

МИНИСТЕРСТВО ЦИФРОВОГО РАЗВИТИЯ, СВЯЗИ И МАССОВЫХ  
КОММУНИКАЦИЙ  
РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное бюджетное образовательное учреждение  
высшего образования  
«Сибирский государственный университет телекоммуникаций и  
информатики»

Кафедра телекоммуникационных систем и вычислительных средств  
(ТС и ВС)

РЕФЕРАТ  
по дисциплине  
*«Моделирование мобильных систем»*

по теме:  
ИМИТАЦИОННАЯ МОДЕЛЬ КАНАЛА СВЯЗИ OFDM В MATLAB

Студент:  
*Группа № ИА-232*

*К.К Ошлаков*

Предподаватель:  
*должность, уч. степень, уч. звание*

*Р.В. Ахнашев*

Новосибирск 2025 г.

## СОДЕРЖАНИЕ

ВВЕДЕНИЕ .....	4
1 РЕАЛИЗАЦИЯ ОСНОВНЫХ БЛОКОВ СИСТЕМЫ СВЯЗИ .....	5
1.1 Практика 1. Знаковое кодирование .....	5
1.1.1 Описание .....	5
1.1.2 Реализация .....	5
1.1.3 Результаты тестирования .....	5
1.1.4 Консольный вывод .....	6
1.2 Практика 2. Помехоустойчивое кодирование .....	6
1.2.1 Описание .....	6
1.2.2 Реализация .....	7
1.2.3 Результаты тестирования .....	7
1.2.4 Консольный вывод .....	8
1.3 Практика 3. Перемежевание .....	8
1.3.1 Описание .....	8
1.3.2 Реализация .....	8
1.3.3 Результаты тестирования .....	9
1.3.4 Консольный вывод .....	9
1.4 Практика 4. QPSK модуляция .....	10
1.4.1 Описание .....	10
1.4.2 Реализация .....	10
1.4.3 Результаты тестирования .....	10
1.4.4 Консольный вывод .....	11
2 ФОРМИРОВАНИЕ И ПЕРЕДАЧА OFDM СИГНАЛА .....	12
2.1 Практика 5. OFDM модуляция .....	12
2.1.1 Описание .....	12
2.1.2 Реализация .....	12
2.1.3 Результаты тестирования .....	13
2.1.4 Консольный вывод .....	13
2.2 Практика 6. Модель канала передачи .....	13
2.2.1 Описание .....	13

2.2.2	Реализация.....	14
2.2.3	Результаты тестирования .....	14
2.2.4	Консольный вывод.....	14
2.3	Практика 7. Эквалайзирование и OFDM демодуляция.....	15
2.3.1	Описание .....	15
2.3.2	Реализация.....	15
2.3.3	Результаты тестирования .....	16
2.3.4	Консольный вывод.....	16
2.4	Практика 8. Расчет характеристик и построение графиков.....	17
2.4.1	Описание .....	17
2.4.2	Реализация.....	17
2.4.3	Результаты тестирования .....	17
2.4.4	Консольный вывод.....	19
2.4.5	Графические результаты .....	19
ЗАКЛЮЧЕНИЕ .....		21

## ВВЕДЕНИЕ

Развитие современных систем связи требует постоянного совершенствования методов передачи информации для обеспечения высокой надежности, эффективности и устойчивости к помехам. Особую роль играют технологии цифровой обработки сигналов, позволяющие реализовать сложные алгоритмы кодирования, модуляции, эквалайзирования и демодуляции.

В рамках данного цикла практических работ проводится комплексное исследование и реализация основных функциональных блоков системы цифровой связи с использованием программной среды MATLAB. Целью работы является изучение принципов построения современных систем связи, а также получение практических навыков по реализации ключевых алгоритмов обработки сигналов.

