

Министерство науки и высшего образования Российской Федерации

Федеральное государственное бюджетное образовательное учреждение
высшего образования

**«ТОМСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ СИСТЕМ
УПРАВЛЕНИЯ И РАДИОЭЛЕКТРОНИКИ» (ТУСУР)**

Е. В. Рогожников

Э. М. Дмитриев

К. В. Диноченко

**ПРОЕКТИРОВАНИЕ СИСТЕМ БЕСПРОВОДНОЙ СВЯЗИ И
ИНТЕРНЕТА ВЕЩЕЙ**

Методические указания для выполнения
лабораторных и практических работ для студентов
технических направлений подготовки и
специальностей квалификации бакалавр по
дисциплине Проектирование систем беспроводной
связи и интернета вещей

Томск
2023

УДК 681.3.068
ББК 32.973.2
Р 598

Рецензент:

Крюков Я.В., доцент кафедры телекоммуникаций и основ радиотехники
ТУСУРа, кандидат технических наук

Р 598 Проектирование систем беспроводной связи и интернета вещей: Методические указания для выполнения лабораторных и практических работ для студентов технических направлений подготовки и специальностей квалификации бакалавр по дисциплине Проектирование систем беспроводной связи и интернета вещей/ Е. В. Рогожников, Э. М. Дмитриев, К. В. Диноченко – Томск: Томск. гос. ун-т систем управления и радиоэлектроники, 2023. – 42 с.

Настоящие учебно-методическое пособие содержит указания по выполнению лабораторных и практических работ по дисциплине Проектирование систем беспроводной связи и интернета вещей. Данный лабораторный практикум имеет цель закрепить, полученные в ходе курса, принципы построения современных систем цифровой радиосвязи, построение беспроводных сетей и систем, а также навыки работы с математическим пакетом GNU Octave.

Одобрено на заседании кафедры ТОР, протокол № 1 от 31 августа 2023 г.

УДК 681.3.068
ББК 32.973.2

© Рогожников Е.В., Дмитриев Э.М.,
Диноченко К.В., 2023
© Томск. гос. ун-т систем управления и
радиоэлектроники, 2023

Оглавление

ВВЕДЕНИЕ.....	4
Работа №1	5
Работа №2	7
Работа №3	11
Работа №4	15
Работа №5	20
Работа №6	25
Работа №7	28
Работа №8	31
Работа №9	34
Работа №10	39
СПИСОК ЛИТЕРАТУРЫ.....	42

ВВЕДЕНИЕ

Практикум по курсу “Проектирование систем беспроводной связи и интернета вещей” предназначен для закрепления и расширения теоретических знаний студентов в области построения современных систем цифровой радиосвязи, построения беспроводных сетей и систем.

Первая часть практикума содержит описание следующих работ:

- 1) Начало работы с Octave;
- 2) Реализация BPSK приемника и передатчика в Octave;
- 3) Реализация QPSK приемника и передатчика в Octave;
- 4) OFDM модуляция;
- 5) Кадровая и частотная синхронизация;
- 6) Технология MIMO и кодирование Аламоути;

Работы данного перечня выполняются на базе микроконтроллеров в среде разработки GNU Octave. Среда разработки GNU Octave – это свободная программная система для математических вычислений, использующая совместимый с MATLAB язык высокого уровня. Данную программу можно скачать с официального сайта <https://www.gnu.org/software/octave/> .

Работа №1 «Начало работы с Octave»

Цель работы: Изучить основные функции и блоки Octave и составить тестовую программу.

Задачи работы:

- 1) Изучить основные функции и блоки Octave.
- 2) Произвести генерацию синусоидального сигнала в Octave. Снять характеристики частоты сгенерированных колебаний.
- 3) Произвести сложение и умножение гармонических сигналов.

Ход выполнения работы

Первые три строки программы, как правило, такие:

```
clc
clear all
close all
```

clc – очищает «Командное окно»; clear all – удаляет все переменные из «Области переменных», очищает память; close all – закрывает все открытые фигуры.

Постройте гармонический сигнал во временной и частотной области со следующими параметрами:

$F_0 = 10$ кГц – несущая частота,
 $F_s = 250$ кГц – частота дискретизации,
 $N = 100$ – количество отсчетов.
Реализация в Octave:

```
clc
clear all
close all
F0 = ...;
Fs = ...;
N = ...;
t = (0:N-1)/Fs; временные отсчеты
sig = sin (2*pi*F0*t); гармонический сигнал
figure
plot(t,sig);
grid on
xlabel('Time, s','fontsize',16);
ylabel('Amplitude','fontsize',16);
```

Результат выполнения кода показана на рисунке 1.1.

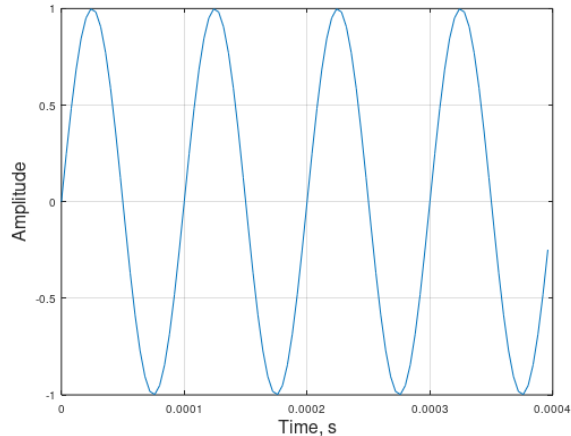


Рисунок 1.1 – Полученный гармонический сигнал

Самостоятельные задачи.

Задача 1. Постройте спектр синусоидального сигнала, используя функцию `fft()`. Задайте ось частот аналогично временным отсчетам. Сформируйте еще один синусоидальный сигнал частотой 50 кГц. Используя цикл `for`, произведите перемножение и суммирование 2 синусоидальных сигналов.

Задача 2. Создайте 2 битовые последовательности (A и B) состоящие из 100 элементов ($N = 100$), используя функцию `randi`. Используя цикл `for` и функцию `if`, произведите суммирование по модулю 2.

Таблица 1.1 – Таблица истинности XOR.

ВХОД		ВЫХОД
0	0	0
0	1	1
1	0	1
1	1	0

Контрольные вопросы к работе:

- 1) Для чего может использоваться среда разработки Octave?
- 2) Как вывести рисунок на экран?
- 3) Как рассчитать спектр сигнала?
- 4) Как задать псевдослучайную битовую последовательность?

Требования к оформлению отчета по работе.

Содержание отчета:

- 1) Введение,
- 2) Описание работы,
- 3) Рисунки:
 - Гармонический сигнал,
 - Спектр гармонического сигнала,
 - Спектр произведения гармонических сигналов,
 - Спектр суммы гармонических сигналов.
- 4) Выводы,
- 5) Полный листинг программы.

Работа №2 «Реализация BPSK передатчика и приёмника в Octave»

Цель работы: Изучить принцип переноса BPSK последовательности на несущую частоту.

Задачи работы:

- 1) Выполнить BPSK модуляцию сигнала;
- 2) Произвести перенос последовательности на несущую частоту;
- 3) Выполнить обработку BPSK модулированного сигнала.

Ход выполнения работы

В данной работе необходимо создать битовую последовательность, произвести BPSK модуляцию, интерполировать ее, перенести на несущую частоту. В приемнике произвести перенос на нулевую частоту, фильтрацию, децимацию, демодуляцию и проверить количество ошибок.

Формирование сигнала. Передатчик.

```
clc
clear all
close all
%Выберите размер BPSK последовательности
size = 1000;
%Установите несущую частоту
fc=5e6;
%Задайте частоту дискретизации
fs=20e6;
%Задайте коэффициент интерполяции
L = 16;
```

Сформируйте битовую последовательность.

```
RANDOMDATA = randi([0 1], 1, 10000);
DATA = RANDOMDATA(1:size);
```

Произведите BPSK модуляцию.

```
for i=1:size
    if (DATA(i)==1)
        MOD_DATA(i) = 1;
    else
        MOD_DATA(i) = -1;
    end;
end
```

Постройте созвездие модулированного сигнала, используя функцию scatterplot(). Для использования этой функции необходимо загрузить ее, написав pkg load communications в командное окно и выполнить.

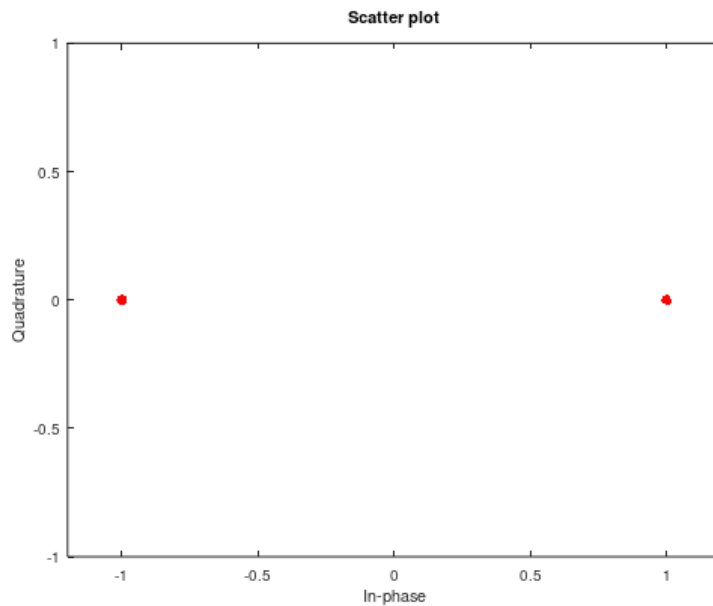


Рисунок 2.1 – Сигнальное созвездие BPSK

Выполните интерполяцию. Если MOD_DATA имеет размерность 1000 символов, на выходе интерполятора должно быть в L раз больше символов. Пример функции интерполяции представлен ниже.

```
p=1; q=1;
for k=1:size
MOD_DATA_interp(q:L*p) = MOD_DATA(k);
q=(p*L+1);
p=p+1;
end
```

Задайте временной диапазон.

```
t = (0:length(MOD_DATA_interp)-1)/fs;
```

Задайте гармонический сигнал с несущей частотой fc .

```
car_sig = sin(...
```

Перенесите BPSK последовательность MOD_DATA_interp на несущую частоту, путем умножения на гармонический сигнал. Постройте ее график с помощью функции plot, опираясь на прошлые работы (рисунок 2.2 С.). Функция xlim используется для ограничения диапазона построения графика.

```
RECO=...
figure
plot(RECO);
xlim([0, 200]);
```

Добавьте белый шум с помощью функции awgn. Отношение сигнал-шум задайте равным 15.

```
RECO_AWGN=awgn(RECO, 15, 'measured')
```


Обработка сигнала. Приёмник.

