

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ

РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное бюджетное образовательное учреждение высшего
образования

РЯЗАНСКИЙ ГОСУДАРСТВЕННЫЙ РАДИОТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ ИМЕНИ В.Ф.
УТКИНА

Кафедра радиоуправления и связи

МЕТОДИЧЕСКИЕ МАТЕРИАЛЫ

по дисциплине (модулю)

«Помехоустойчивые системы передачи информации»

Направление подготовки

11.05.01 «Радиоэлектронные системы и комплексы»

Направленность (профиль) подготовки

Радиосистемы и комплексы управления

Уровень подготовки специалитет

Программа подготовки специалитет

Квалификация выпускника – инженер Форма обучения – очная

Рязань 2025

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ
РЯЗАНСКИЙ ГОСУДАРСТВЕННЫЙ РАДИОТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ
им. В.Ф. УТКИНА

**ПРОЕКТИРОВАНИЕ И МОДЕЛИРОВАНИЕ
РАДИОСИСТЕМ ПЕРЕДАЧИ
ИНФОРМАЦИИ**

Методические указания
к лабораторным работам

Рязань 2019

Проектирование и моделирование радиосистем передачи информации: методические указания к лабораторным работам / Рязан. гос. радиотехн. ун-т; сост. П. С. Покровский. – Рязань, 2019 – 44 с.

Изложены основные элементы теории по дисциплине «Проектирование и моделирование радиосистем передачи информации». Даны указания для подготовки и выполнения лабораторных работ с использованием среды GNU Octave.

Предназначены для студентов 5-го курса факультета радиотехники и телекоммуникаций, обучающихся по направлению «11.05.01. Радиоэлектронные системы и комплексы».

Ил. 20. Библиогр.: 6 назв.

Моделирование радиосистем, радиосигнал, модуляция, передача информации

Печатается по решению редакционно-издательского совета Рязанского государственного радиотехнического университета.

Рецензент: кафедра радиоуправления и связи Рязанского государственного радиотехнического университета (зав. кафедрой д-р техн. наук, проф. С. Н. Кириллов)

Применение среды GNU Octave для моделирования радиосистем передачи информации

Цель работы: изучение основ формирования и обработки сигналов в среде GNU Octave.

Общие сведения о GNU Octave

Условно типы данных языка программирования среды GNU Octave можно разделить на следующие классы:

- числовые (double, complex, int8, ..., int64, uint8, ..., uint64);
- логический (logical);
- текстовые (char);
- записи (struct);
- ячейки (cell);
- указатели на графические объекты и файлы (им в соответствие ставятся скаляры с типом double).

Информацию о типе данных той или иной переменной можно посмотреть во вкладке «Область переменных».

В рамках настоящего курса при моделировании радиосистем передачи информации в основном рассматриваются числовые типы данных. Отличительной особенностью этого языка является его ориентированность на работу не со скалярными величинами и массивами, а с векторами и матрицами. С учетом этой специфики все арифметические операции определяются именно для операций с подобными многомерными наборами данных.

Основными числовыми типами данных в рамках данного курса являются типы вещественных векторов и матриц (double) и комплексных векторов и матриц (complex).

Далее показаны примеры создания некоторой скалярной переменной «a» вещественного типа:

```
>> a = 1; % Создание переменной a типа double и присвоение ей значения 1.0 без эхо-печати выполненной операции в командном окне
>> a = 1 % Создание переменной a типа double и присвоение ей значения 1.0 с эхо-печатью выполненной операции в командном окне
a = 1
```

Здесь и далее сочетание символов «>>» иллюстрирует начало очередной команды среды GNU Octave, генерируемое автоматически графическим интерфейсом, поэтому это сочетание не является частью исполняемого кода и не должно вводиться оператором.

Символ «точка с запятой» в конце предложения языка GNU Octave запрещает эхо-печать результата выполнения операции, после которой он идет.

Для реализации комплексных чисел в рассматриваемом языке зарезервированы имена «i» и «j». Т.е. создание некоторой скалярной переменной «a» комплексного типа можно записать в виде:

>> a = 1+1.1i % Создание переменной a типа complex и присвоение ей значения 1.0 + 1.1j с эхо-печатью выполненной операции в командном окне

a = 1 + 1.1i

>> a = 1+1.1j % Создание переменной a типа complex и присвоение ей значения 1.0 + 1.1j с эхо-печатью выполненной операции в командном окне

a = 1 + 1.1i

Для создания массивов, векторов и матриц используются конструкции с квадратными скобками. При этом используются символы запятой или пробела для отделения элементов разных столбцов, а точки с запятой – для разделения строк в формируемых массивах:

<имя_переменной> = [<выражение 1>, <выражение 2>; <выражение 3>, <выражение 4>];

или

<имя_переменной> = [<выражение 1> <выражение 2>; <выражение 3> <выражение 4>];

В результате выполнения подобной конструкции формируется массив <имя_переменной> вида:

<выражение 1>	<выражение 2>
<выражение 3>	<выражение 4>

Для изменения отдельных элементов массивов, векторов или матриц существует схема обращения по индексу, описываемая в виде:

<имя_переменной>(<номер_строки>, <номер_столбца>)

Данную конструкцию можно применять непосредственно для обращения, т.е. для считывания некоторого элемента из этого набора данных, и для изменения значения этого элемента, тогда требуется выполнить следующую команду:

<имя_переменной>(<номер_строки>, <номер_столбца>) = <новое_значение>

При этом если данное предложение не будет окончено символом точки с запятой, то в процессе эхо-печати будет выполнен вывод в

командное окно всего массива элементов с соответствующим именем переменной, а не одного изменяемого элемента.

Кроме того, в рамках индексации массивов могут применяться векторы, элементы которых являются целыми положительными числами, не превышающими пределов индексируемых размерностей. Данный вид индексации описывается конструкцией:

<имя_переменной>(<вектор_номеров_строк>, <вектор_номеров_столбцов>)

При подобном обращении возвращается массив с количеством строк, соответствующим длине вектора номеров строк, и с количеством столбцов, соответствующим длине вектора номеров столбцов. При этом может использоваться служебное слово «end», обозначающее индекс последнего элемента рассматриваемого массива по заданной размерности. Кроме того, для считывания всех элементов той или иной размерности может использоваться знак двоеточия. Например, для считывания последней строки матрицы «a» и присвоения ее значений вектору «b» может использоваться следующая конструкция:

>> b = a(end, :);

Кроме того, для индексации могут применяться логические матрицы той же размерности, что и индексируемые матрицы. При этом будет возвращен массив элементов индексируемой матрицы, где в логической матрице стоят значения «1», что соответствует логическому значению «истинно».

Конструкции с использованием квадратных скобок для создания массивов, векторов и матриц могут применяться не только над скалярными величинами, но и над матрицами и векторами. Например, допустимо создание матрицы «a» размером 3 на 4 из вектора-строки размером 1 на 4 и двух матриц «b» и «d» размером 2 на 2 согласно следующей записи:

>> a = [c; b, d]

В результате будет создана матрица «a» с элементами:

c(1,1)	c(1,2)	c(1,3)	c(1,4)
b(1,1)	b(1,2)	d(1,1)	d(1,2)
b(2,1)	b(2,2)	d(2,1)	d(2,2)

Для определения размеров массивов, столбцов и матриц используется функция size, синтаксис которой имеет вид
[<количество_строк>, <количество_столбцов>] = size(<имя_переменной>);

При работе с векторами удобнее пользоваться функцией `length`. Данная функция позволяет определить длину массива, т.е. возвращает количество элементов по той размерности, которая имеет их максимальное количество. Таким образом, при выполнении этой функции над матрицей размером 3 на 4 будет возвращено значение 4 [т.к. `max(3, 4) = 4`]. Синтаксис функции `length`:

[т.к. `max(3, 4) = 4`]. Синтаксис функции `length`:

`<длина массива> = length(<имя переменной>);`

В языке GNU Octave существуют встроенные методы формирования стандартных наборов (массивов) величин:

`ones(m, n)` – возвращает массив, состоящий из единиц размером `m` на `n`;

`zeros(m, n)` – возвращает массив, состоящий из нулей размером `m` на `n`;

`rand(m, n)` – возвращает массив, состоящий из случайных величин с нормальным законом распределения (с нулевым математическим ожиданием и дисперсией, равной 1) размером `m` на `n`.

Также в GNU Octave имеется встроенный механизм формирования регулярных векторов-строк, имеющий следующий синтаксис:

`<начальное_значение>:<приращение>:<конечное_значение>`
В качестве примера представлен программный код, формирующий с использованием этого механизма сетку времени от 0 до 10 секунд с дискретом 5 мкс:

```
>> t = [0:5e-6:10];
```

Арифметические операции, определенные в GNU Octave над векторами и матрицами:

- + сложение;
- вычитание;
- * умножение;
- / деление слева направо;
- \ деление справа налево;
- ^ возведение в степень.

Эти операции определены над векторами и матрицами, поэтому выражение `a * b` соответствует выполнению матричного умножения, т.е. при умножении вектора-строки «a» длиной `L` на вектор-столбец «b» той же длиной результат будет равен следующей сумме:

$$a * b = \sum_{i=1}^L a(i,1)b(i,1).$$

В GNU Octave для выполнения поэлементного умножения, деления и возведения в степень элементов массивов введены дополнительные арифметические операции:

- * поэлементное умножение;
- ./ поэлементное деление слева направо;
- .\ поэлементное деление справа налево;
- .^ поэлементное возведение в степень.

При использовании поэлементных операций необходимо гарантировать соответствие размерностей используемых в выражениях операндов. Т.е. вектор-строку длиной «L» можно умножить только на вектор-строку той же длины.

При моделировании радиосистем передачи информации наиболее часто для формирования сигналов используются описанные ниже стандартные функции языка GNU Octave.

Функция `sign` возвращает знак аргумента, т.е.

$$\text{sign}(x) = \begin{cases} 1 & \text{при } x > 0, \\ -1 & \text{при } x < 0, \\ 0 & \text{при } x = 0. \end{cases}$$

Данная функция может применяться к массивам. В этом случае функция `sinc` возвращает матрицу того же размера, что и исходная, т.е.

$$\text{sign}(A) = \begin{bmatrix} \text{sign}(A(1,1)) & \dots & \text{sign}(A(1,\text{end})) \\ \vdots & \dots & \vdots \\ \text{sign}(A(\text{end},1)) & \dots & \text{sign}(A(\text{end},\text{end})) \end{bmatrix}.$$

Синтаксис: `<результат> = sign (<выражение>)`

Пример:

```
>> sign(0.15678)
```

```
ans = 1
```

Функция `sin` вычисляет синус аргумента, представленный в радианах. Данная функция аналогично функции `sign` может применяться к массивам.

Синтаксис: `<результат> = sin(<выражение>)`

Пример:

```
>> sin(pi) % pi есть зарезервированная в GNU Octave константа, равная 3.141...
```

```
ans = 0
```

Функция `sqrt` возвращает корень квадратный из аргумента. Данная функция аналогично функции `sign` может применяться к массивам.

Синтаксис: `<результат> = sqrt(<выражение>)`

Пример:


```
>> sqrt(4)
ans = 2
```

Функция `cos` вычисляет косинус аргумента, представленный в радианах. Данная функция аналогично функции `sign` может применяться к массивам.

