

Мажоритарное кодирование. Код Хэмминга.
Сверточный код. Разбор примеров

Содержание

Номенклатура кодов телекоммуникационных систем

Корректирующая и обнаруживающая способности кодов

Линейный код

Код Хэмминга

Код Хэмминга (7,4). Упражнение

Сверточный код

Сверточный код (5,7)

Сверточный код (5,7). Упражнение

Исправление ошибок. Алгоритм Витерби

Номенклатура кодов телекоммуникационных систем

- ▶ Обнаружение ошибок
 - ▶ Бит четности (асинхронные протоколы: RS-232, RS-485 и др.)
 - ▶ CRC (синхронные протоколы: Ethernet и др.)
- ▶ Исправление ошибок
 - ▶ Мажоритарное кодирование
 - ▶ Блочные коды
 - ▶ Циклические коды
 - ▶ Сверточные коды
 - ▶ Прочие семейства кодов в развитие базовых методов кодирования (полярные, турбокоды, LDPC-коды и др.)

Расстояние (дистанция) Хэмминга

Расстояние Хэмминга (D) — число позиций, в которых соответствующие символы двух кодовых слов одинаковой длины различны. Используем это понятие для двоичных кодовых слов:

Information word	Codeword
0	000
1	111

Рис. 1: $D=1$; при мажоритарном кодировании D увеличивается соответственно числу повторов $D=3$

Information word	Codeword
0	00000
1	11111

Рис. 2: $D=5$

Способность кода к обнаружению и исправлению ошибок

- Способность кода к обнаружению и исправлению ошибок определяется как:

$$C_d = D - 1$$

,

$$C_c = \frac{D - 1}{2}$$

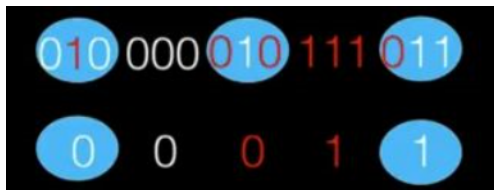


Рис. 3: $C_d = 2$; $C_c = 1$

Код Хэмминга. Идея

Ричард Уэсли Хэмминг (1915-1998)



Код Хэмминга обнаруживает 2 и исправляет 1 ошибку типа замены бита. Является систематическим линейным кодом. Одновременно является блочным БЧХ-кодом, исправляющим 1 ошибку, а также циклическим кодом.

№ проверки	Проверочные позиции	№ контрольного символа
1	1,3,5,7,9,11,...	1
2	2,3,6,7,10,11,14,15,18,19,22,24,...	2
3	4,5,6,7,12,13,14,15,20,21,22,23,...	4
4	8,9,10,11,12,13,14,15,24,25,26,27,28,29,...	8

Код Хэмминга. Подготовка

0	0	0	1	a1
0	0	1	0	a2
0	0	1	1	a3
0	1	0	0	a4
0	1	0	1	a5
0	1	1	0	a6
0	1	1	1	a7
1	0	0	0	a8
1	0	0	1	a9
1	0	1	0	a10
1	0	1	1	a11
1	1	0	0	a12
1	1	0	1	a13
1	1	1	0	a14
1	1	1	1	a15

Код Хэмминга (7,4). Передатчик

1...3...5...7...

2...3...6...7...

4...5...6...7...

Контрольные биты - биты с номерами, которые являются степенью числа 2 – b_1, b_2, b_4

Информационные биты – b_3, b_5, b_6, b_7

$$b_1 = b_3 \oplus b_5 \oplus b_7$$

$$b_2 = b_3 \oplus b_6 \oplus b_7$$

$$b_4 = b_5 \oplus b_6 \oplus b_7$$

Код Хэмминга (7,4). Приемник

Синдром $\{s_1, s_2, s_3\}$:

$$s_1 = b_1 \oplus b_3 \oplus b_5 \oplus b_7$$

$$s_2 = b_2 \oplus b_3 \oplus b_6 \oplus b_7$$

$$s_3 = b_4 \oplus b_5 \oplus b_6 \oplus b_7$$

Синдром указывает номер ошибочного бита.

Если $\{s_1, s_2, s_3\} = \{000\}$, ошибок нет.

