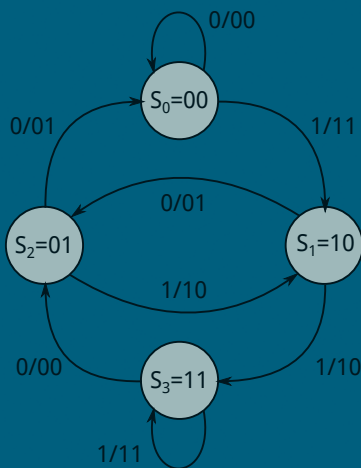


Небаев И.А.

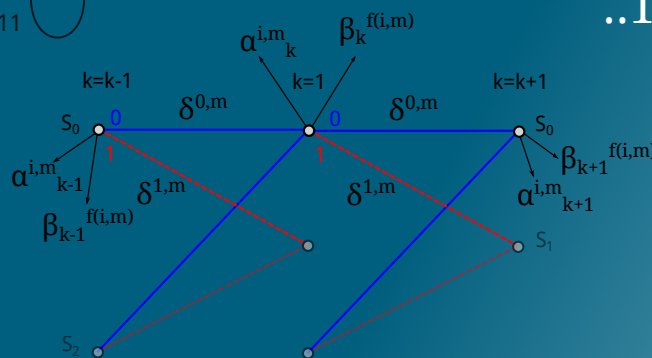
Итеративное декодирование сверточных турбокодов

<http://opds.sut.ru>

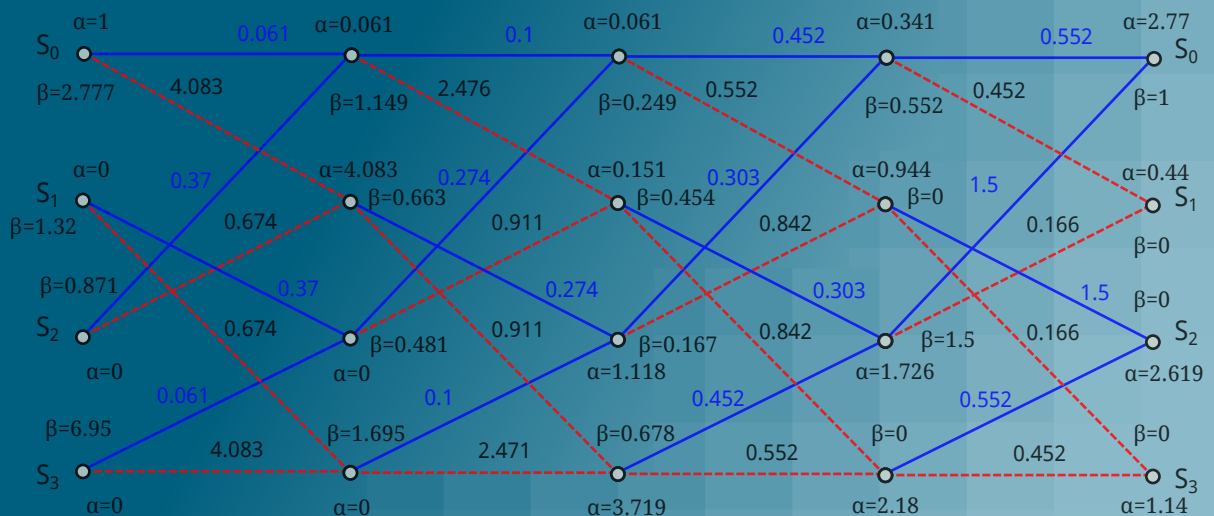


$$r(t) = s_i(t) + n(t)$$

...0001110111011...
00010111100111..
 ..11100100111...



$L(d_1) = 2.33$
 $L(d_2) = 0.43$
 $L(d_3) = -1.1$
 $L(d_4) = -0.19$



Санкт-Петербургский государственный университет телекоммуникаций
им. проф. М. А. Бонч-Бруевича (СПбГУТ)

Кафедра «Обработки и передачи дискретных сообщений» (ОПДС)

Небаев И. А.

