

Министерство образования и науки Российской Федерации
Федеральное государственное бюджетное образовательное
учреждение высшего образования
«ТОМСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ СИСТЕМ
УПРАВЛЕНИЯ И РАДИОЭЛЕКТРОНИКИ»

Кафедра телекоммуникаций и основ радиотехники (ТОР)

Е.В. Рогожников

Телекоммуникационные технологии

Методические указания по выполнению лабораторных работ

Томск – 2018

Рогожников Е.В. Телекоммуникационные технологии: методическое пособие – Томск: Издво ТУСУР, 2018. – 33 с. Методическое пособие для студентов ВУЗов посвящено моделированию систем беспроводной связи в пакете Matlab.

Оглавление

1. Лабораторная работа «Начало работы с Matlab».....	4
2. Лабораторная работа «OFDM модуляция»	6
3. Лабораторная работа «Оценка канала связи/эквалайзирование».....	11
4. Лабораторная работа «Кадровая синхронизация»	17
5. Лабораторная работа «Частотная синхронизация»	20
6. Лабораторная работа «Помехоустойчивое кодирование, линейные блочные коды».....	23
7. Лабораторная работа «Помехоустойчивое кодирование, сверточное кодирование».....	26
8. Лабораторная работа «Кодовое разделение каналов»	30

1. Лабораторная работа «Начало работы с Matlab»

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

Задачи лабораторной работы:

- 1) Изучить основные функции и блоки Matlab.
- 2) Произвести генерацию синусоидального сигнала в Matlab. 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 Command Window, clear all – удаляет все переменные из workspace, очищает память close all – закрывает все открытые фигуры.

Упражнение 1.

Постройте сигнал во временной и частотной области со следующими параметрами:

$F_0 = 10$ kHz; несущая частота
 $F_s = 100$ kHz; частота дискретизации
 $N = 1000$; количество отсчетов

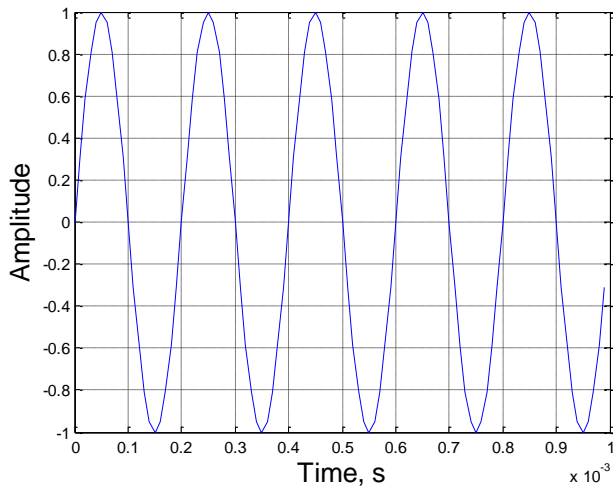
Реализация в Matlab:

```
clc  
clear all  
close all
```

```
F0 = 5e3;  
Fs = 100e3;  
N = 100;  
t = (0:N-1)/Fs;
```

```
sig = sin (2*pi*F0*t);
```

```
figure  
plot(t,sig);  
grid on  
xlabel('Time, s','fontsize',16);  
ylabel('Amplitude','fontsize',16);
```



Задача 1: Постройте спектр синусоидального сигнала, используя функцию `fft()`;

Сформируйте еще один синусоидальный сигнал частотой 10 kHz, используя цикл `for`, произведите перемножение и суммирование 2 синусоидальных сигналов.

Постройте спектр суммы и перемножения двух сигналов.

Задача 2:

Создайте 2 битовые последовательности (A и B) состоящие из 100 элементов ($N = 100$), используя функцию `randint`.

Используя цикл `for`, и функцию `if`, произведите суммирование по модулю 2 (xor operation).

XOR table

Вход		Выход
A	B	
0	0	0
0	1	1
1	0	1
1	1	0

Контрольные вопросы к лабораторной работе

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

2. Лабораторная работа «OFDM модуляция»

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

Задачи лабораторной работы:

- 1) Составить программу для передачи OFDM сигнала в среде Matlab.
- 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 2.1.