Верните BPSK последовательность на нулевую частоту, повторно перемножив BPSK последовательность с гармоническим сигналом. Постройте ее график (Рисунок 2.2 E.).

```
REVIVE =...
```

Выполните фильтрацию полученных данных на нулевой частоте путем интегрирования (Рисунок 2.2 G.).

```
for i=1:length(REVIVE)/L
a(i) = sum(REVIVE(i*L-L+1:i*L))/(L/2);
y(i*L-L+1:L*i) = a(i);
end
```

Произведите децимацию, а затем демодуляцию полученных данных. Цель демодуляции – преобразовать сигнал в битовый поток. Используйте цикл и набор условий. Постройте график децимированной последовательности (Рисунок 2.2 H.).

```
decim = y(L:L:end);
for i=1:length(decim)
if ((decim(i))<0)
demod(i) = 0;
else
demod(i) = 1;
end
end
```

Посчитайте ошибки с помощью функции biterr().

```
err = biterr(DATA, demod);
```

Используя функцию scatterplot(), постройте созвездие сигнала после обработки (Рисунок 2.2 I.). На графиках ниже приведен результат выполнения работы.

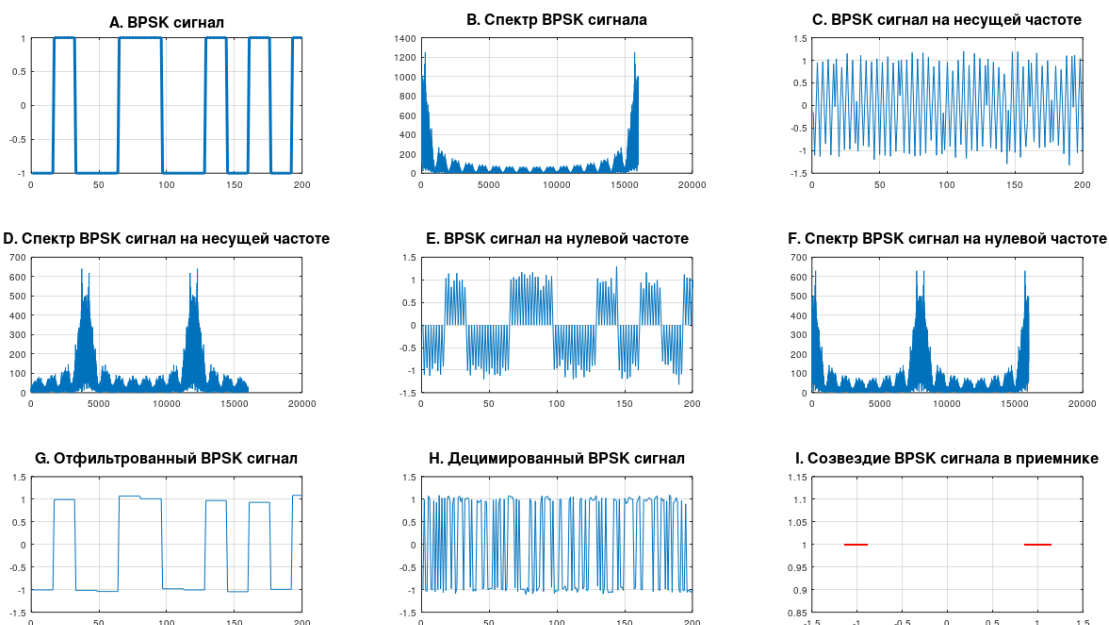


Рисунок 2.2 – Поэтапные результаты выполнения работы

Контрольные вопросы к работе:

- 1) Как произвести интерполяцию?
- 2) Как осуществляется перенос на несущую частоту?
- 3) Как осуществляется возвращение данных на нулевую частоту?

Требования к оформлению отчета по работе.

Содержание отчета:

- 1) Введение,
- 2) Блок схема алгоритма,
- 3) Поэтапное описание выполнения работы,
- 4) Рисунки:
 - BPSK сигнал,
 - BPSK сигнал на несущей частоте,
 - BPSK сигнал на нулевой частоте,
 - Отфильтрованный BPSK сигнал.
- 5) Выводы,
- 6) Листинг программы.

Работа №3 «Реализация QPSK передатчика и приёмника в Octave»

Цель работы: Изучить принцип переноса QPSK последовательности на несущую частоту.

Задачи работы:

- 1) Выполнить QPSK модуляцию сигнала;
- 2) Произвести перенос последовательности на несущую частоту;
- 3) Выполнить обработку QPSK модулированного сигнала.

Ход выполнения работы

В данной работе необходимо создать битовую последовательность, произвести QPSK модуляцию, интерполировать ее, перенести на несущую частоту. В приемнике произвести перенос на нулевую частоту, фильтрацию, децимацию, демодуляцию и проверить количество ошибок.

Формирование сигнала. Передатчик.

```
clc
clear all
close all
%Выберите размер QPSK последовательности
size = 1024;
%Установите несущую частоту
fc=10e6;
%Задайте частоту дискретизации
fs=40e6;
%Задайте коэффициент интерполяции
L = 4;
```

Сформируйте битовую последовательность.

```
DATA = randi([0 M-1], 1, size);
```

Произведите QPSK модуляцию, используя команду `qammod`.

```
MOD_DATA=qammod(. . .
```

Постройте созвездие модулированного сигнала, используя функцию `scatterplot()`. Для использования этой функции необходимо загрузить ее написав `pkg load communications` в командное окно и выполнить (рисунок 3.1).

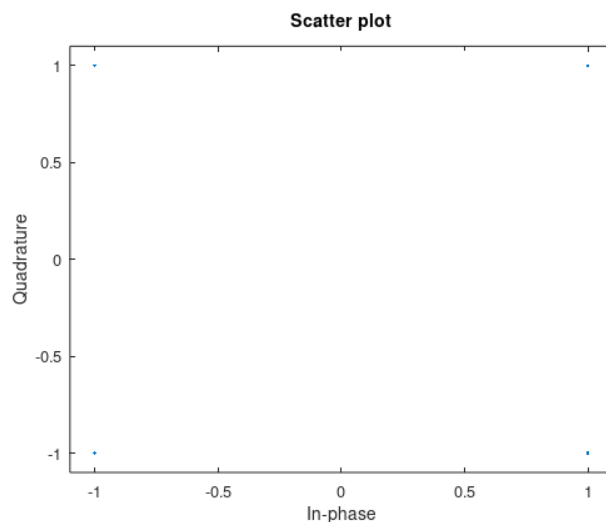


Рисунок 3.1 – Сигнальное созвездие QPSK

Выполните интерполяцию. Если MOD_DATA имеет размерность 1024 символа, на выходе интерполятора должно быть в L раз больше символов. Пример функции интерполяции представлен ниже.

```
p=1; q=1;
for k=1:size
MOD_DATA_interp(q:L*p) = MOD_DATA(k);
q=(p*L+1);
p=p+1;
end
```

Задайте временной диапазон.

```
t = (0:length(MOD_DATA_interp)-1)/fs;
```

Разделите мнимую и действительную части. Постройте их графики с помощью функции plot, опираясь на прошлые работы. Функция xlim используется для ограничения диапазона построения графика (рисунок 3.2).

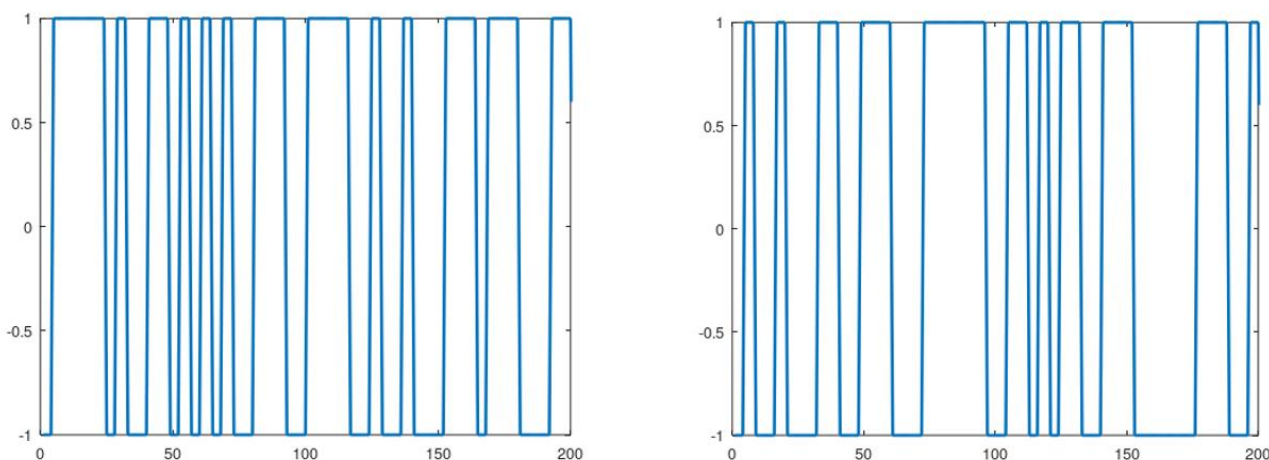


Рисунок 3.2 – Действительная и мнимая составляющая QPSK сигнала

```
Re = real(MOD_DATA_interp);
Im = imag(MOD_DATA_interp);
```

```
figure
plot(Re);
xlim([0, 200]);
```

Задайте гармонический сигнал с несущей частотой f_c .

```
car_sig_re = cos(...
car_sig_im = -sin(...
```

Перенесите мнимую и действительную части QPSK последовательности на несущую частоту, путем умножения на гармонический сигнал. Постройте их графики (Рисунок 3.3).

