

А. А. Бурков, А. М. Тюрликов

ИСПОЛЬЗОВАНИЕ ГИБРИДНОЙ ОБРАТНОЙ СВЯЗИ В БЕСПРОВОДНЫХ СЕТЯХ



МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ
РОССИЙСКОЙ ФЕДЕРАЦИИ
Федеральное государственное автономное образовательное
учреждение высшего образования
САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
АЭРОКОСМИЧЕСКОГО ПРИBORОСТРОЕНИЯ

А. А. Бурков, А. М. Тюрликов

ИСПОЛЬЗОВАНИЕ ГИБРИДНОЙ
ОБРАТНОЙ СВЯЗИ
В БЕСПРОВОДНЫХ СЕТЯХ

Учебно-методическое пособие

УДК 654
ББК 32.884
Б90

Рецензенты:

кандидат технических наук, доцент *А. Д. Жуков*;
кандидат технических наук, доцент *А. А. Овчинников*

Утверждено
редакционно-издательским советом университета
в качестве учебно-методического пособия

Протокол № 1 от 17 января 2022 г.

Бурков, А. А.

Б90 Использование гибридной обратной связи в беспроводных сетях:
учеб.-метод. пособие / А. А. Бурков, А. М. Тюрликов. – СПб.: ГУАП,
2022. – 47 с.

Содержит две лабораторные работы, предназначенные для закрепления теоретического материала лекционных курсов «Сети и системы мобильной связи» и «Сотовые сети», приводятся все необходимые сведения для их выполнения. При выполнении лабораторных работ требуется разработка программ имитационного моделирования.

Предназначено для студентов, обучающихся по направлениям 10.03.01, 10.04.01 «Информационная безопасность», 11.03.02, 11.04.02 «Инфокоммуникационные технологии и системы связи», а также студентов других технических направлений и специальностей, изучающих системы передачи данных.

Подготовлено кафедрой инфокоммуникационных систем.

УДК 654
ББК 32.884

ВВЕДЕНИЕ

В современных сотовых сетях и других беспроводных сетях гибридная обратная связь находит широкое применение. В сочетании с гибридной обратной связью для передачи данных пользователя используются следующие типы помехоустойчивого кодирования: турбокоды в сетях 4G, LDPC-коды в сетях 5G. В данном курсе изучаются основные методы использования помехоустойчивого кодирования в сочетании с гибридной обратной связью. При этом в лабораторных работах используются сверточные коды, так как на них проще продемонстрировать данные методы.

Методические указания содержат две лабораторные работы, предназначенные для закрепления теоретического материала лекционных курсов “Сети и системы мобильной связи” и “Сотовые сети”. Для выполнения лабораторных работ используется среда программирования MatLab. В первом разделе методических указаний приводятся все необходимые сведения для выполнения лабораторных работ. Литература, в которой содержится дополнительный теоретический материал, приведена в библиографическом списке.

Выполнение лабораторных работ можно разбить на следующие этапы:

- сдача двух допусков к лабораторным работам;
- выполнение первой лабораторной работы;
- сдача отчета по первой лабораторной работе;
- выполнение второй лабораторной работы;
- сдача отчета по второй лабораторной работе.

Содержание методического пособия построено по следующему принципу:

- Первый раздел содержит краткие сведения, необходимые для выполнения лабораторных работ;
- Второй и третий разделы содержат описание первого и второго допуска к выполнению лабораторных работ соответственно;
- Четвертый и пятый разделы содержат описание первой и второй лабораторной работы соответственно;
- Шестой раздел содержит список вариантов для выполнения лабораторных работ;
- Приложение 1 и приложение 2 содержат дополнительные сведения для выполнения лабораторных работ.

Авторы хотели бы выразить свою глубокую признательность А. Ю. Глушенковой, В. Ю. Михайлову и Р. О. Рачугину за помощь при подготовке пособия.

1. ПРИНЦИПЫ ПОСТРОЕНИЯ СИСТЕМ С ГИБРИДНОЙ ОБРАТНОЙ СВЯЗЬЮ И СПОСОБЫ АНАЛИЗА ХАРАКТЕРИСТИК ТАКИХ СИСТЕМ

1.1. Системы с гибридной обратной связью

1.1.1. Основная идея систем с обратной связью

Использование системы передачи данных с обратной связью эффективно в случае отправки сообщений по каналу с переменными параметрами, поэтому данный тип систем интересен при беспроводной связи.

Существуют два основных метода использования канала обратной связи. В первом методе решение о повторной передаче принимается приёмником, например, на основе оценки надёжности принятого сообщения или на основе контрольной суммы, передаваемой вместе с сообщением. Такие системы называются системами с решающей обратной связью (РОС) или с автоматическим запросом повторной передачи (ARQ – Automatic Repeat request). Во втором методе решение о повторной передаче принимает передатчик на основе информации, полученной по обратному каналу, например, некоего условного ответа, вычисленного на основе принятого приёмником сигнала. Такие системы называются системами с информационной обратной связью.

Идея схемы гибридной решающей обратной связи (ГРОС, HARQ) заключается в сочетании метода обнаружения ошибок схемы ARQ с исправлением ошибок помехоустойчивого кодирования. Для достижения этой цели передатчик выполняет следующие функции:

- добавление к передаваемому сообщению контрольной суммы (КС), которая может быть вычислена с использованием избыточного циклического кода (CRC);
- кодирование полученных данных помехоустойчивым кодом (свёрточные, турбо-коды, LDPC и др.);
- передача данных в соответствии с используемой модуляцией;
- ожидание квитанции; при получении положительной квитанции (ACK) начинается передача следующего сообщения, при отрицательной квитанции (NACK) повторяется передача текущего сообщения.

Приёмник выполняет следующие функции:

- приём сообщения с используемой модуляцией;
- декодирование помехоустойчивого кода с исправлением ошибок;

- проверка контрольной суммы;
- передачу по обратному каналу в случае совпадения контрольной суммы с заданной положительной квитанции; если контрольная сумма не совпала, то по каналу обратной связи передается отрицательная квитанция, также полученное в текущем приеме сообщение может быть записано в буфер и использовано для повышения надёжности декодирования.

По способу использования в системе неверно принятого сообщения, то есть содержащего ошибки после процедуры исправления ошибок, можно выделить несколько типов ГРОС.

1.1.2. Решающая обратная связь

Решающая обратная связь (РОС) – это система двусторонней связи с автоматическим запросом на повтор. Данные передаются в виде блоков информации, то есть пакетов. Если блок информации получен с ошибками и декодирован unsuccessfully подается запрос на повторную передачу этого же пакета. Обратная связь используется для передачи информации с приемника на передатчик. При успешном декодировании на приемник посылается положительная квитанция (Acknowledgement, ACK), если декодер вынес неверное решение, то на передатчик поступает отрицательная квитанция (Negative-acknowledgement, NACK) после чего будет выполнена повторная передача сообщения. Повторные передачи будут повторяться пока декодер не вынесет верное решение, либо количество повторных передач выйдет за заранее заданный лимит. Схема работы такой системы представлена на рис. 1.1.

Самый простой алгоритм обратной связи – Stop-and-Wait (алгоритм с ожиданием). В период передачи сигнала запроса повторной передачи по каналу обратной связи, могут быть приняты некоторые последующие сообщения, передаваемые по прямому каналу. Доставка сообщений пользователю в исходном порядке требует буферизации. При этом, если необходимо несколько попыток повторной передачи для успешного приема, то размер буфера может стать довольно большим. Метод остановись и жди (SW) позволяет избежать буферизации. Идея метода заключается в том, что передатчик ожидает или положительного подтверждения, или повторного запроса перед тем, как начать передачу следующего блока.

В рамках лабораторных работ рассматривается алгоритм работы с обратной связью «алгоритм с ожиданием». Однако рассматриваемые методики могут быть легко обобщены и на более эффективные

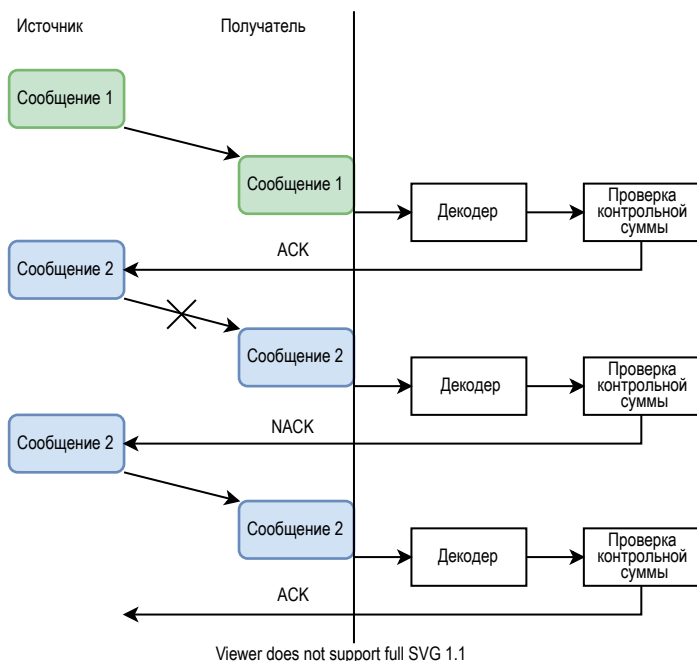


Рис. 1.1 Решающая обратная связь

алгоритмы, учитывающие задержку в получении квитанции по обратной связи [1].

1.1.3. Гибридная решающая обратная связь с комбинированием по Чейзу

Существуют алгоритмы, которые не отбрасывают неверно декодированные сообщения, а сохраняют их и впоследствии используют для декодирования повторной отправки одного и того же сообщения. Одним из таких алгоритмов является второй тип гибридной решающей обратной связи. Он представляет собой более сложную систему, чем описывалась ранее. Помимо повторных передач в данной системе используется накопительный буфер. При неудачном декодировании принятая с ошибками последовательность сохраняется в буфере. При повторной передаче сообщение снова декодируется и при повторной неудаче в буфер записывается вторая принятая с ошибками последовательность. После этого происходит попытка декодирования буфера, в котором хранятся значения надежности для каждого