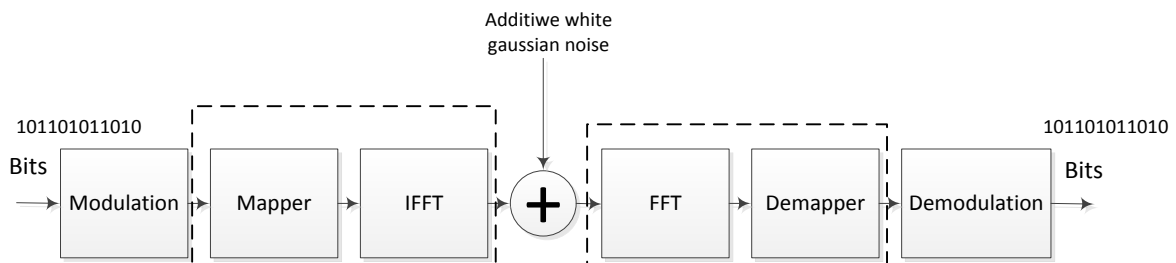


Рисунок 2.1. Формирование и демодуляция OFDM сигнала

В начале программы пропишите 3 строки:

```
clc
clear all
close all
```

Создайте битовую последовательность используя функцию randint:

```
N = 1000;
bits = randint(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;

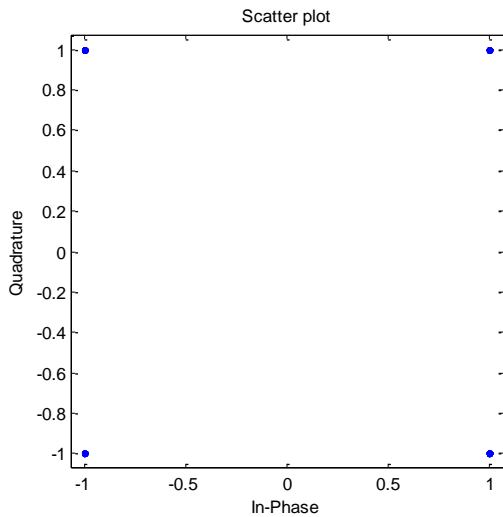
```

```

end

```

Постройте созвездие промодулированного сигнала, используя функцию scatterplot



Имитация канал связи

Используя функцию awgn добавьте шум к промодулированному сигналу

Отношение сигнал/шум SNR = 15dB;

```

SNR = 15;

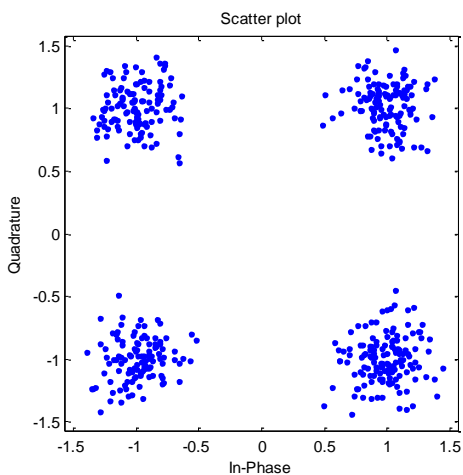
```

```

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

```

Постройте созвездие сигнала с шумом, используя функцию scatterplot



Change the value of SNR for 5, 10, 15, 20 dB, Draw a constellation signal for this values.

Демодуляция

Цель демодуляции – преобразовать сигнал в битовый поток.

Используйте цикл и набор условий.

Ниже приведены только 2 условия из 4. Добавьте еще 2 условия, необходимые 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

Параметры:

Размер преобразования Фурье = 1024;

Защитные интервалы: 100 отсчетов слева и 100 отсчетов справа;

Поднесущие с данными: from 101 to 924, (количество поднесущих с данными: 824);

Модуляция: QPSK

Передатчик

Разместите поднесущие с промодулированными данными, так как показано на рисунке 2.1.

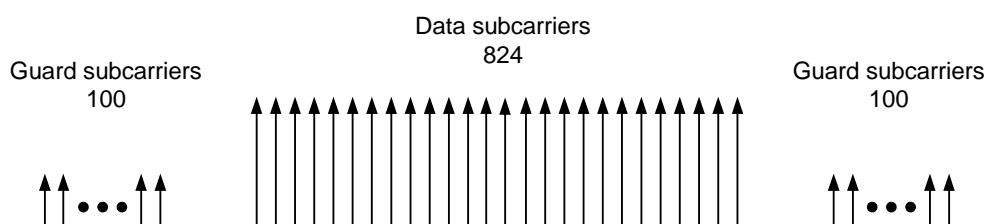


Рисунок 2.1 Расположение поднесущих в OFDM сиволе

Создайте нулевой массив:

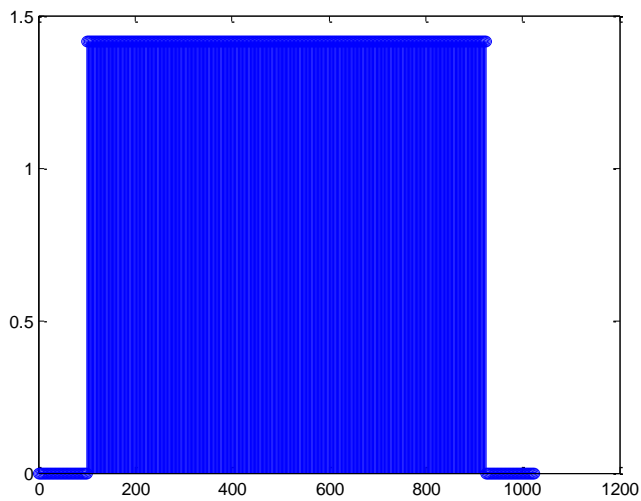
```
spectrum = zeros(1,1024);
```

Разместите поднесущие с промодулированными данными, так как показано на рисунке 2.1.

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

Постройте спектр сформированного OFDM сигнала:

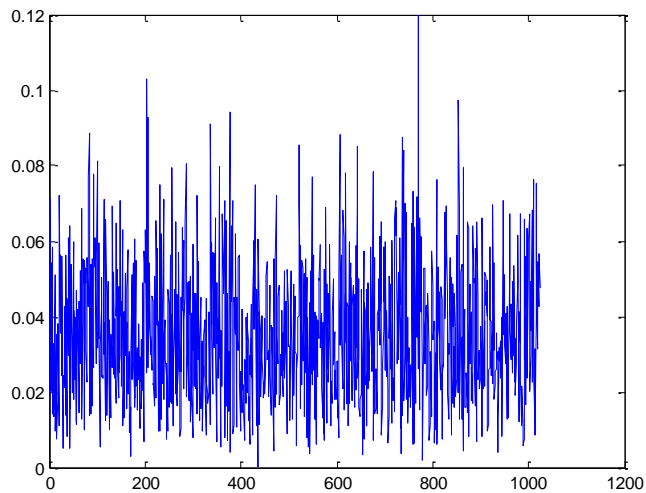
```
figure
stem(abs(spectrum));
```



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

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

Постройте на рисунке OFDM сигнал во временной области



Канал

- Используя функцию `awgn`, добавьте аддитивный шум к сигналу.