Синтаксис: `<результат> = cos(<выражение>)`

Пример:

```
>> cos(pi)
ans = -1
```

Функция `exp` возвращает результат возведения константы $e = 2,718...$ в степень аргумента, т.е.

$$\exp(x) = e^x.$$

Данная функция аналогично функции `sign` может применяться к массивам.

Синтаксис: `<результат> = exp(<выражение>)`

Пример:

```
>> exp(0)
ans = 1
```

Функции `fft` и `ifft` позволяют выполнять соответственно прямое и обратное быстрое преобразование Фурье над входным вектором. Вызовы этих функций производятся следующим образом:

```
<результат> = fft(<вектор>)
```

```
<результат> = ifft(<вектор>)
```

В общем случае `<вектор>` и `<результат>` – это векторы одинаковой размерности, состоящие из комплексных чисел.

Пример:

```
>> s = [ones(1,64), zeros(1,12-64)];
```

```
>> sf = fft(s);
```

```
>> sr = ifft(sf);
```

Функции GNU Octave для визуализации результатов моделирования

Для визуализации результатов моделирования в GNU Octave формируются графические объекты, имеющие строгую иерархическую структуру. На практике зачастую используются три класса (в соответствии с уровнем иерархии): `figure` (рисунки), `axes` (координатная сетка), `line` (кривая зависимости, непосредственно график). При этом каждый объект класса `figure`, представляющий собой отдельное окно операционной системы, может содержать один или несколько объектов класса `axes`. В свою очередь в каждом объекте `axes` может быть несколько объектов `line`, т.е. на каждой координатной сетке может быть построено несколько независимо зависимостей. Пример

сформированных объектов указанных трех классов представлен на рис. 1.

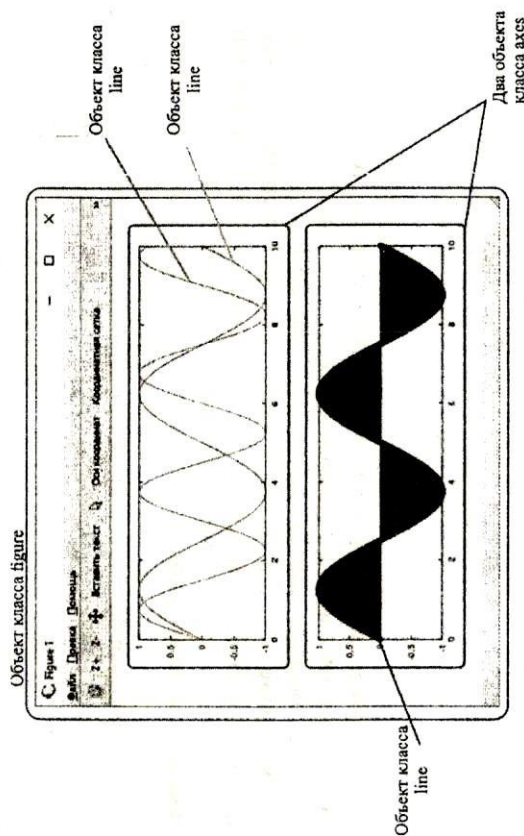


Рис. 1. Графические объекты GNU Octave

С учетом представленной иерархии функции работы с графическими объектами в GNU Octave можно условно поделить на три типа:

- 1) создание и управление объектами класса `figure` (формирование и переклечение между соответствующими графическими окнами ОС);
- 2) создание и управление объектами класса `axes` (формирование и переклечение между координатными сетками в рамках одного графического окна ОС);
- 3) построение непосредственно зависимостей.

Функции первых двух типов являются необязательными, так как при выполнении любой функции третьего типа в случае отсутствия необходимых объектов классов `figure` и `axes` данные объекты будут созданы автоматически.

В этой связи сначала рассмотрим функции третьего типа, наиболее часто употребляемые при моделировании радиосистем передачи информации.

Функция `plot` строит на текущем объекте класса `axes` (при необходимости создаются новые объекты классов `figure` и `axes`) зависимость, представленную в виде двух векторов значений,

соответственно откладываемых по осям абсцисс и ординат. При этом возможно несколько вариантов синтаксиса вызова этой функции:

```
plot(<абсциссы>, <ординаты>)
plot(<ординаты>)
plot(<абсциссы>, <ординаты>, <спецификация линии>)
plot(<абсциссы1>, <ординаты1>, <абсциссы2>, <ординаты2>)
<массив line-объектов> = plot(...)
```

Вариант `plot(<абсциссы>, <ординаты>)` подразумевает наличие двух векторов (неважно -- вектор-столбец или вектор-строка) одинаковой длины `<абсциссы>` и `<ординаты>`. Функция `plot` на текущей координатной сетке сначала строит точку с координатами `(<абсциссы(1)>, <ординаты(1)>)`. Далее ставит вторую точку с координатами `(<абсциссы(2)>, <ординаты(2)>)` и соединяет эти точки отрезком. Этот процесс повторяется, пока не будут перебраны все элементы входных векторов.

Пример:

```
>> t = 0:0.01:10;
>> y = sin(2*pi*t/5);
>> plot(t, y)
```

Вариант `plot(<ординаты>)` аналогичен первому варианту при следующем выборе вектора `<абсциссы>`: `<абсциссы> = 1:length(<ординаты>)`.

Вариант `plot(<абсциссы>, <ординаты>, <спецификация линии>)` позволяет указать цвет, тип маркера и тип линии, задаваемые строкой `<спецификация линии>`. Правила формирования данного параметра:

`<спецификация линии> = 'b1b2b3'`

b ₁	Цвет	b ₂	Маркер	b ₃	Линия
y	жёлтый	+	+	-	сплошная (по умолчанию)
m	розовый	o	o	--	штриховая
c	голубой	*	*	:	пунктирная
r	красный	.	.	-.	штрих-пунктирная
g	зелёный	x	x		
b	синий	s	□		
w	белый	d	◇		

Пример:

```
...
>> plot(t, y, 'ro:')
```

Вариант `plot(<абсциссы1>, <ординаты1>, <абсциссы2>, <ординаты2>)` позволяет сформировать на одной координатной сетке сразу две (или более) зависимости, соответствующие своим наборам значений абсцисс и ординат. При этом для каждой зависимости можно указывать параметр `<спецификация линии>`.

Пример:

```
...
>> y1 = sin(2*pi*t/3);
>> plot(t, y, 'o', t, y1, '+')
```

При этом первая зависимость будет построена с маркером o-типа, а вторая -- с маркером +-типа.

Функция `stem` строит в виде дискретной последовательности на текущем объекте класса `axes` (при необходимости создаются новые объекты классов `figure` и `axes`) зависимость, представленную в виде двух векторов значений, соответственно откладываемых по осям абсцисс и ординат. В отличие от функции `plot` функция `stem` не соединяет соседние точки, а каждой точке ставит в соответствие маркер и отрезок, соединяющий данную точку с осью абсцисс. При этом возможно несколько вариантов синтаксиса вызова данной функции:

```
stem(<абсциссы>, <ординаты>)
stem(<ординаты>)
stem(<абсциссы>, <ординаты>, <спецификация линии>)
```

Варианты вызова данной функции аналогичны синтаксису функции `plot`. Исключением является отсутствие возможности одновременно сформировать несколько зависимостей на одной координатной сетке.

Примеры:

```
>> t = 0:0.5:10;
>> y = sin(2*pi*t/10);
>> stem(t, y);
>> stem(y);
>> stem(t, y, 'ro--');
```

Вызов функции `plot` или `stem` выводит зависимость в активную координатную сетку (объект класса `axes`), которая в свою очередь принадлежит активному окну (объект класса `figure`). При этом происходит стирание уже нарисованных зависимостей на данной координатной сетке зависимостей.

Для возможности одновременной работы с несколькими зависимостями в GNU Octave имеется функция `figure`, которая позволяет сформировать новый объект класса `figure` и/или осуществлять переключение активного окна. Варианты вызова данной функции:

figure(
figure(<номер_figure-объекта>)

Вызов figure() создает новый объект класса figure (т.е. открывается новое окно ОС с названием Figure <очередной_номер>) и делает его активным.

Вызов figure(<номер>) делает активным объект класса figure с порядковым номером <номер> (окно ОС с названием Figure <номер>), при отсутствии соответствующего объекта данный объект автоматически создается. Аргумент <номер> должен принадлежать множеству натуральных чисел.

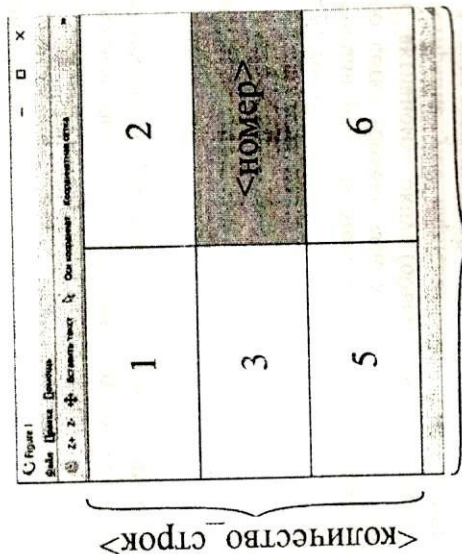
Пример:

```
>> figure(2);
```

Для создания нескольких объектов axes в рамках активного объекта класса figure используется функция subplot. Синтаксис использования функции:

```
subplot(<количество_строк>, <количество_столбцов>, <номер>);
```

Вызов данной функции условно делит активное окно на <количество_строк> строк и <количество_столбцов> столбцов и в рамках данной таблицы создает координатную сетку в позиции, соответствующей номеру <номер> (см. рис. 2).



<количество_столбцов>

Рис. 2. Иллюстрация к описанию функции subplot

Пример:

```
>> t = 0:0.5:10;
```

```
>> y = sin(2*pi*t/10);
```

```
>> subplot(2,1,1); plot(t, y);  
>> subplot(2,1,2); plot(t, -y);
```

Функции GNU Octave для статистической обработки результатов моделирования

Функция sum возвращает вектор-строку значений сумм по столбцам элементов входной матрицы. Т.е. входной массив разбивается на столбцы, для каждого столбца вычисляется сумма элементов, полученные значения образуют вектор-строку. Синтаксис:

```
<результат_размером_1xN> = sum(<массив_размером_MxN>)
```

Пример:

```
>> a = ones(3,4);
```

```
>> b = sum(a)
```

```
b =
```

```
3 3 3 3
```

Функция mean возвращает вектор-строку средних по столбцам значений элементов входной матрицы. Т.е. входной массив разбивается на столбцы, для каждого столбца вычисляется среднее значение, полученные значения образуют вектор-строку. Синтаксис:

```
<результат_размером_1xN> = mean(<массив_размером_MxN>)
```

Пример:

```
>> a = ones(3,4);
```

```
>> b = mean(a)
```

```
b =
```

```
1 1 1 1
```

Функция std возвращает вектор-строку среднеквадратических значений по столбцам элементов входной матрицы. Т.е. входной массив разбивается на столбцы, для каждого столбца вычисляется среднеквадратическое значение, полученные значения образуют вектор-строку. Расчет среднеквадратического значения производится по формуле

$$std(x) = \sqrt{(M-1)^{-1} \sum_{i=1}^M \left(x_i - M^{-1} \sum_{i=1}^M x_i \right)^2}$$

Синтаксис:

```
<результат_размером_1xN> = std(<массив_размером_MxN>)
```

```
>> a = randn(1e6,4);
```

```
>> b = std(a)
```

```
b =
```

```
1.03 1.03 1.01 0.96
```

Функция var возвращает вектор-строку дисперсий столбцов входной матрицы. Т.е. входной массив разбивается на столбцы, для

каждого столбца вычисляется дисперсия, полученные значения образуют вектор-строку. Расчет дисперсии выполняется по формуле

$$\text{var}(\mathbf{x}) = (M - 1)^{-1} \sum_{i=1}^M \left(x_i - M^{-1} \sum_{i=1}^M x_i \right)^2$$

Синтаксис:

<результат_размером_1xN> = std(<массив_размером_MxN>)

>> a = randn(1e6,4);

>> b = var(a)

b = 1.00 1.00 1.00 1.00

Функция **hist** производит расчет гистограммы распределения (т.е. оценки функции плотности вероятности) элементов входного вектора. Имеется несколько способов вызова функции:

hist(<вектор>)

hist(<вектор>, <число_интервалов>)

[<вектор_значений>, <вектор_центров_интервалов>] = hist(...)

