

Радиосеть передачи данных. Часть 4

Исправленная



Сергей Медведев, 26 декабря 2013г.

В данном разделе КП предложено описание разработанной системы в среде MATLAB, демонстрирующей реализацию сценария, описанного в статье 2 (п. 2.3, 3. *Режим передачи*). На рис.1 представлена часть сценария взаимодействия AP и T, которая как раз и будет программно реализована.

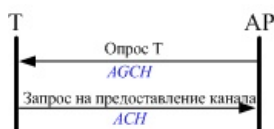


Рис.1. Процедура опроса терминалов и ответ терминала с запросом канала.

Терминалы, прошедшие регистрацию, будут последовательно опрашиваться AP, чтобы точка доступа знала каким терминалам выделить канал. Опрос терминалов осуществляется по каналу AGCH. Сообщение этого канала на канальном уровне состоит из следующих полей:

- TYPE – поле, идентифицирующее принадлежность сообщения определенному логическому каналу(3 бита);
- поле «ID_{Term}» - для адресной передачи сообщений(6 бит);
- DATA – имеет в своем составе четыре нуля(4 бита);
- CRC - необходимо для определения целостности пакета на приемной стороне.(12 бит);
- FILL - для приведения сообщения к стандартной длине(60 бит).

Канальному кодированию, перемежению, модуляции, формированию OFDM-символов - сообщение подвергается на физическом уровне.

В ответ на сообщение канала AGCH, терминал формирует сообщение, содержащее в себе один из вариантов:

% 1 - активен (1) (канал не требуется, пассивный режим)

% 2 - характер трафика пульсирующий(если да, то 1)

% 3 - характер трафика постоянный(если да, то 1)

Данное сообщение передается по каналу ACH и имеет следующие поля:

- TYPE – поле, идентифицирующее принадлежность сообщения определенному логическому каналу(3 бита);
- поле «ID_{Term}» - чтобы AP знала, от какого терминала сообщение(6 бит);
- DATA – имеет в своем составе один из выше перечисленных вариантов(4 бита);
- CRC - необходимо для определения целостности пакета на приемной стороне.(12 бит);
- FILL - для приведения сообщения к стандартной длине(50 бит).

На физическом уровне сообщение подвергается тем же процедуры, что и для сообщения канала AGCH.

Текст программы состоит из двух частей: T- приемная и передающая часть, AP – приемная и передающая

Рассмотрим функцию, которая устанавливает параметры модели.

```
function set = initSettings()
```

```
% параметры модели
```

```

% Идентификаторы сообщений

set.bcch = 0;

set.ach = 1;

set.tch = 2;

set.agch = 3;

set.rach = 4;

%Параметры помехоустойчивого кодека

%Для примера используется код ВЧХ:

%Длина кодового слова

set.n=127;

set.k=85;

% Параметры CRC

% CRC-12

set.poly=[1 1 0 0 0 0 0 0 0 1 1 1 1];

[set.x set.y]=size(set.poly); %размерность полинома

%размер сообщения канального уровня без учета CRC

set.lenght_info=73;

Рассмотрим функцию, которая отражает передатчик AP.

function [phy_frame] = transmitter_AP( TypeMSG ,ID_T, DATA, NUMBER_PROFILE)

% TYPE - Тип сообщения

% DATA - Поле DATA

% ID_Term - Идентификатор терминала

% NUMBER_PROFILE - профиль функционирования

% 1 - BPSK

% 2 - QPSK

% Установка параметров модели

st=initSettings();

global Nc

global pilotIndexes

global ns;

% определение кода типа сообщения (3-х битовое поле)

% Предусмотрены 5 возможных типов сообщений

% BCCH - широковещательное сообщение

% ACH - сообщение ответ

% TCH - сообщение трафика