Установите отношение сигнал/шум 15 дБ;

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

Приемник

- Перейдите к частотной области используя функцию `fft`

```
spectrum_r = fft(sig_noise);
```

Постройте на рисунке спектр принятого сигнала

Выберите только поднесущие с данными.

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

Постройте созвездие, используя функцию `scatterplot`

```
scatterplot(data);
```

- Демодуляция

Произведите демодуляцию используя цикл `for` и условия `if`.

Сравните массивы `bits` и `bits_demod`, они должны совпадать.

Контрольные вопросы к лабораторной работе

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

3. Лабораторная работа «Оценка канала связи/эквалайзирование»

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

Задачи лабораторной работы:

- 1) Сформировать OFDM сигнал.
- 2) Произвести передачу сигнала через канал с многолучевостью.
- 3) Произвести оценку канала связи, эквалайзирование, демодуляцию сигнала.

Ход выполнения работы

В данной работе необходимо реализовать схему, приведенную на рисунке 3.1.

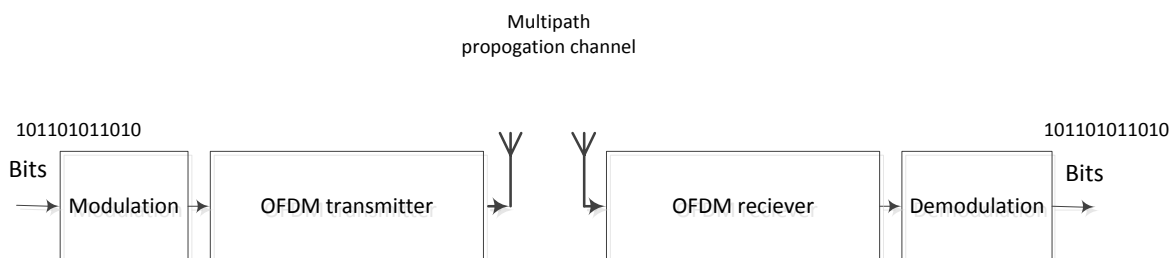


Рисунок 3.1. Формирование сигнала и прохождение через многолучевой канал

Параметры сигнала:

Data subcarriers: from 101 to 924, (number of data subcarriers: 824);

Modulation: Pilot symbol: QPSK;

Data_symbol: QAM 16;

Размер преобразования Фурье = 1024;

Защитные интервалы: 100 отсчетов слева и 100 отсчетов справа;

Поднесущие с данными: from 101 to 924, (количество поднесущих с данными: 824);

Модуляция: Пилотный символ QPSK, символ с данными QAM 16

Параметры многолучевого канала распространения радиоволн:

Количество лучей	Задержка	Ослабление
1	0 ns	1
2	200 ns	0.7
3	400 ns	0.5

Сформируйте 2 OFDM символа. Первый из них - PILOT сигнал, следующий – символ для передачи данных.

Pilot OFDM symbol	Data OFDM symbol
-------------------	------------------

Создаем 2 битовые последовательности:

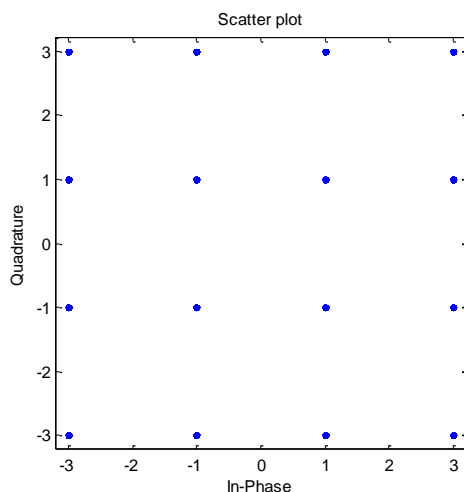
```
N = 824;
M1 = 4;
data_pilot = randint(1,N,M1);
M2 = 16;
data_user = randint(1,N,M2);
```

Далее - модуляция

```
mod_data_pilot = qammod(data_pilot,M1);
mod_data_user = qammod(data_user,M2);
```

Постройте созвездие сигнала, используя функцию scatterplot

```
scatterplot(mod_data_user);
```



Следующий шаг – формирование сетки:

Расположите поднесущие как показано на рис 3.2.