Вызов hist(<вектор>) строит в активном объекте класса axes гистограмму элементов <вектор> для случая разделения всего динамического диапазона на 10 одинаковых интервалов.

Вызов hist(<вектор>, <число_интервалов>) аналогичен выше рассмотренному за исключением возможности задать посредством параметра <число_интервалов> общее количество интервалов, на которые разбивается динамический диапазон значений элементов <вектор>.

Пример:

>> a = randn(1e6,1);

>> hist(a, 20);

Вызов [<вектор_значений>, <вектор_центров_интервалов>] = hist(...) позволяет произвести разделение динамического диапазона элементов <вектор> на интервалы, центры которых возвращаются в выходном параметре <вектор_центров_интервалов>, а также подсчитать число (возвращается в выходном параметре <вектор_значений>) элементов <вектор>, попавших в каждый из этих интервалов. При этом не производится создание графических объектов, т.е. гистограмма в графическом виде не выводится, а возвращается в виде массивов <вектор_значений>, <вектор_центров_интервалов>.

Пример:

>> a = randn(1e6,1);

>> [nn,xx] = hist(a, 20);

>> bar(xx, nn, 1)

Функция **pwelch** производит оценку спектральной плотности мощности (СПМ) входного вектора методом Уэлча. По умолчанию входной вектор разбивается на перекрывающиеся (50 % перекрытия) блоки длиной, равной квадратному корню из длины входного вектора, далее каждый блок умножается на окно Хэмминга соответствующей длины, после чего производится выборочная оценка спектра посредством быстрого преобразования Фурье (БПФ). Затем выборочные оценки усредняются и выводятся в виде зависимости спектральной плотности мощности от частоты, нормированной к частоте дискретизации. Возможно несколько вызовов функции:

pwelch(<вектор>)

pwelch(<вектор>, <длина_окна>, <смещение_окна>, <число_точек_БПФ>, <частота_дискретизации>, <тип_графика>)

[<СПМ>, <вектор_частот>] = pwelch(<вектор>, <длина_окна>, <смещение_окна>, <число_точек_БПФ>, <частота_дискретизации>, <тип_графика>)

Первый способ вызова производит оценку СПМ со стандартными настройками и выводит в активный объект класса axes в виде зависимости, выраженной в размах, от нормированной частоты.

Второй способ вызова позволяет задать длину окна (через скалярный параметр <длина_окна>), смещение окна (скалярный параметр <смещение_окна>, показывающий долю перекрытия относительно длины окна), выбрать количество точек производимого оконного преобразования Фурье (через параметр <число_точек_БПФ>), указать частоту дискретизации, выраженную в Гц (параметр <частота_дискретизации>), а также задать тип выводимого графика (строка <тип_графика>). Наболее часто употребляемыми значениями параметра <тип_графика> являются 'plot' (выводит СПМ, выраженную в стандартном виде) и 'db' (выводит СПМ, выраженную в дБ).

Третий способ вызова аналогичен второму, но позволяет получить СПМ не в виде графической зависимости, а в виде вектора-столбца значений <СПМ> и соответствующих значений частот <вектор_частот>.

Перед первым запуском в текущей сессии работы в среде GNU Octave функции pwelch необходимо подключить пакет работы с сигналами, для чего выполнить следующую команду:

>> pkg load signal

Пример:

>> pkg load signal

>> Fs = 100;

>> t = [0:(1/Fs):10];


```
>> a = sin(2*pi*t/3);
>> rwhch(a, [], [], Fs, 'db');
```

Использование здесь в качестве аргументов пустых массивов [] сигнализирует функции rwhch о необходимости в качестве соответствующих параметров подставлять значения, установленные по умолчанию.

На допуск

Изучить основные конструкции языка программирования, применяемые в среде GNU Octave: типы данных, оператор присвоения, арифметические поэлементные операторы языка, условные операторы, операторы работы с векторами и матрицами, арифметические операторы языка для работы с векторами и матрицами, функции формирования стандартных массивов данных, основные математические функции (в частности, для построения гистограмм, преобразования Фурье, определения знака операнда, взятия модуля и т.п.), функции построения графиков (plot и stem) и управления свойствами этих графиков.

Порядок выполнения работы

1. Сформировать типовые сигналы для интервала времени от 0 до 10 секунд с дискретом 1 мс:

- дельта-функции $\delta(t)$, $\delta(t - 0,5)$ и $\delta(t - 1,5)$;

- гармонические колебания: $A \cos(2\pi ft + \varphi)$ (амплитуда принимает значения 0,1; 1,5 и 10, частота плюс 1,5 Гц, а фаза плюс 30° и 45°)

- комплексную гармонику: $A \exp(j(2\pi ft + \varphi))$ (амплитуда принимает значение 1, частота минус 10 Гц, а фаза плюс 0,35);

- «белый» гауссовский шум со спектральной плотностью мощности 1 Вт/Гц;

- линейно-кусочную функцию согласно формуле:

$$y(t) = \begin{cases} 0, & t < 1 \\ 1, & t \in [1; 2,5] \\ (t - 1,5), & t \in (2,5; 5] \\ 3,5, & t > 5. \end{cases}$$

2. Построить графики этих сигналов на отдельных рисунках и на одном.

3. Построить графики спектральной плотности.

4. Сформировать массив случайных величин длиной 100000 элементов с нормальным законом распределения $m = \langle \text{номер_бригады} \rangle$

и $\sigma = \langle \text{количество_бригад} \rangle - \langle \text{номер_бригады} \rangle$. Построить его гистограмму. Сделать вывод.

5. Сформировать массив случайных величин длиной 100000 элементов с равномерным законом распределения на отрезке от $\langle \text{номер_бригады} \rangle$ до $(\langle \text{количество_бригад} \rangle - \langle \text{номер_бригады} \rangle + 1)$. Построить его гистограмму. Сделать вывод.

6. Сформировать прямоугольный элементарный импульс длительностью 10 с. Построить временную реализацию и спектральную плотность мощности. Сделать вывод.

7. Сформировать гармонический сигнал длительностью 10 с, амплитудой 1 и с частотами $(\langle \text{номер_бригады} \rangle * 0,1)$ Гц, $(\langle \text{номер_бригады} \rangle * 0,1 + 0,15)$ Гц и $(\langle \text{номер_бригады} \rangle * 0,1 + 0,325)$ Гц. Построить временную реализацию и амплитудный спектр. Сделать вывод.

Контрольные вопросы

1. Как сформировать массив единиц заданной преподавателем размерности, как сформировать массив, состоящий из массивов единиц и нулей с размерностями, заданными преподавателем?

2. Как сформировать временную сетку с указанными преподавателем границами и частотой дискретизации?

3. Как сформировать гармонический сигнал с параметрами, заданными преподавателем?

4. Как построить гистограмму случайного процесса?

5. Как построить на одном рисунке три графика тремя способами?

6. Как включить отображение координатной сетки на графиках? Как изменить настройки линий (цвет, тип линии и тип маркера) при построении графиков?

7. Как построить спектральную плотность мощности некоторого сигнала, выраженную в абсолютных единицах и децибелах?

8. Как вычислить преобразование Фурье от некоторого сигнала? Какие есть при этом особенности и ограничения?

9. Какие в языке GNU Octave имеются арифметические операторы, как их применять на примере, указанном преподавателем?

10. Какие в языке GNU Octave имеются типы данных, как и зачем они применяются?

Лабораторная работа № 2 Исследование сигналов основной полосы

Цель работы: освоение навыков моделирования радиосистем передачи информации в основной полосе с применением низкочастотных эквивалентов, а также оценки основных параметров радиосигналов.

Элементы теории

Рассмотрим двоичную радиосистему передачи информации, т.е. систему, в которой используются сигналы с двумя уровнями. Подобные радиосистемы по сравнению с многоуровневыми М-позиционными системами характеризуются большей энергетической эффективностью при меньшей спектральной эффективности. При этом под спектральной эффективностью понимается удельная скорость передачи, т.е. количество переданных бит в секунду на один герц [2].

Упрощенная структурная схема модема двоичной радиосистемы передачи информации приведена на рис. 1. Здесь ФНЧ – фильтр нижних частот, ПФ – полосовой фильтр, СВН – схема восстановления несущей, УВВ – устройство взятия выборки, СВТЧ – схема восстановления тактовой частоты.

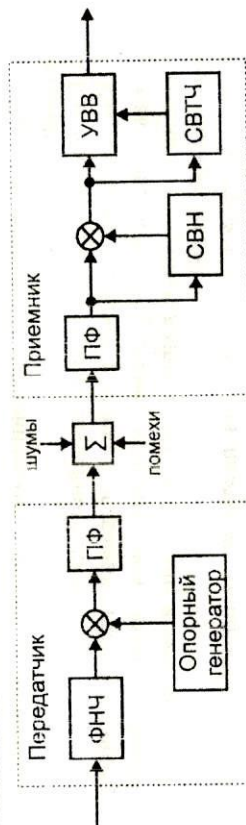


Рис. 1. Структурная схема модема двоичной радиосистемы передачи информации

В данной схеме процесс модуляции выполняется посредством перемножения сигнала опорного генератора с двоичным сигналом, ограниченным по полосе. Если основное ограничение полосы частот сигнала в основной полосе выполняется ФНЧ передатчика, то считается, что выполняется предмодуляционная фильтрация. В противном случае (полоса частот ПФ передатчика уже удвоенной частоты среза ФНЧ) имеет место постмодуляционная фильтрация. Для имитации шумов радиосистемы передачи информации зачастую используют модель линейного широкополосного канала [2]. При этом в качестве основной составляющей помех и шумов считают шум

радиочастотного усилителя приемника (на схеме не показан). С точностью, достаточной для моделирования реальных радиосистем, данный шум можно считать «белым», т.е. величина спектральной плотности мощности данного случайного процесса не зависит от рассматриваемой частоты. На приемной стороне ПФ устраняет внеполосные шумы и помехи. В блоке СВН на основе принимаемого модулированного сигнала формируется несущее колебание, которое в перемножителе приемника используется для демодуляции полезного сигнала. Демодулированный и ограниченный по полосе сигнал поступает на блок УВВ, который по внешнему такту с СВТЧ сравнивает величину принятого сигнала с заданным порогом, принимая решение относительно значения текущего переданного информационного символа. В свою очередь блок СВТЧ обеспечивает восстановление тактовой частоты из принимаемого сигнала.