```

```

% AGCH - сообщение опрос

% RACH - сообщение заявка

tps = my_de2bi (TypeMSG, 'left_msb', 3);

[TYPE_R, ID_Tr, DATA_R, detect] = receiver_T (phy_frame_R, profile);

disp('-----');

disp('Сообщение, полученное T');

disp('-----');

% Отображение отсутствия/наличия битовых ошибок

if detect == 0

disp('Битовые ошибки: не присутствую' );

if TYPE_R==st.agch

disp('Принятое сообщение');

disp('TYPE'); disp(st.agch);

if ID_Tr==ID_Term

disp('получатель:'); disp(['терминал с ID']);disp(ID_Tr);

disp('DATA');disp(DATA_R);

disp('формирование сообщения канала ach')

% ===== Канальный уровень =====

% сборка части L2-сообщения

msg_Lev2 = [tps ID_T DATA FILL];

% получение CRC кода

m = [msg_Lev2 zeros(1,st.y-1)]; %вставка нулей в поле CRC

[q r] = deconv(m,st.poly); % поиск частного и остатка от деления

%информационной части на порождающий полином CRC кода

r = mod(abs(r),2);

%расчет CRC

CRC = r(length(msg_Lev2) + 1:end);

%Сборка сообщения

msg_Lev1 = [msg_Lev2 CRC];

% ===== Физический уровень =====

% Кодирование сообщения кодом BCH (127, 85)

code = encode(msg_Lev1,st.n,st.k, 'bch/binary');

%добавление дополнительных бита для приведения размера сообщения к числу

%кратному 2 или 4 (следует из определения позиционности BPSK и QPSK модуляции

code_word = [0 code];

```

```

%Блочное перемежение

alternation_bits = matintrlv(code_word, 16, 8);

% Выбор профиля функционирования

if NUMBER_PROFILE==1

    % Параметры модуляции

    % h - модем BPSK

    mPos=2;

elseif NUMBER_PROFILE==2

    % Параметры модуляции

    % h - модем QPSK

    % h - модем BPSK

    mPos=4;

end

% номера пилотных поднесущих в OFDM-символе

pilotIndexes = [1 16 31 61 76 91 0];

% количество неиспользуемых поднесущих и приходящихся на защитные интервалы

Ngr = 5;

% Размерность OFDM-символа

Nc=91;

% количество поднесущих с данными

Ndata =84;

% количество битов, передаваемых в одном OFDM-символе

s_length = Ndata * log2(mPos);

% Количество OFDM-символов

ns = floor(length(alternation_bits)/s_length);

% Параметры модуляции

% h - модем BPSK с позиционностью mPos

h = modem.qammod(mPos);

h.inputtype = 'bit';

IQ_signal = modulate(h, alternation_bits');

figure;

plot(IQ_signal, 'xr');

```

В третьей статье говорилось, что передача одного OFDM-символа осуществляется с использованием 96 поднесущих: 84 поднесущие для передачи данных, 6 – для пилот сигнала, 2 и 3 – для использования защитных интервалов и 1 нулевая поднесущая для идентификации нулевой частоты OFDM сигнала.



Рис. 2. Структура поднесущих (OFDM символов).