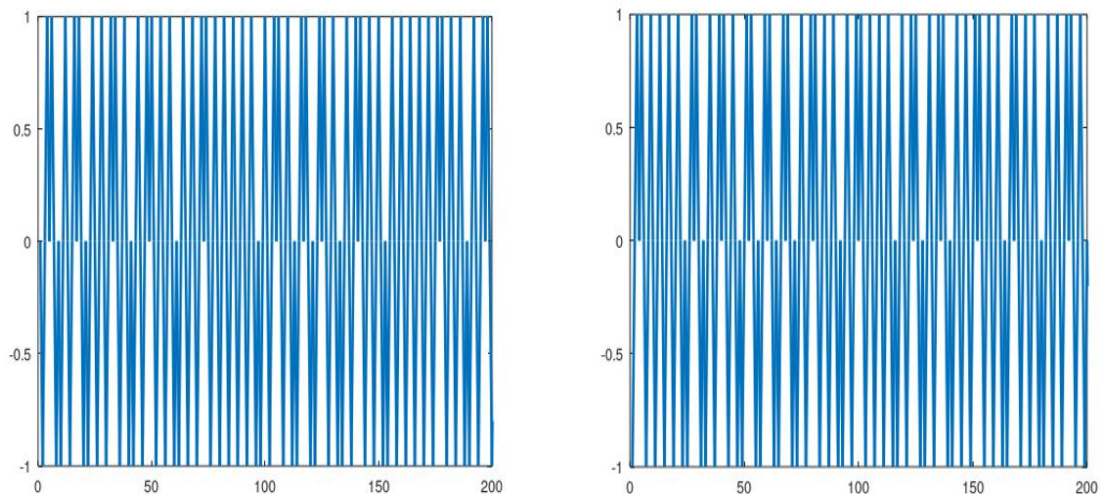


Рисунок 3.3 – Действительная и мнимая составляющая QPSK сигнала на несущей частоте

Обработка сигнала. Приёмник.

Верните QPSK последовательность на нулевую частоту, повторно перемножив мнимую и действительную части QPSK последовательность с гармоническим сигналом. Постройте их графики (Рисунок 3.4).

```
REVIVE =...
IMVIVE =...
```

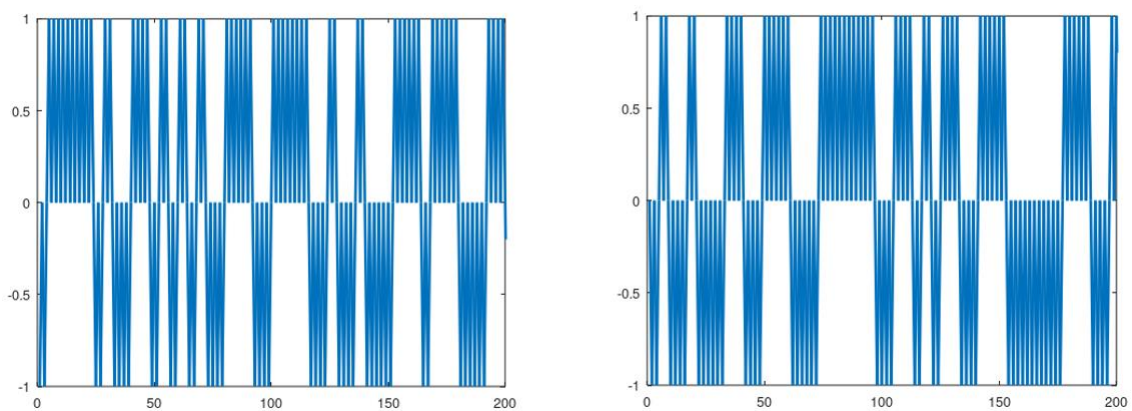


Рисунок 3.4 – действительная и мнимая составляющая QPSK сигнала на нулевой частоте

Выполните фильтрацию полученных данных на нулевой частоте путем интегрирования.

```

for i=1:length(REVIVE)/L
a(i) = sum(REVIVE(i*L-L+1:i*L))/(L/2);
y(i*L-L+1:L*i) = a(i);
end

for i=1:length(IMVIVE)/L
b(i) = sum(IMVIVE(i*L-L+1:i*L))/(L/2);
v(i*L-L+1:L*i) = a(i);
end

```

Сложите действительную и мнимую составляющие (мнимую часть сигнала необходимо умножить на мнимую единицу).

```
Y=
```

Произведите децимацию, а затем демодуляцию полученных данных. Цель демодуляции – преобразовать сигнал в битовый поток. Используйте цикл и набор условий. Постройте график децимированной последовательности.

```

decim = y(L:L:end);
demod = qamdemod(. . .

```

Посчитайте ошибки с помощью функции `biterr()`.

```
err = biterr(DATA, demod);
```

Используя функцию `scatterplot()` постройте созвездие сигнала после обработки.

Контрольные вопросы к работе:

- 1) Как произвести интерполяцию?
- 2) Как осуществляется перенос на несущую частоту?
- 3) Как осуществляется возвращение данных на нулевую частоту?

Требования к оформлению отчета по работе.

Содержание отчета:

- 1) Введение,
- 2) Блок схема алгоритма,
- 3) Поэтапное описание выполнения работы,
- 4) Рисунки:
 - QPSK сигнал,
 - QPSK сигнал на несущей частоте,
 - QPSK сигнал на нулевой частоте,
 - Отфильтрованный QPSK сигнал.
- 5) Выводы,
- 6) Листинг программы.

Работа №4 «OFDM модуляция»

Цель работы: Сформировать OFDM сигнал. Передать сигнал через канал с аддитивным белым гауссовским шумом, произвести его демодуляцию.

Задачи работы:

- 1) Составить программу для передачи OFDM сигнала в среде Octave;
- 2) Произвести модуляцию и демодуляцию сигнала;
- 3) Имитировать передачу сигнала через канал с аддитивным белым гауссовским шумом;
- 4) Построить спектр, созвездие и временное представление сигнала.

Ход выполнения работы

В данной работе необходимо реализовать схему, приведенную на рисунке 4.1.

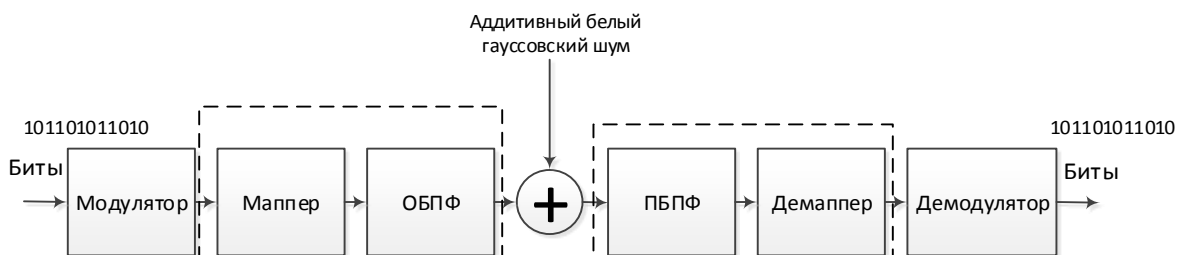


Рисунок 4.1 – Формирование сигнала и прохождение через многолучевой канал

QPSK модуляция и демодуляция.

Создайте битовую последовательность используя функцию `randi()`.

```
clc
clear all
close all
N = 5000;
bits = randi([0, 1], 1, N);
```

Произведите QPSK модуляцию, используя цикл `for` и условие `if`. Ниже приведены 2 условия из 4.

```
k=1;
for i = 1:2:N
    if bits(i) == 1 && bits (i+1) == 1
        mod_data (k) = 1+1i;
    end
    if bits(i) == 1 && bits (i+1) == 0
        mod_data (k) = 1-1i;
    end
    ...
    k=k+1;
end
```

Постройте созвездие модулированного сигнала, используя функцию `scatterplot()`. Для использования этой функции необходимо загрузить ее, написав `pkg load communications` в командное окно, и выполнить.

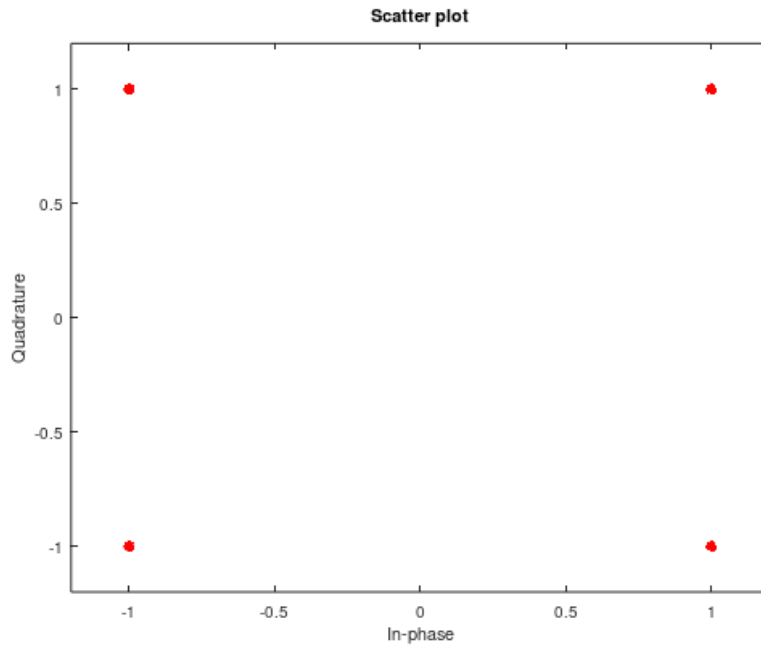


Рисунок 4.2 – Созвездие QPSK сигнала

Используя функцию `awgn()`, добавьте шум к модулированному сигналу. Отношение сигнал/шум $\text{SNR} = 15\text{dB}$.

```
SNR = 15;
mod_data_noise = awgn(mod_data,SNR, 'measured');
```

Постройте созвездие сигнала с шумом, используя функцию `scatterplot()`.

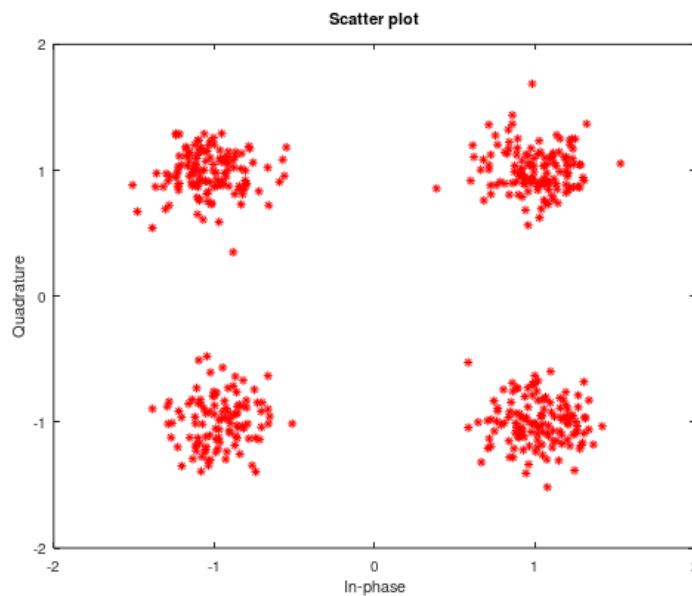


Рисунок 4.3 – Зашумленной воздействие QPSK сигнала

Измените значение SNR с 5, 10, 15, 20 dB. Отобразите сигнальные созвездия для этих значений в отчете.