В ходе выполнения практических работ будут последовательно реализованы и исследованы следующие компоненты: знаковое кодирование и декодирование, помехоустойчивое кодирование (сверточное кодирование и декодирование Витерби), перемежение битовой последовательности, QPSK модуляция, OFDM модуляция, модель канала передачи с замираниями и аддитивным белым гауссовским шумом, а также приемная часть системы, включающая эквалайзирование и OFDM демодуляцию. Завершающий этап работы посвящен анализу производительности реализованной системы путем расчета коэффициента битовых ошибок (BER) и построения графиков.

Каждая практическая работа включает в себя теоретическое описание реализуемого блока, детализацию его программной реализации на языке MATLAB, демонстрацию результатов тестирования в виде консольного вывода и, при необходимости, графических представлений.

Полученные в результате работы знания и навыки будут способствовать пониманию принципов функционирования современных цифровых систем связи и станут основой для дальнейшего изучения и разработки более сложных телекоммуникационных систем.

# 1 РЕАЛИЗАЦИЯ ОСНОВНЫХ БЛОКОВ СИСТЕМЫ СВЯЗИ

## 1.1 Практика 1. Знаковое кодирование

### 1.1.1 Описание

В рамках первой практической работы была реализована система знакового кодирования и декодирования текстового сообщения. Основная задача заключалась в преобразовании текстовой строки в последовательность битов и обратное преобразование битового потока в текст. Каждый символ исходного сообщения кодировался в 8-битную последовательность. Были разработаны функции `sign_encoder.m` и `sign_decoder.m` на языке MATLAB для выполнения соответствующих операций.

### 1.1.2 Реализация

#### Кодирование

Функция `sign_encoder(textMessage)` принимает на вход текстовую строку и преобразует каждый символ в его 8-битное представление. Полученные битовые последовательности конкатенируются в единый битовый поток.

#### Декодирование

Функция `sign_decoder(bitStream)` принимает на вход битовый поток. Поток разбивается на 8-битные блоки, каждый из которых преобразуется обратно в соответствующий символ. Полученные символы объединяются в декодированную текстовую строку.

### 1.1.3 Результаты тестирования

Корректность работы кодера и декодера была проверена путем кодирования тестовой строки и последующего ее декодирования. Консольный вы-

вод ниже демонстрирует исходное сообщение, часть закодированной битовой последовательности и результат декодирования.

```
% START - TASK: 1
disp('TASK 1: ');
originalText = 'Hello World. 1234';
bitStream = sign_encoder(originalText);
decodedText = sign_decoder(bitStream);

disp(['Source: ', originalText]);
disp(['Encoded bits: ', num2str(bitStream)]); %
disp(['Decoded text: ', decodedText]);
% END - TASK: 1
```

#### **1.1.4 Консольный вывод**

```
TASK 1:
Source: Hello World. 1234
Encoded bits: 0 0 0 1 1 1 0 1 ... (    )
Decoded text: Hello World. 1234
```

Закодированная битовая последовательность представляет собой полное битовое представление исходного текста. В документе представлен только фрагмент для наглядности.

## **1.2 Практика 2. Помехоустойчивое кодирование**

### **1.2.1 Описание**

Вторая практическая работа посвящена реализации помехоустойчивого кодирования с использованием сверточного кодера и декодирования с помощью алгоритма Витерби. Данные методы позволяют повысить надежность передачи информации в условиях шума.

## 1.2.2 Реализация

### Сверточный кодер

Был реализован сверточный кодер с параметрами:

- Порождающие полиномы  $G_1 = 171_8$  и  $G_2 = 133_8$ .
- Длина кодового ограничения  $k = 7$ .

Функция `conv_encoder.m` выполняет сверточное кодирование входной битовой последовательности на основе заданных полиномов.

### Декодер Витерби

Для декодирования использован алгоритм Витерби. Декодер `viterbi_decoder.m` находит наиболее вероятную исходную последовательность, анализируя принятую последовательность и используя структуру кодера (треллис). Алгоритм включает вычисление метрик пути и обратное прослеживание для восстановления исходных битов.