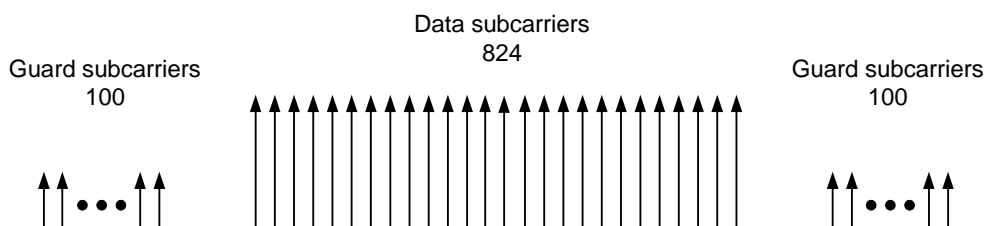


Рисунок 3.2. Формирование сетки

Создайте массив нулевых элементов:

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

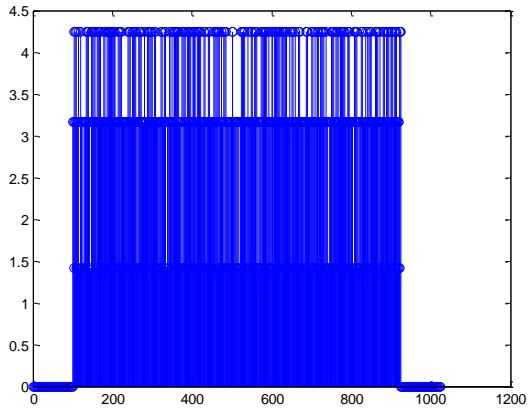
Расположите поднесущие, как показано на рисунке 3.2.:

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

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

Постройте спектр сигнала, используя функцию `fft`

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



Перейдите ко временной области, используя функцию `ifft`

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

Объедините 2 OFDM символа в один массив.

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

Моделирование многолучевого канала распространения радиоволн.

Установите 3 коэффициента ослабления:

```
k1 = 1;
k1 = 0.7;
k1 = 0.5;
```

Сформируйте 3 сигнала. Первый – прямой с нулевой задержкой и ослаблением k_1 . Второй с задержкой в 200нс (2 отсчета) и ослаблением k_2 . Третий с задержкой в 400нс и ослаблением k_3 .

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

1. Используя функцию `AWGN` добавьте шум в сформированный сигнал.

Отношение сигнал/шум SNR = 25dB;

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

Приемник

Запишите в 2 разных массива пилотный сигнал и сигнал с данными

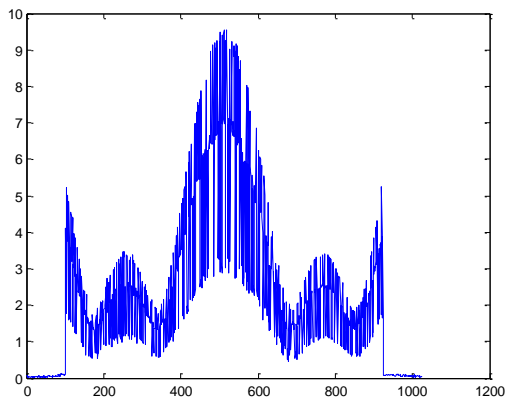
```
pilot_recive = sig_recive_noise(1:1024);
user_data_recive = sig_recive_noise(1025:2048);
```

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 спектр принятого сигнала:

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

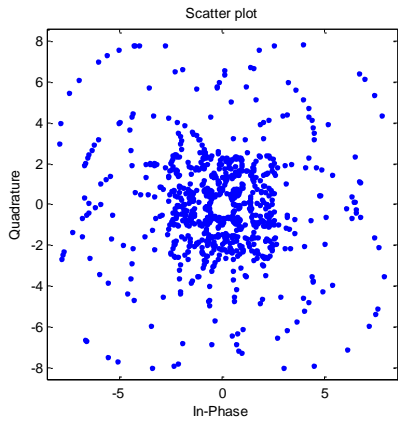


Удалите защитные интервалы в частотной области, выберите только поднесущие с данными.

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

Постройте созвездие сигнала, используя функцию scatterplot

```
scatterplot(user_data_subcarriers);
```



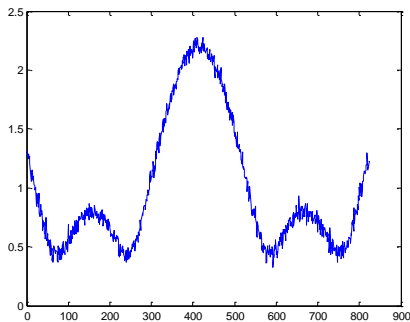
Создайте опорный сигнал: `reference_subcarriers = mod_data_pilot;`

Произведите оценку передаточной функции канала связи

`TF_estimate = pilot_subcarriers./reference_subcarriers;`

Постройте на рисунке передаточную функцию канал распространения радиоволн:

`figure`
`plot(abs(TF_estimate));`

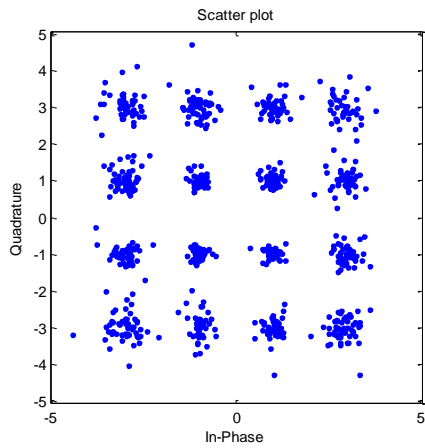


Произведите операции эквалайзирования. Путем деления спектра принятого сигнала на передаточную функцию:

`user_data_estimate = user_data_subcarriers./TF_estimate;`

Выведите на рисунок созвездие сигнала после эквалайзирования, используя функцию `scatterplot`

`scatterplot(user_data_estimate);`



Демодуляция:

```
data_demod = qamdemod(user_data_estimate,M2);
```

Сравните массивы `data_user` и `data_demod`, они должны совпадать.

