмента сигнал имеет большую амплитуду.

4.2. Порядок выполнения работы

4.2.1. Разработайте в Matlab функцию $[P, P_{dB}] = \text{calc_power}(x)$ для вычисления мощности сигнала по формуле (4.1). Функция также должна выдавать значение мощности в децибелах $P_{dB} = 10 \log_{10} P_a$. Посчитайте мощность следующих сигналов при $n = 0, 1 \dots, 127$:

$$\begin{aligned} x_1(n) &= 0,5 \sin\left(2\pi \frac{3}{128} n\right), \\ x_2(n) &= 0,5 \sin\left(2\pi \frac{30}{128} n\right), \\ x_3(n) &= 0,3 \cos\left(2\pi \frac{6}{128} n\right) + 0,4 \cos\left(2\pi \frac{10}{128} n\right). \end{aligned}$$

4.2.2. Разработайте в Matlab функцию для вычисления периодограммы сигнала (см. формулу (4.3)). Постройте периодограммы сигналов из зада-

ния 4.2.1. Вычислите мощность сигналов из полученных периодограмм по формуле(4.2).

4.2.3. Предположим, что зарегистрированный сигнал представляет собой сумму полезного сигнала и белого шума($n = 0, 1, \dots, 511$):

$$x(n) = 0,4 \cos\left(2\pi\frac{9}{128}n\right) + A \cdot r(n).$$

Определите максимальное отношение сигнал/шум (ОСШ), при котором по периодограмме $x(n)$ можно обнаружить в нем наличие полезного сигнала. Отношение сигнал/шум (англ. *signal-to-noise ratio*, *SNR*) – безразмерная величина, равная отношению мощности полезного сигнала к мощности шума:

$$\text{SNR(dB)} = 10 \log_{10} \frac{P_{\text{signal}}}{P_{\text{noise}}}.$$

Для определения требуемого ОСШ необходимо выполнить несколько экспериментов, плавно изменяя параметр A и наблюдая изменение периодограммы сигнала. Будем считать, что сигнал можно обнаружить, если уровень полезного сигнала на 10 дБ больше уровня компонент спектра, образуемых шумовым сигналом. Белый шум $r(n)$ можно получить в Matlab при помощи функции

```
rand(1, 512)
```

4.2.4. Оцените СПМ сигнала на выходе фильтра при помощи метода усреднения периодограмм. Параметры $NSAMP$ и $NSHIFT$ подберите самостоятельно. На вход фильтра поступает 4096 отсчетов белого шума. Коэффициенты фильтра:

```
b = [0.00482434 0 -0.0192973 0 0.02894606 0 -0.0192973 0  
0.00482434];  
a = [1 -2.06958023 3.99771255 -4.3894077 4.45285533 -2.9060422 1.75168470  
- 0.5862147 0.18737949];
```

4.2.5. Разработайте в Matlab функцию вычисления АКФ сигнала. Примените функцию для определения периода основного тона речевого сигнала (wav-файл для анализа хранится в папке "\\1-311-kafevs\Docs\ТиПЦОС\Материалы для ЛРН№4\007.wav"). Wav-файл можно загрузить командой `[x, fs] = wavread('путь_к_файлу')`. Входной сигнал разбивается на перекрывающиеся секции по 320 отсчетов. Перекрытие секций составляет 160 отсчетов. Для каждой секции сигнала вычисляется АКФ и находится номер отсчета, содержащий первый максимум функции. Номер этого отсчета и является *оценкой* периода основного тона. Полученные значения отобразите на графике:

по оси абсцисс – номер сегмента, по оси ординат – значение периода основного тона.

4.2.6. Разработайте в Matlab функцию для отделения вокализованных участков речи от невокализованных:

```
[x v, x u] = vu_separate(x, N, threshold)
```

Сепарация (разделение) происходит на основе анализа кратковременной мощности сигнала (см.(4.7)). На вход функции поступает речевой сигнал x , длина окна анализа на котором считается мощность N и значение порога $threshold$. Если мощность сигнала в момент времени n больше порога, то считается, что отсчет $x(n)$ является вокализованным, в противном случае – невокализованным. Сепарация выполняется следующим образом:

а) для входного сигнала $x(n)$ рассчитывается кратковременная мощность $P(n)$ по выражению (4.7);

б) рассчитывается признак вокализованности:

$$v(n) = \begin{cases} 1, & P(n) > \text{thershold}, \\ 0, & P(n) \leq \text{thershold}; \end{cases}$$

в) вычисляется сепарация сигнала:

$$\begin{aligned} x_v(n) &= x(n) \cdot v(n), \\ x_u(n) &= x(n) \cdot (1 - v(n)). \end{aligned}$$

Удобно мощность сигнала посчитать в децибелах:

$$P_{dB}(n) = 10 \log_{10} P(n)$$

и порог для сепарации также подбирать в децибелах.

ЛАБОРАТОРНАЯ РАБОТА №5. Расчет цифровых фильтров с

бесконечными импульсными характеристиками

ЦЕЛЬ РАБОТЫ – расчет БИХ-фильтров с различными аппроксимациями идеальной АЧХ в пакете Matlab и изучение их свойств.

5.1. Теоретические сведения

Основные свойства БИХ-фильтров

Цифровой фильтр с бесконечной импульсной характеристикой (БИХ-фильтр) является физически реализуемым и устойчивым, если его импульсная характеристика $h(n)$ удовлетворяет следующим условиям:

$$h(n) = 0, \quad n < 0,$$
$$\sum_{n=0}^{\infty} |h(n)| < \infty.$$

Наиболее общая форма записи z -преобразования импульсной характеристики БИХ-фильтров имеет вид

$$H(z) = \sum_{n=0}^{\infty} h(n) z^{-n} = \frac{\sum_{i=0}^M b_i z^{-i}}{1 + \sum_{i=1}^N a_i z^{-i}}.$$

Здесь по крайней мере один из коэффициентов a_i отличен от нуля, причем сразу все корни знаменателя (полюса передаточной функции) не могут в точности компенсироваться корнями числителя (нулями передаточной функции). Нули $H(z)$ могут располагаться на всей z -плоскости, но полюсы в соответствии с условием устойчивости фильтра обязательно должны размещаться внутри круга единичного радиуса. Как правило, число нулей M не превышает числа полюсов N . Системы, удовлетворяющие этому условию, называются системами N -го порядка. При $M > N$ порядок системы становится неопределенным. В этом случае можно считать, что передаточная функция $H(z)$ соответствует последовательному соединению системы N -го порядка и фильтра с конечной импульсной характеристикой $(M - N)$ -го порядка.

Основными функциями, характеризующими фильтр, являются квадрат амплитудной характеристики, фазовая характеристика и характеристика групповой задержки.

При расчете БИХ-фильтра с использованием аппроксимации только амплитудной характеристики (т.е. без учета фазовой характеристики) удобнее всего оперировать квадратом амплитудной характеристики, определяемым следующим образом:

$$|H(e^{j\omega})|^2 = |H(z)H(z^{-1})|_{z=e^{j\omega}}.$$

Расположению нулей и полюсов этой функции в z -плоскости свойственна симметрия с зеркальным отображением относительно единичной окружности. Полюсы $H(z)$ располагаются внутри единичной окружности, поэтому они полностью определяются квадратом амплитудной характеристики фильтра. Нули передаточной функции $H(z)$ чаще всего выбираются таким способом, чтобы соответствующие им нули квадрата амплитудной характеристики располагались на единичной окружности или внутри ее в z -плоскости. Фильтры с такими нулями являются минимально-фазовыми фильтрами.

Так как передаточная функция БИХ-фильтра в общем случае представляет собой комплексную функцию от ω , можно рассматривать и фазовую характеристику фильтра, которая равна

$$\beta(e^{j\omega}) = \operatorname{arctg} \left\{ \frac{\operatorname{Im}[H(z)]}{\operatorname{Re}[H(z)]} \right\}_{z=e^{j\omega}}.$$

Фазовая характеристика БИХ-фильтра, как правило, существенно нелинейна, поэтому для оценки дисперсионного воздействия фильтра на типовой обрабатываемый сигнал часто используется характеристика групповой задержки, записываемая как

$$\tau_g(e^{j\omega}) = -\frac{d\beta(e^{j\omega})}{d\omega} = -jz \frac{d\beta}{dz} \Big|_{z=e^{j\omega}}.$$

Предпочтительна приблизительно постоянная характеристика групповой задержки во всей полосе (или полосах) пропускания фильтра.

Методы расчета коэффициентов БИХ-фильтра

Расчет фильтра сводится к нахождению значений его коэффициентов b_i и a_i , обеспечивающих аппроксимацию заданных характеристик (таких как импульсная, частотная, характеристика групповой задержки и др.) в том или ином смысле (например, в среднеквадратичном или минимаксном). Таким образом, задача расчета фильтра в значительной мере сводится к задаче аппроксимации и может быть решена чисто математическими методами.

Расчет цифровых БИХ-фильтров практически не связан с фильтрами непрерывного времени. Однако вместо того, чтобы заново создавать теорию расчета цифровых фильтров, можно воспользоваться простыми методами отображения, позволяющими преобразовывать аналоговые фильтры в цифровые. Именно эти методы чаще всего применяются при расчете стандартных БИХ-фильтров: нижних и верхних частот, полосовых, режекторных. В таких случаях последовательность расчета должна быть следующей:

- расчет аналогового фильтра-прототипа;
- преобразование полосы частот;
- дискретизация аналогового фильтра.

Другую группу методов расчета цифровых БИХ-фильтров образуют прямые методы расчета в z -плоскости. Часто удается найти такое расположение

полюсов и нулей фильтра, при котором обеспечивается некоторая аппроксимация непосредственно заданной характеристики фильтра.

Третий, также часто встречающийся подход к расчету БИХ-фильтров, заключается в использовании процедур оптимизации для нахождения положения нулей и полюсов в z -плоскости. В этом случае нельзя получить формулы, связывающие коэффициенты фильтра с параметрами заданной характеристики. Расчет производится методом последовательных приближений.