%формирование OFDM-символов

```
ofdmFrame = makeOfdmFrame(IQ_signal, ns);
```

В начале каждого сформированного OFDM-символа находится циклический префикс. Размер циклического префикса составляет $\frac{1}{4}$ часть длины символа.

```
function ofdmFrame = makeOfdmFrame(dataMod, Nofdm)
```

```
global pilotIndexes;
```

```
global Nc;
```

```
global len_cp;
```

```
global len_ofdmSymbol;
```

```
%символы пилотных поднесущих промодулированы псевдослучайным кодом (для
```

```
%примера, код Баркера), как результат - BPSK
```

```
seqBarker = [-1 -1 -1 1 1 1 -1 1 1 -1 1]*4/3;
```

```
ofdmSymbol = [];
```

```
ofdmFrame = [];
```

```
indexOfPilot = 1;
```

```
indexOfData = 1;
```

```
for i=1:Nofdm
```

```
for j=1:Nc % собираются IQ-символы будущего OFDM-символа
```

```
if (j==pilotIndexes(indexOfPilot))
```

```
ofdmSymbol(j)=seqBarker(indexOfPilot);
```

```
indexOfPilot=indexOfPilot+1;
```

```
else
```

```
ofdmSymbol(j) = dataMod(indexOfData);
```

```
indexOfData=indexOfData+1;
```

```
end
```

```
end
```

```
indexOfPilot=1;
```

```
ofdmSymbol = ifft(ofdmSymbol);
```

```
%формирование циклического префикса, как часть OFDM-символа,
```

```
%используется 1/4 часть символа
```

```
cp = ofdmSymbol(Nc-Nc/4+1:Nc);
```

```
len_cp=length(cp);
```

```

% добавление ЦП в начало OFDM-символа

ofdmSymbol=[cp ofdmSymbol];

len_ofdmSymbol=length(ofdmSymbol);

% Сборка информационных OFDM-символов пакета физического уровня

ofdmFrame=[ofdmFrame ofdmSymbol];

ofdmSymbol=[];

end

end

Формирование преамбулы

%Формирование символов преамбулы

%Короткие символы - для подстройки частоты (синхронизация по частоте)

ShortTrainingSymbols = 2*[0 0 1+j 0 0 0 -1-j 0 0 0 1+j 0 0 0 -1-j 0 0 0 -1-j 0 0 0 1+j 0 0 0 0 0 0 -
1-j 0 0 0 -1-j 0 0 0 ...

    1+j 0 0 0 1+j 0 0 0 1+j 0 0 0 1+j 0 0];

short_symbols = ifft(ShortTrainingSymbols);

% выделение первых 15 отсчетов

Strs = short_symbols(1:15);

% Формирование 5 коротких символов преамбулы на основе выбранного набора

short_trs= [Strs Strs Strs Strs Strs];

%длинный символ преамбулы

LongTrainingSymbols=[0 0 0 0 0 0 1 1 -1 -1 1 1 -1 1 -1 1 1 1 1 1 1 -1 -1 1 1 -1 1 -1 1 1 1 ...

1 -1 -1 1 1 -1 1 -1 1 -1 -1 -1 -1 1 1 -1 -1 1 -1 1 -1 1 1 1 0 0 0 0 0];

long_symbol = ifft(LongTrainingSymbols(7:26));

long_trs = [long_symbol];

% Пакет физического уровня (символы преамбулы и OFDM-символы)

phy_frame = [short_trs long_trs ofdmFrame];

figure;

plot(abs(phy_frame));

end

```

Сообщение отправленное AP, должно быть принято и обработано Т. Начинается все с выделения из принятого сигнала символа OFDM, который содержит передаваемые данные. После извлечения полей DATA, содержащего сообщение опроса и TYPE, поле, идентифицирующее принадлежность сообщения определенному логическому каналу.

Передатчик Т формирует сообщение канала АСН, в котором передается запрос на выделение канала для пульсирующего трафика.

Рассмотрим функцию, которая отражает передатчик Т.

```

function [phy_frame] = transmitter_T( TYPE ,ID_T, DATA, NUMBER_PROFILE )

% TYPE - Тип сообщения

% DATA - Поле DATA содержит код запрашиваемой услуги

% ID_Term - Идентификатор терминала

% NUMBER_PROFILE - профиль функционирования

% 1 - BPSK

% 2 - QPSK

% Установка параметров модели

st=initSettings();

global Nc

global pilotIndexes

global ns;

% Прием сообщения AP

[Power_T, TYPE_R_2, ID_Tr, DATA_R_2, detect] = receiver_AP (phy_frame_R_2, profile);

disp('-----');

disp('Сообщение, полученное AP');

disp('-----');

disp('Мощность принятого сигнала');

disp(Power_T);

if detect == 0

disp('Битовые ошибки: не присутствую' );

disp('сообщение от T с ID'); disp(ID_Tr);

disp('TYPE'); disp(st.ach);

if [0 1 0 0]==DATA_R_2

disp('выделение подканала для передачи пульсирующего трафика для T с ID' );

disp(ID_Tr);

%% ===== Канальный уровень =====

% Подготовка к проверке CRC

[q r]=deconv(received_mes_L1,st.poly);

%проверка остатка

r=mod(abs(r),2);

if r == zeros(1,length(received_mes_L1))

detect=0;% ошибок нет

else

detect=1;% обнаружены ошибки

```

```

end;

% Собственно принятое сообщение
received_mes_L2 = received_mes_L1(1:st.lenght_info);

% выделение поля TYPE_R
TYPE_R=my_bi2de(received_mes_L2(1:3), 'left_msb');

% выделение поля ID_Tr
ID_Tr=received_mes_L2(4:9);

% выделение поля DATA_R
DATA_R=received_mes_L2(9:13);

end