Контрольные вопросы к лабораторной работе

- 1) Что такое пилотный сигнал.
- 2) Что такое передатчная функция канала РРВ ?
- 3) Как влияет многолучевость канал на передаваемый сигнал?
- 4) Что такое эквалайзирование?

4. Лабораторная работа «Кадровая синхронизация»

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

Задачи лабораторной работы:

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

Ход выполнения работы

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



Рис. 4.1. Алгоритм кадровой синхронизации

1. Формирование OFDM символов.

Параметры сигнала:

$N_{FFT} = 1024$;

Защитный интервал: 100 отсчетов слева и 100 отсчетов справа;

Поднесущие с данными: с 101 до 924, (количество поднесущих с данными: 824);

Модуляция: пилотных сигналов: QPSK;

Символов с данными: QAM 16;

Ваша задача сформировать 2 разных OFDM символа. Первый из них - пилотный, следующий символом с данными. Structure shown on figure 2.



Рис. 2.

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

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

2. 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];
```

3. Добавляем АБГШ, SNR = 30 дБ;

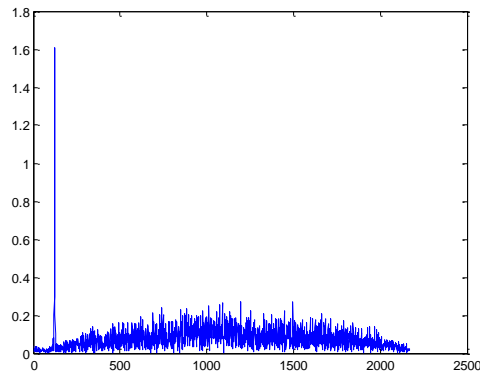
4. Оценка временного смещения

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



Используя функцию `max`, находим максимальное значение ВКФ.

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

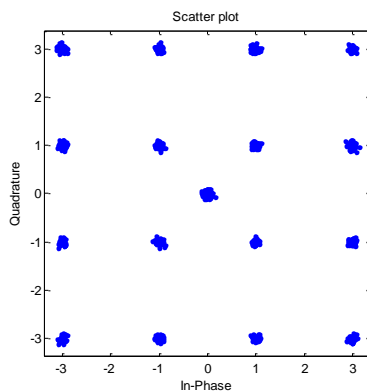
Используем положение максимума ВКФ для определения начала символа с данными.

```
user_data_recive = sig_time_shift_noise(start+1024:end);
```

Постройте спектр и созвездие сигнала пользователя

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

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

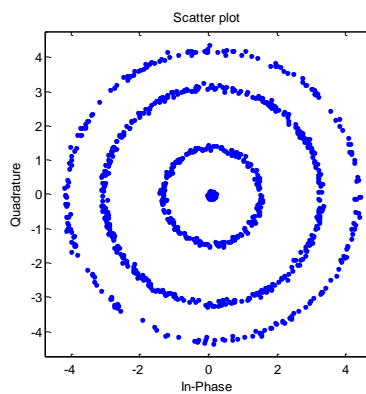


Добавьте ошибку в один отсчет при определении начала OFDM символа, постройте созвездие.

```
user_data_recive = sig_time_shift_noise(start-1+1024:end-1);
```

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

```
scatterplot(spectrum_user_data_r);
```



Контрольные вопросы к лабораторной работе

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

5. Лабораторная работа «Частотная синхронизация»

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

Задачи лабораторной 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,

Частота дискретизации: 10.24 МГц,

```
clc
```

```
clear all
```

```
close all
```

```
NFFT=1024;
```

```
% Формируем OFDM символ
```

```
M = 16;
```

```
data = randint(1,824,M);
```

```
data_mod = qammod(data,M);
```

```
% Добавьте защитные интервалы в частотной области (100 поднесущих слева и 100 справа).
```

```
sp_ofsm = [zeros(1,100),... ];
```

```
sig_vr=ifft(sp_ofsm);
```

```
% дублируем OFDM символ во временной области
```

```
sig_sum=[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;
```

```
% рачитываем 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
```

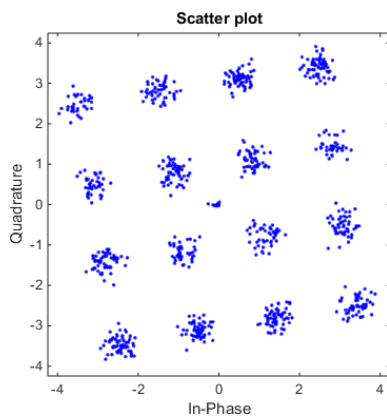
```
sig_freq_shift1=awgn(...);
```

```
% Выбираем отсчеты первого и второго OFDM символа
```

```
Pr_sym1=sig_freq_shift1(...);
```

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

```
% постройте созвездие первого принятого OFDM символа
```



```
% Оценка частотного сдвига
```

```
% Расчет ВКФ между вторым и первым принятым OFDM символом
```

```
corr=xcorr(Pr_sym2, ...);
```

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

```
% Оценка фазового набег
```

```
Dphi=angle(a);
```

```
% рассчитываем длительность OFDM символа
```

```
dt=1/fs;
```

```
tau=(NFFT)*dt;
```

```
ocen_freq=Dphi/(2*pi*tau);
```

```
% сравните оцененное значение частотного сдвига и то которое вы вводили
```

```
% компенсируем частотный сдвиг
```

```
% рассчитываем фазовый набег за интервал дискретизации по полученной оценке
```

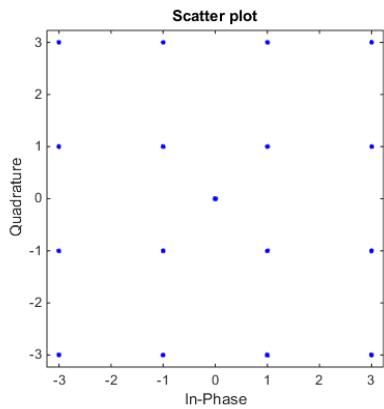
```
d_phi_ocen=(ocen_freq*2*pi)/fs;
```

```
% 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 OFDM символа после компенсации частотного сдвига



Контрольные вопросы к лабораторной 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?

