

Санкт-Петербургский политехнический университет Петра Великого  
Институт компьютерных наук и технологий  
Кафедра компьютерных систем и программных технологий

# Телекоммуникационные технологии

Отчет по лабораторной работе №8  
Модель телекоммуникационного канала

**Работу**  
**выполнил:**  
Графов Д.И.  
Группа: 33531/2  
**Преподаватель:**  
Богач Н.В.

Санкт-Петербург  
2019

# Содержание

<b>1. Цель работы</b>	<b>3</b>
<b>2. Программа работы</b>	<b>3</b>
2.1. Постановка задачи . . . . .	3
2.2. Информация о канале . . . . .	3
<b>3. Теоретическая информация</b>	<b>3</b>
<b>4. Ход выполнения работы</b>	<b>4</b>
4.1. Чтение сообщения . . . . .	4
4.2. Демодуляция . . . . .	4
4.3. Модуляция символов . . . . .	4
4.4. Перемежение бит . . . . .	5
4.5. Декодирование . . . . .	5
<b>5. Выводы</b>	<b>6</b>

# 1. Цель работы

Создание модели телекоммуникационного канала.

## 2. Программа работы

### 2.1. Постановка задачи

По имеющейся записи сигнала из эфира и коду модели передатчика создать модель приемника, в которой найти позицию начала пакета и, выполнив операции демодуляции, дегерметизации и декодирования, получить передаваемые параметры: ID, период, и номер пакета. Запись сделана с передискретизацией 2, т.е. одному BPSK символу соответствуют 2 лежащих друг за другом отсчетов в файле. Запись сделана на нулевой частоте и представляет из себя последовательность 32-х битных комплексных отсчетов, где младшие 16 бит вещественная часть, старшие 16 бит – мнимая часть.

### 2.2. Информация о канале

Пакетный сигнал длительностью 200 мкс состоит из 64 бит полезной информации и 8 нулевых tail-бит. В нулевом 16-битном слове пакета передается ID, в первом - период излучения в мс, во втором – сквозной номер пакета, в третьем - контрольная сумма (CRC-16). На передающей стороне пакет сформированный таким образом проходит следующие этапы обработки:

1. Помехоустойчивое кодирование сверточным кодом с образующими полиномами 753, 561( octal ) и кодовым ограничением 9. На выходе кодера количество бит становится равным 144.
2. Перемежение бит. Количество бит на этом этапе остается неизменным.
3. Модуляция символов. На этом этапе пакет из 144 полученных с выхода перемежителя бит разбивается на 24 символа из 6 бит. Генерируется таблица функций Уолша длиной 64 бита. Каждый 6- битный символ заменяется последовательностью Уолша, номер которой равен значению данных 6-ти бит. Т.о. на выходе модулятора получается  $24 * 64 = 1536$  знаковых символов.
4. Прямое расширение спектра. Полученная последовательность из 1536 символов периодически умножается с учетом знака на ПСП длиной 511 символов. Далее к началу сформированного символьного пакета прикрепляется немодулированная ПСП. Т.о. символьная длина становится равной 2047. Далее полученные символы модулируются методом BPSK.

### 3. Теоретическая информация

В телекоммуникации канал связи — это канал связи, который соединяет два или более общающихся устройств. Этот канал связи может быть физическим или логическим, он может использовать один или более физических каналов связи.

Телекоммуникационный канал связи это, как правило, один из нескольких видов каналов передачи информации, таких как те, которые предусмотрены в спутниковой связи, наземной радио-коммуникационной инфраструктуре и в компьютерных сетях, для подключения двух или более точек.

Термин связь широко используется в компьютерной сети для обозначения средств связи, соединяющих узлы сети. Если канал связи является логическим звеном, типы физического канала всегда должны быть указаны (например, канал передачи данных, канал передачи данных от абонента к базовой станции, канал передачи данных от базовой станции к абоненту, волоконно-оптический канал передачи данных, канал передачи данных точка-точка, и т. д.)

### 4. Ход выполнения работы

#### 4.1. Чтение сообщения

```
1 %read file
2 fid=fopen('test.sig', 'rb');
3 message = fread(fid, 'int16');
4 fclose(fid);
```

Для начала читаем принимаемое сообщение из файла. Далее, зная, что данные сгенерированы с передискретизацией, равной двум (что означает, что одному BPSK символу соответствуют два расположенных друг за другом отчета), считываем каждое второе значение.

```
1 l = length(message) / 2;
2 IQ = zeros(1, l);
3 k = 1;
4 for(i=1:2:length(message))
5 IQ(k) = message(i);
6 IQ(k) = IQ(k) + 1j*(message(i+1));
7 k = k + 1;
8 end
```

#### 4.2. Демодуляция

Производим демодуляцию сигнала.

```
1 demodulated_signal = pskdemod(IQ, 2);
```

#### 4.3. Модуляция символов

С помощью корреляции определяем положение синхропосылки PRS и восстанавливаем последовательность до добавления синхропосылки.

```

1 N = 2^nextpow2(length(demodulated_signal));
2 signalSpectr = fft(demodulated_signal, N);
3 syncSpectr = fft(PRS', N);
4
5 mult_ = signalSpectr.*conj(syncSpectr);
6 result_ = ifft(mult_, N);
7 n_ = find(result_ >= 200);
8 signal_to_modulate2_ = IQdemod((length(PRS)+n(1)):length(IQdemod));
9 signal_to_modulate1_ = signal_to_modulate2_*[PRS' PRS' PRS' PRS(1:3)'];

```

Теперь нужно получить последовательность до применения функции Уолша. В вектор **walsh\_row** записываем индексы из матрицы Уолша, а затем переписываем номера индексов как двоичную последовательность в матрицу **sig\_matrix**.

```

1 signal1 = reshape(signal_to_modulate1, [64 24]);
2 N_ = 64;
3 hadamardMatrix_ = hadamard(N);
4
5 walsh_row_ = zeros(1, 24);
6 for i = 1:24
7 ind = ismember(hadamardMatrix_, signal1(i, :), 'rows');
8 walsh_row(i) = find(ind == 1);
9 end
10 walsh_row_ = (walsh_row_ - 1)';
11 sig_matrix = de2bi(walsh_row, 6, 'left-msb');

```

#### 4.4. Перемежение бит

Теперь необходимо выполнить перемежение бит.

```

1 sig = reshape(sig_matrix', [1 144]);
2 convcode(int16(interleaver+1))_ = sig;

```

#### 4.5. Декодирование

Осталось раскодировать посылку, закодированную сверточным кодом с образующими полиномами 753, 561(octal) и кодовым ограничением 9

```

1 trellis = poly2trellis(9, [753 561]);
2 packet = vitdec(convcode, trellis, 1, 'trunc', 'hard');
3
4 packetR = [packet(1:16); packet(17:32); packet(33:48)];
5 code = bi2de(packetR, 'left-msb');

```

После всех операций можно убедиться, что принятое сообщение совпало с переданным.

```

Исходное сообщение: 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 1 1 0 0
1 0 0 0 0 0 0 0 0 0 0 1 0 1 1 1 0 1 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
Принятое сообщение: 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 1 1 0 0
1 0 0 0 0 0 0 0 0 0 1 0 1 1 1 0 1 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

```

## 5. Выводы

В итоге была получена модель приемника, выполняющая демодуляцию, перемежение и декодирование принятого сигнала. Полученный сигнал полностью соответствует переданному сообщению.