Для упрощения моделирования и сокращения вычислительных затрат на практике структурную схему (см. рис. 1) заменяют более простой эквивалентной моделью, заданной в основной полосе частот, т.е. низкочастотным эквивалентом [2]. Схема данного эквивалента показана на рис. 2.

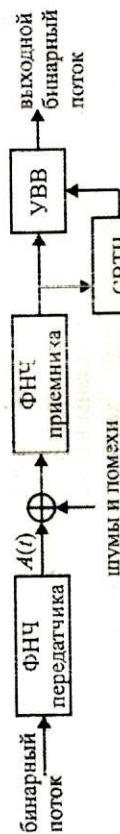


Рис. 2. Структурная схема низкочастотного эквивалента модема двоичной радиосистемы передачи информации

В этой схеме на выходе передатчика формируется сигнал основной полосы:

$$A(t) = \sum_{i=0}^{N-1} b_i u(t - iT_s),$$

где $\{b_i\}_{i=0}^{N-1}$ – последовательность двоичных символов ($b_i = \pm 1$), а $u(t)$ – элементарный импульс.

Наиболее распространенным видом сигналов основной полосы являются сигналы формата «без возврата к нулю» (известны как NRZ-сигналы). Этим сигналам соответствует прямоугольный элементарный импульс $u_{\text{rect}}(t)$ длительностью T_s (рис.3, а).

Однако подобные сигналы характеризуются значительной занимаемой полосой частот, поэтому в современных радиосистемах

передачи информации при формировании радиосигналов NRZ-сигналы подвергнутся предварительной фильтрации.

В общем случае спектральная плотность мощности сигналов основной полосы частот определяется следующим выражением:

$$G(f) = \frac{8}{T_s} p(1-p) |U(f)|^2 + T_s^{-2} |(1-2p)U(0)|^2 \delta(f) + 2T_s^{-2} \sum_{m=1}^{\infty} |(1-2p)U(m/T_s)|^2 \delta(f - m/T_s),$$

где $U(f)$ – спектр элементарного импульса, p – вероятность появления во входном двоичном потоке символов, равных +1.

Другими распространенными элементарными импульсами длительностью T_s являются следующие импульсы:

- 1) треугольный импульс:
 $u_T(t) = \text{rect}(t/(2T_s))(2t/T_s) + \text{rect}((t-T_s/2)/(2T_s))(1-2(t-T_s/2)/T_s)$;
- 2) импульс вида «синус х»:
 $u_{\sin}(t) = \sin(\pi t/T_s) \text{rect}(t/T_s)$;
- 3) импульс вида «синус квадрат х»:
 $u_{\sin^2}(t) = \sin^2(\pi t/T_s) \text{rect}(t/T_s)$.

Сигналы, соответствующие прямоугольному и этим трем типам элементарных импульсов, показаны на рис. 3.

При этом радиосигналы характеризуются наиболее узким основным лепестком спектральной плотности мощности и минимальной скоростью спада уровня боковых лепестков спектра. Импульс вида «синус квадрат х», напротив, обладает наиболее широким спектром с минимальным уровнем внеполосного излучения.

Объединить достоинства этих двух типов элементарных импульсов позволяет переход к формированию радиосигналов с использованием последовательностей «зависимых импульсов». В таких двойных сигналах возврат к нулю происходит только на отрезках времени, когда текущий и последующий информационные символы имеют разные значения, т.е. в моменты перехода через нулевой уровень. В других случаях сигнал не изменяется. Данное правило можно представить в следующем виде:

$$A(t) = \sum_{i=0}^{N-1} \left(\frac{b_i + b_{i+1}}{2} u_{\text{rect}}(t - iT_s) + \frac{b_i - b_{i+1}}{2} u_{\text{tran}}(t - iT_s) \right).$$

Здесь первое слагаемое под знаком суммы отвечает за форму сигнала при отсутствии изменения знака в рассматриваемой паре

символов, а второе – за форму сигнала при переходе от символа «+1» к символу «-1», которая задается функцией $u_{\text{tran}}(t)$.

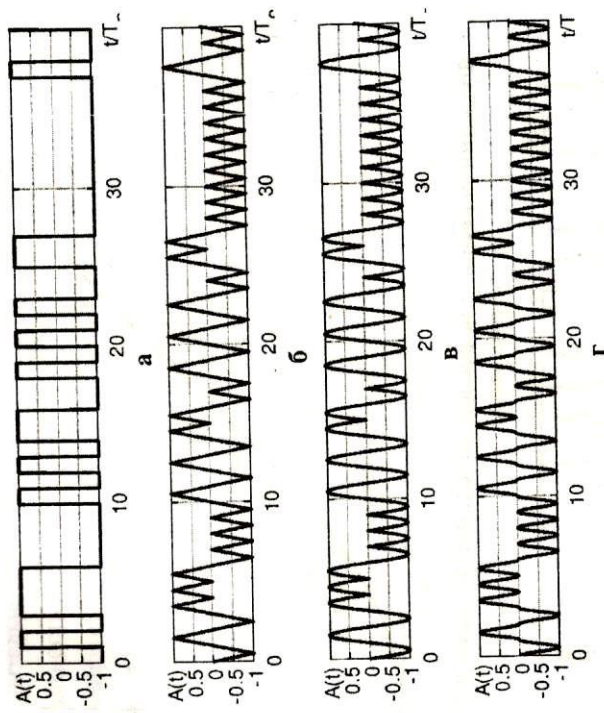


Рис. 3. Сигналы основной полосы, соответствующие прямоугольному (а), треугольному импульсам (б), а также импульсам типа «синус х» (в) и «синус квадрат х» (г)

При этом обратный переход осуществляется по инверсной функции, т.е. $(-u_{\text{tran}}(t))$. Для определения перехода между символами могут использоваться следующие функции:

$$u_{\text{lin_tr}}(t) = u_{\text{rect}}(t)(1 - 2t/T_s),$$

$$u_{\text{sin_tr}}(t) = u_{\text{rect}}(t) \cos(\pi t/T_s).$$

Данные функции показаны на рис. 4, а на рис. 5, а и рис. 5, б приведены зависимости сигналов $A(t)$, соответствующие этим функциям переходов.

Можно показать, что данные последовательности «зависимых импульсов» можно также представить в виде:

$$A(t) = \sum_{i=0}^{N-1} b_i u(t - iT_s). \quad (1)$$

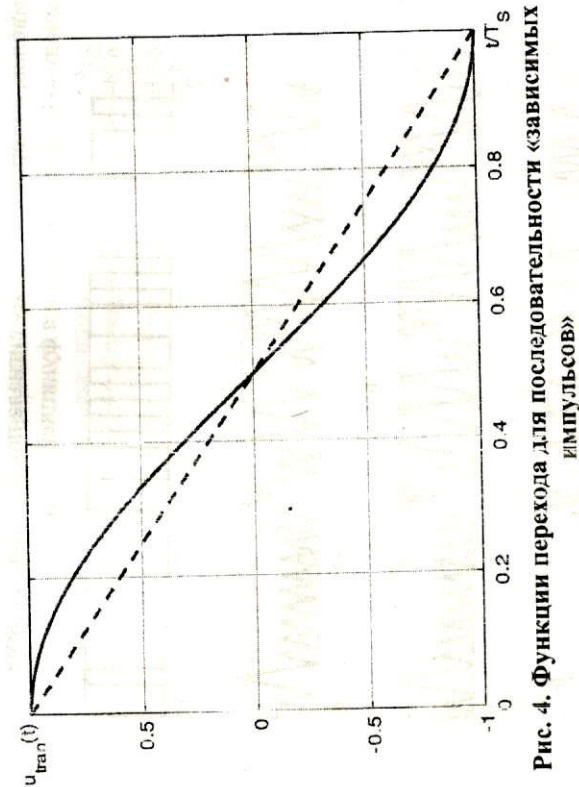


Рис. 4. Функции перехода для последовательности «зависимых импульсов»

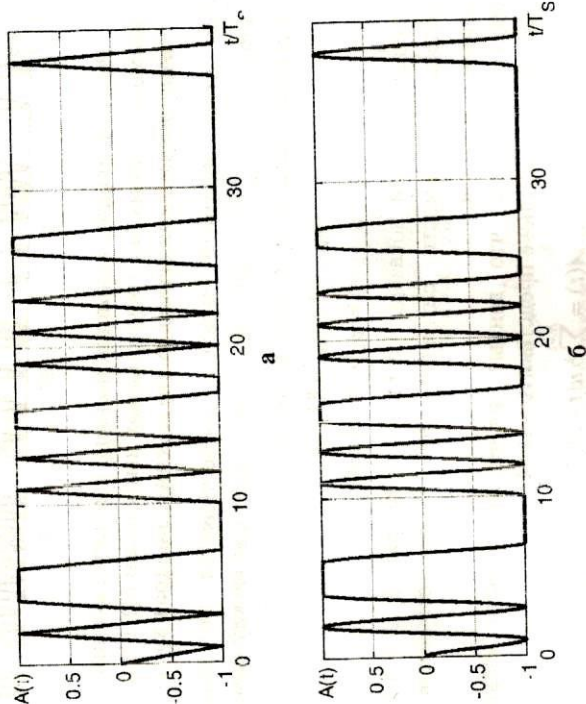


Рис. 5. Последовательности «зависимых импульсов» для рассматриваемых функций перехода

При этом элементарный импульс должен обладать следующими свойствами:

- 1) длительность $2T_s$;
- 2) четность функции $u(t + T_s)$;
- 3) выполнение равенства: $u(t) + u(t + T_s) = 1, t \in [0; T_s]$.

Так, рассмотренные последовательности «зависимых импульсов» можно получить при помощи элементарных импульсов вида:

$$u_{r(2T)}(t) = u_r(t/2);$$

$$u_{\sin(2T)}(t) = u_{\sin 2}(t/2).$$

Кроме того, на практике находят применение последовательности «зависимых импульсов», полученные с использованием формулы (1) на основе импульса:

$$u_s(t) = u_{\sin 2}(t/2) - A(u_{\sin 2}(t) + u_{\sin 2}(t - T_s)),$$

где $A \in [0; 1/2]$ — некоторый весовой множитель.

Для анализа спектральной эффективности того или иного элементарного импульса необходимо оценить занимаемую полосу частот. Обычно в качестве количественной меры данной характеристики сигнала используют полосу частот $\Delta F_{95\%}$, в которой сосредоточено 95 % излучаемой мощности радиосигнала. В рассматриваемом случае данная величина определяется из уравнения вида

$$\int_0^{0.5\Delta F_{95\%}} G(f) df = 0.95 \int_0^{+\infty} G(f) df,$$

где $G(f)$ — спектральная плотность мощности рассматриваемого сигнала.

В качестве примера на рис. 6 приведены зависимости спектральной плотности мощности от нормированной частоты. Здесь data1 соответствует прямоугольному элементарному импульсу, data2 — треугольному импульсу, data3 — импульсу вида «синус х», data4 — импульсу вида «синус квадрат х», data5 — «зависимым импульсам» с линейной функцией перехода, а data6 — «зависимым импульсам» с переходом по закону косинуса.