Учебное пособие к курсовой работе

«Итеративное декодирование сверточных турбокодов»

«Теория кодирования» (ТК)

В учебном пособии рассматриваются практические аспекты применения алгоритмов и методов обработки информации на основе современных итеративно декодируемых кодов. На числовых примерах продемонстрированы принципы кодирования информации с помощью параллельных сверточных турбокодов и алгоритмы мягкого итеративного декодирования. Во второй части пособия размещены практические задания для выполнения курсовой работы и методические указания к их решению.

Учебное пособие предназначено для студентов старших курсов, обучающихся по направлениям 210700 «Инфокоммуникационные технологии и системы связи».

Рецензенты:

Когновицкий О. С., д.т.н., профессор

Буданов А. В., к.т.н., доцент

Санкт-Петербург — 2013г.

Содержание

I Канальное кодирование и формирование избыточной последовательности	4
1. Основные принципы формирования параллельных сверточных турбокодов	4
2. Пример кодирования параллельным сверточным турбокодом	6
II Прием, обработка и декодирование информации	9
3. Итеративная обработка и декодирование сверточных турбокодов	9
4. Алгоритм вероятностного декодирования сверточных турбокодов	14
5. Пример итеративного декодирования сверточных турбокодов	17
III Указания к выполнению курсовой работы	28
6. Цели курсовой работы	28
7. Задачи курсовой работы	28
7.1. Кодирование исходного символа	28
7.2. Передача и прием данных	29
7.3. Итеративное декодирование	31
8. Форма отчета	33
9. Варианты заданий	34
IV Приложение	37

Часть I

Канальное кодирование и формирование избыточной последовательности

1. Основные принципы формирования параллельных сверточных турбокодов

Систематические сверточные коды, применяемые в качестве составных кодов турбокодера, характеризуются параллелизацией (конкатенацией) конечного набора кодирующих устройств. При конструктивном объединении нескольких сверточных кодеров образуется мультикодирующая система, отличительным признаком которой является поточный метод кодирования и модель системы с множеством входов–выходов. Кратко рассмотрим параметры систематических сверточных кодов, оказывающие важное влияние на свойства формируемого составного сверточного турбокодера.

На рис. 1 представлена схема систематического сверточного кодера, характеризующегося степенью кодирования $r = 1/2$, памятью $M_c = 2$ и кодовым ограничением $K = (M_c + 1) = 3$. Для описания всех возможных состояний кодера ($S = M_c^2 = 4$) на рис. 2 представлена диаграмма состояний кодера (а) и кодовая решетка (б).

Структуру связей данного кодера можно представить в виде системы порождающих многочленов для первого ($x^{(0)}$) и второго ($x^{(1)}$) выходов соответственно:

$$\begin{cases} g^{(0)} = 1 \\ g^{(1)} = 1 + x + x^2 \end{cases} \quad (1)$$

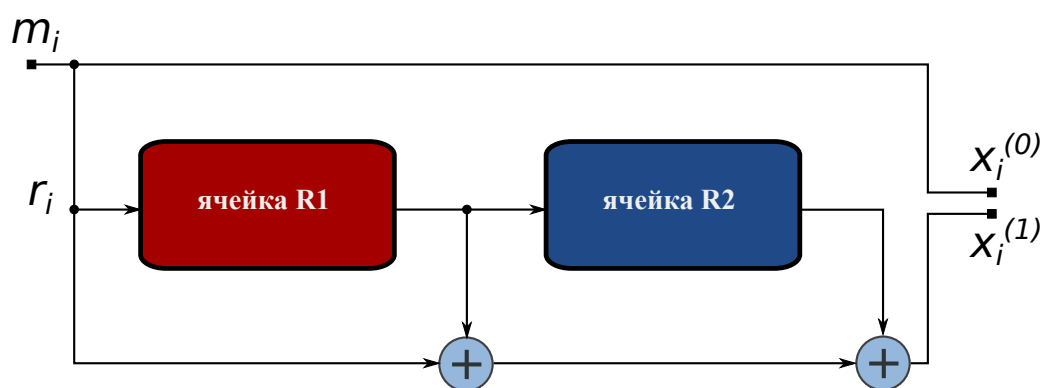


Рисунок 1 — Схема систематического сверточного кодера $r = 1/2$

Выход каждой ветви, формируемый в результате выполнения процедуры кодирования, для сверточного кодера на рис. 1 представляет собой результат сложения значения входного бита m_i и содержимого ячеек памяти кодера в данном такте, и может быть записано в форме:

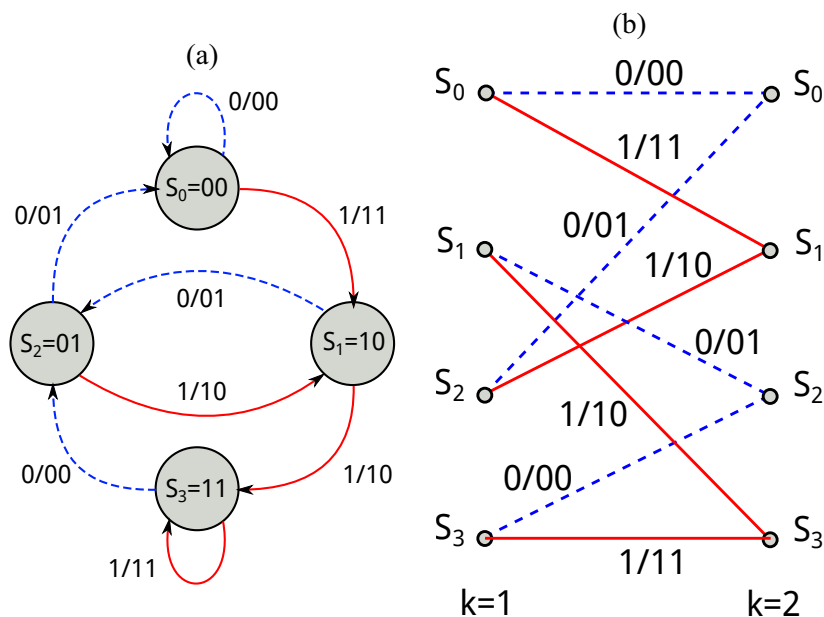


Рисунок 2 — Диаграмма состояний сверточного кодера (a) и кодовая решетка (b)

$$v_i^{(p,q)} = \sum_{j=0}^{M_c} m_{i-j}^{(p)} g_j^{(p,q)}, \quad (2)$$

где (p, q) — номер входа и выхода сверточного кодера, соответственно. Для кодера на рис. 1 $p = 0, q = [0, 1]$. Результирующий кодовый поток сверточного кода представляет собой последовательность бит с каждой ветви кодера:

$$x^{(q)} = \sum_{p=0}^{k-1} v^{(p,q)}, \quad (3)$$

где k — количество информационных входов кодера.

Как уже было указано ранее, систематические сверточные турбокоды конструируются по схеме параллельного построения кодирующего устройства. Параллелизация схемы кодирования преследует цели объединения кодовых комбинаций с каждой ветви кодера для увеличения весовых коэффициентов результирующей кодовой комбинации кодера. Параллельная схема систематического сверточного турбокодера позволяет снизить вероятность образования на выходе кодера кодовых комбинаций с малым весом.

На рис. 3 представлена структура сверточного турбокода. Составляющие кодеры обрабатывают информационные биты параллельно, однако кодирование во втором сверточном кодере, предваряется процедурой перемежения информационной последовательности m_i ($i \in (0, \dots, L - 1)$, где L — длина блока). В задачи блока перемежения входит декорреляция и псевдослучайное размещение информационных символов.

Составная схема турбокодера, изображенного на рис. 3, формализует требования к конкатенируемым кодерам. Все кодирующие блоки должны иметь одинаковую