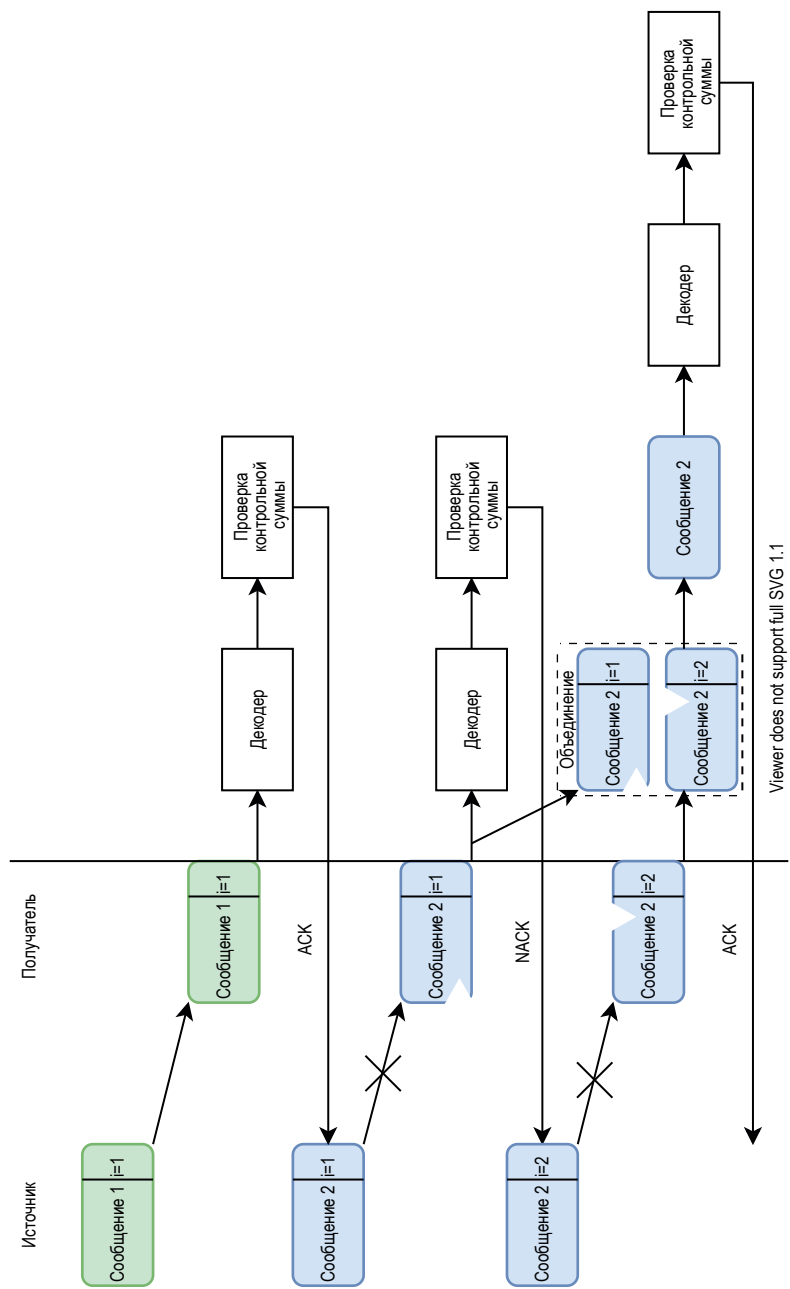


Рис. 1.2 Гибридная решающая обратная связь с комбинированием по Чейзу

символа из принятой последовательности. Наличие буфера и есть главное отличие первого типа гибридной решающей обратной связи от второго. При использовании такой модели системы будет происходить меньшее количество повторных передач. Приемник объединит минимальное необходимое количество пакетов, для получения правильных данных, что соответственно и ускорит работу такой системы. Повторные передачи также будут осуществляться, пока декодер не вынесет верное решение или их количество не превзойдет заданный лимит. Схема работы такой системы представлена на рис. 1.2. Второй тип гибридной решающей обратной связи с накопительным буфером также называется комбинированием по Чейзу.

1.1.4. Гибридная решающая обратная связь с наращиваемой избыточностью

Идея систем типа ГРОС с наращиваемой избыточностью заключается в использовании переменного количества проверочных бит, чтобы обеспечить исправление ошибок на приемной стороне с использованием большего числа проверочных бит, чем на самом деле требуется в текущих условиях канала. Эта схема показана на рис. 1.3. В данной схеме для обозначения проверочных бит используется сокращение «пб».

Во время первой попытки передачи, посылаются только m биты сообщения и n битов четности, то есть используется код со скоростью $R1$. Этот код декодируется на стороне приемника, в случае успеха АСК передается по каналу обратной связи. В противном случае, посылается NACK, и во время второй попытки передаются только $r2$ проверочные биты (без повторения сообщения). В приемнике эти проверочные биты объединяются с m и $r1$, получая таким образом код с скоростью $R2 < R1$ и большей корректирующей способностью (процедуры сочетания и декодирования используются в зависимости от выбранного типа кодов). При необходимости, во время последующих повторных передач передаются только дополнительные (новые) избыточные биты. Таким образом, эта схема часто называется схемой инкрементальной избыточности (IR).

1.1.5. Сравнение типов обратной связи

В данном подразделе были описаны 3 типа обратной связи. Данные виды перечислены в табл. 1.1 с указанием специфики работы приемника и передатчика. Особенности обработки на приемной стороне приведены в табл. 1.2 [4].

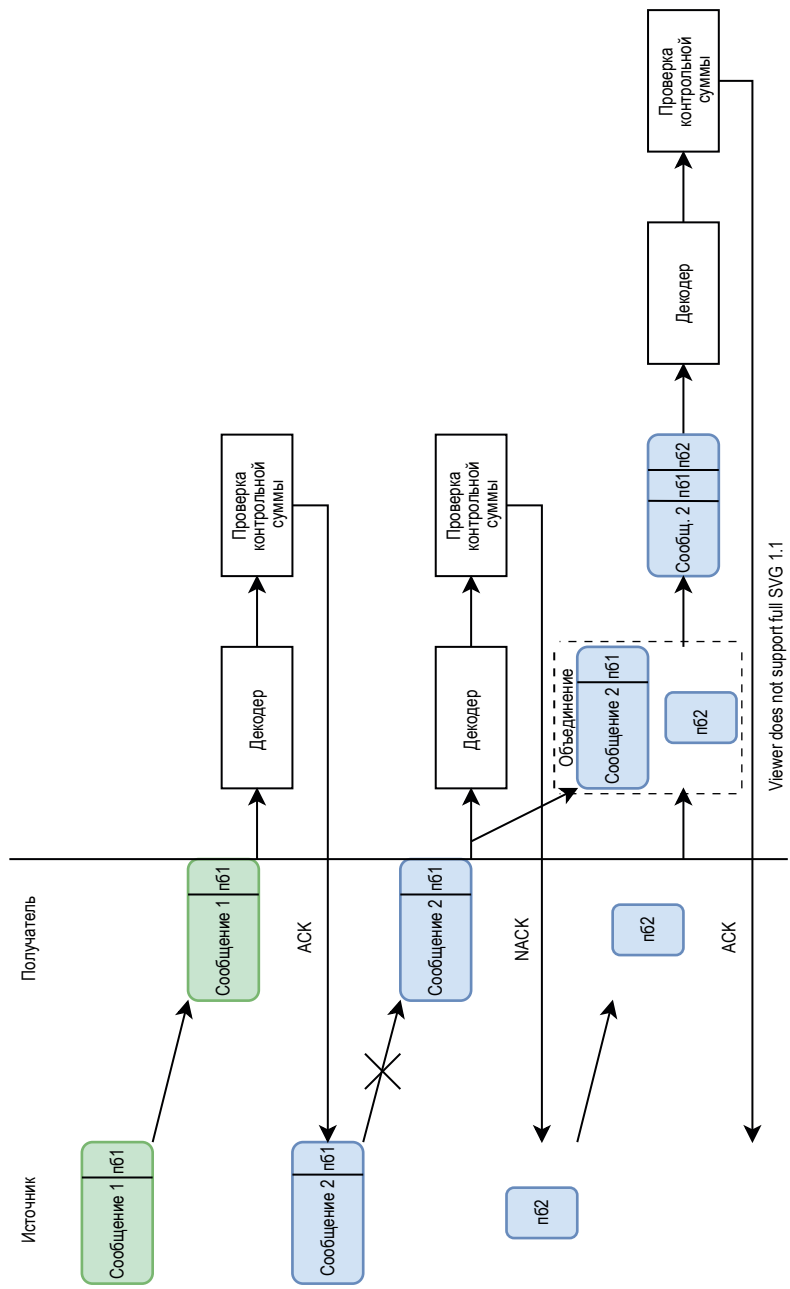


Рис. 1.3. Гибридная решающая обратная связь с наращиваемой избыточностью

Таблица 1.1

Виды гибридной решающей обратной связи

№	Тип связи	Работа передатчика	Работа приемника
1	РОС (ARQ)	Передаётся одно и то же сообщение, пока не будет получена положительная квитанция; используется помехоустойчивый код с выбранной скоростью	Неверно принятые сообщения удаляются
2	ГРОС с комбинированием по Чейзу (<i>HARQ-Chase Combining</i>)		Неверно принятые сообщения сохраняются в буфер и используются для улучшения декодирования последующих повторных передач
3	ГРОС с Нарастиваемой избыточностью (<i>HARQ – Incremental Redundancy</i>)	При передаче может передаваться не всё сообщение, а его фрагмент; скорость кода может меняться	

Таблица 1.2

Особенности работы приведённых типов ГРОС

№	Тип связи	Особенности работы типа связи
1	РОС (ARQ)	не требуется наличия буфера; не используется информация, содержащаяся в неверно принятых сообщениях
2	ГРОС с комбинированием по Чейзу (<i>HARQ-Chase Combining</i>)	требуется наличия буфера; используется информация, содержащаяся в неверно принятых сообщениях; имеет низкую скорость передачи в областях относительно больших соотношений сигнал/шум; сохраненные в буфере неверно принятые сообщения используются при декодировании за счёт комбинирования по Чейзу, что может быть рассмотрено как накопление энергии полезного сигнала
3	ГРОС с нарастиваемой избыточностью (<i>HARQ – Incremental Redundancy</i>)	используется информация, содержащаяся в неверно принятых сообщениях; требуется наличия буфера; имеет низкую скорость передачи в областях низких и средних соотношений сигнал/шум

1.2. Использование циклических кодов для формирования контрольной суммы

1.2.1. Работа кодера и декодера

Кодер хранит порождающий многочлен $g(x)$. Степень многочлена определяется следующим образом $\deg(g(x)) = r$ и определяет количество бит контрольной суммы в кодовом слове. k – число информационных символов передаваемого сообщения \bar{m} .

Передаваемое сообщение рассматривается как вектор длины k . Для каждого сообщения (\bar{m}) кодер выполняет следующие действия:

1. На основе вектора \bar{m} формируется многочлен $m(x)$. Степень многочлена $m(x)$ при этом меньше или равно $k - 1$.

2. Вычисляется многочлен $c(x) = m(x)x^r \bmod g(x)$. Степень многочлена $c(x)$ при этом меньше или равно $r - 1$.

3. Вычисляется многочлен $a(x) = m(x)x^r + c(x)$.

4. На основе многочлена $a(x)$ формируется вектор \bar{a} , длина которого n бит, где $n = k + r$.

Пример:

0. $g(x) = x^3 + x + 1 \rightarrow \deg(g(x)) = 3; \bar{m} = 1010 \rightarrow k = 4;$

1. $m(x) = x^3 + x;$

2. $c(x) = (x^3 + x)x^3 \bmod (x^3 + x + 1) = x + 1;$

3. $a(x) = (x^3 + x)x^3 + x + 1 = x^6 + x^4 + x + 1;$

4. $\bar{a} = 1010011, n = 4 + 3 = 7.$

Стоит отметить, что многочлен, полученный на третьем шаге описанного алгоритма для любого $m(x)$, делится без остатка на порождающий многочлен $g(x)$ то есть $a(x) \bmod g(x) = 0$.

В соответствии с данным свойством может быть предложен следующий алгоритм декодирования. Декодер хранит $g(x)$ (порождающий многочлен) и n (длина кодового слова). Декодер выполняет следующие действия:

1. Принятое сообщение $\bar{b} = \bar{a} + \bar{e}$ переводится в многочлен $b(x)$;

2. Вычисляется синдром: $s(x) = b(x) \bmod g(x)$;

3. Если $s(x) \neq 0$, то декодер выносит решение, что произошли ошибки ($E=1$), иначе декодер выносит решение, что ошибки не произошли ($E=0$):

$$E = \begin{cases} 1, & s(x) \neq 0 \\ 0, & s(x) = 0 \end{cases}.$$

При этом стоит отметить следующее:

$$\text{Ошибка декодирования} = \begin{cases} \bar{e} \neq 0 \\ \bar{e} \in A. \\ E = 0 \end{cases}$$

где e – вектор ошибок, A – множество разрешенных кодовых слов, E – решение принятое декодером.