Порождающая и проверочная матрицы

$$\mathbb{G} = \begin{pmatrix} 7 & 6 & 5 & 3 & \vdots & 4 & 2 & 1 \\ \hline 1 & 0 & 0 & 0 & \vdots & 1 & 1 & 1 \\ 0 & 1 & 0 & 0 & \vdots & 1 & 1 & 0 \\ 0 & 0 & 1 & 0 & \vdots & 1 & 0 & 1 \\ 0 & 0 & 0 & 1 & \vdots & 0 & 1 & 1 \end{pmatrix} \quad \mathbb{H}^T = \begin{pmatrix} s_3 & s_2 & s_1 \\ \hline 1 & 1 & 1 \\ 1 & 1 & 0 \\ 1 & 0 & 1 \\ 0 & 1 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

Код Хэмминга (7,4): Кодирование

$$\mathbf{a} = \begin{bmatrix} 7 & 6 & 5 & 3 \\ 0 & 1 & 0 & 1 \end{bmatrix}$$

$$\mathbf{c} = \mathbf{a} \times \mathbb{G}$$

$$\mathbf{c} = \begin{bmatrix} 7 & 6 & 5 & 3 \\ 0 & 1 & 0 & 1 \end{bmatrix} \times \begin{pmatrix} 7 & 6 & 5 & 3 & \vdots & 4 & 2 & 1 \\ 1 & 0 & 0 & 0 & \vdots & 1 & 1 & 1 \\ 0 & 1 & 0 & 0 & \vdots & 1 & 1 & 0 \\ 0 & 0 & 1 & 0 & \vdots & 1 & 0 & 1 \\ 0 & 0 & 0 & 1 & \vdots & 0 & 1 & 1 \end{pmatrix}$$

$$\mathbf{c} = \begin{bmatrix} 7 & 6 & 5 & 3 & \vdots & 4 & 2 & 1 \\ 0 & 1 & 0 & 1 & \vdots & 1 & 0 & 1 \end{bmatrix}$$

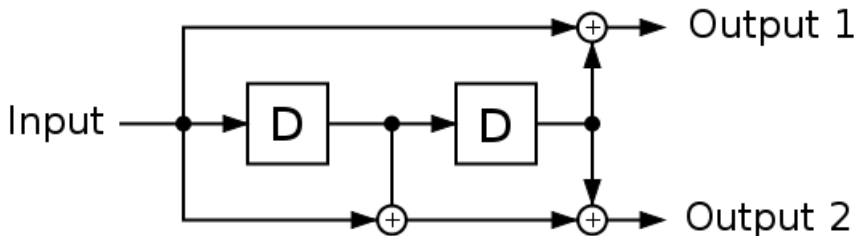
Код Хэмминга (7,4):Вычисление синдрома

$$\left[\begin{array}{cccc|cc} 7 & 6 & 5 & 3 & 4 & 2 & 1 \\ \hline 0 & 1 & 0 & 1 & 1 & 0 & 1 \end{array} \right] \times \begin{pmatrix} s_3 & s_2 & s_1 \\ 1 & 1 & 1 \\ 1 & 1 & 0 \\ 1 & 0 & 1 \\ 0 & 1 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} = [s_3 \quad s_2 \quad s_1]$$

Код Хэмминга (7,4). Упражнение

Инвертируйте один любой бит вектора s и получите двоичный код ошибки в синдроме

Простейший сверточный код. Структура кодера

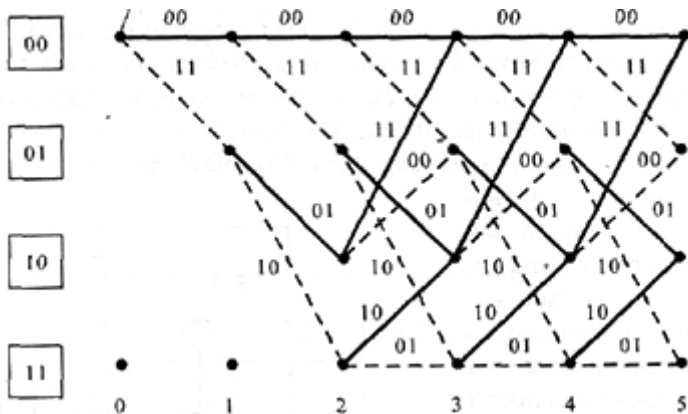


Кодер (5,7). В двоичном представлении (101, 111) или в полиномиальном ($x^2 + 1$, $x^2 + x + 1$). Цифры показывают, какие связи участвуют в формировании сигнала на выходе. Output 1 - только первый и третий (5 или 101). Output 2 - все три выхода (7 или 111). Скорость кодирования $R=1/2$ - на один входной бит - два выходных