```

Программа, которая реализует сценарий рис.1 выглядит следующим образом:

```

clear;%очистка рабочей области

clc; %очистка командного окна

% Установка параметров модели
st=initSettings();

%% Формирование сообщения AP

% вид модуляции

% bpsk - 1
% qpsk - 2

profile=1;

DATA=[0 0 0 0 0 0 0 0 0 0 0 0 0 0 ];

% Идентификатор терминала
ID_Term=[0 0 1 0 1 1 ];

% Типы каналов

% BCCH = 0
% ACB = 1
% TCH = 2
% AGCH = 3
% RACH = 4

% формирование блока OFDM-символов

% для передачи сообщения по радиоканалу
phy_frame = transmitter_AP(st.agch, ID_Term, DATA, profile);

%% Наложение шума на блок символов

phy_frame_R=awgn(phy_frame,st.SNR, 'measured');

```



```

%% Прием сообщения терминалом

[TYPE_R, ID_Tr, DATA_R, detect] = receiver_T (phy_frame_R, profile);

%При передаче сообщений канала ach передатчик терминала выполняет
%те же функции, что и передатчик AP

DATA_2=[0 1 0 0 0 0 0 0 0 0 0 0 0];

%% формирование блока OFDM-символов

% для передачи сообщения по радиоканалу

phy_frame_2=transmitter_AP(st.ach, ID_Term, DATA_2, profile);

phy_frame_R_2=awgn(phy_frame_2,st.SNR,'measured');

% Прием сообщения AP

[Power_T, TYPE_R_2, ID_Tr, DATA_R_2, detect] = receiver_AP (phy_frame_R_2, profile);

end;

```

Продемонстрируем результаты работы модели. Мы наблюдаем сигнально-кодовое созвездие, сформированное AP (рис.3) и OFDM-символы во временной области, соответствующие передаче сообщения канала AGCH.

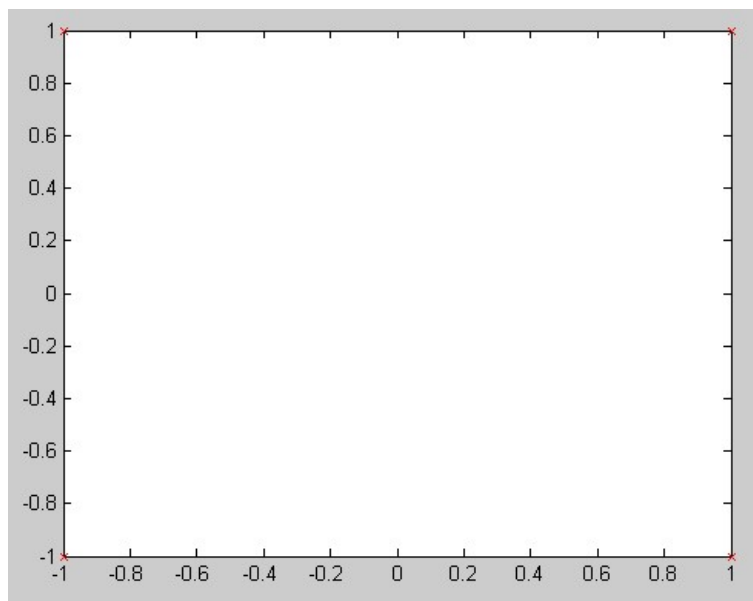


Рис.3. Сигнально-кодовое созвездие, сформированное AP

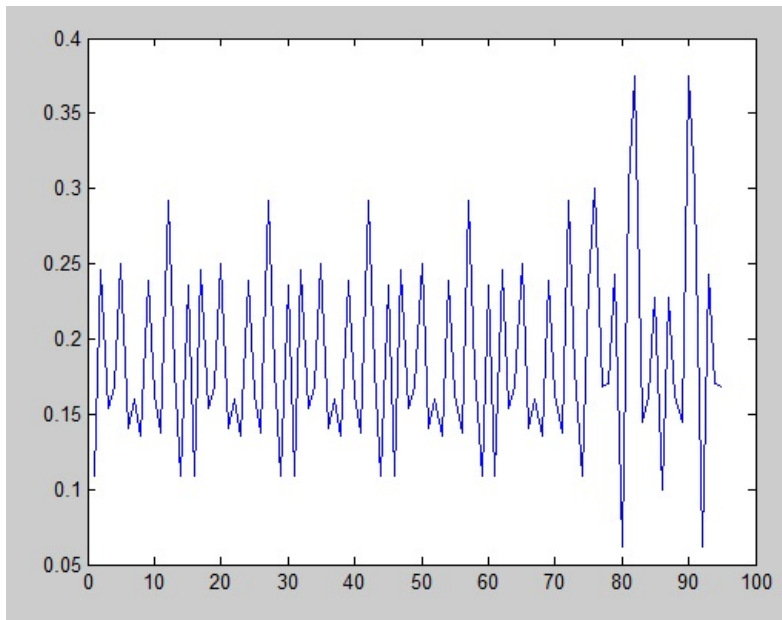


Рис.4. OFDM-символы во временной области, соответствующие передачи сообщения канала AGCH.