Это означает следующее: если вектор ошибок принадлежит множеству разрешенных кодовых слов, то на выходе канала будет другое разрешенное кодовое слово (XOR двух кодовых слов дает кодовое слово), и, следовательно, декодер не обнаружит ошибки.

Кодом называется множество последовательностей, которое может появиться на выходе кодера, при поступлении на вход всех возможных информационных последовательностей. Список всех таких кодовых последовательностей называется множеством разрешенных кодовых слов (A), а мощность этого множества – это мощность кода.

Минимальным расстоянием кода d называется наименьшее число несовпадающих позиций между любыми кодовыми словами. Ранее описанная система кодирования и декодирования позволяет найти количество ошибок, равное $d - 1$ вне зависимости от того, где эти ошибки произошли и того, какое кодовое слово передавалось.

1.2.2. Вычисление верхней оценки ошибки декодирования

Пусть заданы: порождающий многочлен $g(x)$, длина кодируемой последовательности k , минимальное расстояние кода d , вероятность ошибки в канале p .

Тогда асимптотическая верхняя граница вероятности ошибки декодирования CRC определяется как:

$$P_{ED} = \left(\frac{1}{2}\right)^r \quad (1.1)$$

где P_{ED} – вероятность ошибки декодера CRC, r – длина проверочной части CRC ($r = \deg(g(x))$).

Теперь вычислим верхнюю оценку вероятности ошибки декодирования P_e^+ . Рассмотрим два множества векторов ошибок:

$$A = \{\bar{e} : \text{при которых произошла ошибка декодирования}\} = \{\bar{e} \neq 0, E = 0\},$$

$$B = \{\bar{e} : w(\bar{e}) \geq d\},$$

где $w(\bar{e})$ – вес вектора ошибок. Для случая двоичных векторов ошибок вес равен количеству единиц в векторе. Пример: $\bar{e} = 100101 \rightarrow w(\bar{e}) = 3$.



Рис. 1.4. Схематическое представление соотношения между множествами A и B

Графическое представление множеств событий A и B изображено на рис. 1.4.

Мощность множества B больше, чем мощность множества A , где под мощностью множества подразумевается число элементов (векторов ошибок), которое входит в множество. Предположим, что все вектора ошибок с весом $w(\bar{e}) \geq d$ приводят к ошибке декодирования. Используя данный подход, мы можем получить верхнюю границу для вероятности ошибки, для этого нужно найти $Pr\{B\}$ при этом $Pr\{B\} > Pr\{A\}$. Найдем $Pr\{B\}$:

$$\begin{aligned}
 Pr\{B\} &= Pr\{w(\bar{e}) = d \cup w(\bar{e}) = (d+1) \cup \dots \cup w(\bar{e}) = n\} = \\
 &= \sum_{i=d}^n Pr\{w(\bar{e}) = i\} = \sum_{i=d}^n C_n^i p^i (1-p)^{(n-i)}.
 \end{aligned}$$

Тогда верхнюю границу ошибки декодирования можно определить, как вероятность того, что вектор ошибки принадлежит множеству B :

$$P_e^+ = Pr\{B\} = \sum_{i=d}^n C_n^i p^i (1-p)^{(n-i)}. \quad (1.2)$$

Так как

$$\sum_{i=0}^n C_n^i p^i (1-p)^{(n-i)} = 1.$$

Так как d обычно много меньше чем n , то для уменьшения сложности расчета выражение (1) можно переписать следующим образом:

$$P_e^+ = 1 - \sum_{i=0}^{d-1} C_n^i p^i (1-p)^{(n-i)} = f(n, d, p).$$

1.2.3. Вычисление точного значения ошибки декодирования

Пусть заданы: порождающий многочлен $g(x)$, длина кодируемой последовательности k , минимальное расстояние кода d , вероятность ошибки в канале p . Необходимо найти точное значение вероятности ошибки декодирования P_e . Для решения этой задачи рассмотрим каким образом с помощью кодера множество сообщений может быть отображено на множество кодовых слов. Схематично этот процесс представлен на рис. 1.5.

Пусть A – множество кодовых слов, $|A| = 2^k$; B – множество векторов ошибок, $|B| = 2^n$.

Обозначим через A_i число кодовых слов веса i , где i – индекс от 0 до n . На рис. 1.6 показано какие из значения могут принимать A_i для кода с минимальным расстоянием d .

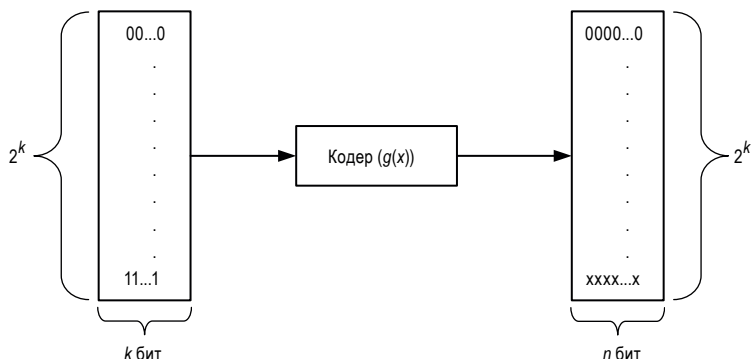


Рис. 1.5. Схематичное представление формирования множества кодовых слов

$$\begin{array}{l}
 A_0=1 \\
 A_1=0 \\
 \dots \\
 A_{d-1}=0 \\
 A_d>0 \\
 \left. \begin{array}{l} A_{d+1} \\ \dots \\ A_n \end{array} \right\} =?
 \end{array}$$

Рис. 1.6 Допустимые значения A_i для кода с минимальным расстоянием d

Ошибка декодирования происходит, если вектор ошибок является кодовым словом (доказательство приводится в курсе лекций). Все слова в коде имеют вес больше либо равный d . Поэтому гарантированно обнаруживаются ошибки число которых $< d$. Для точного определения значения вероятности ошибки декодирования следует посчитать вероятность попадания вектора ошибок в множество A , то есть $P_e = Pr\{A\} = Pr\{\bar{e} \in A : \bar{e} \neq 0\}$. Теперь мы можем записать следующее выражение для вычисления точного значения вероятности ошибки декодирования:

$$P_e = \sum_{i=d}^n A_i p^i (1-p)^{(n-i)}, \quad (1.3)$$

где P_e – вероятность ошибки декодирования, d – минимальное расстояние кода, n – длина кодового слова, p – вероятность ошибки на бит, A_i – количество кодовых слов с весом i (w (кодового слова) = i).

1.3. Коэффициент использования канала

В рамках лабораторной работы путем имитационного моделирования необходимо будет определить коэффициент использования канала для заданных параметров системы. Под коэффициентом использования канала подразумевается характеристика системы, вычисляемая по следующей формуле:

$$\eta = \frac{mN}{nN_t}, \quad (1.4)$$

где m – длина информационных бит при передаче одного сообщения (не учитывая биты CRC); n – количество бит, переданных в канал (информационные биты, биты CRC и биты помехоустойчивого кода); N – количество успешно переданных сообщений; N_t – общее количество совершенных передач сообщений (с учетом неуспешных отправок).

1.4. Использование средств MatLab для моделирования работы кодера и декодера сверточных кодов

1.4.1. Сверточное кодирование с постоянной скоростью кода

Для кодирования и декодирования с помощью сверточных кодов в среде Matlab необходимо начать с получения матрицы переходов решетки Витерби [6]. Для этого используется встроенная функция:

`poly2trellis(ConstraintLength,CodeGenerator),`

где `ConstraintLength` – вектор строка длинны M , количество столбцов в которой означает, сколько входных бит используется для формирования выхода, значения в столбцах означают максимальное количество бит обратной связи (памяти буфера), используемых для бита на текущей позиции. `CodeGenerator` – матрица размерностью $M \times N$, которая содержит в себе информацию об обратных связях (значения в восьмеричной системе счисления).

Данная функция возвращает описание структуры решетки, соответствующее преобразованию для кодера со скоростью K/N , K – это количество входных битовых потоков в кодер, а N – количество выходных соединений.

Для кодирования применяется следующая встроенная функция:

`convenc(msg,trellis),`

где `msg` – кодируемое сообщение в двоичном виде, `trellis` – это описание структуры решетки. Данная функция возвращает вектор длины N , которая определяется по следующей формуле:

$$N = L/R, \quad (1.5)$$

где L – длина `msg`, R – скорость сверточного кода.

Для декодирования применяется следующая встроенная функция:

`vitdec(code, trellis, tble, opmode, dectype),`

где `code` – вектор входных данных в соответствии с параметром `dectype`, `dectype` – принимает 3 значения: 1) ‘hard’ – в этом случае параметр `code` представляет из себя вектор из 0 и 1 (жесткое декодирование); 2) ‘soft’ – `code` должен содержать целые числа в диапазоне от 0 до $2nsdec - 1$, при этом значение 0 представляет бит, с наибольшей достоверностью равный нулю, а значение $2nsdec - 1$ означает бит, с наибольшей достоверностью равный единице, так же нужно передавать данные о количестве квантов (мягкое декодирование); 3) ‘unquant’ – Вектор `code` содержит вещественные числа, при этом 1 соответствует логическому нулю, а -1 представляет логическую единицу (жесткое декодирование). Параметр `trellis` – результат работы функции `poly2trellis`. `Opmode` – может принимать 3 значения, но в данной лабораторной работе будет использоваться только значение ‘trunc’. `Tble` – положительное целое число, задающее глубину просмотра решетки при декодировании (создает задержку вывода решений), в лабораторной работе используется значение параметра равное 1.

1.4.2. Сверточное кодирование с переменной скоростью кода

Семейство сверточных кодов с переменной скоростью кода можно получить, убирая определенные полиномы обратной связи и выкалывая из кода соответствующие им позиции выходных бит. Приведем пример семейства сверточных кодов со скоростями 2/5, 2/4 и 2/3.