В общем случае занимаемая полоса частот частот обратно пропорциональна периоду следования информационных символов. В этой связи, чтобы при сравнительном анализе различных элементарных импульсов, используемых в разных радиосистемах передачи информации, исключить влияние скорости передачи, на

практике применяют относительную полосу занимаемых частот, получаемую в результате произведения $\Delta F_{90\%} T_s$.

Для предварительного анализа помехоустойчивости сигналов, формируемых с использованием того или иного элементарного импульса, на практике принято применять так называемые «глазковые диаграммы». Структурная схема анализа помехоустойчивости сигналов с учетом и без учета искажений в канале связи показана на рис. 7. При этом под искажениями в канале связи понимается воздействие полосовых фильтров, показанных на схеме рис. 1.

Примеры «глазковых диаграмм» для прямоугольного и треугольного элементарного импульса показаны на рис. 8 и рис. 9 соответственно. При этом вносимые искажения моделировались посредством фильтра нижней частоты Баттерворта 4-го порядка с частотой среза, равной $1/T_s$.

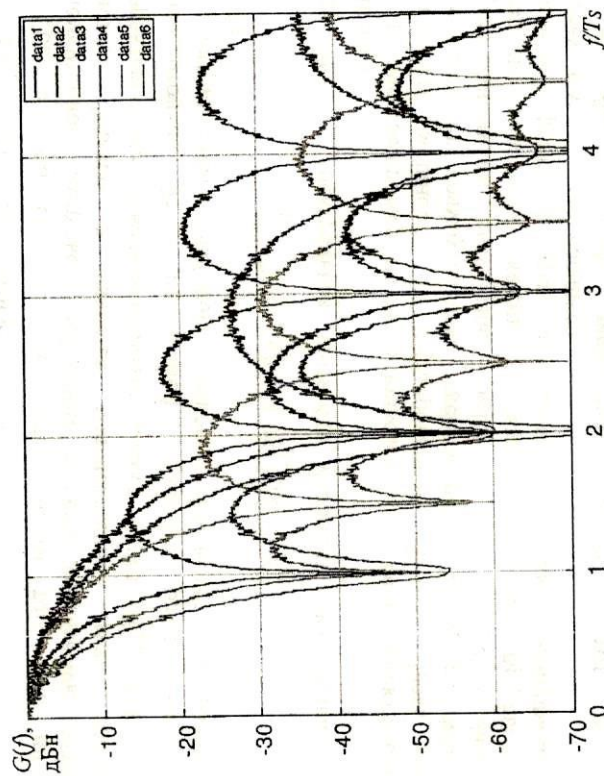


Рис. 6. Спектральная плотность мощности рассматриваемых сигналов

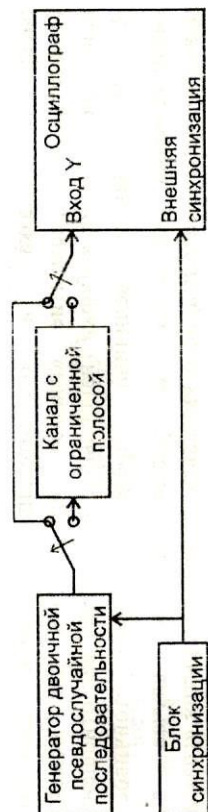


Рис. 7. Схема построения «глазковой диаграммы» для сигналов основной полосы

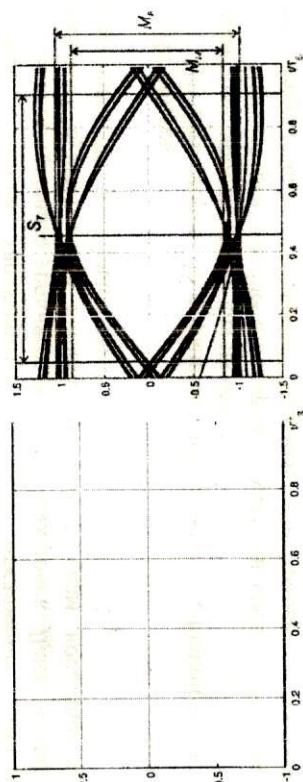


Рис. 8. «Глазковая диаграмма» для прямоугольного элементарного импульса – без искажений (а) и с искажениями (б)

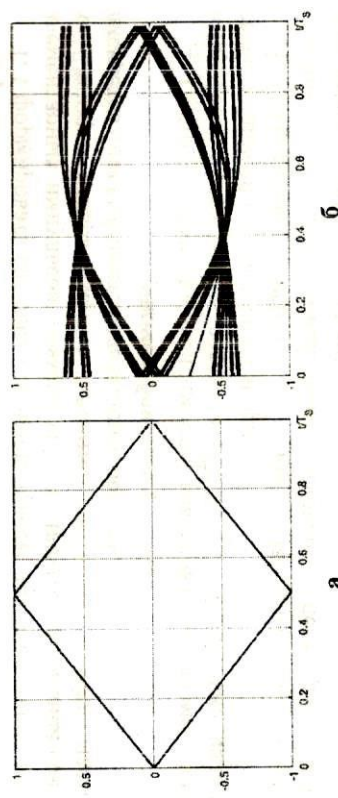


Рис. 9. «Глазковая диаграмма» для треугольного элементарного импульса – без искажений (а) и с искажениями (б)

Для анализа характеристик сигналов на основе «глазковой диаграммы» используются следующие параметры: относительное

раскрытие «глазка» и относительный уровень джиттера. Относительное раскрытие «глазка» определяется для момента наилучшего взятия выборки (наибольшее раскрытие) как отношение M_M / M_P . Данная величина определяет помехоустойчивость анализируемых сигналов. В рассматриваемом случае для прямоугольного элементарного импульса степень раскрытия «глазка» составляет порядка 85 %. Это соответствует снижению помехоустойчивости из-за действия искажений на $20 \lg(1/0,85) \approx 1,4$ дБ.

Уровень джиттера определяется как отношение $(T_s - S_T) / T_s$. Данная величина характеризует степень отклонения моментов пересечения нулевого уровня от номинальных точек. Данный эффект оказывает существенное влияние на работу схем восстановления символической (тактовой) частоты.

На допуск

1. Изучить теорию по формированию сигналов основной полосы и параметры, используемые для их оценки.
2. Изучить реализацию стандартных методов среды GNU Octave для визуализации графиков, оценки спектральной плотности мощности, вычисления интегралов, процедур фильтрации и синтеза цифровых БИХ-фильтров.

Порядок выполнения работы

1. Сформировать сигналы основной полосы для следующих видов элементарных импульсов: прямоугольного, треугольного, вида «синус х», «синус квадрат х». Количество символов 10 000, частота дискретизации выбирается следующим образом:

$$\langle \text{номер_бригады} \rangle * 32 / T_s$$
2. Построить графики зависимости реализованных элементарных импульсов, оформленные согласно ГОСТ, т.е. обязательно наличие сетки, подписей к осям абсцисс и ординат.
3. Сформировать сигналы на основе «зависимых импульсов» с линейным и косинусоидальным законами переходов между информационными символами.
4. Построить графики зависимостей реализованных сигналов для временного интервала от 0 до $20 T_s$, оформленные согласно ГОСТ. Сделать вывод о наличии различий в реализованных сигналах.

5. Построить на одном рисунке зависимости СПМ для реализованных сигналов. Сделать вывод.
6. Определить занимаемую полосу частот и пик-фактор реализованных сигналов. Сделать вывод.

7. Построить «глазковые диаграммы» для реализованных сигналов без искажений и при искажениях, соответствующих действию ФНЧ Баттерворта 4-го порядка с частотой среза $0,55 / T_s$. При построении «глазковых диаграмм» использовать сигналы, ограниченные по времени диапазоном от 0 до $100 T_s$.

8. Измерить относительное раскрытие «глазка» и уровень джиттера, сделать вывод о помехоустойчивости рассматриваемых сигналов.

9. Реализовать сигналы на основе элементарного импульса $u_s(t)$ при $A = 0$, $A = 0,2$ и $A = 0,5$.
10. По семи точкам построить зависимости $\Delta F_{95\%}(A)$ и $\Pi(A)$.

Сделать вывод.

11. Построить «глазковые диаграммы» для последовательностей «зависимых импульсов», полученных на основе $u_s(t)$. Данные диаграммы строятся для случая без искажений и при искажениях, соответствующих действию ФНЧ Баттерворта 4-го порядка с частотой среза $0,55 / T_s$.

12. Измерить относительное раскрытие «глазка» и уровень джиттера, сделать вывод о помехоустойчивости рассматриваемых сигналов.

Контрольные вопросы

1. Понятие низкочастотного эквивалента радиотехнических цепей.
2. Понятие аналитического сигнала. Способы получения аналитического сигнала на основе радиосигнала.
3. Понятие комплексной огибающей. Квадратурное представление. Использование комплексной огибающей для формирования радиосигнала.
4. Какими свойствами должна обладать некоторая функция времени, чтобы ее можно было использовать в качестве элементарного импульса?
5. Сформулируйте требования к элементарному импульсу длительностью T_s и длительностью $> T_s$.
6. Для чего используется предмодуляционный фильтр ФНЧ? Какую функцию выполняет ФНЧ на приемной стороне?

7. Для чего используется полосовой фильтр на передающей стороне? А для чего на приемной?
8. Что понимается под пик-фактором и что он характеризует?
9. Как определяется занимаемая полоса частот с использованием среды GNU Octave?

Лабораторная работа № 3

Исследование алгоритмов восстановления несущей и тактовой частоты сигналов с фазовой манипуляцией

Цель работы: изучение принципов функционирования систем синхронизации, применяемых в радиотехнике с фазоманипулированными сигналами.

Модель радиосистемы передачи информации

Рассмотрим радиосистему передачи информации, использующую сигналы с квадратурной фазовой манипуляцией (QPSK). Общая структурная схема упрощенной модели данной системы представлена на рис. 1.

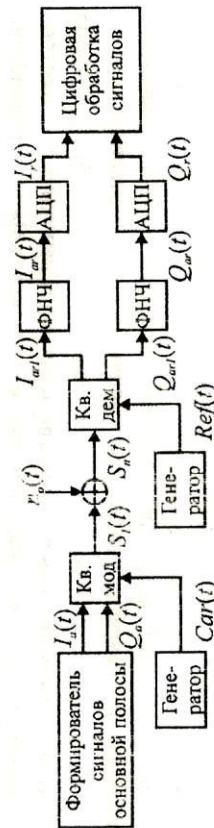


Рис. 1. Структурная схема модели рассматриваемой радиосистемы передачи информации

Здесь на передающей стороне формируются сигналы основной полосы с использованием прямоугольных элементарных импульсов с периодом следования символов T . Далее комплексная огибающая QPSK-сигнала, представленная в виде синфазной и квадратурной составляющих, переносится на несущую частоту, равную $10/T$. В этот сигнал вносится задаваемая пользователем задержка, после чего добавляется «белый» гауссовский шум, а полученная смесь поступает на квадратурный демодулятор приемной стороны. Считается, что в приемнике имеется случайная расстройка от несущего колебания, равномерно распределенная в интервале частот от

-0,5/T до 0,5/T. Кроме того, опорный генератор приемной части характеризуется задаваемой начальной фазой. Сигналы с выхода квадратурного демодулятора поступают на ФНЧ, которые подавляют нежелательные высокочастотные составляющие. В рамках рассматриваемой модели используются фильтры Баттерворта 6-го порядка с частотой среза 2,5/T. Далее синфазная и квадратурная компоненты на квази нулевой частоте подаются на модель АЦП, где посредством децимации уменьшается частота дискретизации в 10 раз. Полученные таким образом сигналы поступают на блок цифровой обработки сигналов, в котором реализованы алгоритмы восстановления несущей и тактовой частоты.