## 1.2.3 Результаты тестирования

Работоспособность сверточного кодирования и декодирования Витерби была проверена путем кодирования битовой последовательности, полученной на предыдущем шаге, и ее последующего декодирования.

```
% START - TASK: 2
disp('TASK 2: ');
codedBits = conv_encoder(bitStream);
decodedBits = viterbi_decoder(codedBits);
decodedText_2 = sign_decoder(bitStream); %

disp(['      : ', num2str(bitStream)]); %
disp(['      : ', num2str(codedBits)]); %
disp(['      : ', num2str(decodedBits)]); %
disp(['Decoded text: ', decodedText_2]);

% END - TASK: 2
```

## 1.2.4 Консольный вывод

TASK 2:

```
      : 0  0  0  1  1  1  0  1 ... (    )  
      : 0  0  0  0  0  0  1  1 ... (    )  
      : 0  0  0  1  1  1  0  1 ... (    )
```

Decoded text: Hello World. 1234

Декодированная битовая последовательность совпадает с исходной, что подтверждает корректность работы сверточного кодера и декодера Витерби в отсутствие ошибок. В документе представлены только фрагменты битовых последовательностей.

## 1.3 Практика 3. Перемежение

### 1.3.1 Описание

Третья практическая работа посвящена реализации перемежения битовой последовательности. Перемежение используется для преобразования пакетных ошибок, которые могут возникать в канале связи, в случайные одиночные ошибки, с которыми помехоустойчивое кодирование справляется более эффективно.

### 1.3.2 Реализация

Были разработаны функции `interleave_forward.m` и `interleave_reverse.m`.

### Перемежение

Функция `interleave_forward(inputBits)` принимает на вход битовую последовательность и случайным образом перемешивает ее элементы, используя случайную перестановку индексов. Генерируемая перестановка сохраняется для последующего восстановления.



## Деперемежение

Функция `interleave_reverse(interleavedBits)` принимает на вход перемешанную битовую последовательность и восстанавливает исходный порядок битов, используя сохраненную обратную перестановку индексов.

### 1.3.3 Результаты тестирования

Корректность работы перемежителя и деперемежителя была проверена путем перемежения закодированной битовой последовательности (из Практики 2) и последующего ее деперемежения.

```
% START - TASK: 3
disp('TASK 3: ');

interleavedBits = interleave_forward(codedBits);

deinterleavedBits = interleave_reverse(interleavedBits);

disp(['          : ', num2str(codedBits)]); %
disp(['          : ', num2str(interleavedBits)]); %
disp(['          : ', num2str(deinterleavedBits)]); %

decodedBits_3 = viterbi_decoder(deinterleavedBits);
decodedText_3 = sign_decoder(decodedBits_3);
disp(['          : ', decodedText_3]);

% END - TASK: 3
```

### 1.3.4 Консольный вывод

TASK 3:

```
          : 0 0 0 0 0 0 1 1 ... (    )
: 0 1 0 1 0 0 0 0 ... (    )
          : 0 0 0 0 0 0 1 1 ... (    )
          : Hello World. 1234
```

Восстановленная битовая последовательность совпадает с исходной закодированной последовательностью, что подтверждает корректность работы перемежителя. Декодированный текст также соответствует исходному. В документе представлены только фрагменты битовых последовательностей.

## 1.4 Практика 4. QPSK модуляция

### 1.4.1 Описание

Четвертая практическая работа посвящена реализации квадратурной фазовой манипуляции (QPSK). QPSK является одной из базовых методов цифровой модуляции, позволяющей передавать два бита информации за один символ, кодируя их фазой несущей.

### 1.4.2 Реализация

Была реализована функция `qpsk_modulator.m`, которая выполняет отображение пар битов в комплексные символы согласно следующему правилу:

- $(0,0) \rightarrow 0.707 + 0.707i$
- $(0,1) \rightarrow 0.707 - 0.707i$
- $(1,0) \rightarrow -0.707 + 0.707i$
- $(1,1) \rightarrow -0.707 - 0.707i$

Также реализована функция `qpsk_demodulator.m` для обратного преобразования комплексных символов в битовую последовательность.