6. Лабораторная работа «Помехоустойчивое кодирование, линейные блочные коды»

Цель работы: произвести помехоустойчивое кодирование с использованием линейного блочного кода и декодирование.

Задачи лабораторной работы:

- 1) Битовую последовательность.
- 2) Произвести кодирование с использованием линейного блочного кода.
- 3) Произвести модуляцию и добавить АБГШ.
- 4) Произвести демодуляцию и декодирование.

Ход выполнения работы

В данной работе необходимо создать битовую последовательность, произвести операцию помехоустойчивого кодирования, произвести QPSK модуляцию, добавить АБГШ. В приемнике произвести демодуляцию, помехоустойчивое декодирование проверить количество ошибок.

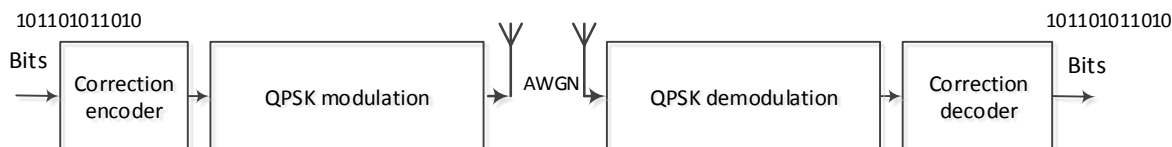


Рисунок 6.1. Формирование сигнала и прохождение через многолучевой канал

Параметры сигнала:

Количество бит = 100;

Модуляция QPSK;

Линейный блочный код (5,2).

```
clc
clear all
close all
```

создайте битовую последовательность

```
N = 100;
bits = randint(1,N);
```

```
Convert a string into a matrix of 2x50:
Bits_2 = reshape (bits,2,50);
Bits_2=Bits_2.';
```

Создайте производящую матрицу для кода 5.2

$$\mathbf{G} = \begin{bmatrix} 1 & 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 1 & 1 \end{bmatrix}$$

```
G(1,:) = [ ... ];
G(2,:) = [ ... ];
```

Произведите операцию кодирования

```
for i=1:50
s(i,:) = Bits_2(i, :)*G;
end
```

Преобразуйте матрицу в строку

```
s=s';
s_str = reshape(s, 1,250);
```

Следующий этап - модуляция

```
mod_data = qammod(s_str,4);
```

1. Используя функцию AWGN добавьте шум

Установите отношение сигнала/шум равное = 25dB;

```
SNR = 25;
```

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

В приемнике произведите демодуляцию

```
bits_demod = qamdemod( ... );
```

Преобразуйте сигнал в матрицу

```
Bits_demod_matr = reshape( ... );
Bits_demod_matr=Bits_demod_matr.';
```

Произведите помехоустойчивое декодирование

```
for i=1:50
bits_decod_matr(i,:) = round(Bits_demod_matr(i, :)/G);
end
```


Преобразуйте матрицу в строку (1x100):

```
bits_decod_matr = bits_decod_matr.';  
bits_decod_str = reshape(    ...    );
```

Проверьте количество ошибок

```
num_errors = sum(abs(bits_decod_str - ...))
```

Изменяйте отношение сигнал/шум от 15 до 5 dB и определите отношение сигнал/шум, при котором появятся ошибки.

Контрольные вопросы к лабораторной работе

- 1) Что такое производящая матрица?
- 2) Что такое проверочная матрица?
- 3) Как производится кодирование бит в линейных блочных кодах?
- 4) Как производится декодирование в линейных блочных кодах?

7. Лабораторная работа «Помехоустойчивое кодирование, сверточное кодирование»

Цель работы: произвести помехоустойчивое кодирование с использованием сверточного кода и декодирование.

Задачи лабораторной работы:

- 1) Сформировать битовую последовательность.
- 2) Произвести кодирование с использованием сверточного кода.
- 3) Произвести модуляцию и добавить АБГШ.
- 4) Произвести демодуляцию и декодирование.

Ход выполнения работы

В данной работе необходимо создать битовую последовательность, произвести операцию помехоустойчивого кодирования с использованием сверточного кода (171, 133, $K = 7$), произвести QPSK модуляцию, добавить АБГШ. В приемнике произвести демодуляцию, помехоустойчивое декодирование проверить количество ошибок.

Параметры сигнала:
 Количество бит = 100;
 Модуляция QPSK;

Прежде всего нарисуйте схему кодера: Сверточный кодер (171, 133, $K=7$).