Расчет аналоговых фильтров-прототипов

В качестве аналогового фильтра-прототипа используется нормированный фильтр нижних частот с частотой среза, равной 1 рад/с. Передаточная функция такого фильтра представляет собой рациональную функцию следующего вида:

$$H(s) = \frac{\sum_{i=0}^m b_i s^i}{1 + \sum_{i=1}^n a_i s^i}.$$

Аппроксимируемой функцией является квадрат амплитудной характеристики. Чаще всего применяются фильтры Баттерворта, Чебышева и эллиптические.

Фильтры Баттерворта имеют максимально гладкую амплитудную характеристику в начале координат в s -плоскости. Ее квадрат равен

$$|H(\Omega)|^2 = \frac{1}{1 + (\Omega^2)^n},$$

где n – порядок фильтра.

Данные фильтры имеют только полюсы (все нули расположены на бесконечности). На частоте $\Omega=1$ рад/с коэффициент передачи равен $1/\sqrt{2}$ (т.е. на частоте среза амплитудная характеристика спадает на 3 дБ). Порядок фильтра полностью определяет весь фильтр.

Отличительной чертой фильтров Чебышева является наименьшая величина максимальной ошибки аппроксимации в заданной полосе частот. Ошибка аппроксимации представляется равновеликими пульсациями, флуктуирует между максимумами и минимумами равной величины. В зависимости от того, где минимизируется ошибка – в полосе пропускания или подавления, различают фильтры Чебышева типа I и II.

Фильтры Чебышева типа I имеют только полюсы и обеспечивают равновеликие пульсации амплитудной характеристики в полосе пропускания и монотонное изменение ослабления в полосе подавления. Квадрат амплитудной характеристики фильтра n -го порядка описывается выражением

$$|H(\Omega)|^2 = \frac{1}{1 + \varepsilon^2 T_n^2(\Omega)},$$

где ε – параметр, характеризующий пульсации в полосе пропускания; $T_n(\Omega)$ – полином Чебышева n -го порядка, равный

$$T_n(\Omega) = \begin{cases} \cos(n \arccos \Omega), & |\Omega| \leq 1, \\ \text{ch}(n \text{ Arch } \Omega), & |\Omega| > 1. \end{cases}$$

Свойство оптимальности фильтров Чебышева типа I заключается в том, что не существует какого-либо другого фильтра n -го порядка, содержащего только полюсы, которые имели бы такие же или лучшие характеристики и в полосе пропускания, и в полосе подавления.

Фильтры Чебышева типа II (иногда их называют обратными фильтрами Чебышева) обеспечивают монотонное изменение ослабления в полосе пропускания (максимально гладкое при $\Omega=0$) и равновеликие пульсации в полосе подавления. Нули фильтров располагаются на мнимой оси в s -плоскости, а полюсы – в левой полуплоскости. Квадрат амплитудной характеристики фильтра n -го порядка можно представить следующим образом:

$$|H(\Omega)|^2 = \frac{1}{1 + \varepsilon^2 [T_n(\Omega_r)/T_n(\Omega_r/\Omega)]^2},$$

где Ω_r – наименьшая частота, на которой в полосе подавления достигается заданный уровень ослабления.

Эллиптические фильтры (называемые также фильтрами Кауэра или Золотарева) характеризуются тем, что их амплитудная характеристика имеет равновеликие пульсации и в полосе пропускания, и в полосе подавления. Данные фильтры оптимальны с точки зрения минимальной ширины переходной полосы. Квадрат амплитудной характеристики записывается в виде

$$|H(\Omega)|^2 = \frac{1}{1 + \varepsilon^2 R_n^2(\Omega, L)},$$

где $R_n(\Omega, L)$ – рациональная функция Чебышева; L – параметр, характеризующий ее пульсации.

Преобразование полосы частот для аналоговых фильтров

Для преобразования фильтра нижних частот (ФНЧ) с частотой среза 1 рад/с в другой фильтр нижних частот (имеющий другую частоту среза), а также в фильтр верхних частот (ФВЧ), полосовой (ПФ) или режекторный (РФ) применяют следующие преобразования:

$$\begin{aligned} \text{ФНЧ} \rightarrow \text{ФНЧ:} & \quad s \rightarrow \frac{s}{\Omega_u}, \\ \text{ФНЧ} \rightarrow \text{ФВЧ:} & \quad s \rightarrow \frac{\Omega_u}{s}, \\ \text{ФНЧ} \rightarrow \text{ПФ:} & \quad s \rightarrow \frac{s^2 + \Omega_u \Omega_l}{s(\Omega_u - \Omega_l)}, \end{aligned}$$

$$\text{ФНЧ} \rightarrow \text{РФ:} \quad s \rightarrow \frac{s(\Omega_u - \Omega_l)}{s^2 + \Omega_u \Omega_l}.$$

Здесь Ω_l – нижняя частота среза; Ω_u – верхняя частота среза.

Данные соотношения имеют нелинейный характер, однако это не создает никаких трудностей, поскольку частотные характеристики преобразуемых фильтров аппроксимируются ступенчатой функцией. Так, нелинейность отображения приводит к изменению взаимного расположения максимумов и минимумов пульсаций характеристик эллиптических фильтров, но не влияет на амплитуду этих пульсаций. Поэтому фильтры, рассчитанные методами преобразования полосы, сохраняют равновеликий характер пульсаций фильтра-прототипа.

Дискретизация аналогового фильтра

Наиболее распространенным методом дискретизации аналоговых фильтров является метод билинейного z -преобразования. Он основан на простом конформном отображении s -плоскости в z -плоскости, получающемся в результате следующей замены:

$$s \rightarrow \frac{2}{T} \frac{(1 - z^{-1})}{(1 + z^{-1})}.$$

При таком преобразовании ось $j\Omega$ из s -плоскости отображается в единичную окружность на z -плоскости; левая полуплоскость s отображается в единичный круг, а правая полуплоскость s – в область вне единичного круга на z -плоскости.

Определенным недостатком билинейного z -преобразования является нелинейность соотношения между частотами аналогового фильтра Ω и цифрового фильтра ω :

$$\Omega \rightarrow \frac{2}{T} \operatorname{tg} \left(\frac{\omega T}{2} \right).$$

Данный недостаток не позволяет, например, использовать билинейное z -преобразование для получения цифрового дифференцирующего фильтра. Существует, правда, большой класс фильтров, для которых частотная деформация может быть скомпенсирована. К ним относятся фильтры нижних и верхних частот, полосовые и режекторные. Метод компенсации прост. Пусть известна совокупность частот среза цифрового фильтра. Используя приведенное соотношение между частотными шкалами, пересчитаем все частоты среза цифрового фильтра в частоты среза аналогового. Теперь рассчитаем аналоговый фильтр, все характерные частоты которого совпадали бы с этими пересчитанными частотами. Выполнив билинейное z -преобразование, получим требуемый цифровой фильтр.

Расчет цифровых фильтров в пакете Matlab

Расчет БИХ-фильтров с аппроксимацией Баттерворта в пакете Matlab осуществляется с помощью функции `butter`. Ее формат следующий:

```
[B,A] = butter(N,Wn)
```

Рассчитывается фильтр нижних частот N -го порядка с частотой среза w_n , заданной в долях частоты Найквиста ($0 < w_n < 1$). В векторах B и A длиной $N+1$ возвращаются коэффициенты числителя и знаменателя передаточной функции, расположенные в порядке уменьшения степеней z . Команда

```
[B,A] = butter(N,Wn,'high')
```

позволяет получить фильтр нижних частот. Если w_n – двухэлементный вектор $w_n = [w_1 \ w_2]$, то будет рассчитан полосовой фильтр порядка $2N$ с полосой пропускания $w_1 < w < w_2$. Для получения режекторного фильтра следует использовать команду

```
[B,A] = BUTTER(N,Wn,'stop')
```

При расчете фильтров в качестве исходных данных обычно задаются граничные частоты полос пропускания w_p и подавления w_s , а также наибольшее допустимое отклонение в полосе пропускания R_p и наименьшее допустимое затухание в полосе подавления R_s . Получить необходимые для функции `butter` порядок фильтра и частоту среза можно с помощью функции `butterord`:

```
[N, Wn] = butterord(Wp, Ws, Rp, Rs)
```

Например, $w_p = 0.2$, $w_s = 0.1$ соответствует фильтру верхних частот, а $w_p = [0.1 \ 0.8]$, $w_s = [0.2 \ 0.7]$ – полосовому.

Для расчета фильтров Чебышева типов I и II используются функции `cheby1` и `cheby2`:

```
[B,A] = cheby1(N,R,Wn)
```

```
[B,A] = cheby2(N,R,Wn)
```

Для `cheby1` параметр R является размахом колебаний в полосе пропускания (в децибелах), а для `cheby2` – в полосе подавления. В остальном данные функции аналогичны `butter`.

Определить порядок фильтров Чебышева типов I и II и их частоту среза можно с помощью функций `cheby1ord` и `cheby2ord`, которые аналогичны функциям `butter`.

Эллиптические фильтры рассчитываются функцией `ellip`:

```
[B,A] = ellip(N,Rp,Rs,Wn)
```

Параметры R_p и R_s обозначают размах колебаний в полосах пропускания и подавления соответственно, выраженные в децибелах. Остальные параметры соответствуют функции `butter`. Для нахождения N и W_n используется функция `ellipord`.

При анализе цифровых БИХ-фильтров часто бывает необходимо построить диаграмму расположения нулей и полюсов передаточной функции на z -плоскости. Для этого служит функция Matlab `zplane`. В качестве параметров ей передаются вектора B и A – коэффициенты числителя и знаменателя передаточной функции. Функция `zplane` строит на графике единичную окружность и отмечает положение нулей с помощью символа 'o' и полюсов с помощью символа 'x'. При кратности нулей и полюсов, большей единицы, она показывается справа вверху.