Схема восстановления несущего колебания

Рассмотрим схему восстановления несущего колебания, основанную на снятии модулирующей информационной последовательности с QPSK-сигнала посредством возведения в четвертую степень. Главная идея этого подхода заключается в том, что QPSK-сигнал на отрезке одного информационного символа равен одному из наборов сигналов:

$$a_1 = A(t) \cos(2\pi f_0 t + 45^\circ), \quad a_2 = A(t) \cos(2\pi f_0 t + 135^\circ),$$

$$a_3 = A(t) \cos(2\pi f_0 t - 45^\circ), \quad a_4 = A(t) \cos(2\pi f_0 t - 135^\circ).$$

Можно показать, что возведение в четвертую степень приводит к следующему результату:

$$a^4 = \frac{1}{4} A^4(t) \left(\frac{3}{2} + 2 \cos(2\pi(2f_0)t \pm \frac{\pi}{2}) - \frac{1}{2} \cos(2\pi(4f_0)t) \right).$$

Таким образом, в спектре данного сигнала присутствует дискретная составляющая на четвертой гармонике несущего колебания. Эту спектральную составляющую обычно выделяют при помощи системы фазовой автоподстройки частоты (ФАПЧ). Искомый сигнал несущего колебания получается при помощи деления фазы генератора, управляемого напряжением (ГУН), который входит в состав подобной схемы восстановления несущей. Схема, реализующая данный алгоритм обработки сигналов, представлена на рис. 2. Здесь ЦКД – цифровой квадратурный модулятор, осуществляющий умножение двух комплексных сигналов.

Подаваемые отсчеты комплексной огибающей, представленные синфазной и квадратурной компонентами, вводятся в четвертую степень. Далее полученный сигнал подается на ЦКД, синфазный выход которого используется в качестве сигнала ошибки схемы ФАПЧ. Этот сигнал ошибки $e(k)$ подается на петлевой фильтр, на

выходе которого формируется сигнал приращения фазы $\xi(k)$ ГУН. Далее на выходе фазового аккумулятора снимается текущее значение фазы $\theta(k)$ ГУН, используемое в петле обратной связи и для получения искомого несущего колебания.

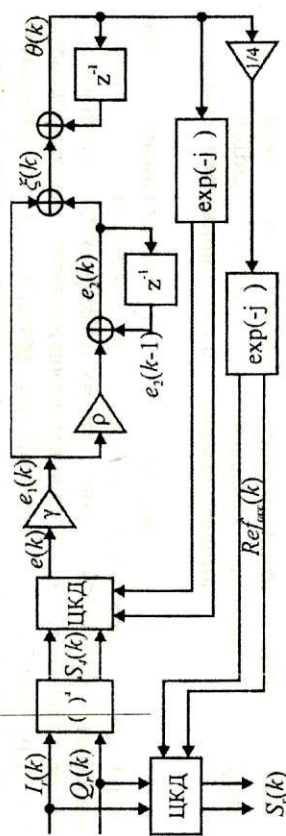


Рис. 2. Функциональная схема алгоритма восстановления несущего колебания QPSK-сигнала

Схема восстановления тактовой частоты
В качестве схемы восстановления тактовой частоты рассмотрим схему, представленную на рис. 3, где ЭЗ – элемент задержки, [] – операция округления до ближайшего минимального целого.

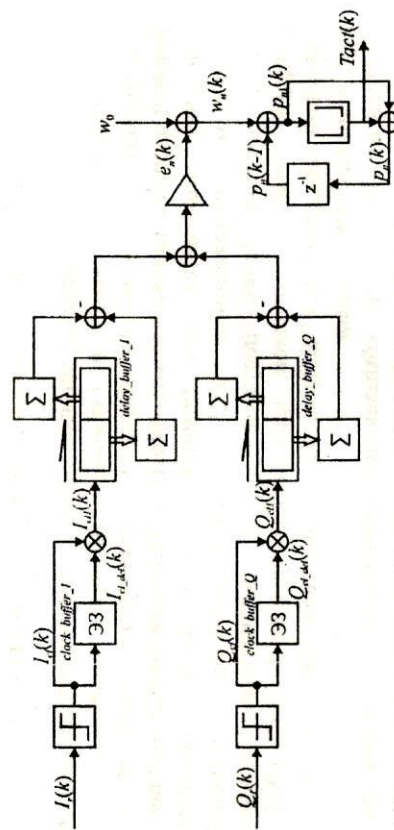


Рис. 3. Функциональная схема алгоритма восстановления тактовой частоты QPSK-сигнала

Здесь отсчеты синфазного и квадратурного каналов проходят через устройства жесткого ограничения, реализующие алгебраическую функцию взятия знака (sign). Далее полученные сигналы поступают на элемент задержки, глубина которого соответствует временному сдвигу $T/2$, после чего осуществляется перемножение задержанных и незадержанных сигналов.

Как известно [1], выходные сигналы соответствующих умножителей содержат в спектре дискретную составляющую на тактовой частоте следования информационных символов. Для выделения данных компонент эти сигналы подаются на схему ФАПЧ. При этом в качестве временного дискриминатора выступают корреляторы, реализованные на основе сдвиговых регистров и специальных схем суммирования. Глубина данных регистров соответствует периоду следования информационных символов. Объединенный сигнал ошибки синфазного и квадратурного каналов ослабляется аттенуатором и подается в виде сигнала $e_n(k)$ на вход генератора, контролируемого кодом (NCO). При этом величина ослабления выбирается таким образом, чтобы $e_n(k) \ll w_0$. Здесь w_0 – величина приращения счетчика NCO, которая обратна пропорциональна количеству отсчетов, приходящихся на один символьный период. Т.е.

$$w_0 = 1/q = 1/(TF_{\text{смп}}).$$

На допуск

1. Изучить теорию и разобраться с порядком выполнения работы.
2. По работам [2...4, 6] изучить понятие дискриминационной характеристики, известной как S-кривая, а также вопросы выбора порядка петлевого фильтра в системах синхронизации.
3. В исходном тексте программы выделить элементы кода, соответствующие всем указанным элементам схем, представленных на рис. 1-3.

Порядок выполнения работы

1. Построить спектральную плотность мощности сигнала $S_4(k)$. Сделать вывод о наличии дискретных составляющих в спектре.
2. Построить S-кривую дискриминатора схемы восстановления несущей частоты. Для этого установить нулевую частотную расстройку между генераторами передающей и приемной частей. Разомкнуть связь между ГУН и ЦКД схемы восстановления несущей частоты. Для не менее 20 различных значений начальной фазы

опорного генератора приемника в интервале от 0° до 360° оценить и записать среднее значение сигнала $e(k)$. Сделать вывод о наличии неопределенности восстановления фазы.

3. Убедиться в работоспособности схемы восстановления несущей частоты в условиях отсутствия и наличия аддитивного «белого» гауссовского шума при различных значениях частотной расстройки: $0,001/T$, $0,002/T$, $0,005/T$. Выполнить демонстрацию преподавателю процесса стабилизации сигнала в процессе создания в процессе вхождения в режим синхронизма.

4. Для различных значений частотной расстройки при различном уровне шума определить длительность переходного процесса вхождения в синхронизм.

5. Построить спектральную плотность мощности сигналов $I_{el}(k)$ и $Q_{el}(k)$ для нулевой и отличной от нуля (значение определяется преподавателем) частотной расстройки несущего и опорного колебания. Сделать вывод о наличии дискретных составляющих в спектре и их зависимости от величины частотной расстройки.

6. Построить S-кривую дискриминатора схемы восстановления тактовой частоты для нулевой частотной и отличной от нуля (значение определяется преподавателем) расстройки. Для этого требуется разомкнуть связь аттенуатора схемы восстановления тактовой частоты с управляющим входом NCO. Для различных значений временного смещения полезного сигнала (не менее 16) оценить и записать среднее значение сигнала $e_n(k)$. Сделать вывод о наличии неопределенности восстановления фазы тактового сигнала.

Контрольные вопросы

1. Объяснить структурную схему модели радиосистемы передачи информации, представленной на рис. 1.

2. Для чего требуется система синхронизации? Из каких компонентов она состоит? В каком блоке структурной схемы (см. рис. 1) она реализуется?

3. Объяснить функциональную схему алгоритма восстановления несущего колебания QPSK-сигнала.

4. Объяснить функциональную схему алгоритма восстановления тактовой частоты QPSK-сигнала.

5. Что такое дискриминатор? Чем он характеризуется?

6. Что такое S-кривая? Как она может быть измерена на практике?

7. Какая неоднозначность возникает при работе указанных алгоритмов восстановления колебаний несущей и тактовой частоты? В

чем ее причина? Какие существуют методы для устранения этой неоднозначности?

8. На схемах приведенных алгоритмов показать следующие элементы: дискриминатор, петлевой фильтр, управляемый генератор.

9. Какого порядка петлевые фильтры используются в схеме восстановления несущей и восстановления тактовой частоты? Почему?

10. Записать в виде математической формулы сигнал $\xi(k)$, выраженный через сигнал ошибки $e(k)$ (см. рис. 2).

11. Записать в виде математической формулы сигнал $Tact(k)$, выраженный через сигнал ошибки $e_n(k)$ (см. рис. 3).

Лабораторная работа № 4

Моделирование процесса обнаружения преамбулы радиосигнала

Цель работы: исследование работы оптимального обнаружителя сигналов на фоне «белого» гауссовского шума.

Теоретическая часть

Большинство современных систем передачи информации функционируют в пакетном режиме. В данном режиме бинарный поток разбивается на пакеты, последовательно излучаемые в радиоэфир. При этом каждый пакет сопровождается заголовком, в котором указывается служебная информация, необходимая для восстановления исходного непрерывного бинарного потока, а между пакетами формируется пауза. Подобная структура радиосигнала позволяет достаточно просто реализовать такие подходы к организации обмена информацией, как TDD, TDMA, CSMA и т.п. Недостатком данного подхода является необходимость для каждого пакета производить установку сеанса связи, что сопряжено с проблемами обнаружения и первичной оценки параметров полезного радиосигнала. Для решения этих задач в структуру пакета вносится специальное поле – преамбула. Состав данного поля является полностью известным на приемной стороне, таким образом исходная задача обнаружения сводится к задаче обнаружения известного сигнала. Структура радиосигнала при пакетном режиме передачи показана на рис. 1.

В рамках алгоритма обнаружения требуется по реализации принимаемого сигнала провести проверку двух гипотез – присутствует (гипотеза H_1) или отсутствует (гипотеза H_0) полезный сигнал в

анализируемой смеси. При этом возможно четыре исхода решения этой задачи:

- 1) сигнал обнаружен при условии гипотезы H_1 (правильное обнаружение);
- 2) сигнал обнаружен при условии гипотезы H_0 (ложная тревога);
- 3) сигнал не обнаружен при условии гипотезы H_1 (пропуск сигнала);
- 4) сигнал не обнаружен при условии гипотезы H_0 (верное необнаружение).