В бинарном виде производящие полиномы: (1111001 , 1011011, $K=7$)

Создайте битовую последовательность:

```
N = 100;
bits = randint(1,N);
```

Создайте сверточный кодер:
 Запишите нули в регистры сдвига

```
s0 = 0;
s1 = 0;
.
.
.
s6 = 0;
```

```

for i=1:N

    s6 = s5;
    .
    .
    .
    s1 = s0;
    s0 = bits(i);

    out1(i) = rem(s0+ ... ..+s6,2);
    out2(i) = rem(s0+... ..s6,2);

end

```

Мультиплексор:

```

for i=1:2:length(bits)*2

    out_bits(i)=out1(j);
    out_bits(i+1)=out2(j);
    j=j+1;
end

```

Создайте модулятор, используя набор условий и цикл for.
В приведенном ниже коде отсутствуют 2 условия, добавьте их.

```

k=1;
for i = 1 : 2 : 2*N

    if out_bits (i) == 1 && out_bits (i+1) == 1
        mod_data (k) = 1+1i;
    end

    if out_bits (i) == 1 && out_bits (i+1) == 0
        mod_data (k) = 1-1i;
    end

    k=k+1;

end

```

Постройте созвездие сигнала, используя функцию scatterplot

```
scatterplot(mod_data);
```

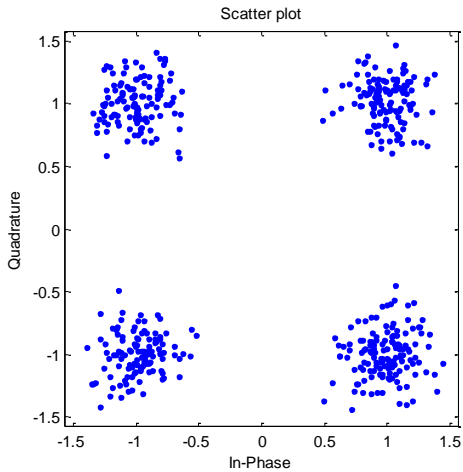
Используя функцию AWGN добавьте шум в сформированный сигнал.

Отношение сигнал/шум SNR = 15dB;

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

Постройте созвездие используя функцию scatterplot

```
scatterplot(mod_data_noise);
```



Демодуляция

Произведите демодуляцию, используя цикл for и набор условий

```
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
```

Декодер

```
out_priem = [bits_demod, bits_demod];
trel = poly2trellis(7, [171 133]);

decoded1 = vitdec(out_priem, trel, 3, 'cont', 'hard');

output_bits = decoded1(4:100+3);
```

Проверьте количество ошибок:

```
num_errors = sum(abs(output_bits - bits))
```

Контрольные вопросы к лабораторной работе

- 1) Что такое производящий полином?
- 2) Что такое глубина декодирования?
- 3) Каково начальное состояние регистров сдвига?
- 4) На что влияет глубина декодирования?

8. Лабораторная работа «Кодовое разделение каналов»

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

Задачи лабораторной 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

Лабораторная работа кадровая синхронизация.

В данной работе необходимо реализовать схему, приведенную на рисунке 8.1.

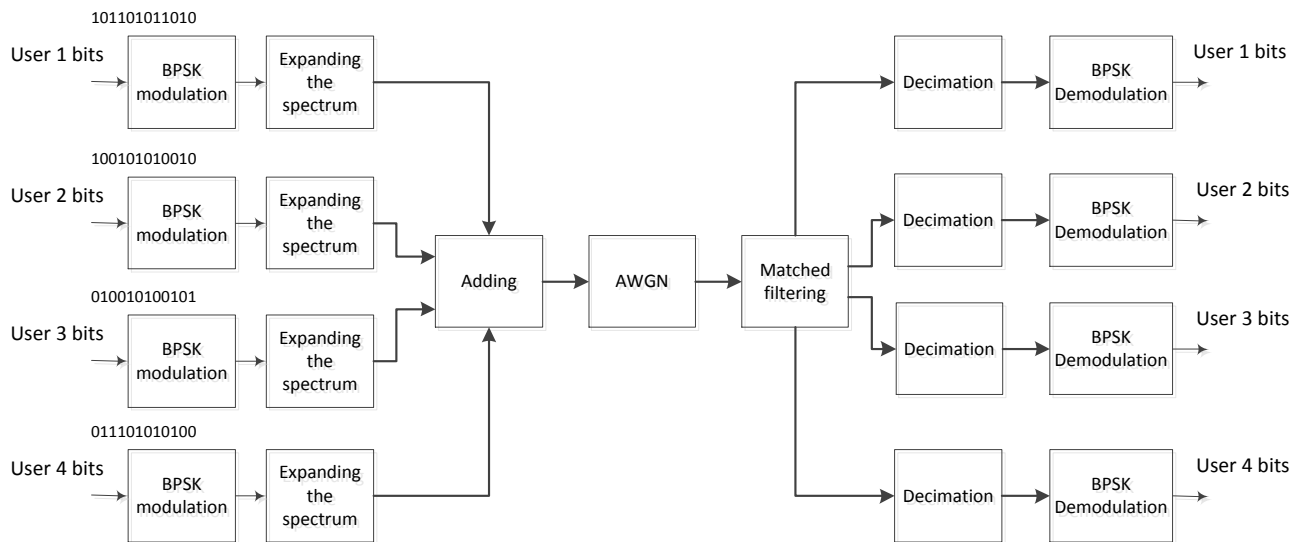


Рисунок 8.1 – Кодовое разделение каналов

Параметры сигналов

Количество пользователей: 4;

Количество бит = 100;

Модуляция BPSK;