Основным кодом (с медленной скоростью) является код 2/5, получить решетку такого кода можно с помощью функции:

$$\text{Trell2_5} = \text{poly2trellis}([5 \ 4], [23 \ 35 \ 00 \ 32 \ 07; 00 \ 05 \ 13 \ 07 \ 13]).$$

Чтобы получить решетку со скоростью 2/4 из того же семейства что и предыдущая решетка, то из предыдущей матрицы обратных связей нужно удалить один из столбцов (в нашем случае последний):

$$\text{Trell2_4} = \text{poly2trellis}([5 \ 4], [23 \ 35 \ 00 \ 32; 00 \ 05 \ 13 \ 07]).$$

Что бы получить код со скоростью 2/3 удалим из предыдущей матрицы еще один столбец:

$$\text{Trell2_3} = \text{poly2trellis}([5 \ 4], [23 \ 35 \ 00; 00 \ 05 \ 13]).$$

Далее кодируем сообщение с помощью решетки с наименьшей скоростью кода (2/5). Закодируем сообщение из 4 бит с использование решетки Trell2_5 .

$$\text{Получим code} = [a1 \ a2 \ a3 \ a4 \ a5 \ b1 \ b2 \ b3 \ b4 \ b5].$$

Затем сформируем первый пакет для передачи со скоростью кода (2/3). Имея сообщение code нам нужно удалить биты, соответствующие полиномам связи которых нет в этом коде, тогда первый пакет будет:

$$P1 = [a1 \ a2 \ a3 \ b1 \ b2 \ b3]. \text{ (6 бит)}$$

Передаём пакет по каналу. Приёмник декодирует принятый пакет при помощи Trell2_3 . После выполняется проверка CRC, если произошла ошибка, то передатчик передает недостающие биты с меньшей скоростью (2/4). Тогда второй пакет выглядит как:

$$P2 = [a4 \ b4]. \text{ (2 бита)}$$

Далее выполняется передача пакета через канал. Приемник формирует из первого и второго пакета последовательность, соответствующую коду 2/4:

$$NP = [a1' \ a2' \ a3' \ a4' \ b1' \ b2' \ b3' \ b4'].$$

Далее приёмник декодирует его при помощи решетки Trell2_4 и проверяет CRC. Если произошла ошибка, то передатчик посылает

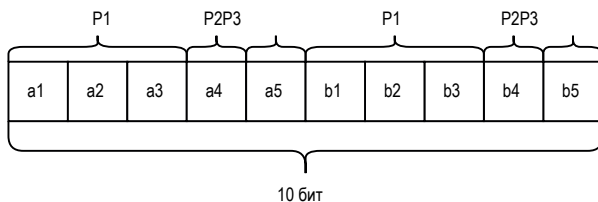


Рис. 1.7 Разбиение закодированной последовательности на пакеты

ет недостающие биты со скоростью $(2/5)$. Третий пакет выглядит как:

$$P3 = [a5 \ b5]. \text{ (2 бита)}$$

Пакеты посылаются через канал. Приёмник формирует последовательность, принятую на предыдущем шаге с принятым пакетом. Далее приёмник декодирует его при помощи решетки Trell2_5 и проверяет CRC. Если после этой передачи не удалось принять сообщение, то процесс повторяется заново. Такой подход позволяет плавно подстраиваться под качество канала, передавая с максимальной для него скоростью кода.

1.4.3. Примеры применения сверточного кодирования в среде MatLab

Пример 1

Закодируем сообщение с помощью сверточного кода и скоростью $R=1/2$. Используем декодирование с жестким решением.

Схема сверточного кода будет иметь следующий вид (рис. 1.8).

Закодируем сообщение длиной 8 бит.

Модуляция сообщения сделана согласно правилу: $\begin{cases} 0 \rightarrow 1 \\ 1 \rightarrow -1 \end{cases}$.

Листинг кода и результат выполнения представлен на рис. 1.9 и 1.10.

Пример 2

Закодируем сообщение с помощью сверточного кода и скоростью $R=1/2$. Используем декодирование с мягким решением. Схема сверточного кода будет аналогична примеру 1.

Модуляция сообщения сделана согласно правилу: $\begin{cases} 0 \rightarrow 1 \\ 1 \rightarrow -1 \end{cases}$.

Листинг кода и результат выполнения представлен на рис. 1.11 и 1.12.

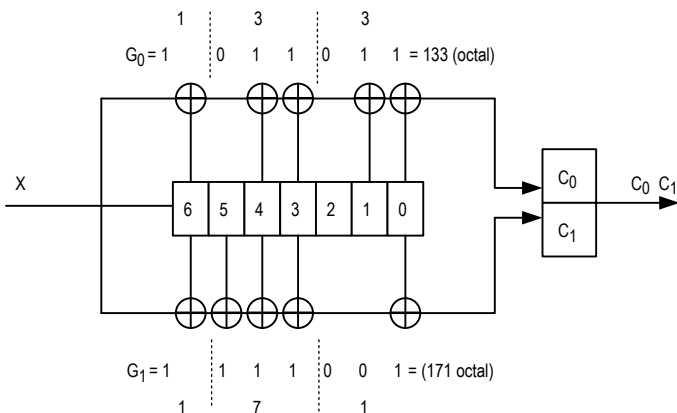


Рис. 1.8. Схема сверточного кода со скоростью $R=1/2$

```

13  %получение описание структуры решетки
14  trellis = poly2trellis(7,[133 171]);
15  % формирование сообщения
16  message = de2bi(125,8);
17  fprintf("message:");
18  fprintf("%d ",message);
19  fprintf("\ncodeword:");
20  % кодирование сообщения
21  codeword = convenc(message,trellis);
22  fprintf("%d ",codeword);
23  fprintf("\nmodulword:");
24  % модулирование сообщения
25  modulword = codeword.*(-2)+1;
26  fprintf("%d ",modulword);
27  fprintf("\nmodulword_rec:");
28  % передача по каналу
29  modulword_rec = modulword + sigma*randn(1,length(modulword));
30  fprintf("%0.2f ",modulword_rec(1:7));
31  fprintf("\n");
32  fprintf("%0.2f ",modulword_rec(8:end));
33  fprintf("\ndemodulword:");
34  % демодулирование
35  demodulword = modulword_rec<0;
36  fprintf("%d ",demodulword);
37  fprintf("\ne:");
38  e = xor(codeword,demodulword);
39  fprintf("%d ",e);
40  fprintf("\ncodeword_rec:");
41  % декодирование сообщения
42  codeword_rec = vitdec(demodulword,trellis, 1, 'trunc','hard');
43  fprintf("%d ",codeword_rec);
44  fprintf("\n");

```

Рис. 1.9. Листинг кода кодирования и декодирования (жесткое решение) с использованием сверточного кода и скоростью $R=1/2$

```

message:1 0 1 1 1 1 1 0
codeword:1 1 0 1 0 0 0 1 0 1 0 0 0 1 1 1
modulword:-1 -1 1 -1 1 1 1 -1 1 -1 1 1 1 -1 -1 -1
modulword_rec:0.61 -2.79 2.77 -0.73 1.79 -0.32 0.53
-1.22 1.34 -2.33 1.37 0.04 1.05 -0.48 -0.74 -0.14
demodulword:0 1 0 1 0 1 0 1 0 1 0 0 0 1 1 1
e:1 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0
codeword_rec:1 0 1 1 1 1 1 0

```

Рис. 1.10. Результат кодирования и декодирования (жесткое решение) с использованием сверточного кода и скоростью $R=1/2$

```

13 %получение описание структуры решетки
14 trellis = poly2trellis(7,[133 171]);
15 % формирование сообщения
16 message = de2bi(125,8);
17 fprintf("message:");
18 fprintf("%d ",message);
19 fprintf("\ncodeword:");
20 % кодирование сообщения
21 codeword = convenc(message,trellis);
22 fprintf("%d ",codeword);
23 fprintf("\nmodulword:");
24 % модулирование сообщения
25 modulword = codeword.*(-2)+1;
26 fprintf("%d ",modulword);
27 fprintf("\nmodulword_rec:");
28 % передача по каналу
29 modulword_rec = modulword + sigma*randn(1,length(modulword));
30 fprintf("%0.2f ",modulword_rec(1:7));
31 fprintf("\n");
32 fprintf("%0.2f ",modulword_rec(8:end));
33 fprintf("\ncodeword_rec:");
34 % декодирование сообщения
35 codeword_rec = vitdec(modulword_rec,trellis, 1, 'trunc','unquant');
36 fprintf("%d ",codeword_rec);
37 fprintf("\n");

```

Рис. 1.11. Листинг кода кодирования и декодирования (мягкое решение) с использованием сверточного кода и скоростью $R=1/2$

```

message:1 0 1 1 1 1 1 0
codeword:1 1 0 1 0 0 0 1 0 1 0 0 0 1 1 1
modulword:-1 -1 1 -1 1 1 1 -1 1 -1 1 1 1 -1 -1 -1
modulword_rec:-0.20 -1.52 1.20 -1.75 -0.05 1.73 1.00
-1.04 1.72 -0.53 1.28 1.99 1.74 -0.81 -1.55 -1.52
codeword_rec:1 0 1 1 1 1 1 0

```

Рис. 1.12. Результат кодирования и декодирования (мягкое решение) с использованием сверточного кода и скоростью $R=1/2$

Пример 3

Закодируем сообщение с помощью сверточного кода и переменной скоростью.

Модуляция сообщения сделана согласно правилу: $\begin{cases} 0 \rightarrow 1 \\ 1 \rightarrow -1 \end{cases}$.

На рис. 1.14 код в строках (49-60) выполняется только в том случае, если декодирование в строке 47 было неуспешным. Аналогично, код в строках (62-73) выполняется только в том случае, если декодирование в 59 строке было неуспешным.

Листинг кода и результат выполнения представлен на рис. 1.13, 1.14 и 1.15.

```
13      % формирование сообщения
14      message = de2bi(125,8);
15      fprintf("message:");
16      fprintf("%d ",message);
17      %получение описание структуры решетки скорости кода 2/5.
18      trellis2_5 = poly2trellis([5 4], [23 35 00 32 07; 00 05 13 07 13]);
19      %получение описание структуры решетки скорости кода 2/4.
20      trellis2_4 = poly2trellis([5 4], [23 35 00 32; 00 05 13 07]);
21      %получение описание структуры решетки скорости кода 2/3.
22      trellis2_3 = poly2trellis([5 4], [23 35 00; 00 05 13]);
23      % кодирование сообщения.
24      fprintf("\ncodeword:");
25      codeword = convenc(message,trellis2_5);
26      fprintf("%d ",codeword);
27      fprintf("\npl:");
28      %2/3 %12 bit
29      p1 = [codeword(1:6) codeword(11:16)];
30      fprintf("%d ",p1);
31      fprintf("\nnp2:");
32      %2/4 %4 bit
33      p2 = [codeword(7:8) codeword(17:18)];
34      fprintf("%d ",p2);
35      fprintf("\nnp3:");
36      %2/5 %4 bit
37      p3 = [codeword(9:10) codeword(19:20)];
38      fprintf("%d ",p3);
39      %модуляция 1 пакета
40      modul2_3 = p1.*(-2)+1;
41      fprintf("\nmodul_rec2_3:");
42      % передача по каналу 1 пакета
43      modul_rec2_3 = modul2_3 + sigma*randn(1,length(modul2_3));
44      fprintf("%0.2f ",modul_rec2_3);
45      fprintf("\ncodeoutput2_3:");
46      % декодирование кода со скоростью 2/3.
47      codeoutput2_3 = vitdec(modul_rec2_3,trellis2_3, 1, 'trunc','unquant');
48      fprintf("%d ",codeoutput2_3);
```

Рис. 1.13. Листинг кода кодирования и декодирования с использованием сверточного кода и переменной скоростью

```

49 %если при декодировании сообщения со скоростью 2\3 произошла ошибка,
50 %посылаем недостающие биты со скоростью 2\4.
51 %модуляция 2 пакета и передача его по каналу.
52 - modul2_4 = (p2.*(-2)+1) + sigma*randn(1,length(p2));
53 - fprintf("\nmodul_rec2_4:");
54 %формирование кода со скоростью 2/4
55 - modul_rec2_4 = [modul_rec2_3(1:(end/2)) modul2_4(1:(end/2)) modul_rec2_3((end/2)+1:end) modul2_4((end/2)+1:end)];
56 - fprintf("%0.2f ",modul_rec2_4);
57 - fprintf("\ncodeoutput2_4:");
58 % декодирование кода со скоростью 2/4.
59 - codeoutput2_4 = vitdec(modul_rec2_4,trellis2_4, 1, 'trunc','unquant');
60 - fprintf("%d ",codeoutput2_4);
61
62 %если при декодировании сообщения со скоростью 2\4 произошла ошибка,
63 %посылаем недостающие биты со скоростью 2\5.
64 %модуляция 3 пакета и передача его по каналу.
65 - modul2_5 = (p3.*(-2)+1) + sigma*randn(1,length(p3));
66 - fprintf("\nmodul_rec2_5:");
67 %формирование кода со скоростью 2/5.
68 - modul_rec2_5 = [modul_rec2_4(1:(end/2)) modul2_5(1:(end/2)) modul_rec2_4((end/2)+1:end) modul2_5((end/2)+1:end)];
69 - fprintf("%0.2f ",modul_rec2_5);
70 - fprintf("\ncodeoutput2_5:");
71 % декодирование кода со скоростью 2/5.
72 - codeoutput2_5 = vitdec(modul_rec2_5,trellis2_5, 1, 'trunc','unquant');
73 - fprintf("%d ",codeoutput2_5);