### 1.4.3 Результаты тестирования

Корректность модуляции и демодуляции была проверена путем модуляции перемешанной битовой последовательности (из Практики 3) и последующей ее демодуляции.

```
% START - TASK: 4  
disp('TASK 4: ');
```

```

modulatedSymbols = qpsk_modulator(interleavedBits);
demodulatedBits = qpsk_demodulator(modulatedSymbols);

disp(['          : ', num2str(interleavedBits)]); %
disp(['          : ', num2str(demodulatedBits)]); %

deinterleavedBits_4 = interleave_reverse(demodulatedBits);
decodedBits_4 = viterbi_decoder(deinterleavedBits_4);
disp(['decodedBits_4: ', num2str(decodedBits_4)]); %
decodedText_4 = sign_decoder(decodedBits_4);

disp(['Decoded text after modulation/demodulation: ', decodedText_4])

% END - TASK: 4

```

#### 1.4.4 Консольный вывод

```

TASK 4:
          : 0  1  0  1  0  0  0  0 ... (    )
          : 0  1  0  1  0  0  0  0 ... (    )
decodedBits_4: 0  0  0  1  1  1  0  1 ... (    )
Decoded text after modulation/demodulation: Hello World. 1234

```

Демодулированная битовая последовательность совпадает с перемешанной, а декодированный текст совпадает с исходным, что подтверждает корректность работы QPSK модулятора и демодулятора в отсутствие ошибок. В документе представлены только фрагменты битовых последовательностей.

## 2 ФОРМИРОВАНИЕ И ПЕРЕДАЧА OFDM СИГНАЛА

### 2.1 Практика 5. OFDM модуляция

#### 2.1.1 Описание

Пятая практическая работа посвящена реализации модуляции по принципу OFDM (Orthogonal Frequency-Division Multiplexing - ортогональное частотное мультиплексирование). OFDM является широко используемым методом модуляции в современных системах связи благодаря своей устойчивости к многолучевому распространению и интерференции.

#### 2.1.2 Реализация

Был разработан OFDM модулятор (`ofdm_modulator.m`) со следующими ключевыми компонентами:

- **Вставка пилотных символов:** В сигнал добавляются известные пилотные символы для последующей оценки канала.
- **Добавление нулевых поднесущих:** Добавляются нулевые поднесущие по краям спектра для обеспечения защиты от внеполосных излучений и облегчения фильтрации.
- **Применение ОБПФ:** Выполняется обратное быстрое преобразование Фурье для формирования временного OFDM символа.
- **Добавление циклического префикса:** В начало OFDM символа добавляется копия его конца для борьбы с межсимвольной интерференцией.

Параметры модуляции, такие как интервал пилотных символов ( $\Delta R_s$ ), длина циклического префикса ( $T_{CP}$ ), процент нулевых поднесущих ( $C$ ) и общее число поднесущих ( $N_{subcarrier}$ ), сохраняются для использования в демодуляторе.

### 2.1.3 Результаты тестирования

В данной практической работе производится только модуляция, без обратного преобразования. Проверка корректности производится на последующих этапах, когда сигнал пройдет через канал и будет демодулирован. Консольный вывод показывает фрагмент сгенерированного OFDM символа.

```
% START - TASK: 5
```

```
disp('TASK 5: ');
```

```
delta_Rs = 6;
```

```
T_CP = 16;
```

```
C = 0.25;
```

```
N_subcarrier = 128;
```

```
ofdm_symbol = ofdm_modulator(modulatedSymbols, delta_Rs, T_CP, C, N_s
```

```
% disp(ofdm_symbol) %
```

```
% END - TASK: 5
```

### 2.1.4 Консольный вывод

```
TASK 5:
```

(Консольный вывод для этой задачи пустой, так как `disp(ofdm_symbol)` закомментировано)

## 2.2 Практика 6. Модель канала передачи

### 2.2.1 Описание

Шестая практическая работа посвящена реализации модели канала передачи с учетом замираний и аддитивного белого гауссовского шума (АБГШ). Реалистичная модель канала необходима для оценки производительности системы связи в условиях, приближенных к реальным.