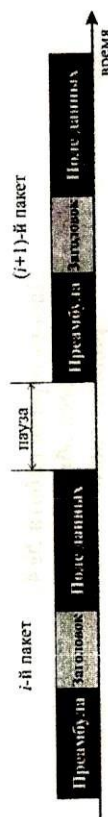


Рис. 1. Структура радиосигнала при пакетной передаче информации

Одним из подходов к построению алгоритмов обнаружения является использование критерия Неймана – Пирсона, основанное на максимизации вероятности P_D правильного обнаружения при ограничении на вероятность P_F ложной тревоги. По этому критерию оптимальный обнаружитель реализует следующее решающее правило [4]:

$$H_0: \frac{p(\mathbf{r}|H_1)}{p(\mathbf{r}|H_0)} < \lambda, \quad H_1: \frac{p(\mathbf{r}|H_1)}{p(\mathbf{r}|H_0)} \geq \lambda. \quad (1)$$

Здесь $p(\mathbf{r}|H_0)$ и $p(\mathbf{r}|H_1)$ – функция плотности вероятности распределения анализируемой смеси $\mathbf{r} = \{r(n)\}_n$ при гипотезах H_0 и H_1 соответственно, а λ – некоторый порог, который зависит от допустимого значения вероятности P_F ложной тревоги.

Отношение $\Lambda(\mathbf{r}) = \frac{p(\mathbf{r}|H_1)}{p(\mathbf{r}|H_0)}$ называется отношением (или функцией) правдоподобия.

В случае канала с АБГШ для гипотез H_0 и H_1 анализируемый сигнал соответственно равен

$$H_0: r(t) = n(t); \quad H_1: r(t) = \sqrt{E_s}(t) + n(t).$$

Здесь $n(t)$ – комплексный «белый» гауссовский шум со спектральной плотностью мощности N_0 , E – энергия полезного сигнала, $s(t)$ – нормированный опорный сигнал:

$$\int |s(t)|^2 dt = 1. \quad (2)$$

С учетом этого в случае цифровой обработки сигналов плотности $p(\mathbf{r}|H_0)$ и $p(\mathbf{r}|H_1)$ условной вероятности можно представить в виде

$$p(\mathbf{r}|H_0) = \frac{1}{(\pi N_0)^{N/2}} \exp \left(-\sum_{n=0}^{N-1} \frac{|r(n)|^2}{N_0} \right),$$

$$p(\mathbf{r}|H_1) = \frac{1}{(\pi N_0)^{N/2}} \exp \left(-\sum_{n=0}^{N-1} \frac{|r(n) - s(n)|^2}{N_0} \right).$$

$$\Lambda(\mathbf{r}) = \exp \left(\sum_{n=0}^{N-1} \frac{|r(n)|^2}{N_0} - \sum_{n=0}^{N-1} \frac{|r(n) - s(n)|^2}{N_0} \right).$$

Так как $|r(n) - s(n)|^2 = |r(n)|^2 - 2\operatorname{Re}\{r(n)s^*(n)\} + |s(n)|^2$, то функцию правдоподобия можно представить в виде

$$\Lambda(\mathbf{r}) = \exp \left(2\operatorname{Re} \sum_{n=0}^{N-1} \frac{r(n)s^*(n)}{N_0} - \sum_{n=0}^{N-1} \frac{|s(n)|^2}{N_0} \right).$$

Тогда с учетом монотонности функции натурального логарифма решающее правило (1) можно представить в виде

$$H_0: \operatorname{Re} \sum_{n=0}^{N-1} r(n)s^*(n) < \frac{N_0}{2} \ln \lambda + \frac{1}{2} \sum_{n=0}^{N-1} |s(n)|^2,$$

$$H_1: \operatorname{Re} \sum_{n=0}^{N-1} r(n)s^*(n) \geq \frac{N_0}{2} \ln \lambda + \frac{1}{2} \sum_{n=0}^{N-1} |s(n)|^2.$$

Здесь правая часть неравенств представляет собой некоторый порог γ , с которым сравнивается значение действительной части взаимной корреляционной функции (левая часть неравенств):

$$\gamma = \frac{N_0}{2} \ln \lambda + \frac{1}{2} \sum_{n=0}^{N-1} |s(n)|^2.$$

Данный порог не зависит от мощности принимаемого полезного сигнала. Второе слагаемое здесь определяется исключительно эталонным сигналом.

Если значение ρ_I действительной части взаимной корреляции превышает данный порог, то принимается решение о наличии во

входной смеси $r(n)$ полезного сигнала. В противном случае принимается решение об отсутствии искомого сигнала.

На практике принимаемый сигнал $r(n)$ представляется в виде синфазной $I(n)$ и квадратурной $Q(n)$ составляющих, аналогично эталонный сигнал $s(n)$ описывается квадратурными компонентами $I_s(n)$ и $Q_s(n)$. Обозначим взаимную корреляцию буквой ρ , а ее действительную и квадратурную составляющие соответственно ρ_I и ρ_Q , т.е.

$$\rho = \sum_{n=0}^{N-1} r(n)s^*(n) = \rho_I + j\rho_Q.$$

Тогда с учетом введенных обозначений левую часть неравенств можно представить в виде

$$\rho_I = \operatorname{Re} \sum_{n=0}^{N-1} r(n)s^*(n) = \operatorname{Re} \sum_{n=0}^{N-1} (I(n) + jQ(n))(I_s(n) + jQ_s(n))^*, \quad (3)$$

$$\rho_I = \sum_{n=0}^{N-1} I(n)I_s(n) + \sum_{n=0}^{N-1} Q(n)Q_s(n).$$

Само решающее правило описывается следующим способом:

$$H_0: \rho_I < \gamma, H_1: \rho_I \geq \gamma.$$

Таким образом, с учетом сделанных допущений оптимальным по критерию Неймана – Пирсона обнаружителем является линейное устройство, реализующее функцию вычисления действительной части взаимной корреляции ρ_I принятой смеси $r(n)$ и ожидаемого полезного сигнала $s(n)$. Как известно [2, 3], подобное устройство может быть построено на основе коррелятора или согласованного фильтра. Следует отметить, что вероятность правильного обнаружения при фиксированной вероятности ложной тревоги определяется только отношением сигнал-шум и не зависит от корреляционных свойств полезного радиосигнала.

На рис. 2 приведена схема одного из возможных вариантов реализации данного обнаружителя.

Здесь отсчеты синфазного и квадратурного канала попадают в сдвиговые регистры, для содержимого которых при помощи умножителей и сумматоров рассчитываются частные взаимные корреляции соответствующих действительных сигналов. В результате

формируется сигнал $\rho_I(n)$, сравниваемый с порогом. Данный сигнал описывается выражением

$$\rho_I(n) = \sum_{k=0}^{N-1} I(n-k)I_s(N-(k+1)) + \sum_{k=0}^{N-1} Q(n-k)Q_s(N-(k+1)) \quad (4)$$

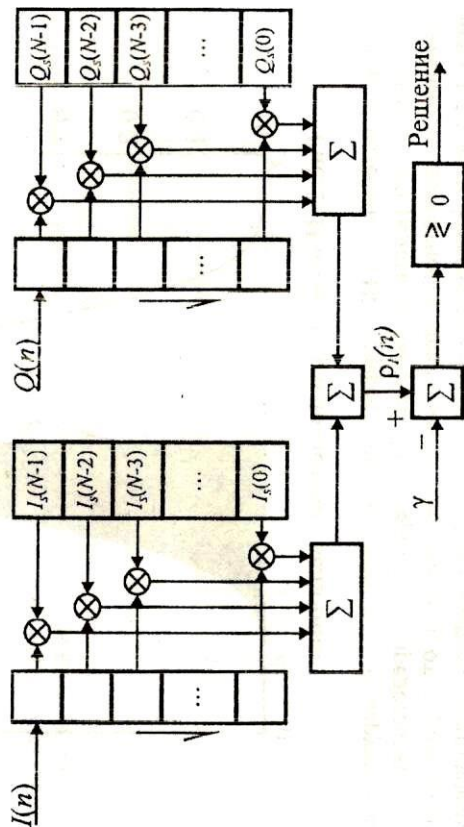


Рис. 2. Схема корреляционного обнаружителя

Гипотезам H_0 и H_1 соответствуют плотности вероятности $p(\rho_I | H_0)$ и $p(\rho_I | H_1)$, которые представляют собой гауссианы, смещенные относительно друг друга. Данные плотности вероятности проиллюстрированы на рис. 3, где сплошной линией показана функция $p(\rho_I | H_0)$, пунктирной линией – $p(\rho_I | H_1)$, серой заливке соответствует вероятность P_D правильного обнаружения для порога $\gamma = 1$ (величина порога показана вертикальной сплошной линией), штриховке – вероятность ложной тревоги P_F для того же порога. «Ширина» гауссианов определяется спектральной плотностью мощности шума, т.е. $\sigma = \sqrt{N_0/2}$. Смещение гауссиана, соответствующего гипотезе H_1 , равно $\sqrt{E}$.

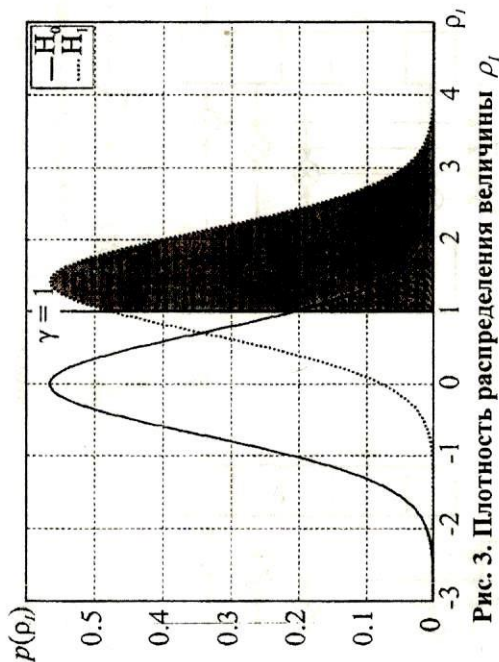


Рис. 3. Плотность распределения величины ρ_I

Как известно [4], эффективность работы обнаружителя описывается посредством рабочей характеристики, представляющей зависимость вероятности правильного обнаружения от вероятности ложной тревоги. Каждая рабочая характеристика соответствует своему отношению d величины смещения функции $p(\rho_I | H_1)$ к «ширине» гауссианов:

$$d = \sqrt{E} / \sigma = \sqrt{2E / N_0}.$$

Для данного значения отношения d вычисляются зависимости вероятности P_D правильного обнаружения и вероятности ложной тревоги P_F от величины порога γ . Далее эти функции $P_D(\gamma)$ и $P_F(\gamma)$ пересчитываются в зависимость $P_D(P_F)$.

На рис. 4 в качестве примера приведены рабочие характеристики для разных значений отношения d . Здесь прямая $d = 0$ соответствует предельному случаю, когда устройство обработки с одинаковой вероятностью принимает решение о наличии полезного сигнала, когда тот присутствует или отсутствует. Чем дальше от этой прямой проходит рабочая характеристика, тем качественнее обнаружитель.