```

Рис. 1.14. Листинг кода кодирования и декодирования с использованием сверточного кода и переменной скоростью

```

message:1 0 1 1 1 1 1 0
codeword:1 1 0 1 0 1 0 1 0 1 1 0 1 1 0 0 0 1 1 1
p1:1 1 0 1 0 1 1 0 1 1 0 0
p2:0 1 0 1
p3:0 1 1 1
modul_rec2_3:-1.92 0.67 1.84 -0.30 0.22 -0.72
-1.50 0.72 -1.51 -1.91 0.19 0.81
codeoutput2_3:1 0 1 1 1 1 0 1
modul_rec2_4:-1.92 0.67 1.84 -0.30 0.22 -0.72 -0.51 -0.46
-1.50 0.72 -1.51 -1.91 0.19 0.81 0.90 -0.38
codeoutput2_4:1 0 0 0 1 1 0 0
modul_rec2_5:-1.92 0.67 1.84 -0.30 0.22 -0.72 -0.51 -0.46 1.24 -0.56
-1.50 0.72 -1.51 -1.91 0.19 0.81 0.90 -0.38 -2.32 -1.91
codeoutput2_5:1 0 1 1 1 1 1 0

```

Рис. 1.15. Результат кодирования и декодирования с использованием сверточного кода и переменной скоростью

2. ПЕРВЫЙ ДОПУСК К ВЫПОЛНЕНИЮ ЛАБОРАТОРНЫХ РАБОТ. СИСТЕМА ПЕРЕДАЧИ ДАННЫХ С КОНТРОЛЬНОЙ СУММОЙ

Цель допуска: Изучение способов моделирования системы передачи данных по каналу с АБГШ с использованием CRC.

2.1. Описание моделируемой системы

В допуске нужно реализовать систему передачи данных по каналу с АБГШ, с использованием CRC (англ. *Cyclic redundancy check*, Циклический избыточный код), для проверки целостности данных.

На рис. 2.1 приведена схема реализуемой системы.

Источник формирует данные m – битовую последовательность длины l . В блоке CRC- r к данным добавляется избыточная проверочная часть длины r . Полученное кодовое слово модулируется при помощи BPSK, ноль после модуляции преобразуется в +1, единица в -1. Модулированное кодовое слово передается по каналу с АБГШ. Демодулятор сравнивает полученные из канала данные с порогом и ставит в соответствие 0 и 1. Полученные данные проверяются на наличие ошибок, вычисляя синдром s . Возможны три варианта: ошибка произошла, и она обнаружена ($s \neq 0$); ошибка произошла, но не была обнаружена ($s = 0, m \neq m'$); ошибок не было ($s = 0, m = m'$).

2.2. Порядок выполнения допуска

1. Необходимо реализовать систему передачи данных по каналу с АБГШ с использованием CRC (рис. 2.1), вариант CRC многочлена выдается преподавателем.

2. Провести моделирование передачи N пакетов данных по реализованной системе при различных SNR.

3. В ходе имитационного моделирования построить графики следующих значений от отношения сигнал/шум в децибелах:

а) оценка вероятности ошибки на бит (отношение числа ошибочных бит к общему числу бит),

б) оценка вероятности ошибки декодирования CRC (такое событие, что $s = 0, m \neq m'$).

4. Построить теоретические графики следующих значений от отношения сигнал/шум в децибелах:

а) вероятность ошибки на бит для BPSK по формуле (П1.1) (см. прил. 1);

б) вероятность ошибки декодирования CRC (см. разд. 1.2).

5. Продемонстрировать работу моделирующей программы и построенные графики преподавателю.

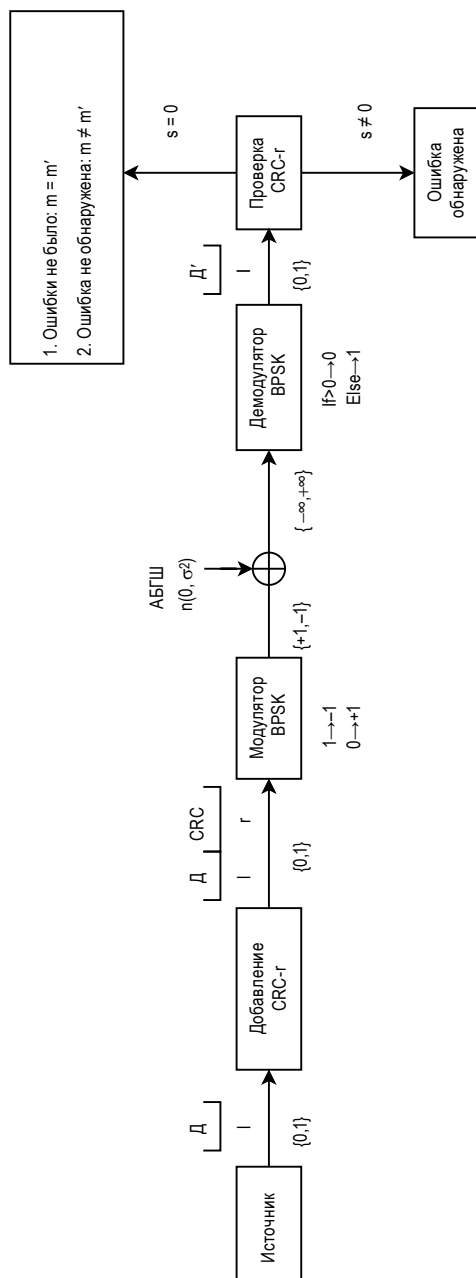


Рис. 2.1. Схема передачи данных с проверкой CRC

2.3. Варианты многочленов CRC для допуска

Таблица 2.1

Варианты многочленов CRC

№	Полином CRC	r	m	k	d
1	$x^3 + x + 1$	3	4	7	3
2	$x^3 + x^2 + 1$	3	4	7	3
3	$x^4 + x^3 + x^2 + 1$	4	3	7	4
4	$x^4 + x^2 + x + 1$	5	3	7	4
5	$x^8 + x^7 + x^6 + x^4 + x^2 + 1$	8	4	12	—
6	$x^{16} + x^{13} + x^{12} + x^{11} + x^{10} + x^8 + x^6 + x^5 + x^2 + 1$	16	8	24	—

3. ВТОРОЙ ДОПУСК К ВЫПОЛНЕНИЮ ЛАБОРАТОРНЫХ РАБОТ. СИСТЕМА ПЕРЕДАЧИ ДАННЫХ С ОБРАТНОЙ СВЯЗЬЮ

Цель допуска: изучение способов моделирования с обратной связью, помехоустойчивым кодированием для исправления ошибок и CRC для обнаружения ошибок.

3.1. Описание моделируемой системы

Реализовать систему передачи данных по каналу с АБГП. Для кодирования данных использовать код Хэмминга (n , m , d), n – длина кодового слова, m – длина информационных данных, d – расстояние кода (для кодов Хэмминга является постоянным и равно 3). Длина информационных данных 8 бит. Для обнаружения ошибок используется CRC-16 с порождающим многочленом $x^{16} + x^{13} + x^{12} + x^{11} + x^{10} + x^8 + x^6 + x^5 + x^2 + 1$.

3.2. Порядок выполнения допуска

1. Написать и отладить моделирующую программу в соответствии с полученным от преподавателя вариантом.

2. Провести моделирование передачи N пакетов данных по реализованной системе при различных SNR.

3. Построить графики следующих значений от отношения сигнал/шум в децибелах:

а) вероятность ошибки на бит, рассчитанной с помощью имитационного моделирования в точках А и А' по следующему принципу: число ошибочных бит разделить на общее число переданных бит;

б) вероятность ошибки на бит, рассчитанной с помощью имитационного моделирования в точках В и В': число ошибочных бит разделить на общее число переданных бит;

в) вероятность ошибки на бит, рассчитанной по формуле (П1.1) (см. прил. 1);

г) вероятность ошибки декодирования, рассчитанной с помощью имитационного моделирования;

д) вероятность ошибки декодирования, рассчитанной по формуле (1.3) (см. разд. 1.2), где вероятность ошибки на бит рассчитана с помощью имитационного моделирования в точках А и А';

е) вероятность ошибки декодирования, рассчитанной по формуле (1.3) (см. разд. 1.2), где вероятность ошибки на бит рассчитана по (П1.1) (см. прил. 1);

ж) значение средней скорости передачи, рассчитанной по формуле (1.4) (см. разд.1.3).

4. Продемонстрировать работу моделирующей программы и построенные графики преподавателю.

3.3. Варианты заданий

3.3.1. Описание вариантов

Рассматриваются три варианта заданий: система передачи с кодом Хэмминга (31, 26, 3), система передачи с кодом Хэмминга (15, 11, 3), система передачи с кодом Хэмминга (7, 4, 3). В последующих пунктах будет подробно рассмотрен каждый из вариантов.

3.3.2. Моделирование системы передачи с кодом Хэмминга (31, 26, 3)

На рис. 3.1 изображена схема передачи для первого варианта. В соответствии с данной схемой для передачи каждого сообщения при текущем SNR выполняются следующие действия:

- 1) сгенерировать сообщение длиной 8 бит;
- 2) добавить контрольную сумму к сообщению, тогда длина сообщения с контрольной суммой будет представлять последовательность длиной 24 бита;
- 3) в соответствии с кодом Хэмминга длина информационной части равна 26 бит, тогда к выбранному сообщению добавляется 2 нулевых бита для выравнивания длины. Получается 1 пакет длиной 26 бит;
- 4) далее пакет кодируется помехоустойчивым кодом, и его длина становится равна 31 бит. Из пакета удаляется 2 добавленных нулевых бита. Пакет длиной 29 бит модулируется в соответствии с правилом $0 \rightarrow 1$; $1 \rightarrow -1$ и передается в канал, где на него накладывается шум с параметрами $(0; \sigma^2)$;
- 5) на приемной стороне каждый пакет демодулируется, добавляется 2 нулевых бита, далее пакет декодируется по Хэммингу, и у пакета отбрасываются добавленные нулевые биты, после чего проверяется контрольная сумма;
- 6) если контрольная сумма совпала, то сообщение доставляется получателю. Также проверяется, совпало ли доставленное сообщение с отправленным, если нет, то прибавляется счетчик ошибок декодирования. Если контрольная сумма не совпала, то такое сообщение отправляется заново, т. е. переходим к шагу 5. Отметим, что при отправке каждого сообщения прибавляется счетчик отправленных сообщений для текущего SNR.