5.2. Порядок выполнения работы

5.2.1. В соответствии с номером варианта рассчитайте ФНЧ с аппроксимацией Баттерворта (табл. 5.1).

Таблица 5.1

Варианты для задания 5.2.1

Номер варианта	1	2	3	4	5	6
Порядок фильтра	7	6	5	8	9	5
Частота дискретизации, Гц	200	2000	16000	8000	10000	20
Частота среза, Гц	60	400	5000	1900	2000	5

Постройте АЧХ и ФЧХ, диаграмму расположения нулей и полюсов передаточной функции, значимую часть импульсной характеристики. Каким образом можно получить каждые два графика из третьего?

5.2.2. Выполните задание 5.2.1, воспользовавшись функцией `cheby1` и приняв допустимую неравномерность АЧХ в полосе пропускания 0,5 дБ.

5.2.3. Выполните задание 5.2.2, воспользовавшись функцией `ellip` и приняв минимальное затухание АЧХ в полосе подавления 30 дБ.

5.2.4. Требуется цифровой ФВЧ со следующими параметрами (табл. 5.2):

Таблица 5.2

Варианты для задания 5.2.4 (частоты определены в долях от частоты Найквиста)

Номер варианта	1	2	3	4	5	6
Граничная частота подавления	0,64	0,28	0,4	0,1	0,86	0,72
Граничная частота пропускания	0,7	0,32	0,6	0,15	0,9	0,73
Допустимая неравномерность в полосе пропускания, дБ	0,05	0,1	$1 \cdot 10^{-5}$	0,15	0,11	1
Минимальное затухание в полосе подавления, дБ	40	30	100	85	57	30

Какой порядок будет иметь такой фильтр с аппроксимациями Баттерворта, Чебышева типа I, Чебышева типа II, эллиптической? Сравните эффективность различных аппроксимаций при более жестких и более мягких требованиях к АЧХ.

5.2.5. Исследуйте, как скажется на АЧХ и ФЧХ фильтров из заданий 5.2.1–5.2.3 усечение коэффициентов передаточной функции до четырех десятичных разрядов, до двух разрядов? Сделайте выводы.

ЛАБОРАТОРНАЯ РАБОТА №6. Расчет цифровых КИХ-фильтров с линейной фазовой характеристикой методом взвешивания

ЦЕЛЬ РАБОТЫ – расчет КИХ-фильтров с линейной фазовой характеристикой в пакете Matlab и изучение их свойств, изучение различных типов временных окон.

6.1. Теоретические сведения

Представление о КИХ-фильтре

Представление о цифровом фильтре с конечной импульсной характеристикой можно получить, если рассмотреть задачу вычисления скользящего значения в массиве данных. Предположим, есть массив данных $x(n), n = 0, 1, \dots$. Необходимо сформировать массив выходных (сглаженных) данных

$$y(n) = \frac{1}{5} \sum_{m=-2}^2 x(n-m). \quad (6.1)$$

В данном случае сглаживание состоит в том, что для получения выходного значения в каждый момент времени n берется среднее значение от пяти точек входных данных $x(n-2), x(n-1), x(n), x(n+1)$ и $x(n+2)$. Заметим, что с помощью этой формулы невозможно сгладить по два значения в начале и в конце потока данных. На рис. Рис. 6.16 поясняется работа формулы (6.1).

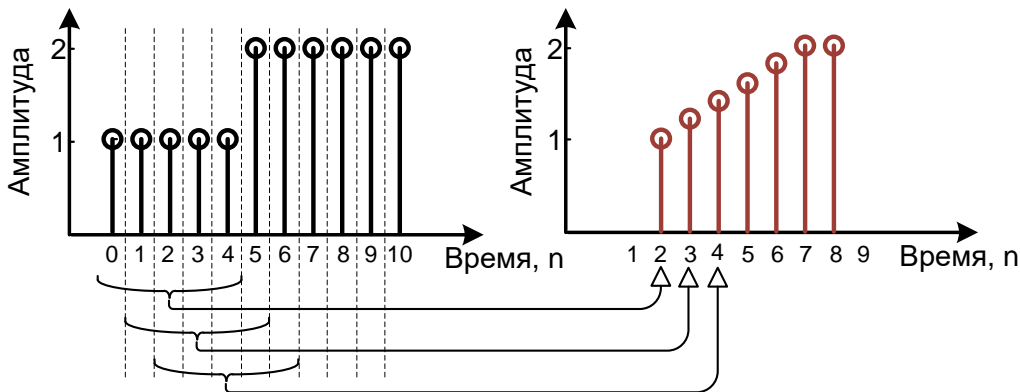


Рис. 6.1. Процесс вычисления скользящего среднего значения массива данных

Формула (6.1) представляет собой цифровой КИХ-фильтр ($N = 4$)

$$y(n) = \sum_{m=0}^N x(n-m)h(m), \quad (6.2)$$

у которого все коэффициенты $h(m)$ имеют значения $1/5$.

Как выглядит формула (6.1) с частотной точки зрения? Предположим, что входная функция представляет собой комплексную синусоиду $e^{j\omega n}$. Поскольку формула линейна по отношению к данным, то такая же функция будет всегда на выходе, за исключением того, что она умножается на множитель $H(\omega)$, который называется частотной характеристикой фильтра (см. лабораторную работу №2). Подставив комплексную экспоненту $e^{j\omega n}$ в формулу сглаживания пятерками, получим

$$H(\omega) = \frac{1}{5}(e^{-2j\omega} + e^{-j\omega} + 1 + e^{j\omega} + e^{2j\omega}).$$

Отметим, что отрицательные и положительные частоты имеют одинаковые коэффициенты и, следовательно, попарно могут быть заменены соответствующими косинусными функциями:

$$H(\omega) = \frac{1}{5}(1 + 2 \cos \omega + 2 \cos 2\omega).$$

Вид этой функции показан на рис. 6.2. Частотная характеристика $H(\omega)$ показывает, что происходит, когда усредняются данные с частотой ω . Очевидно, что сглаживание можно выполнять и по 7, и по 9, и т.д. точкам.

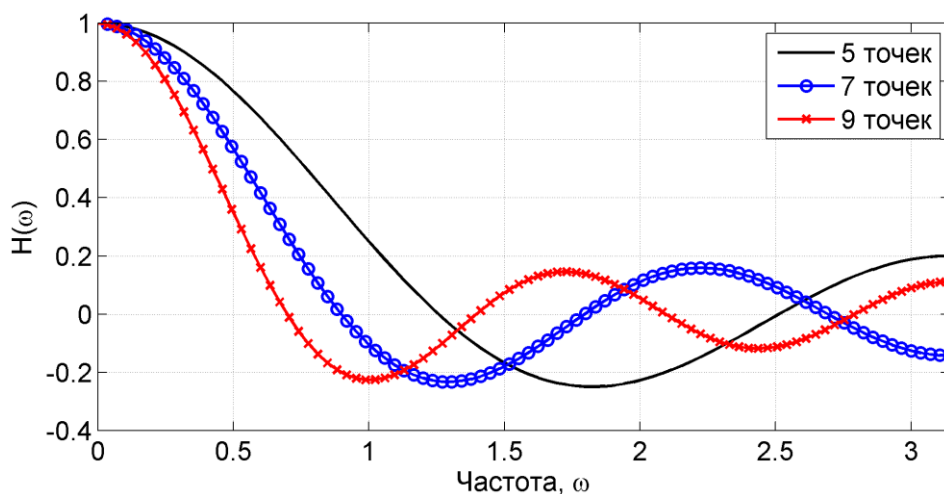


Рис. 6.2. КИХ-фильтр для сглаживания данных в частотной области

На рис. 6.3 и 6.4 показано вычисление скользящего среднего для данных, содержащих колебания различных частот. Из приведенных иллюстраций можно сделать вывод, что увеличение частоты колебаний во входных данных привело к уменьшению амплитуды колебаний в выходном массиве данных, что находится в полном соответствии с графиком частотной характеристики.

Таким образом, мы рассмотрели частный случай КИХ-фильтра (6.2), у которого все коэффициенты являются постоянными. В общем случае проектирование КИХ-фильтра состоит в расчете таких коэффициентов $h(m)$, которые обеспечивали бы требуемый вид частотной характеристики.

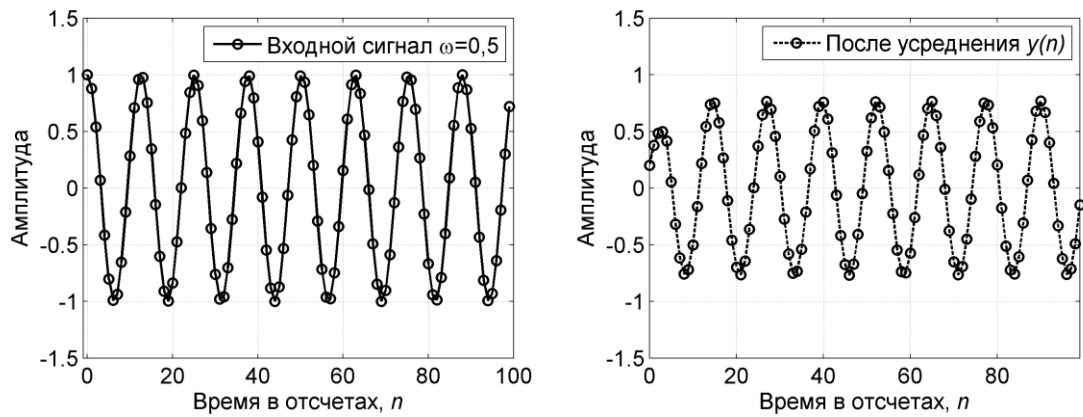


Рис. 6.3. Усреднение по 5точкам данных, содержащих колебание частотой $\omega = 0,5$ рад/с