### 2.2.2 Реализация

Была реализована функция `channel_model.m`, моделирующая канал со следующими характеристиками:

- **Многолучевое распространение:** Сигнал достигает приемника по нескольким путям ( $N_b$  лучей) с разными задержками и ослаблениями.
- **Случайные задержки лучей:** Задержки для каждого луча генерируются случайным образом.
- **Добавление АБГШ:** К сигналу примешивается аддитивный белый гауссовский шум с заданной мощностью ( $N_0$ ), определяемой отношением сигнал/шум (SNR).

Модель канала принимает на вход OFDM символ и возвращает принятый сигнал, прошедший через канал.

### 2.2.3 Результаты тестирования

Работа модели канала была проверена путем пропускания OFDM символа, сгенерированного на предыдущем этапе, через модель. Консольный вывод показывает фрагмент принятого сигнала.

```
% START - TASK: 6
disp('TASK 6: ');

N_path = 3;
NO_dB = 15;
max_delay = 6;
rx_signal = channel_model(ofdm_symbol, N_path, NO_dB, max_delay);

disp(rx_signal); %
% END - TASK: 6
```

### 2.2.4 Консольный вывод

TASK 6:

Columns 1 through 13

1.3812 - 0.8223i    0.0111 - 0.1080i    -1.0138 + 0.7830i ... (    )

Columns 14 through 26

0.7438 - 1.3265i    -3.3089 + 0.5977i    1.1598 + 0.3892i ... (    )

Columns ... (    )

... 0.0007 + 0.0138i

Принятый сигнал представляет собой комплексные значения, которые включают в себя искажения от канала и шум. В документе представлен только фрагмент для наглядности.

## 2.3 Практика 7. Эквалайзирование и OFDM демодуляция

### 2.3.1 Описание

Седьмая практическая работа посвящена реализации приемной части системы связи, включающей OFDM демодуляцию и эквалайзирование. Эти этапы позволяют восстановить исходные символы из принятого OFDM сигнала, прошедшего через канал.

### 2.3.2 Реализация

Был разработан OFDM демодулятор (`ofdm_demodulator.m`), выполняющий следующие действия:

- **Удаление циклического префикса:** Удаляется добавленный на передаче циклический префикс.
- **БПФ:** Выполняется быстрое преобразование Фурье для перехода из временной области в частотную.
- **Оценка канала по пилотным символам:** Используются принятые пилотные символы для оценки частотной характеристики канала.

- **Эквалайзирование:** Выполняется компенсация искажений, внесенных каналом, на основе оценки канала.
- **Удаление нулевых поднесущих:** Удаляются нулевые поднесущие, добавленные на передаче.

Полученные в результате демодуляции символы (QPSK символы) далее передаются на демодулятор QPSK.

### 2.3.3 Результаты тестирования

Корректность работы OFDM демодулятора и эквалайзера оценивается на следующем этапе, где происходит демодуляция полученных символов и декодирование битовой последовательности.

```
% START - TASK: 7
```

```
disp('TASK 7: ');
```

```
received_symbols = ofdm_demodulator(rx_signal, delta_Rs, T_CP, C, N_s
% received_symbols = ofdm_demodulator(ofdm_symbol, delta_Rs, T_CP, C,
demodulatedBits = qpsk_demodulator(received_symbols);
```

```
% disp('Original QPSK symbols:'); %
```

```
% disp(modulatedSymbols);
```

```
% disp('Received QPSK symbols:'); %
```

```
% disp(received_symbols);
```

```
% END - TASK: 7
```

### 2.3.4 Консольный вывод

```
TASK 7:
```

(Консольный вывод для этой задачи пустой)



## **2.4 Практика 8. Расчет характеристик и построение графиков**

### **2.4.1 Описание**

Восьмая практическая работа посвящена анализу производительности разработанной системы связи путем расчета коэффициента битовых ошибок (BER) и построения графиков, иллюстрирующих спектр сигналов и сигнальные созвездия.