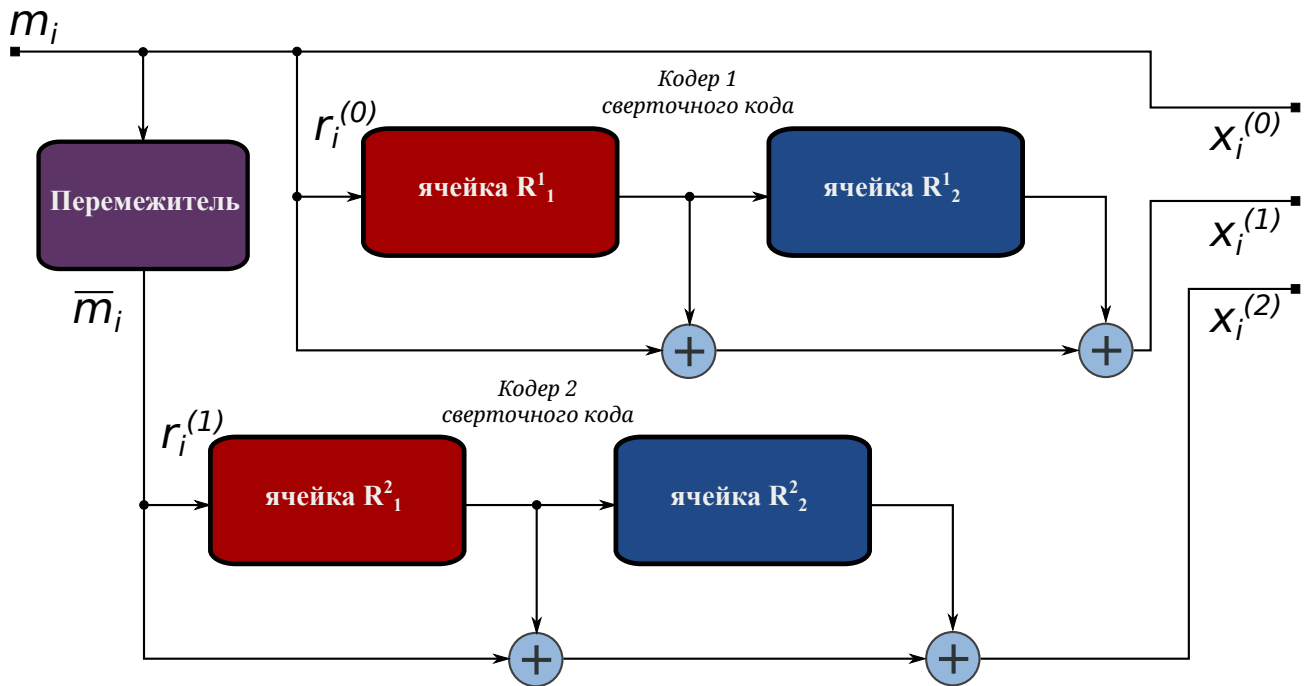


Рисунок 3 – Структура сверточного турбокодера $r = 1/3$

длину кодового ограничения и кодовую скорость. Кодовое слово кодера на рис. 3, состоит из информационного бита и двух проверочных бит, и имеет вид $x = (x_0^{(0)}, x_0^{(1)}, x_0^{(2)}, \dots, x_{L-1}^{(0)}, x_{L-1}^{(1)}, x_{L-1}^{(2)})$, что в результате дает кодовую скорость $r = 1/3$.

Для случая большего числа составляющих (внутренних) кодеров турбокода, изображенных на рис. 4, результирующая кодовая скорость может быть найдена с помощью выражения

$$r = \frac{k}{n(N+1)} \quad (4)$$

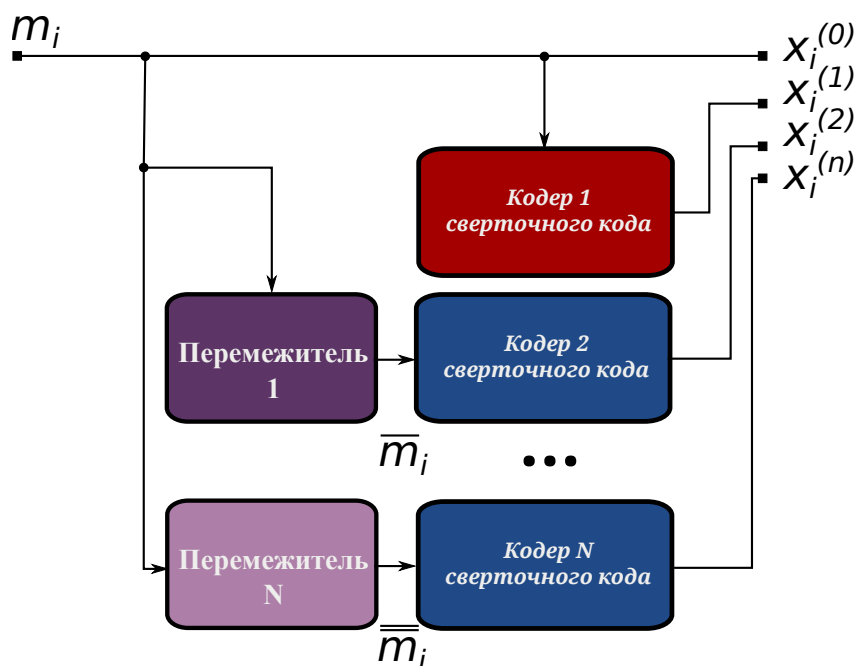
где N – количество составляющих кодеров. При скорости составляющих кодеров $r = 1/2$, общая степень кодирования равна