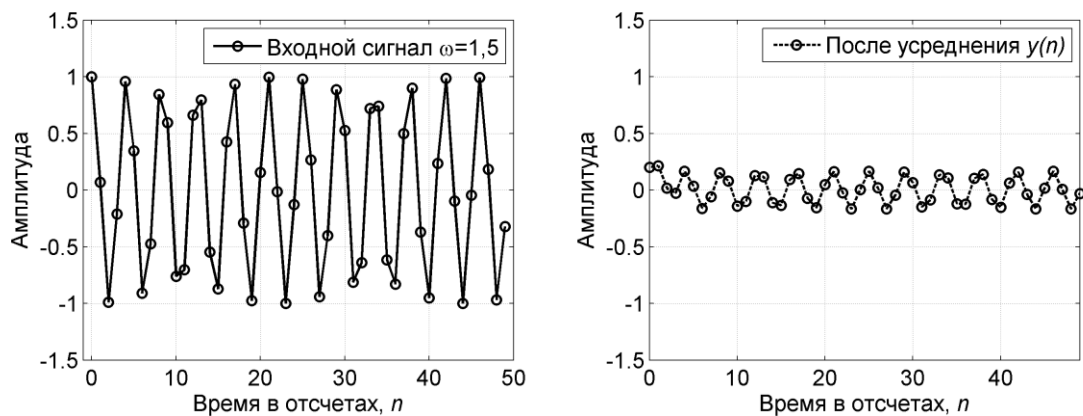


Рис. 6.4. Усреднение по 5 точкам данных, содержащих колебание частотой $\omega = 1,5$ рад/с

Основные свойства КИХ-фильтров

Пусть импульсная характеристика КИХ-фильтра $h(n)$ задана на интервале $0 \leq n \leq N-1$. Ее преобразование Фурье равно

$$H(e^{j\omega}) = \sum_{n=0}^{N-1} h(n) e^{-j\omega n}.$$

Рассматривая только действительные $h(n)$, получим дополнительные ограничения на функцию $H(e^{j\omega})$, представив ее через амплитуду и фазу:

$$H(e^{j\omega}) = \pm |H(e^{j\omega})| e^{j\theta(\omega)}.$$

Модуль преобразования Фурье является симметричной функцией, а фаза – антисимметричной функцией частоты, т. е.

$$|H(e^{j\omega})| = |H(e^{-j\omega})|, \quad 0 \leq \omega \leq \pi, \quad \theta(\omega) = -\theta(-\omega).$$

На практике при расчете КИХ-фильтров часто требуется строго линейная фазовая характеристика. Линейность фазы является еще одним ограничением; она означает, что фазовая характеристика $\theta(\omega)$ имеет вид

$$\theta(\omega) = -\alpha\omega, \quad -\pi \leq \omega \leq \pi,$$

где α – постоянная фазовая задержка, выраженная через число интервалов дискретизации.

Выражение для преобразования Фурье от импульсной характеристики теперь можно переписать следующим образом:

$$H(e^{j\omega}) = \sum_{n=0}^{N-1} h(n) e^{-j\omega n} = \pm |H(e^{j\omega})| e^{-j\alpha\omega} = H^*(e^{j\omega}) e^{-j\alpha\omega},$$

где $H^*(e^{j\omega})$ – действительная функция.

Для каждого N существует только одна фазовая задержка α , при которой достигается строгая линейность фазовой характеристики фильтра:

$$\alpha = \frac{N-1}{2}.$$

При этом импульсная характеристика должна обладать вполне определенной симметрией:

$$h(n) = h(N-1-n), \quad 0 \leq n \leq N-1.$$

Если N – нечетное, то α – целое, т.е. задержка в фильтре равна целому числу интервалов дискретизации. Центр симметрии импульсной характеристики приходится на центральный отсчет, для которого нет пары. Функция $H^*(e^{j\omega})$ в этом случае симметрична относительно π .

При четном N задержка становится дробной. Центр симметрии импульсной характеристики располагается между двумя центральными отсчетами, и каждый отсчет имеет парный. Функция $H^*(e^{j\omega})$ асимметрична относительно π и обращается в 0 при $\omega = \pi$. Отсюда следует, что такие фильтры нельзя использовать, например, для проектирования фильтров верхних частот.

Согласно условию линейности фазовой характеристики, требуется, чтобы фильтр имел постоянную как групповую, так и фазовую задержку. Если достаточно, чтобы только групповая задержка была постоянной (как это часто бывает), можно определить еще один тип фильтра с линейной фазой, фазовая характеристика которого является кусочно-линейной функцией частоты:

$$H(e^{j\omega}) = \pm |H(e^{j\omega})| e^{j(\beta - \alpha\omega)}.$$

Линейность фазы в этом случае достигается при следующих условиях:

$$\alpha = \frac{N-1}{2},$$

$$\beta = \pm \frac{\pi}{2},$$

$$h(n) = -h(N-1-n), \quad 0 \leq n \leq N-1.$$

Импульсные характеристики фильтров данного типа, в отличие от предыдущих, антисимметричны относительно центра.

Для нечетных значений N средний отсчет импульсной характеристики всегда равен нулю. Функция $H^*(e^{j\omega})$ антисимметрична относительно π и обращается в 0 при $\omega = 0$ и $\omega = \pi$. Без учета множителя с линейным изменением фазы частотная характеристика является чисто мнимой, поэтому такие фильтры

наиболее пригодны для проектирования преобразователей Гильберта и дифференциаторов.

В случае, если N четно, $H^*(e^{j\omega})$ симметрична относительно π и обращается в 0 при $\omega=0$.

Расчет КИХ-фильтров с линейной фазой методом взвешивания

Суть метода взвешивания (метод окна) сводится к получению конечной импульсной характеристики путем усечения последовательности бесконечной импульсной характеристики длины. Поскольку частотная характеристика требуемого идеального цифрового фильтра $H_d(e^{j\omega})$ является периодической функцией частоты, ее можно представить рядом Фурье:

$$H_d(e^{j\omega}) = \sum_{n=-\infty}^{\infty} h_d(n)e^{-j\omega n}, \quad (6.3)$$

где $h_d(n)$ – соответствующая последовательность отсчетов импульсной характеристики, т.е.

$$h_d(n) = (1/2\pi) \int_{-\pi}^{\pi} H_d(e^{j\omega}) e^{j\omega n} d\omega. \quad (6.4)$$

Использование соотношений (6.3) и (6.4) для проектирования КИХ-фильтров связано с двумя трудностями. Во-первых, импульсная характеристика фильтра в общем случае имеет бесконечную длину, поскольку суммирование в (6.3) производится в бесконечных пределах. Во-вторых, такой фильтр физически нереализуем, т. к. импульсная характеристика начинается в $-\infty$, т.е. никакая конечная задержка не сделает фильтр физически реализуемым. Итак, фильтр, рассчитываемый на основе представления функции $H_d(e^{j\omega})$ рядом Фурье, оказывается физически нереализуемым БИХ-фильтром.

Один из возможных методов получения КИХ-фильтра, аппроксимирующего заданную функцию $H_d(e^{j\omega})$, заключается в усечении бесконечного ряда Фурье (6.3) за пределами $n = \pm(N-1)/2$. Однако простое усечение ряда приводит к хорошо известному явлению Гиббса, которое проявляется в виде выбросов и пульсаций определенного уровня до и после точки разрыва в аппроксимируемой частотной характеристике. Так, например, при аппроксимации стандартных фильтров типа идеального фильтра нижних частот или полосового фильтра максимальная амплитуда пульсаций частотной характеристики составляет около 9% и не уменьшается с увеличением длины импульсной характеристики, т.е. учет все большего числа членов ряда Фурье не приводит к уменьшению максимальной амплитуды пульсаций. Вместо этого по мере увеличения уменьшается ширина выброса.

Лучшие результаты дает метод проектирования КИХ-фильтров, основанный на использовании весовой последовательности конечной длины $w(n)$, называемой окном, для модификации коэффициентов Фурье $h_d(n)$ в формуле (6.3), с тем чтобы управлять сходимостью ряда Фурье.

Метод взвешивания иллюстрируется на рис. 6.5: заданная периодическая частотная характеристика $H_d(e^{j\omega})$ и ее коэффициенты Фурье $h_d(n)$ – на рис. 6.5, а, весовая последовательность конечной длины $w(n)$ и ее преобразование Фурье $W(e^{j\omega})$ – на рис. 6.5, б. Для большинства приемлемых окон функция $W(e^{j\omega})$ имеет главный лепесток, содержащий почти всю энергию окна, и боковые лепестки, которые обычно быстро затухают. Чтобы получить аппроксимацию функции $H_d(e^{j\omega})$, формируется последовательность $h(n) = h_d(n)w(n)$, в точности равная нулю за пределами интервала $-(N-1)/2 \leq n \leq (N-1)/2$.