Осуществите демодуляцию, ниже приведены только 2 условия из 4. Добавьте еще 2 условия, необходимых для демодуляции.


```

k=1;
for i = 1:length(mod_data_noise)
    if real(mod_data_noise (i)) >0 && imag(mod_data_noise (i)) >0
        bits_demod (k) = 1; bits_demod (k+1) = 1;
    end
    if real(mod_data_noise (i)) <0 && imag(mod_data_noise (i)) >0
        bits_demod (k) = 0; bits_demod (k+1) = 1;
    end
    ...
    k=k+2;
end

```

Сравните массивы bits и bits_demod, они должны совпадать.

Формирование OFDM сигнала.

Используя сформированную битовую последовательность и модулятор, произведите формирование OFDM сигнала.

Задайте следующие параметры:

- Размер преобразования Фурье - 1024;
- Защитные интервалы - 100 отсчетов слева и 100 отсчетов справа;
- Поднесущие с данными - с 101 по 924, (количество поднесущих с данными: 824);
- Модуляция – QPSK;
- Разместите поднесущие с модулированными данными, так как показано на рисунке

4.4.

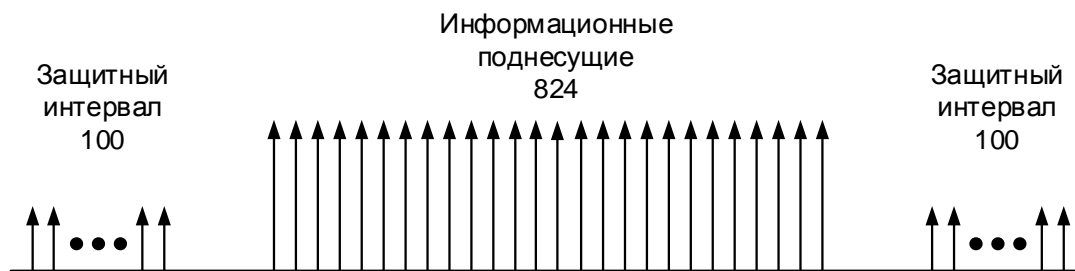


Рисунок 4.4 – Расположение поднесущих в OFDM символе

Создайте нулевой массив:

```
spectrum = zeros(1,1024);
```

Разместите поднесущие с модулированными данными, так как показано на рисунке

3.4.

```
spectrum(101:924) = mod_data(1:824);
```

Постройте спектр сформированного OFDM сигнала.

```
figure
stem(abs(spectrum));
```

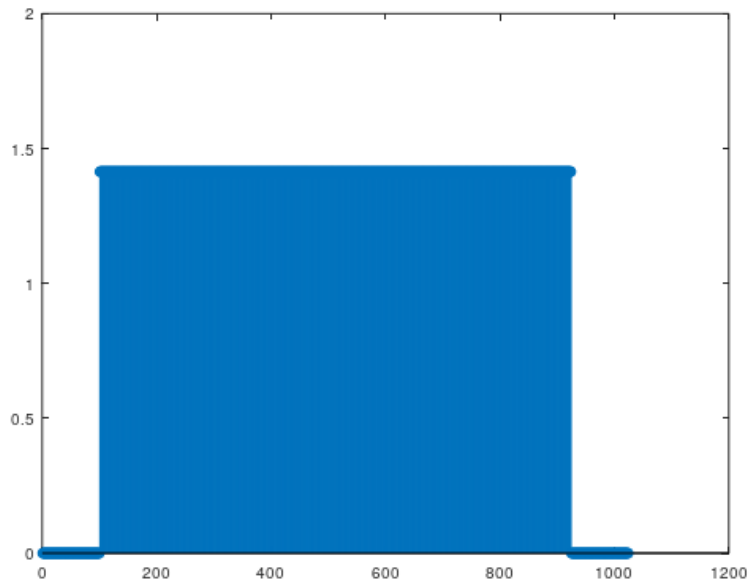


Рисунок 4.5 – Спектр OFDM символа

Сформируйте OFDM сигнал во временной области, используя функцию `ifft()`.

```
sig_time = ifft(spectrum);
```

Постройте на рисунке OFDM сигнал во временной области.

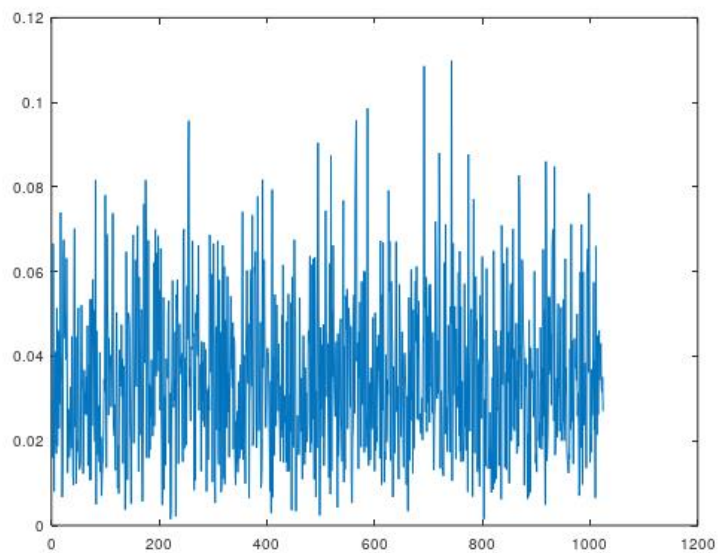


Рисунок 4.6 – OFDM символ во временной области

Канал передачи.

Используя функцию `awgn()`, добавьте аддитивный шум к сигналу. Установите отношение сигнал/шум 15 дБ.

```
SNR = 15;
sig_noise = awgn(sig_time,SNR, 'measured');
```

Приемник.

Перейдите к частотной области, используя функцию `fft()`. Постройте на рисунке спектр принятого сигнала. Выберите только поднесущие с данными.

```
spectrum_r = fft(sig_noise);  
mod_data_r = spectrum_r(101:924);
```

Постройте созвездие, используя функцию `scatterplot()`.

```
scatterplot(mod_data_r,1,0,"r*");
```

Произведите демодуляцию используя цикл `for` и условия `if`. Сравните массивы `bits` и `bits_demod`, они должны совпадать.

Контрольные вопросы к работе:

- 1) Что такое OFDM сигнал?
- 2) Для чего нужны защитные интервалы в частотной области?
- 3) Как связаны временное и частотное представление сигнала?
- 4) Формирование OFDM символа производится во временной или частотной области?

Требования к оформлению отчета по работе.

Содержание отчета:

- 1) Введение,
- 2) Блок схема алгоритма, который вы применили в работе,
- 3) Параметры OFDM символа,
- 4) Рисунки:
 - Созвездие QPSK сигнала при разном отношении сигнал шум,
 - Спектр OFDM символа,
 - OFDM символ во временной области,
- 5) Выводы,
- 6) Листинг программы.

Работа №5 «Оценка канала связи/эквалайзирование»

Цель работы: разработать модель системы передачи данных с использованием OFDM модуляции и передачи данных через многолучевой канал.

Задачи работы:

- сформировать OFDM сигнал;
- произвести передачу сигнала через канал с многолучевостью;
- произвести оценку канала связи, эквалайзирование, демодуляцию сигнала.

Ход выполнения работы

В данной работе необходимо реализовать схему, приведенную на рисунке 5.1.

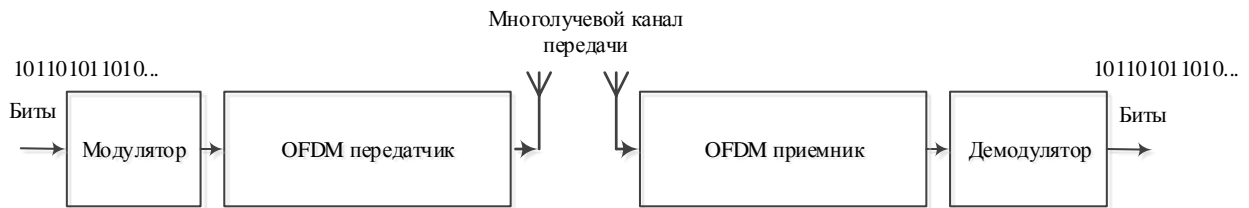


Рисунок 5.1 – Формирование сигнала и прохождение через многолучевой канал

Параметры сигналов приведены в таблице 5.1.

Таблица 5.1 – Параметры сигналов

Параметр	Обозначение	Значение	Пояснения
Размер преобразования Фурье	FFT_size	1024	Общее количество отсчетов на символ
Защитные интервалы	GI	100	100 отсчетов слева и 100 отсчетов справа
Поднесущие с данными	used_subcar	824	С 101 по 924
Модуляция информационного символа	index_mod_data	16	16-QAM
Модуляция пилотного символа	index_mod_pilot	4	QPSK

Параметры канала передачи приведены в таблице 5.2.

Таблица 5.2 – Многолучевого канала передачи

Номер луча	Задержка, отсч.	Ослабление, раз
1	0	1
2	2	0.7
3	4	0.5

Формирование сигнала. Передатчик.

Сформируйте 2 OFDM символа, так как это показано на рисунке 5.2. Первый из них – пилот сигнал, следующий – символ для передачи данных.

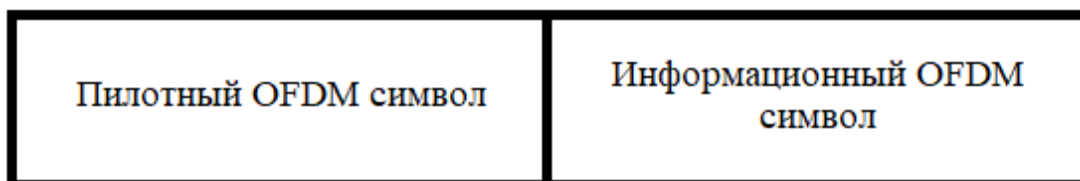


Рисунок 5.2 – Структура формирования сигнала

Создаем 2 битовые последовательности.

```
used_subcar = ...;  
index_mod_pilot = ...;  
index_mod_data = ...;  
data_pilot = randint(1, used_subcar, index_mod_pilot);  
data_user = randint(...);
```

Далее выполните модуляцию сигнала.

```
mod_data_pilot = qammod(data_pilot, index_mod_pilot);  
mod_data_user = qammod(...);
```

Постройте созвездия сигналов, используя функцию scatterplot().

```
scatterplot(mod_data_user);  
scatterplot(...);
```

Созвездие для информационного символа показана на рисунке 5.3.