Решетка кодера

- ▶ Структуре сверточного кодера однозначно соответствует решетка кодера - направленный граф, с помощью которого производится кодирование и декодирование.
- ▶ Узлы решетки - содержимое ячеек памяти кодера (00, 01, 10, 11). Ребра описывают, в какое следующее состояние переходят ячейки кодера. Вес ребра - состояние на выходе кодера (в положениях ключа 2 и 3)
- ▶ Из каждого узла решетки ведет два пути.
- ▶ "Нулевой" путь показывает, что произойдет (что запишется в ячейки кодера и что будет на выходе), если на вход кодера приходит новый бит, равный 0.
- ▶ "Единичный" путь показывает, что произойдет, если на вход кодера приходит новый бит, равный 1.

Пусть кодер задан решеткой:



Начальное состояние 00. Нулевой путь показан сплошной линией, единичный - штриховой

Кодирование 0101

Закодируем посылку 0101.

1. Начальное состояние 00. Поскольку первый бит заданной посылки равен 0, то смотрим, куда из узла 00 (начальное состояние) ведет нулевой путь. Переходим по сплошной линии в состояние 00, отдавая на выход код 00 (вес этого ребра).
2. Следующий бит на входе 1. После шага 1 мы находимся в состоянии 00. Смотрим, куда из 00 ведет единичный путь. Переходим по штриховой линии в узел 01, отдавая на выход код 11
3. Следующий бит на входе снова 0. Мы находимся в состоянии 01. Смотрим, куда из 01 ведет нулевой путь. Переходим по сплошной линии в узел 10, отдавая на выход код 01
4. И последний бит посылки 1. Мы находимся в состоянии 10. Смотрим, куда из 10 ведет единичный путь. Переходим по штриховой линии в узел 01, отдавая на выход код 00

Выходная посылка 00110100

Декодирование 00110100

1. Декодирование происходит на той же решетке в том же направлении слева направо.
2. Находясь в начальном узле 00, смотрим, какому пути соответствуют первые два бита принятой посылки - нулевому или единичному? Видим, что вес 00 имеет сплошная линия, т.е. нулевой путь. **Декодируем 0.** Этот путь ведет в состояние тоже 00.
3. Находимся в узле 00, в который перешли на шаге 1. Смотрим, какому пути соответствуют вторые два бита принятой посылки - нулевому или единичному? Видим, что вес 11 имеет штриховая линия, т. е. единичный путь. **Декодируем 1.** Этот путь ведет в состояние 01.
4. Продолжая аналогично, из узла 01, декодируем посылку 0101

Как получить решетку кодера по структуре или полиномиальному описанию?

- ▶ Разрядность узла соответствует числу ячеек кодера (ячеек памяти, звеньев задержки, длине регистра), она равна степени полинома (или числу связей -1). В нашем случае линия состоит из двух ячеек.
- ▶ Задавшись начальным состоянием (обычно 00) моделируют реакцию кодера для нулевого и единичного входа, т.е. сдвигают регистр нулем и единицей последовательно для всех комбинаций в ячейках кодера (в нашем случае 00, 01, 10, 11)

Например, пусть в ячейках кодера находится комбинация 01. Рассмотрим, как поведет себя кодер при приходе 0. Кодер сдвинули нулем ($0 \rightarrow 01$), он перейдет в состояние 00.

Рассчитаем вес этого пути $\{out_1, out_2\}$:

$$out_1 = 1 + x^2 \quad out_1 = 0 \oplus 1 = 1$$

$$out_2 = 1 + x + x^2 \quad out_2 = 0 \oplus 0 \oplus 1 = 1$$

Значит, нулевой путь из 01 ведет в 00 с весом 11

Пусть теперь кодер сдвинули единицей ($1 \rightarrow 01$), он перейдет в состояние 10. Рассчитаем вес этого пути $\{out_1, out_2\}$:

$$out_1 = 1 + x^2 \quad out_1 = 1 \oplus 1 = 0$$

$$out_2 = 1 + x + x^2 \quad out_2 = 1 \oplus 0 \oplus 1 = 0$$

Значит, единичный путь из 01 ведет в 10 с весом 00

Обратите внимание! Дистанция Хэмминга между весами нулевого и единичного путей максимальна для данной разрядности кодера

Сверточный код (5,7). Упражнение

Закодируйте и декодируйте на решетке кодера, приведенной выше, произвольную 4-х битовую комбинацию.