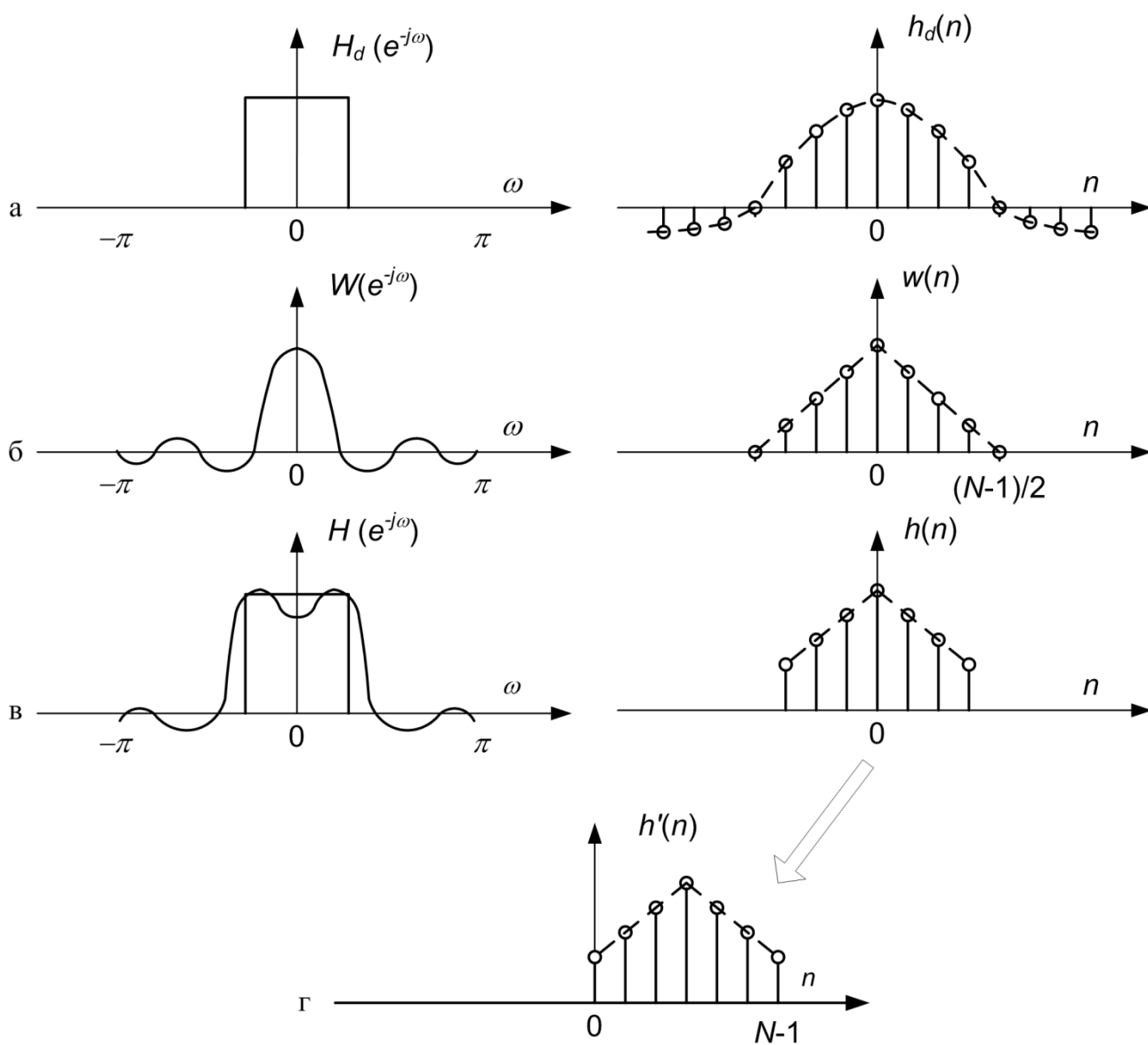


Рис. 6.5. Иллюстрация метода взвешивания

На рис. 6.5, в представлена последовательность $h(n)$ и ее преобразование Фурье $H(e^{j\omega})$, равное, очевидно, круговой свертке функций $H_d(e^{j\omega})$ и $W(e^{j\omega})$, поскольку $h(n)$ является произведением $h_d(n)w(n)$:

$$H(e^{j\omega}) = \frac{1}{2\pi} \int_{-\pi}^{\pi} H_d(e^{j\theta}) W(e^{j(\omega-\theta)}) d\theta. \quad (6.5)$$

Физически реализуемая последовательность $h'(n) = h(n - (N - 1)/2)$, которая равна задержанной последовательности $h(n)$ и может быть использована в качестве искомой импульсной характеристики фильтра показана на рис. 6.5, г. При этом частотная характеристика физически реализуемого фильтра будет равна $H(e^{j\omega})e^{j\omega(N-1)/2}$ и при соблюдении условий симметрии импульсной характеристики: $h'(n) = \pm h'(N - 1 - n)$ будет иметь линейную ФЧХ.

На примере, представленном на рис. 6.5, можно проследить влияние операции взвешивания коэффициентов Фурье фильтра на его частотную характеристику.

Прежде всего, по обе стороны от точек разрыва заданной функции $H_d(e^{j\omega})$ появляются переходные полосы. Ясно, что поскольку результирующая частотная характеристика фильтра равна круговой свертке идеальной частотной характеристики и частотной характеристики окна, то ширина переходных полос зависит от ширины главного лепестка функции $W(e^{j\omega})$. Кроме того, на всех частотах ω возникают ошибки аппроксимации, имеющие вид пульсаций частотной характеристики, которые обусловлены боковыми лепестками функции $W(e^{j\omega})$. Ясно, наконец, и то, что получаемые фильтры ни в каком смысле *не являются оптимальными* (даже если окна и удовлетворяют тому или иному критерию оптимальности), поскольку их частотные характеристики рассчитываются через свертку.

После общего рассмотрения метода взвешивания возникают два вопроса: какими свойствами должны обладать окна и насколько точно они могут быть реализованы на практике? Ответ на первый вопрос относительно прост. Желательно, чтобы окно обладало следующими свойствами:

- 1) ширина главного лепестка частотной характеристики окна, содержащего по возможности большую часть общей энергии, должна быть малой;
- 2) энергия в боковых лепестках частотной характеристики окна должна быстро уменьшаться при приближении ω к π .

Было предложено много окон, аппроксимирующих заданные характеристики. В качестве примеров мы рассмотрим три окна, а именно: прямоугольное окно, «обобщенное» окно Хэмминга и окно Кайзера. Эти окна обладают свойствами всех возможных видов окон и позволяют достаточно хорошо понять преимущества и недостатки метода взвешивания.

Прямоугольное окно

N -точечное *прямоугольное окно*, соответствующее простому усечению (без модификации) ряда Фурье, описывается весовой функцией

$$w_R(n) = \begin{cases} 1, & -\frac{N-1}{2} \leq n \leq \frac{N-1}{2}, \\ 0, & \text{при других } n. \end{cases}$$

Здесь и далее предполагается, что N – нечетное число. С помощью простой модификации аналогичные результаты могут быть получены и для четного N . Предполагается также, что последовательность окна имеет нулевую задержку. Частотная характеристика прямоугольного окна описывается соотношением

$$\begin{aligned} W(e^{j\omega}) &= \sum_{n=-(N-1)/2}^{(N-1)/2} e^{-j\omega n} = \frac{e^{j\omega(N-1)/2}(1 - e^{-j\omega N})}{(1 - e^{-j\omega})} = \\ &= \frac{e^{j\omega N/2} - e^{-j\omega N/2}}{e^{j\omega/2} - e^{-j\omega/2}} = \frac{\sin(\omega N/2)}{\sin(\omega/2)}. \end{aligned}$$

Для вычисления N -точечного прямоугольного окна в пакете Matlab можно использовать функцию `boxcar(N)`. График прямоугольного окна и его АЧХ для $N=23$ в диапазоне от $-\pi$ до π представлен на рис. 6.6.

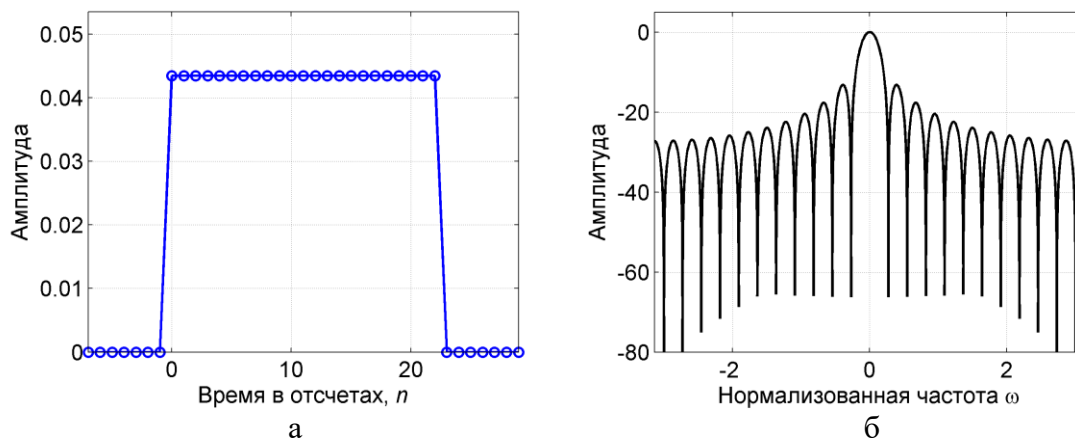


Рис. 6.6. Прямоугольное окно: представление во временной (а) и частотной (б) областях

Окно Бартлетта

N -точечное *окно Бартлетта* – треугольное окно. Коэффициенты окна Бартлетта вычисляются по следующим формулам:

$$w(k) = \begin{cases} \frac{2k}{n-1}, & 0 \leq k \leq \frac{n-1}{2}, \\ 2 - \frac{2k}{n-1}, & \frac{n-1}{2} \leq k \leq n-1, \end{cases} \quad (\text{для нечетного } n);$$

$$w(k) = \begin{cases} \frac{2k}{n-1}, & 0 \leq k \leq \frac{n}{2} - 1, \\ \frac{2(n-k-1)}{n-1}, & \frac{n}{2} \leq k \leq n-1, \end{cases} \quad (\text{для четного } n).$$

Окно Бартлетта почти полностью совпадает с треугольным окном. Отличие от треугольного окна заключается лишь в том, что значение окна Бартлетта при $k = 1$ и $k = n$ равно нулю.

Для вычисления N -точечных окон Бартлетта и треугольного в Matlab можно использовать функции `bartlett(N)` и `triang(N)` соответственно. Графики АЧХ окна Бартлетта и треугольного окна для $N = 23$ в диапазоне от $-\pi$ до π представлены на рис. 6.7 и 6.8 соответственно.