### **2.4.2 Реализация**

#### **Расчет BER**

Производится сравнение принятой и демодулированной битовой последовательности с исходной битовой последовательностью, полученной после перемежения. Подсчитывается количество ошибочных битов и вычисляется BER как отношение количества ошибок к общему числу переданных битов.

#### **Построение графиков**

Строятся следующие графики:

- Спектр переданного OFDM символа.
- Спектр принятого OFDM символа (до эквалайзирования).
- Сигнальное созвездие QPSK символов в передатчике.
- Сигнальное созвездие QPSK символов в приемнике (после OFDM демодуляции и эквалайзирования).

Эти графики позволяют визуально оценить влияние канала на сигнал и эффективность эквалайзирования.

### **2.4.3 Результаты тестирования**

Ниже представлены консольный вывод с рассчитанным BER и сгенерированные графики.

% START – TASK: 8

```

disp('TASK 8: ');

min_length = min(length(demodulatedBits), length(interleavedBits));

demodulatedBits_truncated = demodulatedBits(1:min_length);
interleavedBits_truncated = interleavedBits(1:min_length);

%      BER
bit_errors = sum(demodulatedBits_truncated ~= interleavedBits_truncated);
total_bits = length(demodulatedBits_truncated);
BER = bit_errors / total_bits;

disp(['          : ', num2str(bit_errors)]);
disp(['          : ', num2str(total_bits)]);
disp(['BER: ', num2str(BER)]);

figure(1);

subplot(2, 2, 1);
plot(abs(fft(ofdm_symbol)));
title('          ');
xlabel('      ');
ylabel('      ');

subplot(2, 2, 2);
plot(abs(fft(rx_signal)));
title('          ');
xlabel('      ');
ylabel('      ');

subplot(2, 2, 3);
scatter(real(modulatedSymbols), imag(modulatedSymbols), 'filled')
title('          ');
xlabel('Re');
ylabel('Im');

```

```

subplot(2, 2, 4);
scatter(real(received_symbols), imag(received_symbols), 'filled');
title('');
xlabel('Re');
ylabel('Im');

% END - TASK: 8

```

#### 2.4.4 Консольный вывод

TASK 8:

```

: 1
: 204
BER: 0.004902

```

#### 2.4.5 Графические результаты

Рисунок 1 демонстрирует спектры переданного и принятого сигналов, а также сигнальные созвездия до и после передачи через канал. Спектр принятого сигнала демонстрирует влияние многолучевости и шума. Сигнальное созвездие в приемнике показывает, как шум и искажения канала влияют на расположение принятых символов по сравнению с идеальным созвездием на передатчике. Наличие ошибок ( $BER > 0$ ) подтверждает влияние канала на качество передачи.