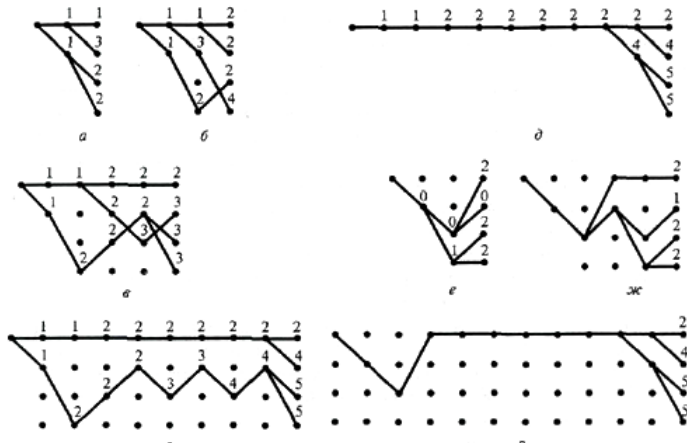
Исправление ошибок. Алгоритм Витерби

- ▶ Для исправления ошибок также используется решетка кодера
- ▶ Одиночная ошибка в канале приводит к тому, что при декодировании не находится вес узла, соответствующий принятым двум битам (в нашем примере).
- ▶ Алгоритм декодирования разветвляется и образует два гипотетических пути. Далее переходит к следующей паре бит. Если гипотетический путь ведет в узел из которого есть путь, соответствующий следующей паре, штраф не увеличивается (угадали ветку на решетке, можно исправить ошибку).

Исправление ошибок. Алгоритм Витерби

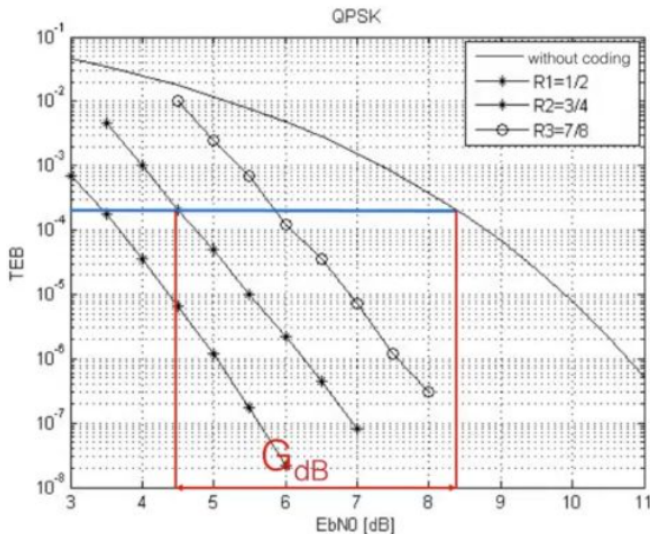
- ▶ Если снова нет правильного веса, алгоритм снова разветвляется, а штраф увеличивается.
- ▶ В результате алгоритм Витерби декодирует переданное сообщение наиболее вероятным способом вдоль пути с наименьшим штрафом, исправляя по возможности ошибки.
- ▶ Для борьбы с пачками ошибок сверточные кодеры обычно используют совместно с перемежителями. Перемежители изменяют порядок следования бит, чтобы, тем самым, при обратном перемежении на приемнике превратить пачки ошибок в одиночные ошибки.
- ▶ Глубина просмотра решетки является задаваемым параметром.

Исправление ошибок. Алгоритм Витерби



Ветвление путей при различных местах расположения ошибки

Канал с помехоустойчивым кодированием



Изменение наклона кривой $P_{err} = f(SNR)$ QPSK - модулятора при внедрении в канал сверточного кодера

Алгоритм Витерби. Упражнение

В примере выше сделайте 1 ошибку в любом бите 8-битовой посылки и исправьте ее, пользуясь решеткой кодера.

Материалы для изучения

- ▶ **Digital Signal Processing.** *École Polytechnique Fédérale de Lausanne.* M. Vetterli, P. Prandoni
- ▶ **Introduction to Satellite Communications.** *Institut Mines-Telecom.* L. Franck et al.
- ▶ **Блейхут Р.** Теория и практика кодов, контролирующих ошибки. – Мир, 1986. – Т. 576.