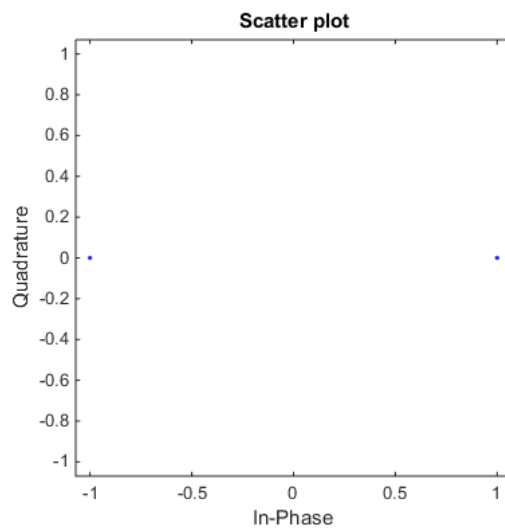
Сформируйте битовую последовательность для каждого пользователя

```
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
end
```

Постройте созвездие, используя функцию scatterplot:



Сформируйте матрицу Адамара

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

Произведите кодирование сформированных данных, используя функции Уолша:

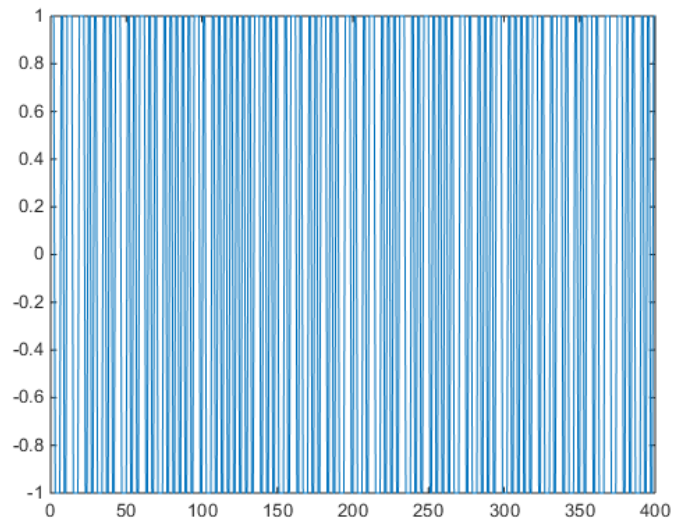
```
j=1;
for i=1:N
    if mod_bits1 (i) == 1
        mod_bits_1_code (j:j+k-1)= mtx(1, :);
    else
        mod_bits_1_code (j:j+k-1)= mtx(1, :)*(-1);
    end
    .
    .
    .
```

```

    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



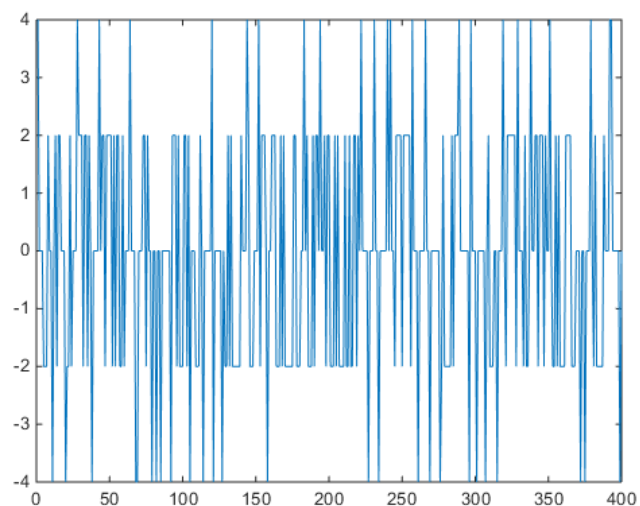
Следующий этап – сложение данных всех пользователей.

```

sum_of_signals =
mod_bits_1_code+mod_bits_2_code+mod_bits_3_code+mod_bits_4_code;

```

Постройте суммарный сигнал, используя функцию plot



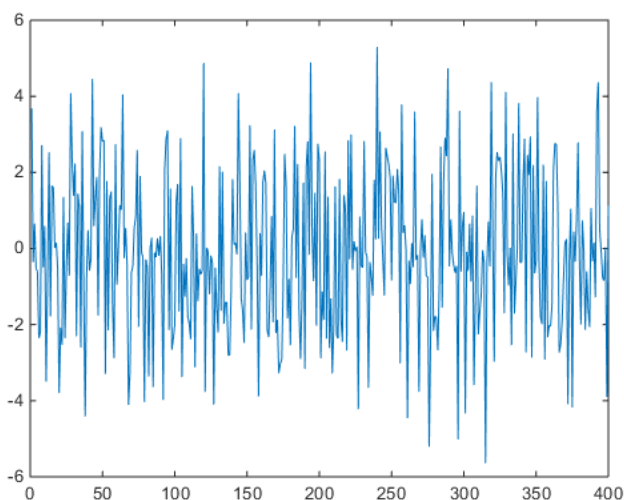
Добавьте аддитивный белый гауссовский шум

```

sum_of_signals_noise = awgn(sum_of_signals, 10, 'measured');

```


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)));
```

Децимация

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

Демодуляция

```
for i=1:N
    if korr_decim(i) > 0
        demod_bits(i) = 1;
    else
        demod_bits(i) = 0 ;
    end
end
```

```
end
```

Посчитайте количество ошибок после демодуляции:

```
num_errors = sum(abs(demod_bits - bits3))
```

Контрольные вопросы к лабораторной работе

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