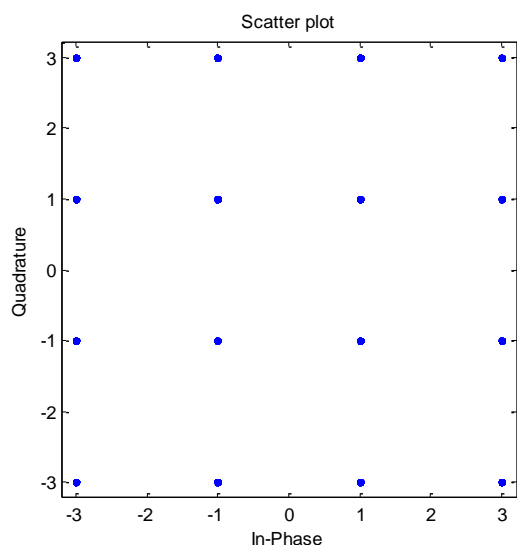


Рисунок 5.3 – Созвездие информационного символа

Следующий шаг – формирование спектра: Расположите поднесущие, как показано на рисунке 5.4.



Рисунок 5.4 – Формирование спектра

Создайте массив нулевых элементов:

```
spectrum_pilot = zeros(1,1024);  
spectrum_user = zeros(1,1024);
```

Расположите поднесущие, как показано на рисунке 5.4.

```
spectrum_pilot(101:924) = mod_data_pilot;  
spectrum_user(101:924) = mod_data_user;
```

Постройте спектры сигналов, используя функцию plot() и abs().

```
figure  
plot(abs(spectrum_user));  
...
```

Графическое отображение спектра сигнала с информационными данными представлено на рисунке 5.5.

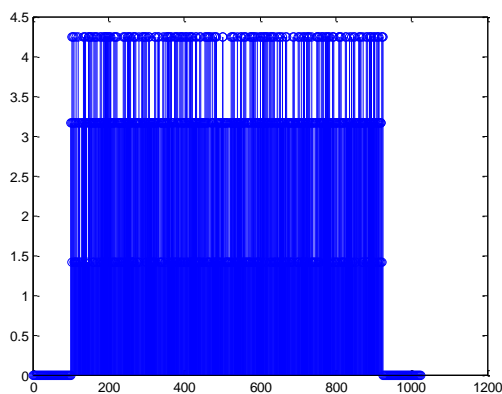


Рисунок 5.5 – Спектр информационного OFDM символа

Перейдите к временной области, используя функцию ifft().

```
sig_time_user = ifft(spectrum_user);  
sig_time_pilot = ...;
```

Объедините два OFDM символа в один массив.

```
sig_time = [sig_time_pilot, sig_time_user];
```

Формирование канала передачи.

Далее выполните моделирование многолучевого канала распространения радиоволн. Установите 3 коэффициента ослабления согласно таблице 5.2

```
k1 = ...;  
k2 = ...;  
k3 = ...;
```

Сформируйте 3 сигнала. Первый – прямой с нулевой задержкой и ослаблением k1. Второй с задержкой в 2 отсчета и ослаблением k2. Третий с задержкой в 4 отсчета и ослаблением k3.

```
sig1 = [sig_time,zeros(1,4)]*k1;  
sig2 = [zeros(1,2),sig_time,zeros(1,2)]*k2;  
sig3 = [zeros(1,4),sig_time]*k3;
```

Далее сигналы суммируются в приемной антенне.

```
sig_time_channel = sig1+sig2+sig3;
```

Используя функцию `awgn()`, добавьте шум в сформированный сигнал. Отношение сигнал/шум установите равным 25 дБ.

```
SNR = 25;  
sig_recive_noise = awgn(sig_time,SNR, 'measured');
```

Обработка сигнала. Приёмник.

Запишите в двух разных массива пилотный сигнал и сигнал с данными.

```
pilot_recive = sig_recive_noise(1:1024);  
user_data_recive = sig_recive_noise(1025:2048);
```

Используя `fft` функцию получите спектр пилотного символа и символа с данными.

```
spectrum_pilot_r = fft(pilot_recive);  
spectrum_user_data_r = fft(user_data_recive);
```

Постройте спектры принятых символов.

```
figure  
plot(abs(spectrum_user_data_r));  
...
```

Спектр сигнала после канала показан на рисунке 5.6.

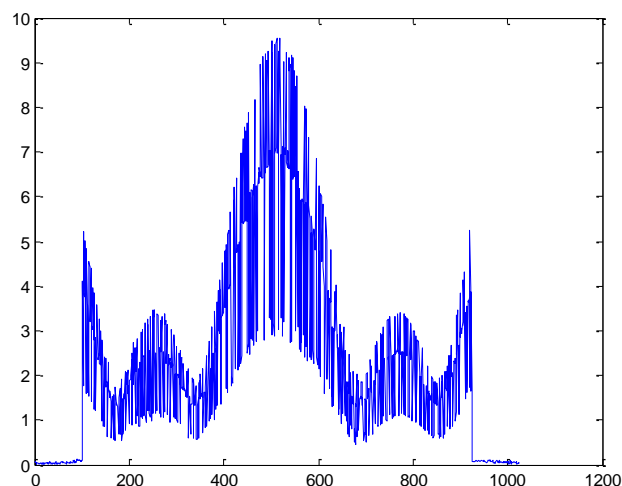


Рисунок 5.6 – Спектр сигнала, прошедшего через канал

Удалите защитные интервалы в частотной области и выберите только поднесущие с данными.

```
pilot_subcarriers = spectrum_pilot_r(101:924);  
user_data_subcarriers = ...;
```

Постройте созвездие сигнала, используя функцию `scatterplot()`.
Создайте опорный сигнал.

```
reference_subcarriers = mod_data_pilot;
```

Произведите оценку передаточной функции канала связи.

```
TF_estimate = pilot_subcarriers./reference_subcarriers;
```

Постройте передаточную функцию канала распространения радиоволн.
Произведите операцию эквалайзирования, путем деления спектра принятого сигнала на передаточную функцию.

```
user_data_estimate = user_data_subcarriers./TF_estimate;
```

Выведите на рисунок созвездие сигнала после эквалайзирования.
Выполните демодуляцию сигнала.

```
data_demod = qamdemod(user_data_estimate, index_mod_data);
```

Сравните массивы: `data_user` и `data_demod`. Они должны совпадать.

Контрольные вопросы к работе

- 1) Что такое пилотный сигнал?
- 2) Что такое передаточная функция канала РРВ?
- 3) Как влияет многолучевость в канале на передаваемый сигнал?
- 4) Что такое эквалайзирование?

Требования к оформлению отчета по работе.

Содержание отчета:

- 1) Введение,
- 2) Блок схема алгоритма, который вы применили в работе,
- 3) Параметры многолучевого канала,
- 4) Рисунки:
 - Спектр сигнала в передатчике,
 - Спектр сигнала после прохождения многолучевого канала,
 - Созвездие сигнала до эквалайзера,
 - Созвездие сигнала после эквалайзера.
- 6) Выводы.

Работа №6 «Кадровая синхронизация»

Цель работы: определить начало кадра в канале с случайной задержкой.

Задачи работы:

- Сформировать кадр из 2 OFDM символов;
- Произвести передачу сигнала через канал с случайной задержкой;
- Произвести оценку начала кадра.

Ход выполнения работы

В данной работе необходимо реализовать схему, приведенную на рисунке 6.1.



Рисунок 6.1 – Формирование сигнала и прохождение через многолучевой канал

Параметры OFDM символов приведены в таблице 6.1.

Таблица 6.1 – Параметры сигналов

Параметр	Обозначение	Значение	Пояснения
Размер преобразования Фурье	NFFT	1024	Общее количество отсчетов на символ
Защитные интервалы	GI	100	100 отсчетов слева и 100 отсчетов справа
Поднесущие с данными	used_subcar	824	С 101 по 924
Модуляция информационного символа	Sig_time_user		16-QAM
Модуляция пилотного символа	Sig_time_pilot		QPSK

Ваша задача сформировать 2 разных OFDM символа. Первый из них – пилотный, следующий символ с данными.



Рисунок 6.2. – Структура сигнала

Формирование сигналов.

Сформируйте два различных OFDM символа и объедините их в один массив во временной области.

```
sig_time = [sig_time_pilot, sig_time_user];
```

Добавляем случайную задержку. Используем для этого функцию randint.

```
delay = randint(1,1, 200);
```

Добавляем в начало массива нулевые отсчеты, количество которых равно случайному значению.

```
sig_time_shift = [..., sig_time];
```

Добавляем АБГШ, $\text{SNR} = 30$ дБ.

Далее, рассчитываем взаимокорреляционную функцию между пилотным сигналом и принятым сигналом с задержкой.

```
sig_time_pilot = [sig_time_pilot,zeros(1,delay+1024) ];  
korr = ifft(fft(sig_time_shift).*conj(fft(sig_time_pilot)));
```

Пример ВКФ приведен на рисунке 6.3.