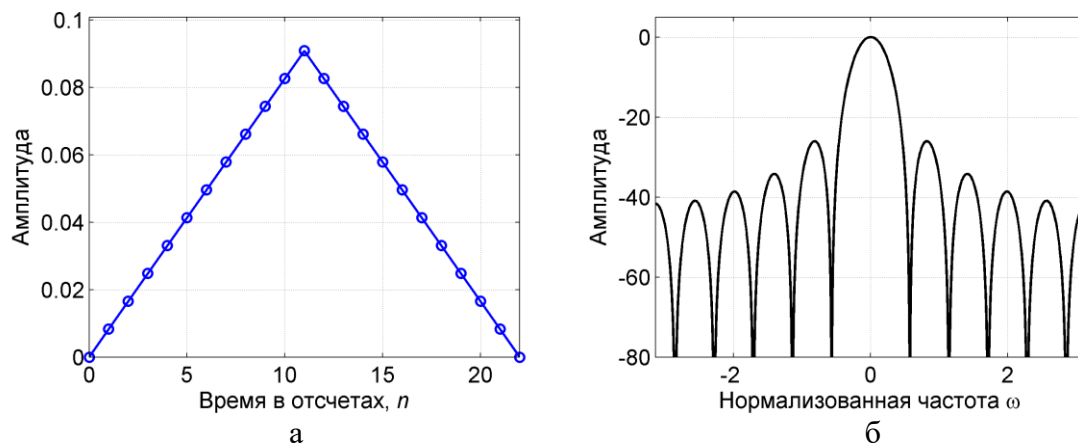


Рис. 6.7. Окно Бартлетта: представление во временной (а) и частотной (б) областях

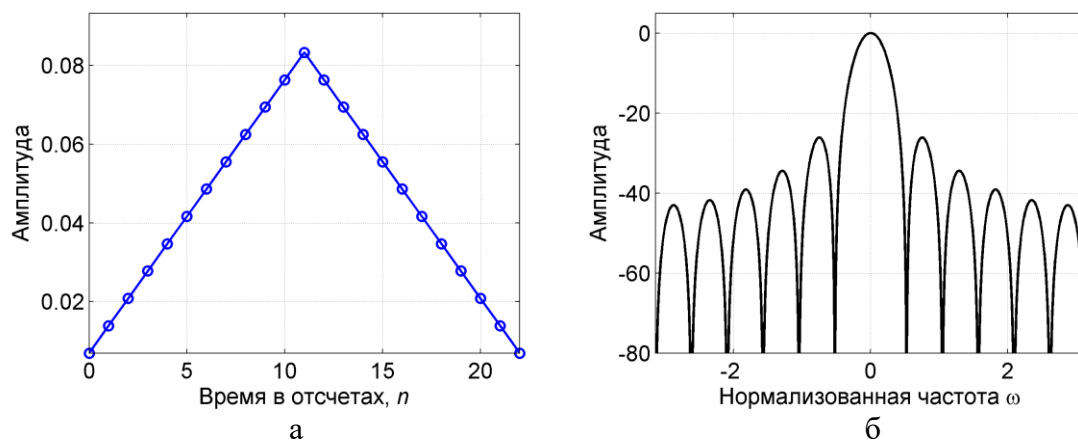


Рис. 6.8. Треугольное окна: представление во временной (а) и частотной (б) областях

Обобщенное окно Хэмминга

Окно, называемое *обобщенным окном Хэмминга*, имеет вид

$$w_H(n) = \begin{cases} \alpha + (1 - \alpha) \cos\left(\frac{2\pi n}{N-1}\right), & -\frac{N-1}{2} \leq n \leq \frac{N-1}{2}, \\ 0, & \text{при других } n, \end{cases} \quad (6.6)$$

причем α лежит в пределах $0 \leq \alpha \leq 1$. Случай, когда $\alpha = 0,5$ соответствует окну Ханна, когда $\alpha = 0,54$ – окну Хэмминга. Частотную характеристику рассматриваемого окна легко получить, если учесть, что оно может быть представлено в виде произведения прямоугольного окна и окна, определяемого формулой (6.6), но для всех n , т. е.

$$w_H(n) = w_R(n) \left[\alpha + (1 - \alpha) \cos\left(\frac{2\pi n}{N-1}\right) \right].$$

Следовательно, частотная характеристика обобщенного окна Хэмминга может быть записана в следующем виде:

$$W_H(e^{j\omega}) = \alpha W_R(e^{j\omega}) + \frac{1-\alpha}{2} W_R(e^{j[\omega - (2\pi/(N-1))]} + \frac{1-\alpha}{2} W_R(e^{j[\omega + (2\pi/(N-1))]}).$$

Для вычисления N -точечного окна Ханна в Matlab можно использовать функцию `hann(N)`, окна Хэмминга – функцию `hamming(N)`, окна Блэкмана – функцию `blackman(N)`. Эти окна являются частным случаем обобщенного окна Хэмминга. Графики и АЧХ окон Ханна, Хэмминга и Блэкмана для $N = 23$ в диапазоне от $-\pi$ до π представлены на рис. 6.9, 6.10 и 6.11 соответственно.

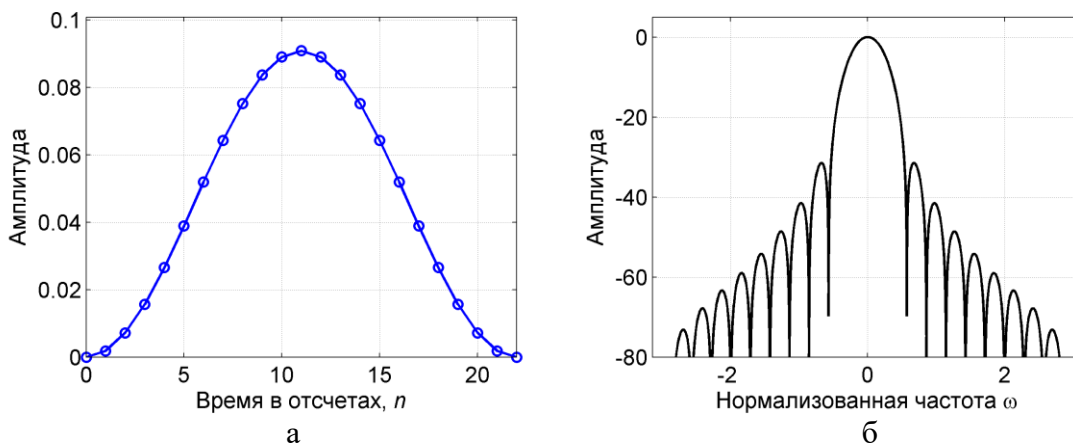


Рис. 6.9. Окно Ханна: представление во временной (а) и частотной (б) областях

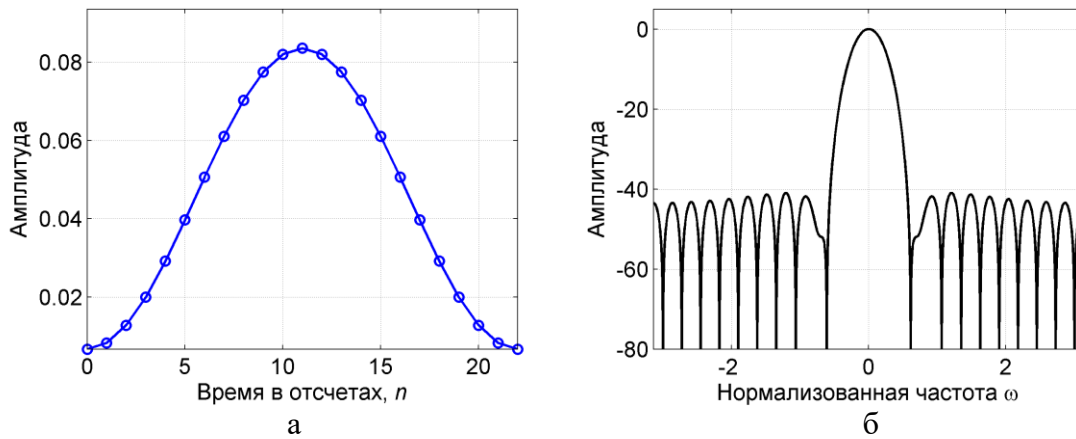


Рис. 6.10.Окно Хэмминга: представление во временной (а) и частотной (б) областях

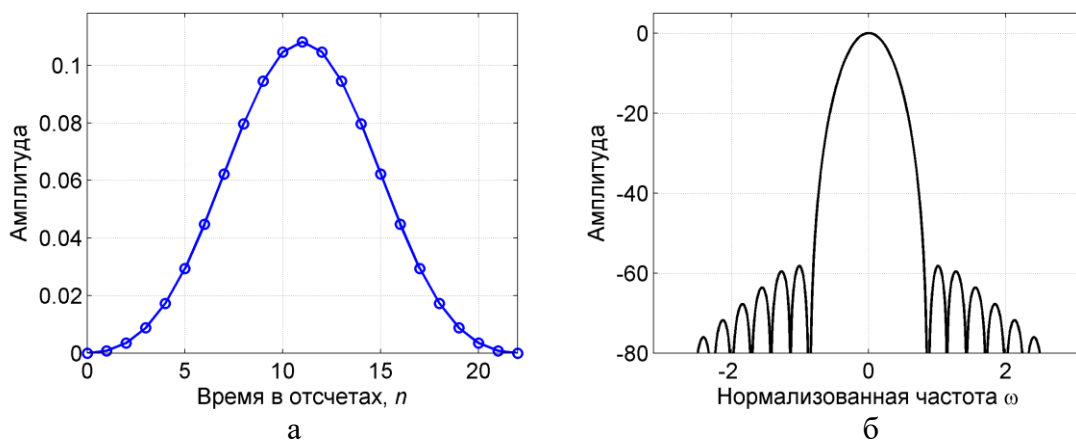


Рис. 6.11.Окно Блэкмана: представление во временной (а) и частотной (б) областях

Из рисунков видно, что ширина главного лепестка частотной характеристики окна Хэмминга в два раза больше, чем для прямоугольного окна. Однако уровень боковых лепестков в случае окна Хэмминга значительно ниже, чем у прямоугольного окна. При $\alpha = 0,54$, т. е. для обычного окна Хэмминга, 99,96 % общей энергии спектра содержится в главном лепестке, а максимумы боковых лепестков намного ниже главного максимума. В отличие от окна Хэмминга максимум боковых лепестков в спектре прямоугольного окна ниже главного максимума всего на 10–20 дБ.

