

САНКТ-ПЕТЕРБУРГСКИЙ
ПОЛИТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ
ПЕТРА ВЕЛИКОГО

ИНСТИТУТ КОМПЬЮТЕРНЫХ НАУК И ТЕХНОЛОГИЙ

КАФЕДРА КОМПЬЮТЕРНЫХ СИСТЕМ И ПРОГРАММНЫХ ТЕХНОЛОГИЙ

Отчет
по лабораторной работе №8
на тему: "Модель телекоммуникационного канала"

выполнила:
Шевченко А.С.
группа: 33501/1
преподаватель:
Богач Н.В.

1. Цель работы

По имеющейся записи сигнала из эфира и коду модели передатчика создать модель приемника, в которой найти позицию начала пакета и, выполнив операции демодуляции, демultiplexирования и декодирования, получить передаваемые параметры: ID, период, и номер пакета. Известно, что ID = 4, период 100 мс, номер пакета 373. 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. Постановка задачи

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. Перемещение бит. Количество бит на этом этапе остается неизменным.
3. Модуляция символов. На этом этапе пакет из 144 полученных с выхода перемещителя бит разбивается на 24 символа из 6 бит. Генерируется таблица функций Уолша длиной 64 бита. Каждый 6битный 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 началу сформированного символического пакета прикрепляется немодулированная ПСП. Т.о. символьная длина становится равной 1747. Далее полученные символы модулируются методом BPSK.

3. Ход работы

1. Чтение данных

```
1 - fid=fopen('D:\!PROJECTS\Matlab\Telecom\Lab8\test.sig', 'rb');
2 - A = fread(fid, '2*int16=>int16', 4)';
3 - fclose(fid);
4
5 %
6 - im_part = zeros(1, length(A)/2);
7 - re_part = zeros(1, length(A)/2);
8 - k = 1;
9 - for i = 1:2:length(A)
10 -     re_part(k) = A(i);
11 -     im_part(k) = A(i+1);
12 -     k = k + 1;
13 - end
```

Рис. 1: Фрагмент кода приемника, в котором осуществляется чтение данных

Здесь мы считали наши данные, сгенерированные передатчиком. В связи с тем, что данные сгенерированы с передискретизацией равной 2, то есть одному BPSK символу соответствуют два лежащих друг за другом отсчета, нам нужно считывать каждое второе значение. Далее необходимо разделить их на действительную и мнимую части.

2. Демодуляция

Теперь произведем демодуляцию сигнала, модулированного методом BPSK.

```

15 - IQ = zeros(1, length(im_part));
16 - for i = 1:length(im_part)
17 -     IQ(i) = re_part(i) + 1j*im_part(i);
18 - end
19 - IQdemod = pskdemod(IQ, 2);
20 -

```

Рис. 2: Фрагмент кода приемника, в котором осуществляется демодуляция

3. Синхропосылка

Нам дана синхропосылка PRS, ее положение определим с помощью алгоритма корреляции, а далее восстановим последовательность до добавления синхропосылки.

```

48 - N = 2^nextpow2(length(IQdemod));
49 - signalSpectr = fft(IQdemod, N);
50 - syncSpectr = fft(PRS', N);
51 -
52 - mult = signalSpectr.*conj(syncSpectr);
53 - result = ifft(mult, N);
54 -
55 - n = find(result >= 200);
56 -
57 - signal_to_modulate2 = IQdemod((length(PRS)+n(1)):length(IQdemod));
58 - signal_to_modulatel = signal_to_modulate2.*[PRS' PRS' PRS' PRS(1:3)'];
59 -
60 - signall = reshape(signal_to_modulatel, [64 24])';

```

Рис. 3: Фрагмент кода приемника, в котором осуществляется обработка синхропосылки

4. Функция Уолша

Восстанавливаем последовательность после применения функций Уолша.

```

65 - N=64;
66 - hadamardMatrix=hadamard(N);
67 -
68 - walsh_row = zeros(1,24);
69 - for i = 1:24
70 -     ind = ismember(hadamardMatrix, signal(i,:), 'rows');
71 -     walsh_row(i) = find(ind == 1);
72 - end
73 - walsh_row = (walsh_row - 1)';
74 - sig_matrix = de2bi(walsh_row, 6, 'left-msb');
75 -

```

Рис. 4: Фрагмент кода приемника, в котором осуществляется восстановление последовательность после функции Уолша в источнике

5. Перемежитель

Далее необходимо сделать обратное перемежение с помощью матрицы интерливера, которую применял источник.

```

77 - interleaver = [0; 133; 122; 111; 100; 89; 78; 67; 56; 45; 34; 23; 12; 1; 134; 123;
78     112; 101; 90; 79; 68; 57; 46; 35; 24; 13; 2; 135; 124; 113; 102; 91;
79     80; 69; 58; 47; 36; 25; 14; 3; 136; 125; 114; 103; 92; 81; 70; 59;
80     48; 37; 26; 15; 4; 137; 126; 115; 104; 93; 82; 71; 60; 49; 38; 27;
81     16; 5; 138; 127; 116; 105; 94; 83; 72; 61; 50; 39; 28; 17; 6; 139;
82     128; 117; 106; 95; 84; 73; 62; 51; 40; 29; 18; 7; 140; 129; 118; 107;
83     96; 85; 74; 63; 52; 41; 30; 19; 8; 141; 130; 119; 108; 97; 86; 75;
84     64; 53; 42; 31; 20; 9; 142; 131; 120; 109; 98; 87; 76; 65; 54; 43;
85     32; 21; 10; 143; 132; 121; 110; 99; 88; 77; 66; 55; 44; 33; 22; 11];
86
87
88 - sig = reshape(sig_matrix', [1 144]);|
89 - convcode(intl6(interleaver+1)) = sig;
90

```

Рис. 5: Фрагмент кода приемника, в котором осуществляется восстановление последовательности после перемежения в источнике

6. Сверточный код

Окончательно раскодируем нашу последовательность.

```

91     %% convolution encoder
92 -   trl=poly2trellis(9,[753 561]);
93 -   packet=vitdec(convcode,trl, 1, 'trunc', 'hard');
94

```

Рис. 6: Фрагмент кода приемника, в котором осуществляется раскодирование последовательности

7. Проверка

Проверим наш результат, для этого вычислим контрольную сумму, если она совпадет с суммой, сгенерированной передатчиком, то посылка принята без ошибок.

```

98 -   hDetect=crc.generator('Polynomial', '0x1021');
99 -   decode_with_crc=generate(hDetect,packet');
100 -   crc_send = decode_with_crc(73:88);
101
102 -   packetR = [packet(1:16); packet(17:32); packet(33:48); packet(49:64)];
103 -   code = bi2de(packetR, 'left-msb');
104
105

```

Рис. 7: Фрагмент кода приемника, в котором осуществляется проверка полученной последовательности

4. Выводы

Таким образом, мы описали модель приемника, в которой осуществлялись операции демодуляции, де-перемежения, декодирования. В результате обработки заданного сигнала, мы получили результат, соответствующий передаваемым данным: ID = 4, период = 100мс, номер пакета = 373.