Для ускорения работы моделирующей программы можно заранее подготовить кодовую книгу, состоящую из всех сообщений

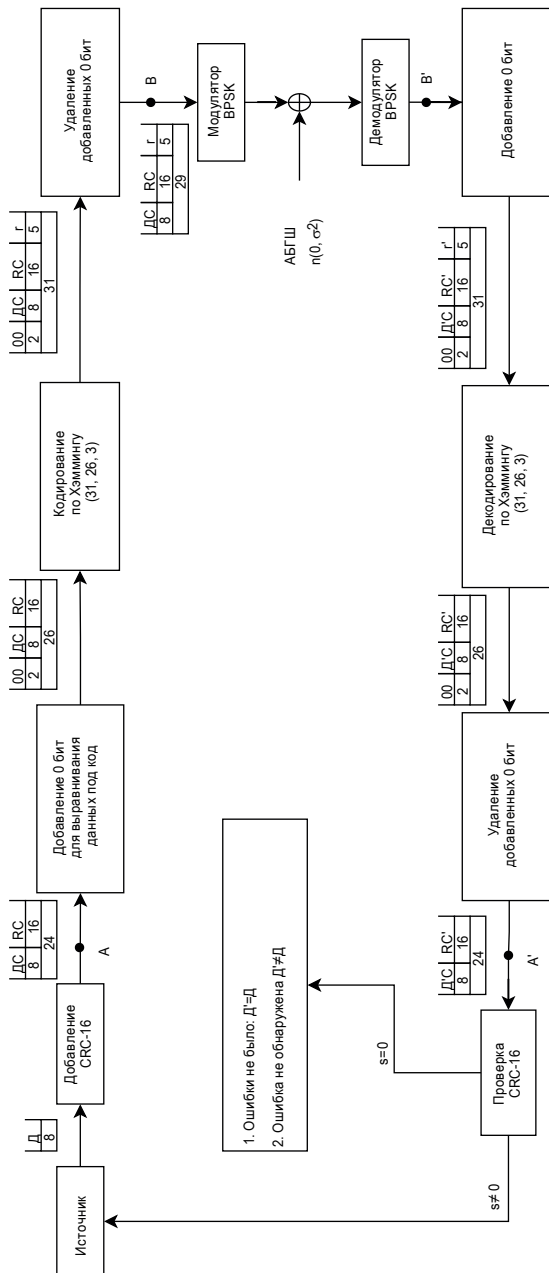


Рис. 3.1. Схема передачи данных для первого вариант

с добавленными проверочными битами. Тогда выполнение 1 и 2 шага можно заменить на случайный выбор из кодовой книги.

3.3.3. Система передачи с кодом Хэмминга (15, 11, 3)

На рис. 3.2 изображена схема передачи для второго варианта. В соответствии с данной схемой для передачи каждого сообщения при текущем SNR выполняются следующие действия:

- 1) сгенерировать сообщение длиной 8 бит;
- 2) добавить контрольную сумму к сообщению, тогда длина сообщения с контрольной суммой будет представлять последовательность длиной 24 бита;
- 3) в соответствии с кодом Хэмминга длина информационной части равна 11 бит, тогда выбранное сообщение разбивается на 3 пакета по 8 бит, и к каждому пакету добавляется по 3 нулевых бита для выравнивания длины. Получается 3 пакета по 11 бит;
- 4) далее каждый пакет кодируется помехоустойчивым кодом, и его длина становится равна 15 бит. Из каждого пакета удаляется по 3 добавленных нулевых бита. Каждый пакет длиной 12 бит модулируется в соответствии с правилом $0 \rightarrow 1$; $1 \rightarrow -1$ и передается в канал, где на него накладывается шум с параметрами $(0; \sigma^2)$;
- 5) на приемной стороне каждый пакет демодулируется, добавляется по 3 нулевых бита, далее пакет декодируется по Хэммингу, и у каждого пакета отбрасываются добавленные нулевые биты, после чего пакеты объединяются в один и проверяется контрольная сумма;
- 6) если контрольная сумма совпала, то сообщение доставляется получателю. Также проверяется, совпало ли доставленное сообщение с отправленным, если нет, то прибавляется счетчик ошибок декодирования. Если контрольная сумма не совпала, то такое сообщение отправляется заново, т. е. переходим к шагу 5. Отметим, что при отправке каждого сообщения прибавляется счетчик отправленных сообщений для текущего SNR.

Для ускорения работы моделирующей программы можно заранее подготовить кодовую книгу, состоящую из всех сообщений с добавленными проверочными битами. Тогда выполнение 1 и 2 шага можно заменить на случайный выбор из кодовой книги.

3.3.4. Система передачи с кодом Хэмминга (7, 4, 3)

На рис. 3.3 изображена схема передачи для третьего варианта. В соответствии с данной схемой для передачи каждого сообщения при текущем SNR выполняются следующие действия:

- 1) сгенерировать сообщение длиной 8 бит;

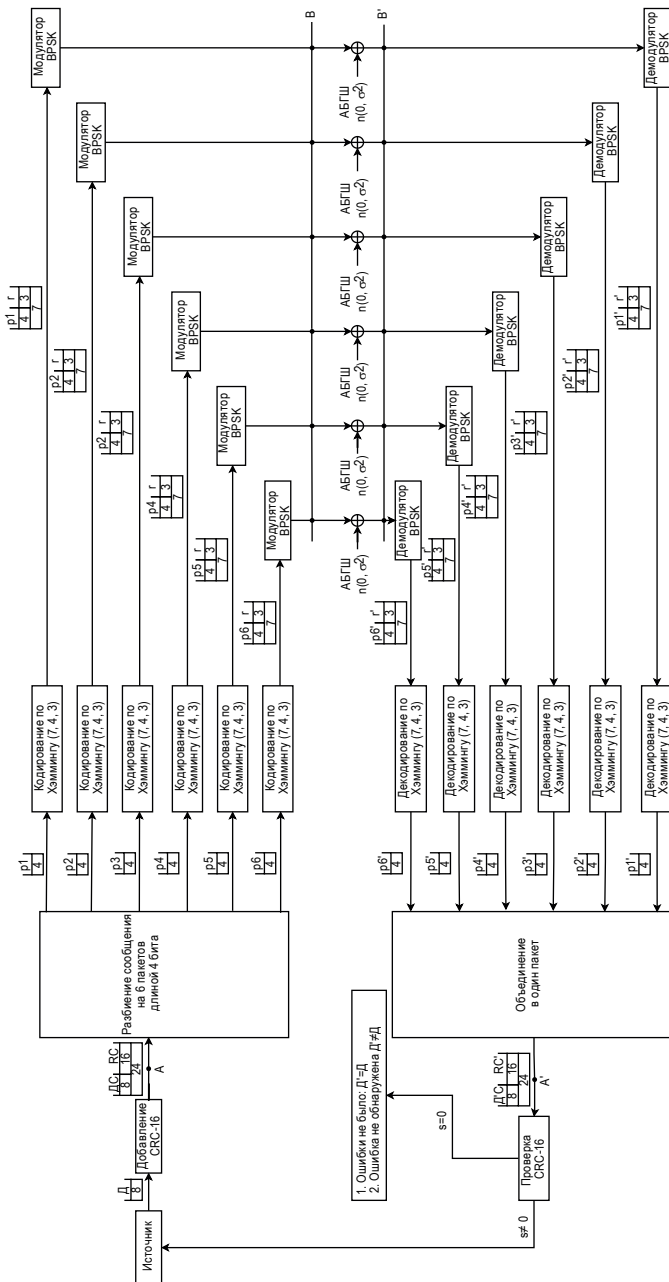


Рис. 3.3. Схема передачи данных для третьего варианта

2) добавить контрольную сумму к сообщению, тогда длина сообщения с контрольной суммой будет представлять последовательность длиной 24 бита;

3) в соответствии с кодом Хэмминга длина информационной части равна 4 бита, тогда выбранное сообщение разбивается на 6 пакетов по 4 бита;

4) далее каждый пакет кодируется помехоустойчивым кодом, и его длина становится равна 7 бит. Каждый пакет модулируется в соответствии с правилом $0 \rightarrow 1$; $1 \rightarrow -1$ и передается в канал, где на него накладывается шум с параметрами $(0; \sigma^2)$;

5) на приемной стороне каждый пакет длиной 7 бит демодулируется, пакеты декодируются по Хэммингу, после чего их длина становится равна 4 бита, далее пакеты объединяются в один и проверяется контрольная сумма;

6) если контрольная сумма совпала, то сообщение доставляется получателю. Также проверяется, совпало ли доставленное сообщение с отправленным, если нет, то прибавляется счетчик ошибок декодирования. Если контрольная сумма не совпала, то такое сообщение отправляется заново, т. е. переходим к шагу 5. Отметим, что при отправке каждого сообщения прибавляется счетчик отправленных сообщений для текущего SNR.

Для ускорения работы моделирующей программы можно заранее подготовить кодовую книгу, состоящую из всех сообщений с добавленными проверочными битами. Тогда выполнение 1 и 2 шага можно заменить на случайный выбор из кодовой книги.

4. ЛАБОРАТОРНАЯ РАБОТА №1 «ПЕРЕДАЧА ДАННЫХ В СИСТЕМАХ С ОБРАТНОЙ СВЯЗЬЮ HARQ TYPE 1 (CHASE COMBINING)»

Цель работы: исследование кодирования данных с помощью сверточного кода и последующего декодирования данных после передачи по каналу с АБГШ и замираниями.

4.1. Описание задания

Реализовать систему передачи данных по каналу с АБГШ и замираниями, используя CRC-16 для проверки целостности данных с порождающим многочленом $x^{16} + x^{13} + x^{12} + x^{11} + x^{10} + x^8 + x^6 + x^5 + x^2 + 1$. Для кодирования данных использовать сверточный код и полиномы обратной связи. Длина информационных данных 32 бита. Требуется модернизировать систему, полученную во втором допуске, заменить кодирование по Хэммингу на сверточный код.

4.2. Моделирование системы передачи с применением гибридной обратной связи

Для моделирования системы с гибридной обратной связью, описанной в разд. 1.1.3, будет описана следующая схема системы (см. рис. 4.1). В соответствии с данной схемой для передачи каждого сообщения при текущем SNR выполняются следующие действия:

- 1) сгенерировать сообщение длиной 32 бита;
- 2) к сообщению добавляются проверочные биты CRC-16. Длина сообщения с контрольной суммой представляет собой последовательность 48 бит;
- 3) сообщение с контрольной суммой кодируется сверточным кодом с заданной скоростью кода. Например, если скорость $1/3$, то длина закодированного сообщения $48 \cdot 3 / 1 = 144$;
- 4) сообщение модулируется в соответствии с правилом: $0 \rightarrow 1$; $1 \rightarrow -1$;
- 5) модулированный пакет передается в канал с замираниями и АБГШ. Замирания вычисляются следующим образом

$$H = \sqrt{randn^2 + randn^2},$$

где $randn$ – случайная величина, распределенная по нормальному закону с параметрами (0;1). Сигнал на выходе канала имеет вид $r = s \cdot H + n$, где r – выход канала; s – модулированный сигнал; H – ко-

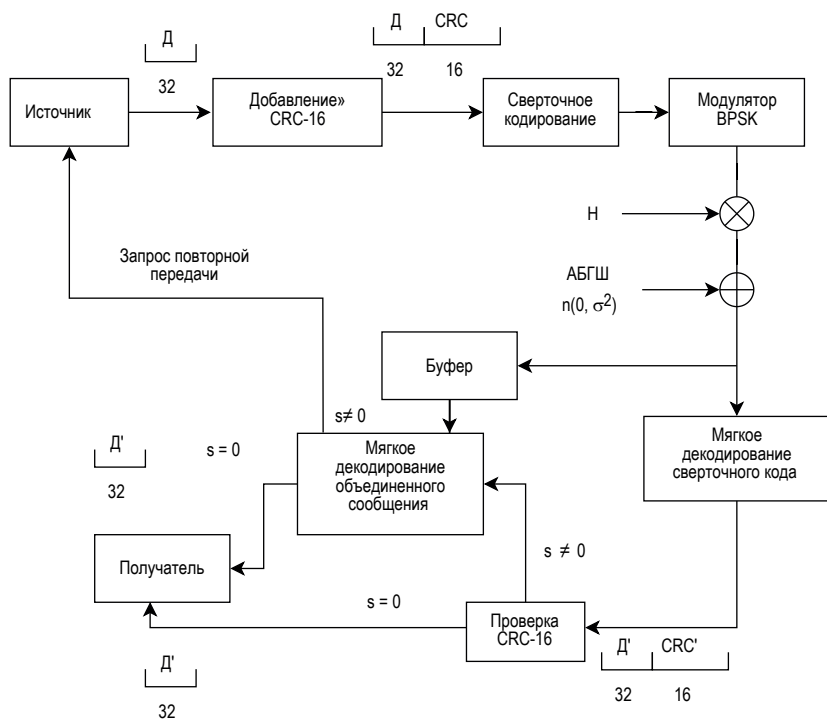


Рис. 4.1. Схема моделирования HARQ Type I

эффицент релейского замирания в канале; n – величина, распределенная по нормальному закону с параметрами $(0; \sigma^2)$.