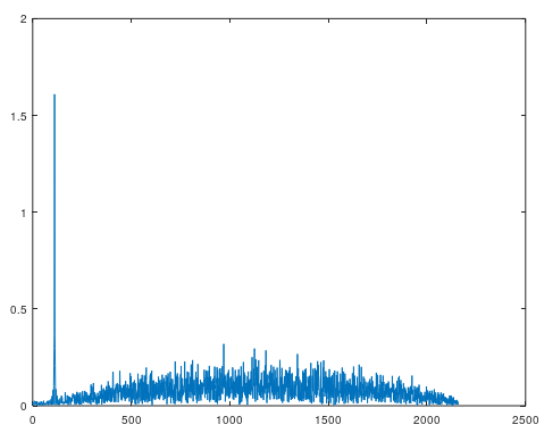


Рисунок 6.3 – ВКФ

Используя функцию `max`, находим максимальное значение ВКФ:

```
[value, position] = max(...);
```

Используем положение максимума ВКФ для определения начала символа с данными.

```
user_data_recive = sig_time_shift(position+1024:end);
```

Постройте спектр и созвездие сигнала пользователя.

```
spectrum_user_data_r = fft(.....);  
scatterplot(.....)  
...
```

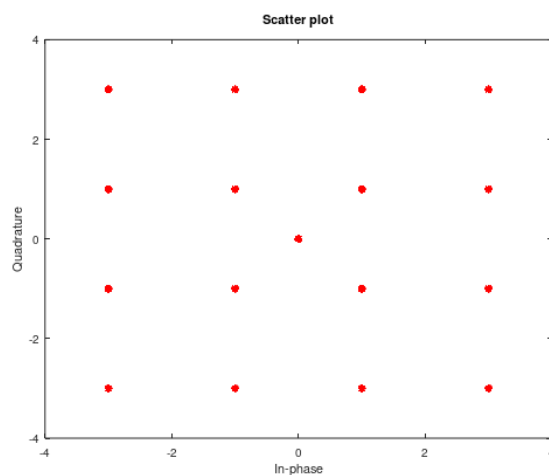


Рисунок 6.4 – Созвездие OFDM символа

Добавьте ошибку в один отсчет при определении начала OFDM символа, постройте созвездие.

```
user_data_recive = sig_time_shift(position-1+1024:end-1);  
spectrum_user_data_r = fft(user_data_recive);  
scatterplot(spectrum_user_data_r);...
```

Созвездие сигнала при ошибке синхронизации показано на рисунке 6.5.

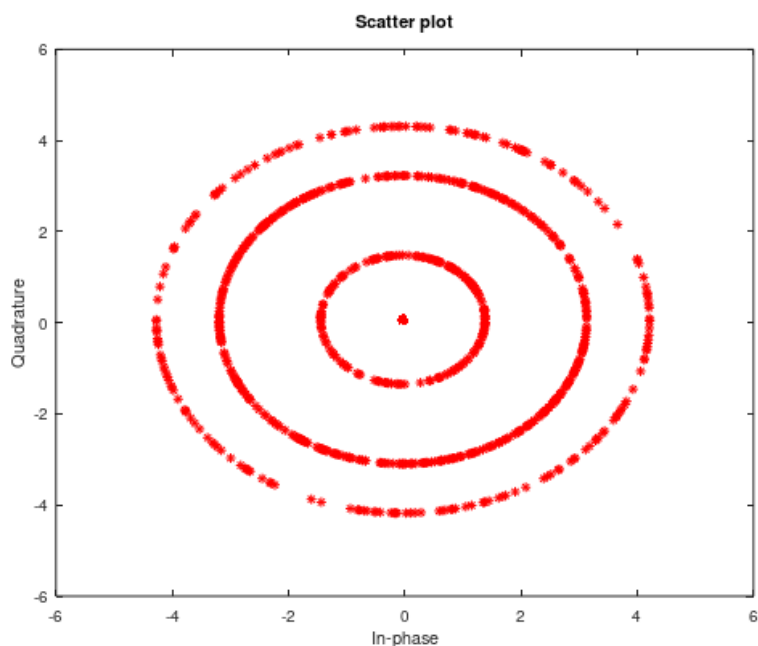


Рисунок 6.5 – Созвездие при ошибке синхронизации

Контрольные вопросы к работе:

- 1) Что такое кадровая синхронизация?
- 2) К чему приводит ошибка кадровой (временной) синхронизации?
- 3) Что такое циклический префикс?
- 4) Каково назначение пилотного символа?

Требования к оформлению отчета по работе.

Содержание отчета:

- 1) Введение,
- 2) Блок схема алгоритма, который вы применили в работе,
- 3) Рисунки:
 - Корреляционная функция,
 - Созвездие при правильной синхронизации,
 - Созвездие сигнала при ошибке в 1 отсчет.
- 4) Выводы,
- 5) Листинг программы.

Работа №7 «Частотная синхронизация»

Цель работы: произвести оценку и устранение частотного сдвига в системе связи с OFDM модуляцией.

Задачи работы:

- 1) Сформировать кадр из 2 OFDM символов;
- 2) Ввести частотный сдвиг;
- 3) Произвести оценку частотного сдвига;
- 4) Произвести коррекцию частотного сдвига.

Ход выполнения работы

В данной работе необходимо ввести частотный сдвиг в OFDM сигнал. Затем в приемнике произвести оценку частотного сдвига и произвести коррекцию частотного сдвига.

Параметры сигнала:

- Количество OFDM символов: 2;
- Размер преобразования Фурье: 1024;
- Модуляция: QAM 16;
- Значение частотного сдвига: 500 Гц.

Поэтапное выполнение.

Сформируйте 2 OFDM символа, далее задайте величину частотного сдвига.

```
clc
clear all
close all
NFFT=1024
M = 16;
data = randint(1,824,M);
data_mod = qammod(data,M);
sp_ofsm = [zeros(1,100), data_mod, zeros(1,100)];
sig_vr=ifft(sp_ofsm);
sig_sum=[sig_vr, sig_vr];
% Вводим частотный сдвиг
NN=NFFT*2;
fs=10.24e6;
T=NN/fs;
D_Freq=500;
```

Рассчитываем фазовый набег за время 2 OFDM символов.

```
D_Phi=D_Freq*2*pi*T;
```

Рассчитываем фазовый набег за интервал дискретизации.

```
dphi=D_Phi/NN;
```

Вводим частотный сдвиг.

```
for j=1:NN
sig_freq_shift(j)=sig_sum(j)*exp(1i*dphi*j);
end
```

Добавляем аддитивный шум 35 дБ.

```
sig_freq_shift1=awgn(...);
```

Выбираем отсчеты первого и второго OFDM символа.

```
Pr_sym1=sig_freq_shift1(...);  
Pr_sym2=sig_freq_shift1(NFFT+1:2*NFFT);
```

Построим созвездие первого принятого символа 7.1.

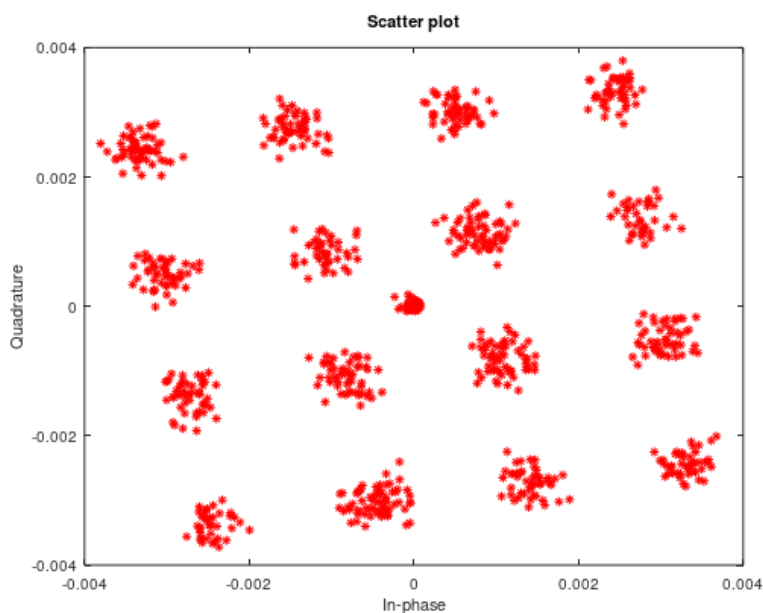


Рисунок 7.1 – Созвездие первого принятого символа

Расчет ВКФ между вторым и первым, принятым OFDM символом.

```
corr=xcorr(Pr_sym2, ...);  
[a,b]=max(corr);
```

Оценка фазового набега.

```
Dphi=angle(a);  
Dt=1/fs; %расчет длительности символа  
Tau=(NFFT)*dt;  
Ocen_freq=Dphi/(2*pi*tau);
```

Сравните оценку и вводимый частотный сдвиг, они должны быть близки по величине. Далее рассчитайте фазовый набег за интервал дискретизации по полученной оценке.

```
d_phi_ocen=(ocen_freq*2*pi)/fs;
```

Компенсируем частотный, вводя фазовый набег, но с отрицательным знаком.

```
for j=1:1024  
Pr_sym1_komp(j)=Pr_sym1(j)*exp(1i*j*(-...));  
end
```

Постройте созвездие первого символа после компенсации.

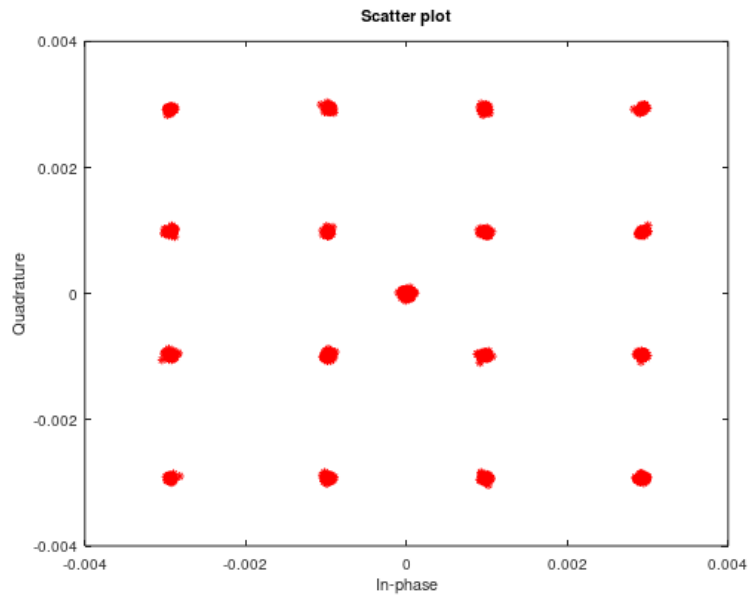


Рисунок 7.2 – Созвездие первого символа после компенсации

Контрольные вопросы к работе:

- 1) Что такое частотная синхронизация?
- 2) К чему приводит ошибка частотной синхронизации?
- 3) Какие сигнальные конструкции используются для частотной синхронизации?
- 4) Как компенсировать частотный сдвиг в цифровом виде?

Требования к оформлению отчета по работе.

Содержание отчета:

- 1) Введение,
- 2) Блок схема алгоритма, который вы применили в работе,
- 3) Рисунки:
 - Корреляционная функция,
 - Созвездие после введения частотного сдвига;
 - Созвездие после устранения частотного сдвига;
- 4) Выводы;
- 5) Листинг программы.

Работа №8 «Технология MIMO»

Цель работы: произвести моделирование технологии MIMO с пространственным мультиплексированием для 2 передающих и 2 приемных каналов.

Задачи работы:

- 1) Сформировать 2 OFDM символа;
- 2) Задать 4 канал связи (прямые и перекрестные);
- 3) Добавить аддитивный белый гауссовский шум;
- 4) Произвести обработку MIMO сигнала для каждого канала.