$$r = \frac{1}{N+1}. \quad (5)$$

2. Пример кодирования параллельным сверточным турбокодом

Рассмотрим метод кодирования данных с помощью систематического сверточного турбокода на примере составного кодера рис. 3. Пусть исходная информационная последовательность представляет собой вектор длиной $L = 4$ бита:

$$[1 \ 1 \ 0 \ 0] \quad (6)$$

Рисунок 4 — Структура N -размерного сверточного турбокодера

Условимся, что кодирование начинается в нулевом состоянии, т. е. оба кодера обнулены. Для формирования входной последовательности на втором кодере используется перемежитель размерностью 2×2 , который читает информационный блок по 2 бита по строкам, а выдает кодеру по столбцам. Т. о., содержимое матрицы перемежителя имеет вид:

$$\begin{bmatrix} 1 & 0 \\ 1 & 0 \end{bmatrix} \quad (7)$$

Следовательно входные биты, подаваемые на второй кодер представляют собой вектор:

$$[1 \ 0 \ 1 \ 0] \quad (8)$$

Руководствуясь диаграммой состояний рис. 2 (а) для каждого из составных кодеров, результаты кодирования представлены в табл. 1.

Поскольку систематический выход x_i^0 второго (внутреннего) составного сверточного кодера формируется из перемеженной последовательности и может быть восстановлен на принимающей стороне исходя из систематических бит первого кодера (внешнего), то биты x_i^0 , сформированные вторым кодером в канал не передаются. Исходя из выражения (3) и параллельной конструкции турбокодера, результирующая выходная кодовая комбинация будет иметь вид $x = (x_0^{(0)}, x_0^{(1)}, x_0^{(2)}, \dots, x_3^{(0)}, x_3^{(1)}, x_3^{(2)})$:

$$[1 \ 1 \ 1 \ 1 \ 0 \ 1 \ 0 \ 0 \ 0 \ 0 \ 1 \ 1] \quad (9)$$

Таблица 1 – Процесс кодирования исходных последовательностей (6) и (8)

Входной бит, m_i	Регистр R_1^1	Регистр R_2^1	Выход x_i^0	Выход x_i^1
-	0	0	-	-
$m_0 = 1$	1	0	1	1
$m_1 = 1$	1	1	1	0
$m_2 = 0$	0	1	0	0
$m_3 = 0$	0	0	0	1
Входной бит, \bar{m}_i	Регистр R_1^2	Регистр R_2^2	Выход x_i^0	Выход x_i^2
-	0	0	-	-
$\bar{m}_0 = 1$	1	0	1	1
$\bar{m}_1 = 0$	0	1	0	1
$\bar{m}_2 = 1$	1	0	1	0
$\bar{m}_3 = 0$	0	1	0	1

Часть II

Прием, обработка и декодирование информации

3. Итеративная обработка и декодирование сверточных турбокодов

Распределение вероятностей при передаче сигналов по каналу с аддитивным белым гауссовым шумом (АБГШ) можно представить в виде функций правдоподобия, изображенных на рис. 5. Функция $p(x|d=0)$ выражает распределение случайной непрерывной переменной x , при условии передачи бита данных $d=0$. Соответственно, функция $p(x|d=1)$ — распределение x при передаче бита $d=1$.