1) полученное на выходе канала сообщение подается на декодер сверточного кода с мягкими решениями;

2) проверка CRC, если синдром равен 0, то передается следующее сообщение, иначе принятое сообщение аккумулируется в буфер, и производится повторная попытка декодирования данных из буфера и проверка CRC, в случае если синдром не равен 0, то текущее сообщение передается заново.

Для ускорения моделирования можно использовать нулевую информационную последовательность. Тогда шаги 1-3 можно пропустить, поскольку сообщение, закодированное сверточным кодом с скоростью R , будет представлять собой нулевую последовательность длиной $48 \cdot 1/R$.

Для примера опишем работу временной диаграммы на рис. 1.2. При первой передаче первого сообщения ошибок не происходит, $s = 0$,

сообщение успешно декодируется. При первой передаче второго сообщения произошли ошибки, $s \neq 0$, требуется повторная передача, данное сообщение сохраняется в буфер. При второй передаче второго сообщения снова происходит ошибка, $s \neq 0$. Теперь, так как в буфере уже есть сообщение, сообщения из буфера и из канала суммируются. Полученное таким образом сообщение проходит мягкое декодирование, $s = 0$, сообщение от второго абонента успешно принято.

4.3. Порядок выполнения лабораторной работы

1. Написать и отладить моделирующую программу в соответствии с полученным от преподавателя вариантом.

2. Построить графики следующих значений от отношения сигнал/шум в децибелах:

а) вероятность ошибки на бит при передаче данных со сверточным кодированием, рассчитанной с помощью имитационного моделирования.

б) вероятность ошибки на бит при передаче данных без кодирования, рассчитанной с помощью имитационного моделирования.

в) вероятность ошибки декодирования при передаче данных со сверточным кодированием, рассчитанной с помощью имитационного моделирования.

г) вероятность ошибки декодирования при передаче данных без кодирования, рассчитанной с помощью имитационного моделирования.

д) верхнюю границу ошибки декодирования, рассчитанной по формуле (1.1) (см. разд. 1.2).

е) значение средней скорости передачи, рассчитанной по формуле (1.4) (см. разд. 1.3).

3. Продемонстрировать работу моделирующей программы и построенные графики преподавателю.

4.4. Номера наборов параметров сверточного кода и полиномов обратной связи

Таблица 4.1

Наборы параметров для HARQ T1

Номер набора параметров	Скорость для HARQ T1	Полиномы обратной связи
1	1/2	[13 17]
2	1/3	[13 17 11]
3	1/4	[13 17 11 03]

5. ЛАБОРАТОРНАЯ РАБОТА №2 «ПЕРЕДАЧА ДАННЫХ В СИСТЕМАХ С ОБРАТНОЙ СВЯЗЬЮ HARQ TYPE 2 (INCREMENT REDUNDANCY)»

Цель работы: исследование системы передачи данных с применением гибридной обратной связи на основе наращиваемой избыточности.

5.1. Описание задания

Реализовать систему передачи данных по каналу с АБГШ и замираниями, используя CRC-16 для проверки целостности данных с порождающим многочленом $x^{16} + x^{13} + x^{12} + x^{11} + x^{10} + x^8 + x^6 + x^5 + x^2 + 1$. Для кодирования данных использовать сверточный код с переменной скоростью и полиномы обратной связи. Длина информационных данных 32 бита.

5.2. Моделирование системы передачи с применением гибридной обратной связи на основе наращиваемой избыточности

На рис. 5.3 изображена схема системы. В соответствии с данной схемой для передачи каждого сообщения при текущем SNR выполняются следующие действия:

- 1) сгенерировать сообщение длиной 32 бита;
- 2) к сообщению добавляются проверочные биты CRC-16. Длина сообщения с контрольной суммой представляет собой последовательность 48 бит;
- 3) сообщение с контрольной суммой кодируется сверточным кодом с соответствующей минимальной скоростью кода. Например, если минимальная скорость $2/8$, то по формуле 1.5 длина закодированного сообщения $48 \cdot 8 / 2 = 192$;
- 4) из полученного сообщения формируются пакеты для работы на разных скоростях кода. Рассмотрим пример аналогично пункту 1.6. Тогда для скорости кода $2/3$ длина пакета будет 72 бита,

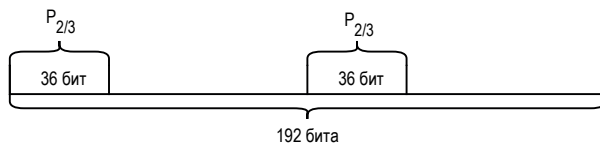


Рис. 5.1. Пакет с данными при скорости $2/3$

$48 \cdot 3/2 = 72$. Последовательность разбивается следующим образом: так как необходимо получить 72 бита, берется 36 битов сначала и 36 битов после половины.

Пакет для скорости 2/3 имеет вид:

$$P_{2/3} = [m(1:36) \ m(97:132)]$$

$$P_{res} = [P_{2/3}]$$

Для скорости 2/4 длина пакета $48 \cdot 4/2 = 96$. Необходимо добавить по 12 бит с каждой стороны.

Тогда пакет, который будет передан при скорости кода 2/4, имеет вид:

$$P_{2/3} = [m(1:36) \ m(97:132)]$$

$$P_{2/4} = [m(37:48) \ m(133:144)]$$

$$P_{res} = [P_{2/3} \ P_{2/4}]$$

Для остальных скоростей формирование пакетов происходит аналогично.

1) скорость кода первой передачи для текущего сообщения равна максимальной скорости из заданного набора;

2) сообщение модулируется в соответствии с правилом: $0 \rightarrow 1$; $1 \rightarrow -1$;

3) передается модулированный пакет, соответствующий текущей скорости кода, в канал с замираниями и АБГШ. Замирания вычисляются следующим образом

$$H = \sqrt{randn^2 + randn^2},$$

4) где $randn$ – случайная величина, распределенная по нормальному закону с параметрами $(0; 1)$. Сигнал на выходе канала имеет вид $r = s \cdot H + n$, где r – выход канала; s – модулированный сигнал; H – коэффициент релейевского замирания в канале; n – величина, распределенная по нормальному закону с параметрами $(0; \sigma^2)$ (см. прил. 1);

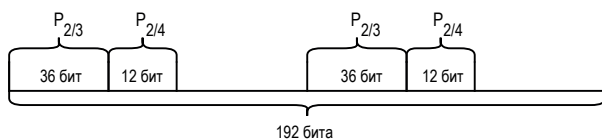


Рис. 5.2. Пакет с данными при скорости 2/4

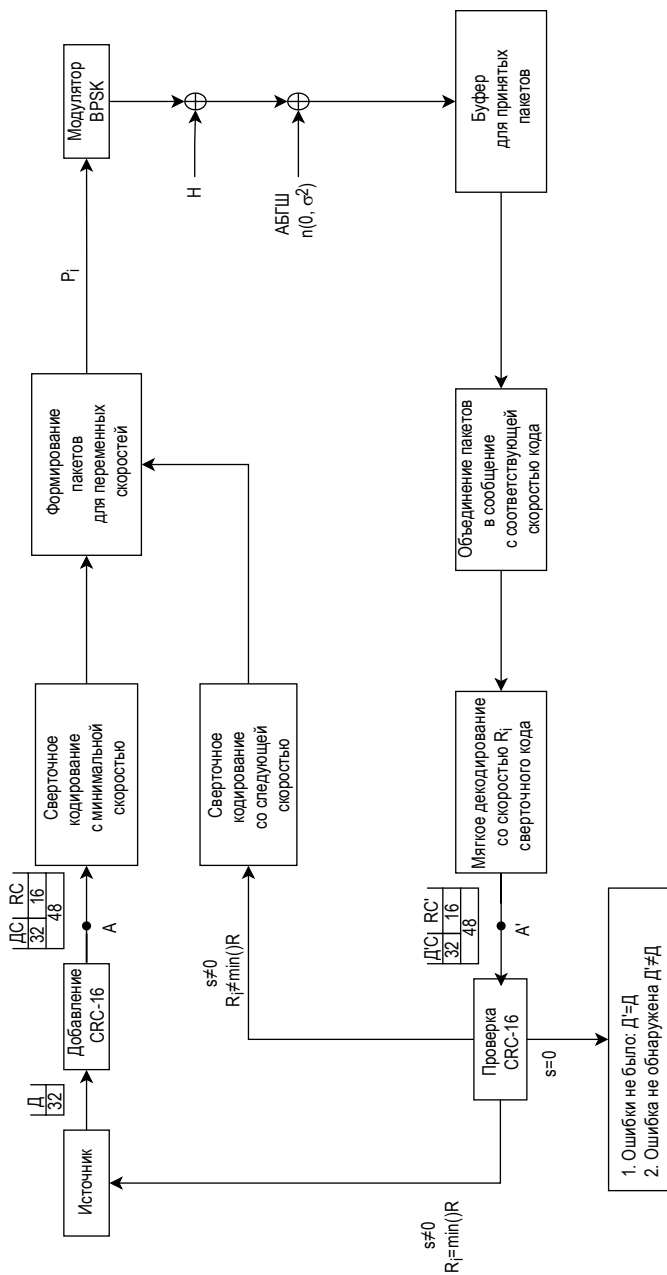


Рис. 5.3. Схема системы передачи данных HARQ Type 2

5) полученное на выходе канала сообщение подается на декодер сверточного кода с мягкими решениями;

6) проверка CRC, если синдром равен 0, то передается следующее сообщение, иначе передаются дополнительные биты для текущего сообщения, соответствующие более медленной скорости кода, т. е. переход к шагу 7. Если были переданы проверочные биты для всех скоростей (что соответствует минимальной скорости кода) и сообщение так и не было принято, то все передачи повторяются заново, т. е. переход к шагу 3. Отметим, что при отправке сообщения на заданной скорости прибавляется счетчик отправленных сообщений для текущего SNR.

Для ускорения моделирования можно использовать нулевую информационную последовательность. Тогда шаги 1, 2 и 3 можно пропустить, поскольку сообщение, закодированное сверточным кодом с минимальной скоростью, будет представлять собой нулевую последовательность длиной $48 \cdot 1/R_{\min}$. Также заранее рекомендуется создать решетки для всех скоростей с помощью функции *poly2trellis*.

5.3. Порядок выполнения лабораторной работы

1. Написать и отладить моделирующую программу в соответствии с полученным от преподавателя вариантом.