Ход выполнения работы

В данной работе необходимо реализовать пространственное разделение каналов применяя OFDM модуляцию. Сформируйте 2 OFDM символа с QPSK и QAM-16.

Выполнение работы.

Сформируем 2 OFDM символа во временной области, первый с модуляцией QPSK, второй с модуляцией QAM 16 опираясь на материал прошлых работ.

```
clc
clear all
close all
NFFT = 1024;
index_mod1 = 4;
index_mod2 = 16;
sym1 = ...;
sym2 = ...;
```

Формируем 4 передаточные функции для канальной матрицы.

```
h11 = zeros(1,NFFT);
h11(1) = 1;
h11(2) = 0.5;
h11(5) = 0.3;
H11 = fft(h11);
h12 = zeros(1,NFFT);
h12(1) = 1;
h12(2) = 0.5;
h12(4) = 0.3;
H12 = fft(h12);
h21 = zeros(1,NFFT);
.
.
.
H21 = fft(h21);
h22 = zeros(1,NFFT);
.
.
.
H22 = fft(h22);
```

Записываем сигналы принимаемые первой и второй приемной антеннами MIMO 2x2 системы.

```
y1 = ifft((sym1).*H11) + ifft((sym2).*H12);  
y2 = . . .;
```

Имитируем передачу двух независимых сигналов в случае SISO системы.

```
y_asiso1 = ifft((sym1).*H11);  
y_asiso2 = ifft((sym2).*H22);
```

Добавляем шум и возвращаем сигналы в частотную область.

```
snr = 35;  
y1 = awgn(y1, snr, 'measured');  
y2 = awgn(y2, snr, 'measured');  
y_asiso1 = awgn(y_asiso1, snr, 'measured');  
y_asiso2 = awgn(y_asiso2, snr, 'measured');  
  
% рассчитаем спектр принятого SISO сигнала  
sp_asiso1 = fft(y_asiso1);  
sp_asiso2 = fft(y_asiso2);  
  
% Разбираем сигналы в приемнике  
sp_y1 = fft(y1);  
sp_y2 = fft(y2);
```

Найдем определитель матрицы и выполним оценку.

```
H_det = H11.*H22-H21.*H12;  
x=1./(H_det);  
x1_ocen = H22.*x.*sp_y1 + (-1)*H12.*x.*sp_y2;  
x2_ocen = (-1)*H21.*x.*sp_y1 + H11.*x.*sp_y2;  
x1_asiso = sp_asiso1./H11;  
x2_asiso = sp_asiso2./H22;
```

Постройте созвездия 1-го и 2-го сигнала для MIMO и для SISO до и после оценок.

```
% созвездия до оценок MIMO  
scatterplot(sp_y1,1,0,"r*") %  
scatterplot(sp_y2,1,0,"r*") %  
% созвездия после оценок MIMO  
scatterplot(x1_ocen,1,0,"r*") %  
scatterplot(x2_ocen,1,0,"r*") %  
% созвездия до оценок SISO  
scatterplot(sp_asiso1,1,0,"r*") %  
scatterplot(sp_asiso2,1,0,"r*") %  
% созвездия после оценок SISO  
scatterplot(x1_asiso,1,0,"r*") %  
scatterplot(x2_asiso,1,0,"r*")
```

Измените параметры канальной матрицы, посмотрите, как меняются созвездия.

Контрольные вопросы к работе:

- 1) Что такое частотная синхронизация?
- 2) К чему приводит ошибка частотной синхронизации?
- 3) Какие сигнальные конструкции используются для частотной синхронизации?
- 4) Как компенсировать частотный сдвиг в цифровом виде?

Требования к оформлению отчета по работе.

Содержание отчета:

- 1) Введение,
- 2) Схема системы ММО,
- 3) Блок схема алгоритма, который вы применили в работе,
- 4) Параметры канальной матрицы,
- 5) Рисунки:
 - Созвездия принятых сигналов до и после обработки.
- 6) Выводы,
- 7) Листинг программы.

Работа №9 «Кодовое разделение каналов»

Цель работы: произвести моделирование системы связи с кодовым разделением каналов.

Задачи работы:

- 1) Сформировать битовые последовательности, произвести их модуляцию.
- 2) Используя коды Уолша, произвести кодирование модулированных данных.
- 3) Пропустить сигнал через канал с АБГШ.
- 4) Произвести разделение сигналов в приемнике.

Ход выполнения работы

В данной работе необходимо реализовать схему, приведенную на рисунке 9.1.

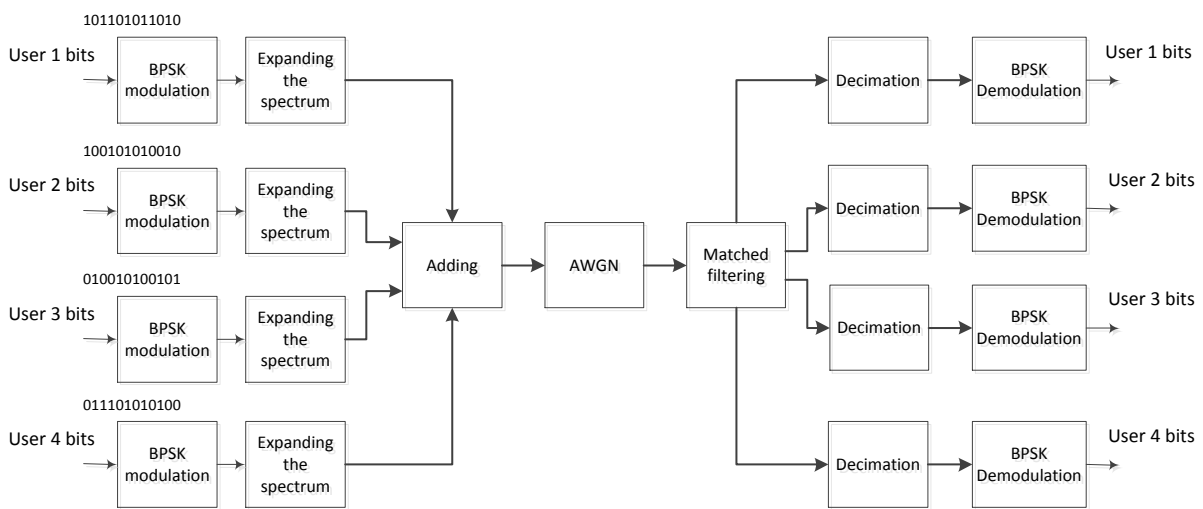


Рисунок 9.1 – Кодовое разделение каналов

Параметры сигналов:

- 1) Количество пользователей – 4;
- 2) Количество бит – 100;
- 3) Модуляция – BPSK.

Этапы выполнения работы.

Сформируйте битовую последовательность для каждого пользователя.

```
N = 100;  
User_id = 3;  
N_users = 4;  
k=4;  
bits1 = randint(1,N);  
.  
.  
.  
.  
bits4 = randint(1,N);
```

Произведите BPSK модуляцию.

```
for i=1:N
    if bits1(i) == 1
        mod_bits1(i) = 1;
    else
        mod_bits1(i) = -1;
    end
    .
    .
    .
    .
    if bits4(i) == 1
        mod_bits4(i) = 1;
    else
        mod_bits4(i) = -1;
    end
end
```

Постройте созвездия, используя функцию scatterplot().

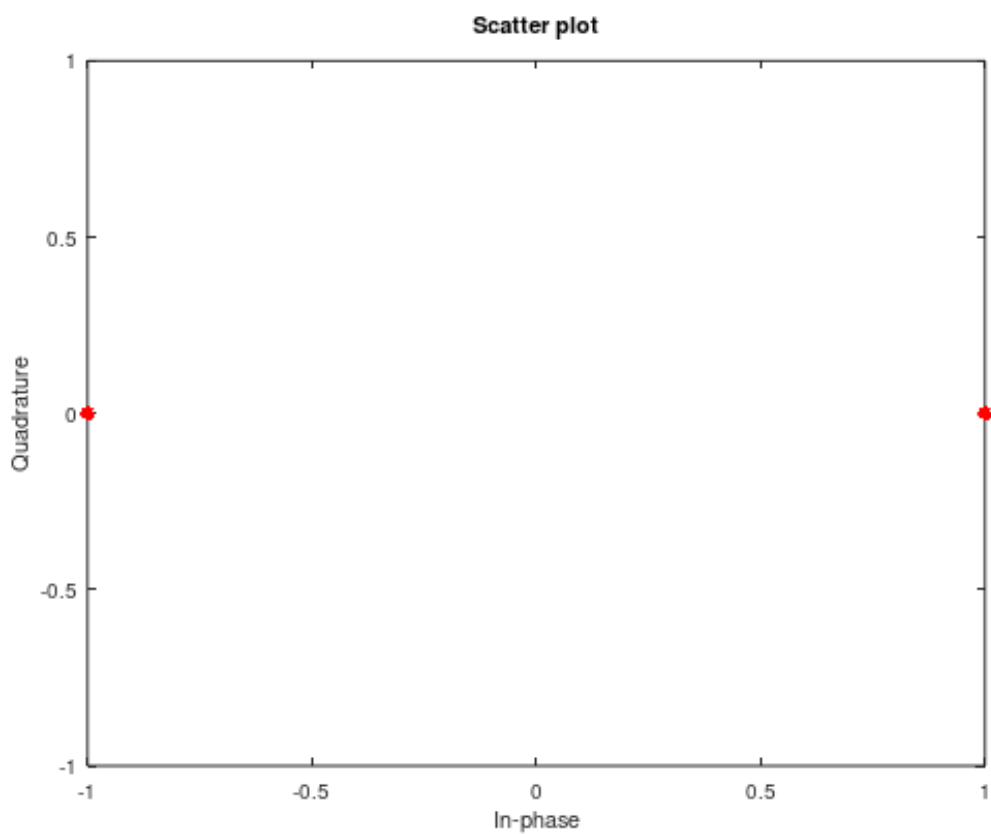


Рисунок 9.2 – Созвездие BPSK сигнала

Сформируйте матрицу Адамара.

```
mtx=hadamard(N_users);
```

Произведите кодирование сформированных данных, используя функции Уолша.

```
j=1;
for i=1:N
    if mod_bits1 (i) == 1
        mod_bits_1_code (j:j+k-1)= mtx(1,:);
    else
        mod_bits_1_code (j:j+k-1)= mtx(1,)*(-1);
    end
    .
    .
    .
    if mod_bits4 (i) == 1
        mod_bits_4_code (j:j+k-1)= mtx(4,:);
    else
        mod_bits_4_code (j:j+k-1)= mtx(4,)*(-1);
    end
    j=j+k;
end
```

Постройте на рисунке один из сигналов, используя функцию plot. На рисунке 9.3 показан пример модулированного сигнала.