При расчете фильтра нижних частот расширение главного лепестка соответствует расширению переходной полосы между полосами пропускания и непропускания, тогда как уменьшение уровня боковых лепестков соответствует меньшим пульсациям в полосе пропускания и лучшему подавлению в полосе непропускания фильтра.

Окно Кайзера

Задача расчета хороших окон фактически сводится к математической задаче отыскания ограниченных во времени функций, преобразования Фурье которых наилучшим образом аппроксимируют функции, ограниченные по частоте.

те, т.е. имеют минимальную энергию за пределами заданного интервала частот. При решении этой задачи в замкнутой форме для непрерывных функций времени был введен класс так называемых *вытянутых сфероидальных волновых функций*. Эти функции имеют достаточно сложный вид. Поэтому Кайзер в качестве наилучшего окна предложил относительно простую аппроксимацию этих функций, названную окном Кайзера, которая записывается в следующем виде:

$$w_K(n) = \begin{cases} \frac{I_0\left(\beta\sqrt{1-[2n/(N-1)]^2}\right)}{I_0(\beta)}, & \text{при } -\left(\frac{N-1}{2}\right) \leq n \leq \frac{N-1}{2}, \\ 0, & \text{при других } n, \end{cases}$$

где β – константа, определяющая компромисс между максимальным уровнем боковых лепестков и шириной главного лепестка (или долей общей энергии в главном лепестке) частотной характеристики окна, а модифицированная функция Бесселя первого рода нулевого порядка вычисляется согласно следующему выражению:

$$I_0(x) = 1 + \sum_{k=1}^{\infty} \left[\frac{1}{k!} \left(\frac{x}{2}\right)^k \right]^2.$$

Как именно величина β оказывает влияние на этот компромисс, видно из табл.6.1.

Таблица 6.1
Влияние коэффициента β

β	$D = \Delta F N$ (ΔF – ширина переходной полосы, N – длина окна)	Пульсация в по- лосе пропуска- ния, дБ	Пульсация в полосе непропускания, дБ
2,120	1,50	$\pm 0,27$	-30
3,384	2,23	$\pm 0,0864$	-40
4,538	2,93	$\pm 0,0274$	-50
5,658	3,62	$\pm 0,00868$	-60
6,764	4,32	$\pm 0,00275$	-70
7,865	5,0	$\pm 0,000868$	-80

Для вычисления N -точечного окна Кайзера сопредельной величиной β в Matlab можно использовать функцию `kaiser(N, β)`. График и АЧХ окна Кайзера для $N = 23$ в диапазоне от $-\pi$ до π представлен нарис. 6.12.

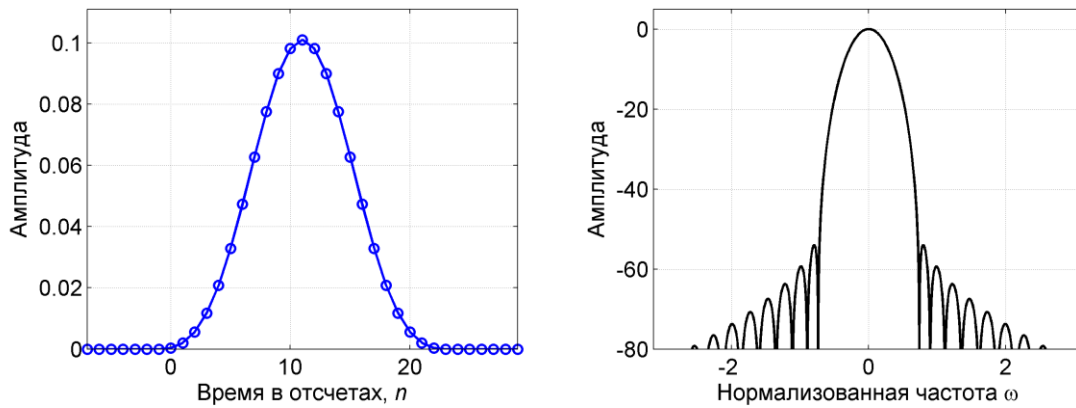


Рис. 6.12. Окно Кайзера (для $\beta=7,5$): представление во временной и частотной области

Окно Кайзера является по существу оптимальным окном, т.к. представляет последовательность конечной длины, которая имеет минимум энергии спектра за пределами некоторой заданной частоты.

Как отмечено выше, метод взвешивания предполагает предварительное вычисление импульсной характеристики $h_d(n)$ (6.4) требуемого фильтра как коэффициентов ряда Фурье (6.3) его частотной характеристики. В тех случаях, когда аналитическое вычисление затруднено, можно осуществить дискретизацию по частоте функции $H_d(e^{j\omega})$ с очень малым интервалом дискретизации $2\pi/M$ ($M \gg N$), а затем вычислить обратное дискретное преобразование Фурье полученной последовательности выборок:

$$h(n) = \frac{1}{N} \sum_{k=0}^{M-1} H_d \left(e^{j(2\pi k/M)} \right) e^{j(2\pi kn/M)} .$$

Тогда функция $h(n) \approx h_d(n)$ в диапазоне $|n| \leq (N-1)/2$.

Аналитическое определение импульсных характеристик не представляет труда для большинства широко используемых типов фильтров, т.е. для ФНЧ, ФВЧ, фильтра полосно-пропускающего (ФПП) и фильтра полосно-задерживающего (ФПЗ). В табл.6.2 сведены импульсные характеристики перечисленных выше идеальных, физически нереализуемых фильтров, для которых f_c обозначает безразмерную частоту среза для ФНЧ и ФВЧ, а f_1 и f_2 – граничные частоты полосы пропускания для ФПП, либо граничные частоты полосы задерживания (непропускания) для ФПЗ.

Итак, порядок расчета фильтра методом взвешивания следующий:

1. В зависимости от вида фильтра по формулам из табл.6.2 вычисляется импульсная характеристика фильтра.
2. Полученная последовательность взвешивается с помощью временного окна.

Выражения для идеальных импульсных характеристик различных типов фильтров

Тип фильтра	Формулы для нечетное N	Формулы для четного N
ФНЧ	$h(0) = 2f_c$ $h(n) = \frac{\sin(2\pi n f_c)}{n\pi}$	$h(n) = \frac{\sin 2\pi(n - 0,5)f_c}{(n - 0,5)\pi}$
ФВЧ	$h(0) = 1 - 2f_c$ $h(n) = -\frac{\sin(2\pi n f_c)}{n\pi}$	$h(n) = \frac{1}{(n - 0,5)\pi} [\sin \pi(n - 0,5) - \sin 2\pi(n - 0,5)f_c]$
ФПП	$h(0) = 2(f_2 - f_1)$ $h(n) = \frac{1}{n\pi} [\sin(2\pi n f_2) - \sin(2\pi n f_1)]$	$h(n) = \frac{1}{(n - 0,5)\pi} [\sin 2\pi(n - 0,5)f_2 - \sin 2\pi(n - 0,5)f_1]$
ФПЗ	$h(0) = 1 - 2(f_2 - f_1)$ $h(n) = \frac{1}{n\pi} [\sin(2\pi n f_1) - \sin(2\pi n f_2)]$	$h(n) = \frac{1}{(n - 0,5)\pi} [\sin \pi(n - 0,5)f_c - \sin 2\pi(n - 0,5)f_c]$

Пример 6.1. Расчета КИХ-фильтра

Используя окно Хэмминга, необходимо рассчитать КИХ-фильтр нижних частот линейной фазой с длиной импульсной характеристики 23 отсчета и нормированной частотой среза $\omega_0 = 0,3$. Вычислить и отобразить на графиках идеальную и реальную (полученную) АЧХ, АЧХ в логарифмическом масштабе, импульсную характеристику и расположение нулей на единичной окружности.

При расчетах будем учитывать, что проектируемый фильтр нижних частот имеет коэффициенты ряда Фурье (т.е. отсчеты импульсной характеристики) вида

$$h(0) = 2f_c, \quad h(n) = \frac{\sin(2\pi n f_c)}{n\pi}.$$

Для вычисления последовательности вида $\frac{\sin(xn)}{xn}$ в Matlab может быть использована стандартная функция $y = \text{sinc}(x*n)$. Для расчета заданных выше фильтров в Matlab используется следующая программа

```
N = 23;
w0 = 0.3;
M = (N - 1) / 2;
n = floor(-M) : floor(M);
```

```

L = 1024;
IdealW = 0 : 1/L : (L - 1) / L;
IdealH = zeros(L, 1);
IdealH(1 : w0 * L + 1) = 1;
h = w0 * sinc(w0 * n);
w = hamming(N);
h = h .* w';
[H, wi] = freqz(h, 1, L);
figure;
plot(wi/pi, abs(H), 'k'); hold on;
plot(IdealW, IdealH, 'b--');
grid on;
figure;
plot(wi/pi, 20 * log10(abs(H)), 'k');
grid on; zoom on;
figure;
impz(h, 1);
grid on;
figure;
zplane(h, 1);
grid on;

```

Результаты работы программы представлены на рис.6.13 –Рис. 6.16.