2. Построить графики следующих значений от отношения сигнал/шум в децибелах:

а) вероятность ошибки на бит при передаче данных со сверточным кодированием, рассчитанной с помощью имитационного моделирования в точках А и А' по следующему принципу: число ошибочных бит разделить на общее число переданных бит;

б) вероятность ошибки на бит при передаче данных без кодирования, рассчитанной с помощью имитационного моделирования по следующему принципу: число ошибочных бит разделить на общее число переданных бит;

в) вероятность ошибки декодирования при передаче данных со сверточным кодированием, рассчитанной с помощью имитационного моделирования;

г) вероятность ошибки декодирования при передаче данных без кодирования, рассчитанной с помощью имитационного моделирования;

д) верхнюю границу ошибки декодирования, рассчитанной по формуле (1.1) (см. разд. 1.2);

е) значение пропускной способности, рассчитанной по формуле (1.4) (см. разд. 1.3);

3. Продемонстрировать работу моделирующей программы и построенные графики преподавателю.

5.4. Номера наборов параметров кода и полиномов обратной связи

Таблица 5.1

Наборы параметров для HARQ T2

Номер набора параметров	Скорость для HARQ T2	Полиномы обратной связи
1	1/4, 1/3, 1/2	[13 17 11 03]
2	1/4, 1/3, 1/2, 1	[13 17 11 03]
3	2/8, 2/7, 2/6, 2/5, 2/4, 2/3	[23 35 0 32 07 16 14 06; 0 5 13 07 13 05 10 01]
4	2/7, 2/5, 2/3	[23 35 0 32 07 16 14; 0 5 13 07 13 05 10]

6. ВАРИАНТЫ ЗАДАНИЙ

Наборы параметров приводились в табл. 4.1 и 5.1.

Таблица 6.1

Варианты лабораторных работ

Вариант	Набор параметров для HARQ T1	Набор параметров для HARQ T2
1	1	1
2	2	2
3	3	3
4	1	4
5	2	1
6	3	2
7	1	3
8	2	4
9	3	1
10	1	2
11	2	3
12	3	4

ДВОИЧНАЯ ФАЗОВАЯ МОДУЛЯЦИЯ

Сигналы фазовой модуляции имеют вид

$$s_i(t) = \sqrt{2E/T} \cos(2\pi f_0 t - \theta_i), \quad 0 < t < T$$

где E – энергия сигнала; T – период следования сигнала; f_0 – несущая частота; θ_i – фаза i -го сигнала.

Вероятность ошибки для такой системы вычисляется следующим образом

$$P_e = Q(\sqrt{2E/N_0}), \quad (\text{П1.1})$$

где P_e – вероятность ошибки на бит, E – средняя энергия сигнала, N_0 – энергия шума, а $Q(x)$ имеет вид

$$Q(x) = \int_x^{\infty} \frac{1}{\sqrt{2\pi}} e^{-z^2/2} dz.$$

Расчет дисперсии шума для имитационного моделирования.

Для передачи сигнала используется двоичная модуляция, следовательно, энергия сигнала в расчете на 1 отсчет (бит) равна 1. Тогда можно воспользоваться формулой $E/N_0 = 10^{(E/N_0)_{\text{дБ}}/10}$ для нахождения N_0 . Далее, используя формулу $\sigma^2 = N_0/2$ необходимо найти σ .

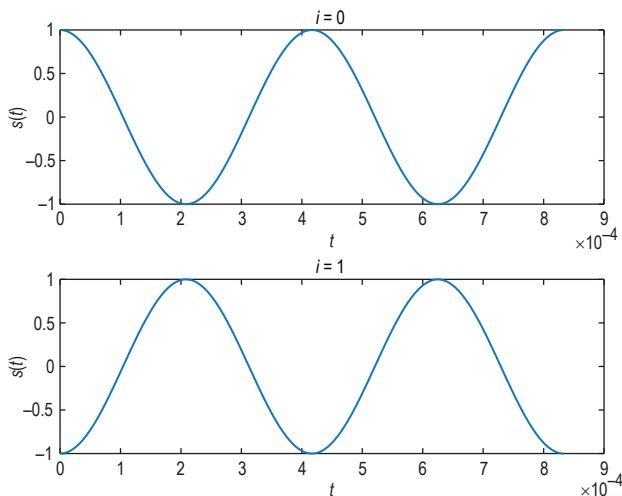


Рис. П.1 Сигналы ФМ-2

ИДЕЯ МЯГКОГО И ЖЕСТКОГО ДЕКОДИРОВАНИЯ

Жесткий декодер представляет собой декодер максимального правдоподобия. Его суть заключается в поиске кодового слова, которое имеет наименьшее значение расстояния Хэмминга с принятой последовательностью.

В качестве примера будет приведен код Хэмминга (7, 4). Рассмотрим шаги одной из итерации декодирования:

1. Для передачи из кодовой книги было выбрано следующее слово [0 0 1 1 0 1 0].

2. Промодулировав данное кодовое слово получили следующую последовательность [1 1 -1 -1 1 -1 1]

3. Промодулированная последовательность поступила в канале.

4. На вход демодулятор пришла последовательность из значений от $+\infty$ до $-\infty$. Демодулятор преобразует ее в 1 и -1.

5. На вход декодера поступила следующая последовательность [0 0 0 1 0 1 0]. Далее декодер прошел по всей кодовой книге и подсчитал для каждого кодового слова расстояние Хэмминга до полученной из демодулятора последовательности и выбрал кодовое слово с наименьшим расстоянием. В табл. П.1 рассмотрен пример такого перебора.

Декодирование с мягким решением использует принятые из канала величины для формирования оценок кодой последовательности. Мягкое декодирование подразумевает под собой поиск кодового слова, находящегося на минимальном евклидовом расстоянии от принятой последовательности. Последовательность из канала поступает сразу на декодер.

В качестве примера будет приведен код Хэмминга (7, 4). Рассмотрим шаги одной из итерации декодирования:

1. Для передачи из кодовой книги было выбрано следующее слово [0 0 1 1 0 1 0].

2. Промодулировав данное кодовое слово получили следующую последовательность [1 1 -1 -1 1 -1 1]

3. Промодулированная последовательность поступила в канале.

4. На вход декодера поступила следующая последовательность [0.587, 1.19, -0.884, -3.515, 2.226, -0.238, 0.089]. Далее декодер сравнил евклидово расстояние для всех замодулированных кодовых слов и выбрал с наименьшим расстоянием. В табл. П.2 рассмотрен пример такого перебора.

Стоит отметить, что процедура мягкого декодирования вычислительно сложнее жесткого декодирования, за счет применения

Таблица П.1

Поиск наименьшего расстояния Хэмминга

y	a		$d(a, y)$
0001010	0000000	0001010	2
	1010001	1011011	5
	1110010	1111000	4
	0100011	0101001	3
	0110100	0111110	3
	1100101	1101111	6
	1000110	1001100	3
	0010111	0011101	4
	1101000	1100010	3
	0111001	0110011	4
	0011010	0010000	1
	1001011	1000001	2
	1011100	1010110	4
	0001101	0000110	3
	0101110	0100100	2
	1111111	1110101	5

Таблица П.2

Подсчет Евклидовых расстояний

y	x	
0.587 1.19 -0.884 -3.515 2.226 -0.238 0.089	1111111	5.308
	-11-1111-1	5.599
	-1-1-111-11	5.899
	1-1111-1-1	5.687
	1-1-11-111	6.189
	-1-111-11-1	6.968
	-1111-1-11	6.518
	11-11-1-1-1	5.740
	-1-11-1111	5.023
	1-1-1-111-1	3.962
	11-1-11-11	3.103
	-111-11-1-1	4.458
	-11-1-1-111	5.083
	111-1-11-1	4.835
	1-11-1-1-11	5.180
	-1-1-1-1-1-1-1	5.477

вычисления Евклидовых расстояний. Однако данный подход позволяет уменьшить вероятность ошибки на бит за счет использования дополнительной информации о надежности принятого сигнала [6, 7].

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. *Бурков А. А.* Основы построения инфокоммуникационных систем и сетей: лаб. практикум / А. А. Бурков, А. М. Тюрликов. СПб.: ГУАП, 2018. 50 с.
2. *Трофимов А. Н.* Основы теории цифровой связи: учеб. пособие / А. Н. Трофимов. СПб.: ГУАП, 2014. 184 с.
3. *Трофимов А. Н.* Основы теории цифровой связи: учеб.-метод. пособие / А. Н. Трофимов. СПб.: ГУАП, 2018. – 53 с.
4. *Бурков А. А., Тюрликов А. М.* Верхняя оценка спектральной эффективности для систем с гибридной решающей обратной связью при ограничении на вид модуляции / Вопросы радиоэлектроники. Сер.: Техника телевидения. – 2020. – № 1. – С. 74–83.
5. *Evgenii Krouk, Sergei Semenov.* Modulation and coding techniques in wireless communications / A John Wiley and Sons, Ltd, Publication, 2011. P. 682.
6. *Морелос-Сарагоса Р.* Искусство помехоустойчивого кодирования. Методы, алгоритмы, применение / Р. Морелос-Сарагоса. М.: Техносфера, 2005. 320 с.
7. *Кудряшов Б. Д.* Основы теории кодирования: учеб. пособие / Б. Д. Кудряшов СПб.: БХВ-Петербург, 2016. 400 с.

СОДЕРЖАНИЕ

Введение.....	3
1. Принципы построения систем с гибридной обратной связью и способы анализа характеристик таких систем	4
1.1. Системы с гибридной обратной связью	4
1.2. Использование циклических кодов для формирования контрольной суммы	11
1.3. Коэффициент использования канала.....	15
1.4. Использование средств MatLab для моделирования работы кодера и декодера сверточных кодов	15
2. Первый допуск к выполнению лабораторных работ. Система передачи данных с контрольной суммой	24
2.1. Описание моделируемой системы	24
2.2. Порядок выполнения допуска	24
2.3. Варианты многочленов CRC для допуска.....	26
3. Второй допуск к выполнению лабораторных работ. Система передачи данных с обратной связью	27
3.1. Описание моделируемой системы	27
3.2. Порядок выполнения допуска.....	27
3.3. Варианты заданий	28
4. Лабораторная работа №1 «Передача данных в системах с обратной связью HARQ Type 1 (Chase Combining)»	34
4.1. Описание задания.....	34
4.2. Моделирование системы передачи с применением гибридной обратной связи	34
4.3. Порядок выполнения лабораторной работы	36
4.4. Номера наборов параметров сверточного кода и полиномов обратной связи	36
5. Лабораторная работа №2 «Передача данных в системах с обратной связью HARQ Type 2 (Increment Redundancy)»	37
5.1. Описание задания.....	37
5.2. Моделирование системы передачи с применением гибридной обратной связи на основе наращиваемой избыточности	37
5.3. Порядок выполнения лабораторной работы	40
5.4. Номера наборов параметров кода и полиномов обратной связи	41
6. Варианты заданий	42
Приложение 1. Двоичная фазовая модуляция	43
Приложение 2. Идея мягкого и жесткого декодирования	44
Библиографический список	46

Учебное издание

**Бурков Артём Андреевич,
Тюрликов Андрей Михайлович**

**ИСПОЛЬЗОВАНИЕ ГИБРИДНОЙ
ОБРАТНОЙ СВЯЗИ
В БЕСПРОВОДНЫХ СЕТЯХ**

Учебно-методическое пособие

Публикуется в авторской редакции
Компьютерная верстка *С. Б. Мацапуры*

Подписано к печати 25.02.2022. Формат 60 × 84 1/16.

Усл. печ. л. 2,8. Уч.-изд. л. 3,0.

Тираж 50 экз. Заказ № 69.

Редакционно-издательский центр ГУАП
190000, г. Санкт-Петербург, ул. Б. Морская, д. 67, лит. А