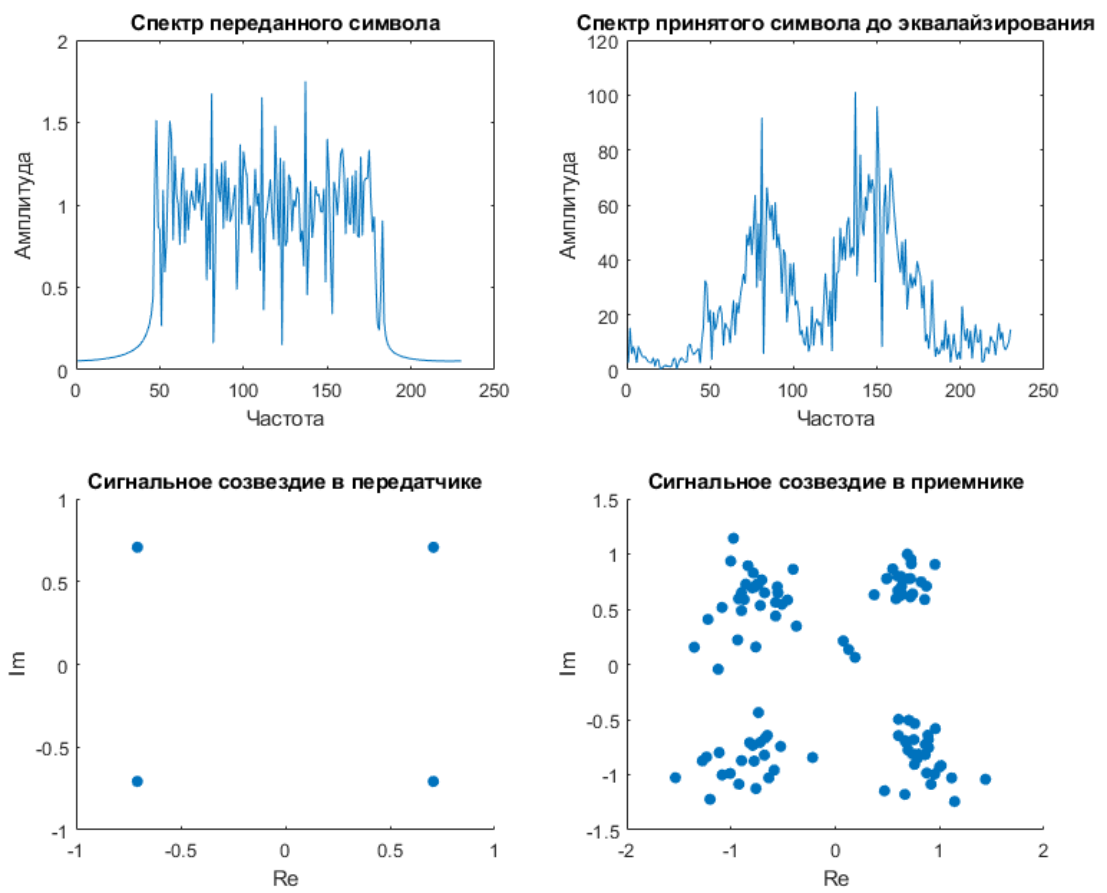


Рисунок 1 — Результаты моделирования

## ЗАКЛЮЧЕНИЕ

В результате выполнения цикла практических работ была реализована полная модель системы цифровой связи. В процессе выполнения задач были освоены и применены следующие основные блоки и технологии:

- **Кодирование источника:** Преобразование текстовой информации в битовую последовательность и обратное преобразование.
- **Помехоустойчивое кодирование:** Использование сверточного кодера и декодера Витерби для защиты информации от ошибок. **Перемежение:** Применение перемежителя для преобразования пакетных ошибок в случайные.
- **QPSK модуляция:** Отображение битовой последовательности в комплексные символы для передачи по каналу.
- **OFDM модуляция:** Формирование OFDM символа с пилотными символами, нулевыми поднесущими и циклическим префиксом.
- **Модель канала с замираниями и шумом:** Моделирование реальных условий распространения сигнала с учетом многолучевости и АБГШ.
- **Демодуляция и декодирование:** Восстановление исходных данных из принятого сигнала, включая OFDM демодуляцию, эквалайзирование, QPSK демодуляцию, депережежение и декодирование Витерби.

В ходе тестирования была продемонстрирована работоспособность каждого из реализованных блоков, а также всей системы в целом. Расчет BER и построение графиков позволили оценить влияние модельного канала на качество передачи и увидеть эффект помехоустойчивого кодирования и эквалайзирования.

Полученные результаты показывают, что реализованная система способна передавать информацию через модельный канал.