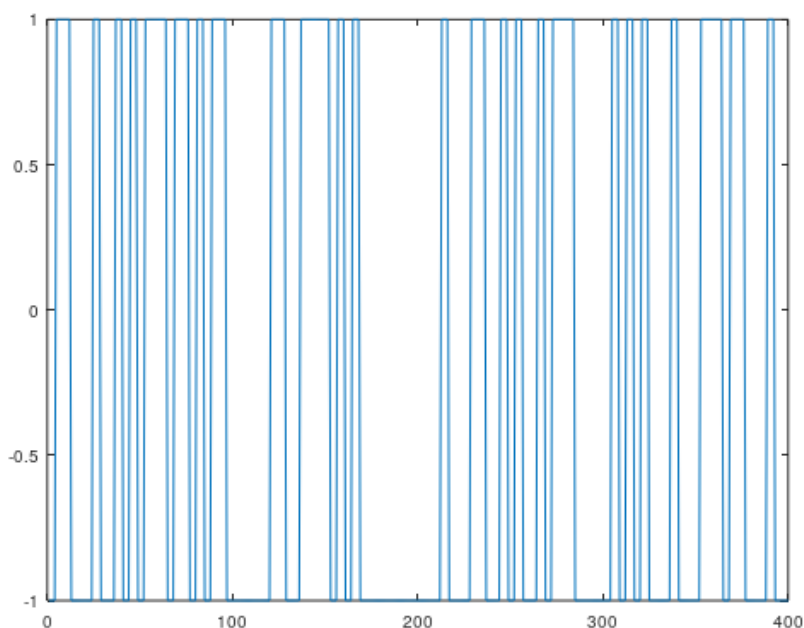


Рисунок 9.3 – Модулированный сигнал

Следующий этап – сложение данных всех пользователей.

```
sum_of_signals = mod_bits_1_code+mod_bits_2_code+...;
...
```

Постройте суммарный сигнал, как показано на рисунке 9.4.

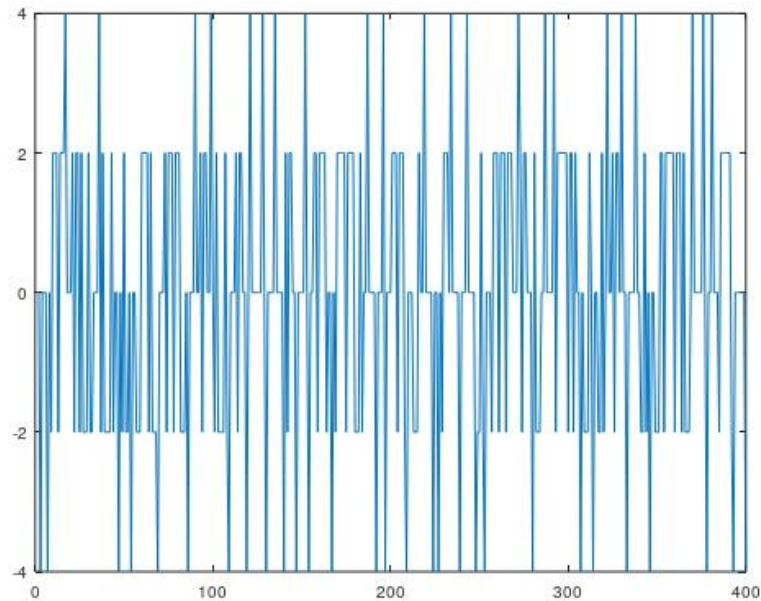


Рисунок 9.4 – Суммарный сигнал от четырёх пользователей

Добавьте аддитивный белый гауссовский шум.

```
sum_of_signals_noise = ...
```

Постройте сигнал после добавления шума.

В приемной части производим корреляционный прием, в качестве опорного сигнала используйте код одного из пользователей.

```
B_long = zeros(1,length(mod_bits_1_code));
B_long(1:k)=mtx(User_id,:);
korr = ifft(fft(sum_of_signals_noise).*conj(fft(B_long)));
```

Выполните децимацию сигнала.

```
for i=1:k:N*k
    korr_decim(j) = korr(i)/k;
    j=j+1;
end
```

Демодулируйте принятый сигнал.

```
for i=1:N
    if korr_decim(i)...
        .
        .
        .
    end
```

Посчитайте количество ошибок после демодуляции.

```
num_errors = ...
```

Контрольные вопросы к работе:

- 1) Что такое матрица Адамара?
- 2) Что такое коды Уолша?
- 3) Как разделяются сигналы в приемнике при кодовом разделении каналов?

Требования к оформлению отчета по работе.

Содержание отчета:

- 1) Введение,
- 2) Блок схема алгоритма, который вы применили в работе,
- 3) Рисунки:
 - Переданные сигналы,
 - Суммарный сигнал,
 - Принятый сигнал одного из пользователей.
- 4) Выводы,
- 5) Листинг программы.

Работа №10
«Кодирование Аламоути в технологии MIMO»

Цель работы: Произвести моделирование MIMO технологии для двух передающих и приемных каналов с кодированием Аламоути.

Задачи работы:

- 1) Сформировать два OFDM символа.
- 2) Задать MIMO и SISO каналы связи.
- 3) Реализовать схему передачи при помощи кодирования Аламоути.
- 4) Построить спектр и созвездие сигнала.

Ход выполнения работы

В данной работе необходимо произвести кодирование Аламоути для MIMO технологии и сравнить влияние SNR на число ошибок для данного способа передачи данных и для SISO технологии. Проанализировать результаты и сделать выводы.

Задайте исходные данные.

```
NFFT = 1024;  
index_mod1 = 4;  
index_mod2 = 4;  
gi = 199;  
conv = (NFFT-gi-1);  
snr = 20;  
...  
...
```

Задайте битовую последовательность.

```
data_bit = ...
```

Осуществите QPSK модуляцию data_bit.

```
modqpsk = ...
```

Сформируйте два OFDM символа.

```
sym1 = ...  
sym2 = ...
```

Задайте еще два OFDM символа следующим образом.

```
sym1z = conj(sym1);  
sym2z = -1.*conj(sym2);
```

Сформируйте 4 передаточные функции для канальной матрицы.

```
h11 = zeros(1,NFFT);  
h11(1) = 1;  
h11(2) = 0.5;  
h11(5) = 0.3;  
H11 = fft(h11);
```

```

h12 = zeros(1,NFFT);
h12(1) = 1;
h12(2) = 0.5;
h12(4) = 0.3;
H12 = fft(h12);
...
...
...

```

Запишите сигналы, принимаемые 1 и 2 антенной.

```

y_asiso1 = ifft((sym1).*H11);
y_asiso2 = ...

```

Добавьте к ним белый шум.

```

y_asiso1=...
y_asiso2=...

```

Выполнение кодирования и декодирования Аламоути.

Реализуйте передачу символов, используя кодирование Аламоути.

Кодирование Аламоути заключается в том, что мы одновременно передаем в момент времени $t1$ первый символ, а вместе с ним комплексно-сопряженный второму с отрицательным знаком. В момент времени $t2$ мы передаем второй символ и комплексно сопряженный первому символу символ.

При таком способе передаче скалярное произведение первой переданной последовательности и второй равно нулю. Что позволяет лучше различать сигналы в приемнике и соответственно повысить помехозащищенность системы передачи данных, а также уменьшить влияние шума. В данной работе реализуется передача 2 на 2.

Сформируйте сигналы, принимаемые антеннами 1 и 2 в промежутки времени $t1$ и $t2$.

```

y1 = ifft((sym1).*H11) + ifft((sym2).*H12);
y2 = ifft((sym2z).*(H11)) + ifft((sym1z).*H12);
y3 = ifft(...
y4 = ifft(...

```

Добавляем каждому символу шум и переносим их в область частот.

```

n1 = fft(...
n2 = fft(...
n3 = fft(...
n4 = fft(...

```

Оцените переданные символы для схемы Аламоути для случая MIMO и SISO.

```

x1_ocen=(conj(H11).*n1+H12.*conj(n2)+conj(H21).*n3+H22.*conj(n4))./(abs(H11).^2+abs(H12).^2+abs(H21).^2+abs(H22).^2);
...
...
x1_asiso = y_asiso1./H11;
x2_asiso = y_asiso2./H22;

```


Демодулируйте принятые символы.

rxData1 = ... rxData2 = ...

Посчитайте ошибки после демодуляции.

Errors_MIMO = ... Errors_SISO = ...
--

Постройте созвездия 1-го и 2-го OFDM сигнала для MIMO и для SISO, а также их спектр.

Изменяйте значение SNR от 5 до 30 и сравните результат моделирования технологии с кодированием Аламоути MIMO и технологии SISO.

Контрольные вопросы к работе:

- 1) В чём ключевое отличие MIMO с кодированием Аламоути от обычного MIMO?
- 2) Почему мы передаем вместе с обычным OFDM символом комплексно сопряженный?
- 3) Какая схема передачи более устойчива к воздействию шума?
- 4) Какая технология по вашему мнению более эффективна – MIMO с Аламоути или классическое MIMO?

Требования к оформлению отчета по работе.

Содержание отчета:

- 1) Введение,
- 2) Блок схема работы алгоритма, который вы применили в работе,
- 3) Рисунки:
 - Спектр сигнала в приемнике для MIMO и SISO,
 - Созвездия сигнала в приемнике для MIMO и SISO.
- 4) Выводы,
- 5) Листинг программы.

СПИСОК ЛИТЕРАТУРЫ

1. Документация по MATLAB на русском языке – URL: <https://docs.exponenta.ru/> [дата обращения 28.03.2021].
2. Дьяконов В.П. MATLAB. Полный самоучитель – М.: ДМК Пресс, 2012. – 768 с.
3. Щербаков В.С., Руппель А.А., Глушец В.А. ОСНОВЫ МОДЕЛИРОВАНИЯ СИСТЕМ АВТОМАТИЧЕСКОГО РЕГУЛИРОВАНИЯ И ЭЛЕКТРОТЕХНИЧЕСКИХ СИСТЕМ В СРЕДЕ MATLAB И SIMULINK: Учебное пособие. – Омск: Изд-во СибАДИ, 2003. – 160 с.