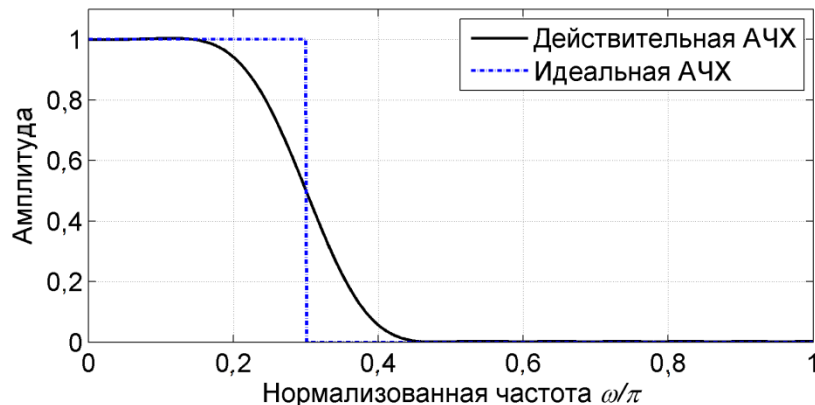


Рис. 6.13. Амплитудно-частотные характеристики фильтра

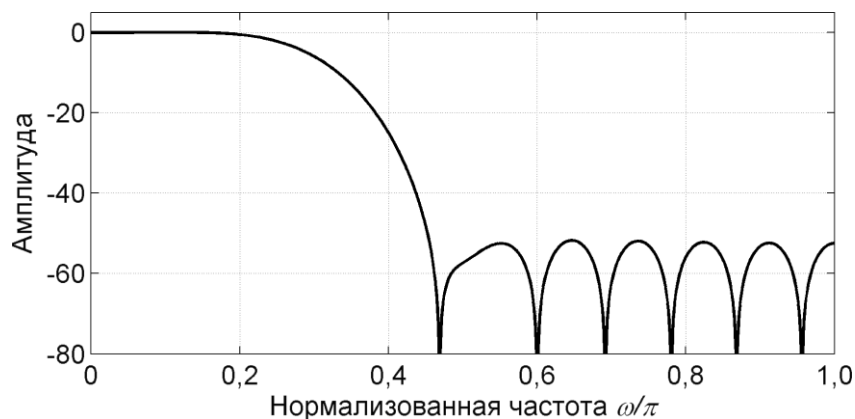


Рис. 6.14. Амплитудно-частотная характеристика в логарифмическом масштабе

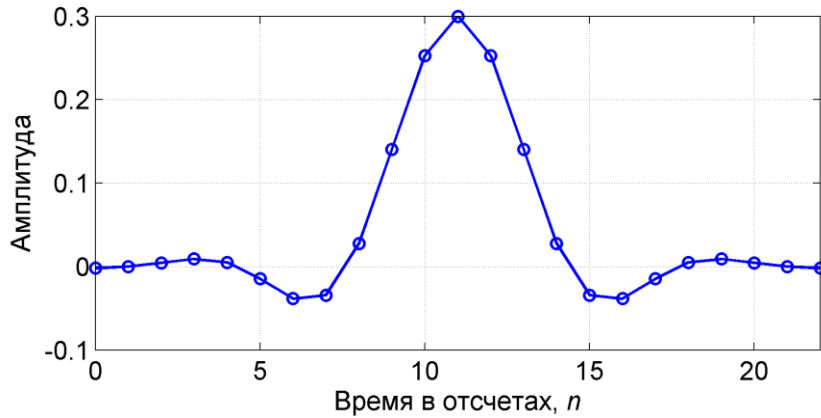


Рис. 6.15. Импульсная характеристика фильтра

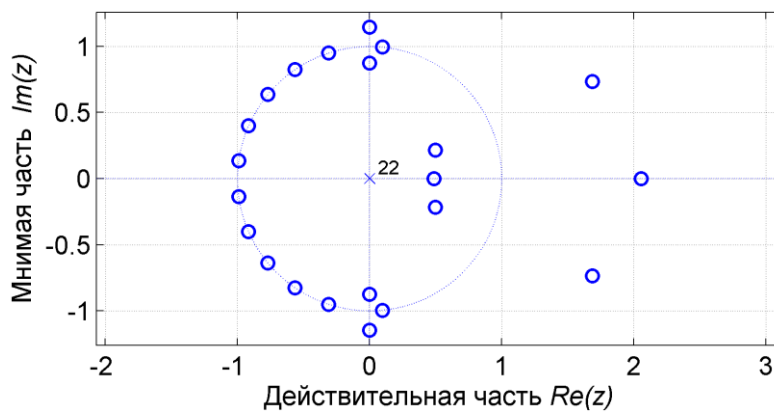


Рис. 6.16 – Диаграмма расположения нулей

Одним из главных преимуществ метода взвешивания является относительная простота расчета импульсных характеристик с минимальным числом вычислительных операций. Основной недостаток метода состоит в том, что получаемые КИХ-фильтры не удовлетворяют всем известным критериям оптимальности, и, следовательно, в большинстве случаев они могут быть существенно улучшены.

1.5 Порядок выполнения работы

6.2.1. Используя прямоугольное окно, рассчитайте КИХ-фильтр нижних частот с линейной фазой в соответствии с вариантом (табл. 6.3).

Таблица 6.3

Варианты для задания 6.2.1 (частота определена в долях от частоты Найквиста)

Номер варианта	1	2	3	4	5	6
Длина импульсной характеристики	23	25	27	29	31	27
Нормированная граничная частота пропускания	0,3	0,7	0,5	0,4	0,6	0,4

Постройте график импульсной характеристики, идеальные и действительные АЧХ (в логарифмическом масштабе) и ФЧХ, а также диаграмму расположения нулей. Для устранения скачков фазы воспользуйтесь функцией `unwrap`.

Как объяснить отклонение АЧХ и ФЧХ фильтра от идеальных? Что изменится при использовании нулевой фазы, линейной фазы, рассчитанной для длины импульсной характеристики в 11 отсчетов?

2. Выполните задание 6.2.1 для длины импульсной характеристики, большей на единицу. Что можно сказать о симметрии импульсной характеристики? Сравните поведение АЧХ двух фильтров в области границы полосы пропускания.

3. Постройте следующие временные окна и их АЧХ: треугольное, Хэмминга, Блэкмана. Используйте функции `triang`, `hamming`, `blackman`.

4. Выполните задание 6.2.1 с использованием временных окон: треугольного, Хэмминга, Блэкмана. Сравните АЧХ этих фильтров. Какой из них обеспечивает наименьшую среднеквадратичную погрешность аппроксимации АЧХ?

5. Выполните задание 1 с использованием временного окна Кайзера. Определите значение β , обеспечивающее минимальную среднеквадратичную погрешность аппроксимации АЧХ.

6. Используя окно Хэмминга, рассчитайте следующий многополосный КИХ-фильтр с линейной фазой, используя данные табл. 6.4.

Таблица 6.4

Варианты для задания 6.2.1 (частота определена в долях от частоты Найквиста)

Номер варианта	1	2	3	4	5	6
Длина импульсной характеристики	61	53	59	51	57	55
Первая полоса пропускания	0...0,2	0,3...0,4	0...0,3	0,4...0,6	0...0,25	0,5...0,6
Вторая полоса пропускания	0,3...0,4	0,8...1,0	0,7...0,85	0,9...1,0	0,4...0,6	0,75...1,0

Коэффициент передачи в первой полосе пропускания равен единице, а во второй полосе – 0,5. Постройте идеальную и полученную АЧХ.

ЛИТЕРАТУРА

1. Оппенгейм, А. Цифровая обработка сигналов / А. Оппенгейм, Р. Шафер.– М.: Техносфера, 2006. – 853 с.
2. Лайонс, Р. Цифровая обработка сигналов; пер. с англ. – 2-е изд. / Р. Лайонс – М.: ООО «Бином-Пресс», 2006. – 656 с.
3. Рабинер, Л. Теория и применение цифровой обработки сигналов / Л. Рабинер, Б. Гоулд. – М.: Мир, 1978. – 848 с.
4. Сергиенко, А.Б. Цифровая обработка сигналов / А. Б. Сергиенко. – СПб. : Питер, 2002. – 608 с.
5. Солонина, А.И. Алгоритмы и процессоры цифровой обработки сигналов / А. И. Солонина, Д.А. Улахович, Л.А. Яковлев. – СПб. : БХВ-Петербург, 2002. – 464 с.
6. Вашкевич, М.И. Косинусно-модулированные банки фильтров с фазовым преобразованием: реализация и применение в слуховых аппаратах / М. И. Вашкевич, И.С. Азаров, А.А. Петровский – М. : Горячая линия-Телеком, 2014. – 210 с.
7. Гольденберг, Л.М., Цифровая обработка сигналов: учеб. пособие для вузов/ Л. М. Гольденберг, В. Д. Матюшкин, М. Н. Поляк – М.: Радио и связь, 1990. – 315 с.
8. Каппелини, В. Цифровые фильтры и их применение / В. Каппелини, А. Константинидис, П. Эмилиани.– М.: Радио и связь, 1983. – 350 с.
9. Хемминг, Р. В. Цифровые фильтры. / Р. В. Хемминг – М.: Сов. Радио, 1980. – 224 с.
10. Голд, Б. Цифровая обработка сигналов / Б. Голд, Ч. Рэйдер.– М.: Сов. радио, 1973. – 368 с.

Учебное издание

Петровский Александр Александрович
Вашкевич Максим Иосифович
Азаров Илья Сергеевич

ТЕОРИЯ И ПРИМЕНЕНИЕ ЦИФРОВОЙ ОБРАБОТКИ СИГНАЛОВ
ЛАБОРАТОРНЫЙ ПРАКТИКУМ
ПОСОБИЕ

Редактор *М. А. Зайцева*
Корректор *Е. Н. Батурчик*
Компьютерная правка, оригинал-макет

Подписано в печать . Формат 60x84 1/16. Бумага офсетная. Гарнитура «Таймс».
Отпечатано на ризографе. Усл. печ. л. . Уч-изд. л. . Тираж 70 экз. Заказ 247.

Издатель и полиграфическое исполнение: учреждение образования
«Белорусский государственный университет информатики и радиоэлектроники».
220013, Минск, П. Бровки, 6