Таким образом, эффективность обнаружения зависит от величины d , то есть при заданном уровне шумов определяется только энергией полезного сигнала и не зависит от формы самого сигнала. В частности, на рабочую характеристику не влияет база сигнала. Однако сужение ширины основного пика автокорреляционной функции полезного

сигнала позволяет повысить точность локализации обнаруженного сигнала на временной оси. В этой связи на практике в системах передачи информации в качестве преамбулы зачастую используются фазоманипулированные сигналы, образованные специальными кодовыми последовательностями. Данные последовательности выбираются по критериям минимума ширины основного пика и минимума уровня боковых выбросов автокорреляционной функции. В частности, широкое распространение получило использование последовательностей Баркера и m -последовательностей.

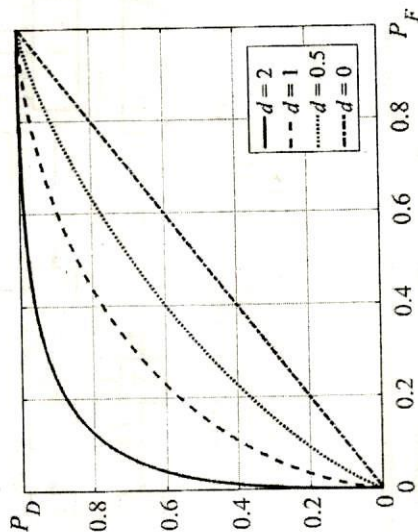


Рис. 4. Рабочие характеристики обнаружителя сигналов

На практике из-за частотного рассогласования задающих генераторов приемного и передающего устройства полезный радиосигнал поступает на решающее устройство со случайной фазой. В этих условиях в рамках введенных обозначений оптимальный обнаружитель реализует следующее правило:

$$H_0: |\rho| = |\rho_I + \rho_Q| < \gamma, \quad H_1: |\rho| = |\rho_I + \rho_Q| \geq \gamma.$$

Здесь квадратурная составляющая взаимной корреляции определяется выражением

$$\rho_Q = \sum_{n=0}^{N-1} Q(n) I_s(n) - \sum_{n=0}^{N-1} I(n) Q_s(n).$$

На рис. 5 представлена схема обнаружителя, реализующего данное решающее правило.

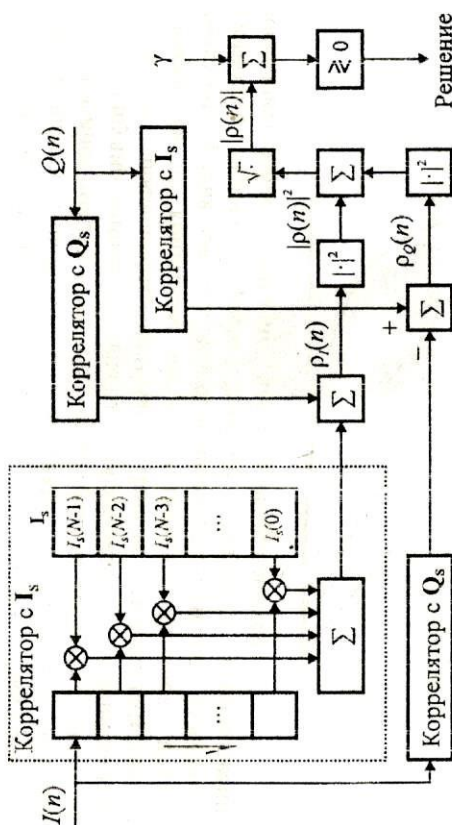


Рис. 5. Схема корреляционного обнаружителя для сигналов со случайным сдвигом фазы

Оценка $\hat{\phi}$ неизвестной фазы ϕ обнаруженного таким образом сигнала вычисляется в следующем виде:

$$\hat{\phi} = \arctg(\rho_Q / \rho_I).$$

Наличие в решающем правиле нелинейной операции взятия модуля приводит к тому, что законы распределения величины $|\rho|$ являются раисовскими, то есть отличаются от гауссовских. Следствием данного обстоятельства является ухудшение помехоустойчивости, т.е. увеличение числа ошибок обнаружения в условиях низкого отношения сигнал-шум. При этом рабочие характеристики для устройства, реализующего данную схему, могут быть также получены посредством компьютерного моделирования.

На допуск

1. Изучить теорию и разобраться с порядком выполнения работы.
2. Реализовать в GNU Octave в соответствии с рис. 6 модель, вычисляющую значение ρ_I действительной части взаимной корреляции между принимаемой смесью $r(n)$ и полезным сигналом $s(n) = I_s(n) + jQ(n)$.

Данная схема реализует последовательность действий:

- А. Сформировать кодовую последовательность $\mathbf{x} = \{x_i\}_{i=0}^{63}$.

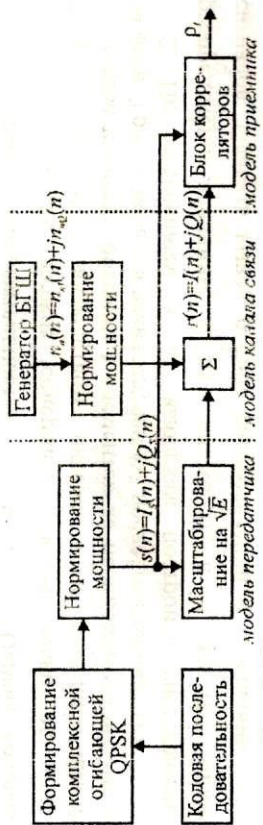


Рис. 6. Модель для получения рабочих характеристик

Б. Сформировать комплексную огибающую QPSK-сигнала с прямоугольным элементарным импульсом. Период следования символов $T = 1$ мкс, а частота дискретизации определяется по следующему правилу:

$$F_d = 8 * (\text{номер_бригады} + 1) / T.$$

В. Выполнить нормировку мощности комплексной огибающей в соответствии с формулой (2) с получением опорного сигнала $s(n)$.

Г. Умножить комплексную огибающую на величину $\sqrt{E} = \sqrt{2}$ (соответствует значению $d = 2$): $s_T(n) = \sqrt{E} \cdot s(n)$.

Д. Сформировать реализацию $n(n)$ комплексного «белого» гауссовского шума длины, аналогичной длине сформированного сигнала $s_T(n)$.

Е. Произвести нормировку шума для приведения спектральной плотности мощности к значению 1 Вт/Гц.

Ж. Сформировать смесь $r(n)$.

З. Вычислить и сохранить значение ρ_I действительной части взаимной корреляции сигналов $r(n)$ и $s(n)$. Расчет производится по формуле (3). Это соответствует гипотезе H_1 .

И. Вычислить и сохранить значение ρ_I действительной части взаимной корреляции сигналов $n(n)$ и $s(n)$, что соответствует гипотезе H_0 .

Порядок выполнения работы

1. На основе построенной модели сформировать выборку, содержащую не менее 1000 значений ρ_I для гипотез H_0 и H_1 для

трех кодовых последовательностей x_0 , x_1 и x_2 . Первая кодовая последовательность должна содержать исключительно единицы. Вторая образуется посредством интерполяции псевдослучайной последовательности объемом 8 элементов до необходимого объема в 64 бита. Третья является псевдослучайной последовательностью.

2. По полученным в п.1 выборкам построить гистограммы распределения величины ρ_1 для гипотез H_0 и H_1 для трех типов сигналов. Сделать вывод о типе распределений и о влиянии типа сигнала на функцию распределения величины ρ_1 .

3. Построить рабочие характеристики $P_D(P_F)$ для трех типов сигналов. Сделать вывод о влиянии типа сигнала на характеристики обнаружения.

4. Для одного из сигналов провести моделирование и построить рабочие характеристики для $d = 4$, $d = 1$, $d = 0.25$. Сделать выводы.

5. Переделать схему моделирования, внося следующие изменения:
- добавить блок внесения случайной задержки в точке А (см. рис. 6), величина данной задержки должна быть равномерно распределена в пределах от 33 до 77 мкс;

- проверить, что длительность генерируемой реализации шума соответствует длительности сигнала после внесения задержки;

- переделать блок корреляторов, с тем чтобы на выходе формировалось не одно значение ρ_1 , а функция времени $\rho_1(n)$; для этого необходимо реализовать вычисление по формуле (4).

6. Для каждого типа опорного сигнала построить автокорреляционную функцию.

7. Для каждого из трех сигналов построить по одной временной реализации $\rho_1(n)$. Сравнить с соответствующей автокорреляционной функцией. Сделать вывод о точности локализации различных типов сигналов на временной оси.

Контрольные вопросы

1. Что такое пакетный режим передачи? Объяснить состав пакетов и назначение отдельных полей.

2. Постановка задачи обнаружения. Сформулировать критерий Неймана – Пирсона.

3. Решающее правило оптимального обнаружителя в общем случае.

4. Функция правдоподобия в случае действия в канале аддитивного «белого» гауссовского шума.

5. Решающее правило оптимального обнаружителя в случае действия в канале аддитивного «белого» гауссовского шума.

6. Схема корреляционного обнаружителя в случае известной фазы принимаемого сигнала.

7. Как производится выбор порога обнаружения? Как связаны вероятности правильного обнаружения и ложной тревоги с порогом, спектральной плотностью мощности действующего шума и энергией сигнала?

8. Что такое рабочая характеристика? Как производится расчет рабочих характеристик? Какие предельные случаи рабочих характеристик вам знакомы?

9. Чем определяются характеристики обнаружения радиосигналов? Как влияют корреляционные свойства на качество обнаружения и локализацию сигналов во времени?

10. Обнаружение сигналов в условиях случайного сдвига фазы. Объяснить работу соответствующей схемы.

Библиографический список

1. Введение в Octave для инженеров и математиков / Е.Р. Алексеев, О.В. Чеснокова – М.: AIT Linux, 2012. – 368 с.
2. Феер К. Беспроводная цифровая связь. Методы модуляции и расширения спектра. – М.: Радио и связь, 2000. – 520 с.
3. Прокис Дж. Цифровая связь: пер. с англ. / под ред. Д.Д. Кловского. – М.: Радио и связь, 2000. – 800 с.
4. Ван Трис Г. Теория обнаружения, оценок и модуляции. Том 1. Теория обнаружения, оценок и линейной модуляции. – М.: Советское радио, 1972. – 744 с.
5. Склад Б. Цифровая связь. Теоретические основы и практическое применение: пер. с англ. – М.: Издательский дом «Вильямс», 2003. – 1104 с.
6. Mengali U., D'Andrea A.N. Synchronization Techniques for Digital Receivers. – New York: Plenum Press, 1997. – 520 p.

Содержание

Лабораторная работа № 1. Применение среды GNU Octave для моделирования радиосистем передачи информации	1
Лабораторная работа № 2. Исследование сигналов основной полосы	16
Лабораторная работа № 3. Исследование алгоритмов восстановления несущей и тактовой частоты сигналов с фазовой манипуляцией	26
Лабораторная работа № 4. Моделирование процесса обнаружения преамбулы радиосигнала	31
Библиографический список	42

Проектирование и моделирование радиосистем передачи информации

Составитель Покровский Павел Сергеевич

Редактор Р.К. Мангутова

Корректор С.В. Макушина

Подписано в печать 05.09.19. Формат бумаги 60x84 1/16.
Бумага писчая. Печать трафаретная. Усл. печ. л. 2,75.

Тираж 50 экз. Заказ 3712.

Рязанский государственный радиотехнический университет.
390005, Рязань, ул. Гагарина, 59/1.

Редакционно-издательский центр РГРТУ.