Сигнально-кодовое созвездие после демодуляции OFDM-символов изображено на рис.5. В результате демодуляции полученных IQ символов и функций, выполняемых приемником Т, терминалом получено сообщение

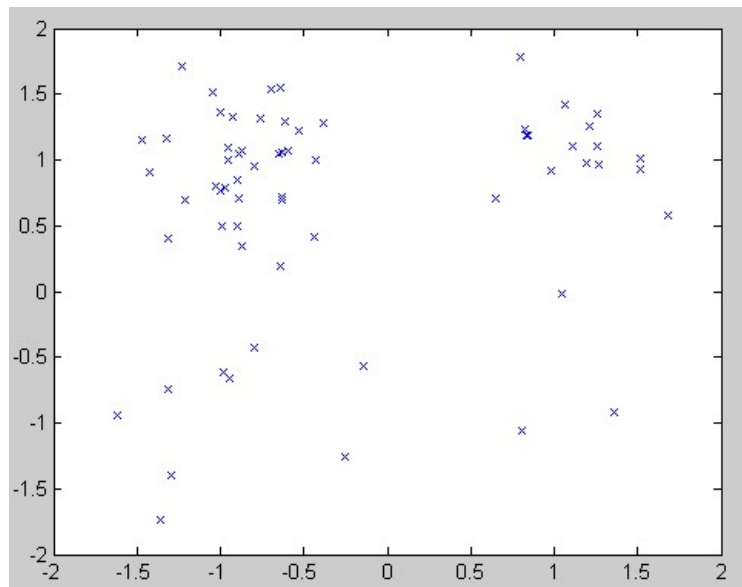


Рис.5. Сигнально-кодовое созвездие после демодуляции OFDM-символов (ОСШ = 8дБ)

Далее, сформированное сообщение будет проходить через канал с АБГШ и поступает на приемник АР. Сигнально-кодовое созвездие после демодуляции OFDM-символов изображено на рис.6. В результате демодуляции полученных IQ символов и функций, выполняемых приемником Т, терминалом получено сообщение

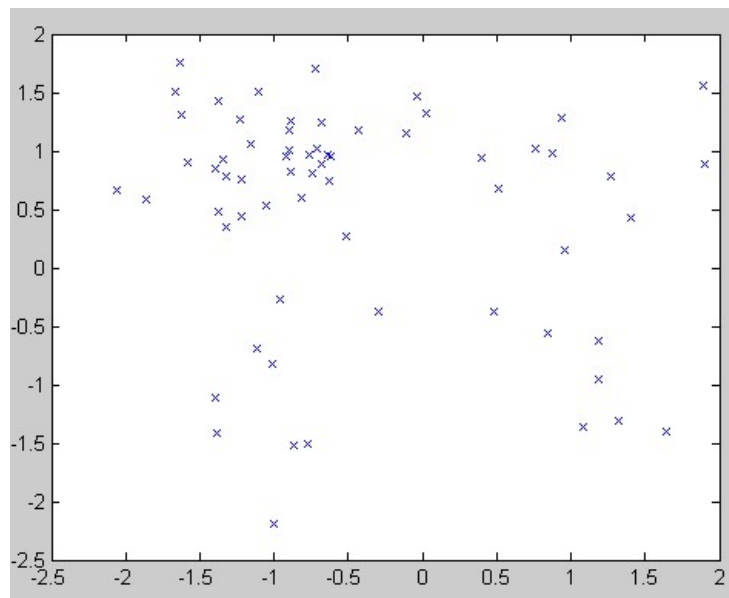


Рис.6. Сигнально-кодое созвездие после демодуляции OFDM-символов (ОСШ = 8дБ).

В ходе моделирования мы добились всех поставленных задач.