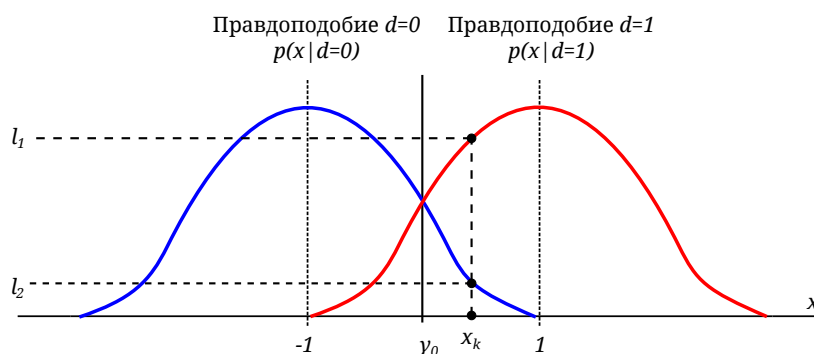


Рисунок 5 — Функция распределения вероятностей принятого сигнала x_k

При детектировании значения сигнала x_k (рис. 5), в момент времени k , необходимо определить значение функции правдоподобия для полученного x_k . Расположение x_k вдоль оси x , не позволяет однозначно определить значение передаваемого бита, поскольку отмеченной точке x_k соответствуют два значения правдоподобия — l_1 для $d=1$ и l_2 для $d=0$. Основываясь на принципах жесткого решения и максимального правдоподобия, приемник должен сравнить значения l_1 и l_2 и вынести решение о принятом бите: $d|_{l_1 > l_2} = 1$ или $d|_{l_2 > l_1} = 0$. Т. о. в жесткой схеме принятия решения, при расположении переменной x_k правее порога γ_0 , принятый сигнал переносит бит $d=1$, а при расположении x_k левее γ_0 , сигнал переносит бит $d=0$.

При декодировании турбокодов используется алгоритм максимума апостериорной вероятности (MAP — *Maximum A posteriori Probability*), исполняющий правило минимизации вероятности ошибки и учитывающий значение априорной вероятности данных. Принцип расчета апостериорной вероятности основан на обновлении данных полученных при детектировании исходного сигнала. Выбор значения переданного бита осуществляется согласно неравенству наибольшей апостериорной вероятности:

$$\begin{cases} d = 1 : P(d = 1|x) > P(d = 0|x) \\ d = 0 : P(d = 1|x) < P(d = 0|x), \end{cases} \quad (10)$$

где $P(d|x)$ — апостериорная вероятность принятого сигнала.

Выражение (10) можно переписать, используя одну из форм теоремы Байеса (см. приложение ??) и априорную вероятность появления сигнала:

$$\begin{cases} d = 1 : p(x|d = 1)P(d = 1) > p(x|d = 0)P(d = 0) \\ d = 0 : p(x|d = 1)P(d = 1) < p(x|d = 0)P(d = 0) \end{cases} \quad (11)$$

Однако, на практике для решения задач декодирования используется отношение функций правдоподобия выраженное в логарифмической форме, и ведущее к сокращению машинных вычислений. Сама функция отношения правдоподобий представляет собой дробь от (11):

$$\begin{cases} d = 1 : \frac{p(x|d=1)P(d=1)}{p(x|d=0)P(d=0)} > 1 \\ d = 0 : \frac{p(x|d=1)P(d=1)}{p(x|d=0)P(d=0)} < 1 \end{cases}, \quad (12)$$

а логарифмическое отношение функций правдоподобия (LLR — *Log-Likelihood Ratio*):

$$\begin{aligned} L(d|x) &= \ln \left(\frac{P(d = 1|x)}{P(d = 0|x)} \right) = \\ &= \ln \left(\frac{p(x|d = 1)P(d = 1)}{p(x|d = 0)P(d = 0)} \right) = \\ &= \ln \left(\frac{p(x|d = 1)}{p(x|d = 0)} \right) + \ln \left(\frac{P(d = 1)}{P(d = 0)} \right). \end{aligned} \quad (13)$$

Первое слагаемое суммы в выражении (13) представляет собой логарифмическое отношение функций правдоподобия (LLR) тестовой статистики x , полученное путем измерений x на выходе канала передачи данных при попеременной передаче бит $d = 0$ и $d = 1$. Второе слагаемое представляет собой априорное логарифмическое отношение функций правдоподобия бита данных d . В укороченной форме сумма (13) принимает вид:

$$L(d|x) = L(x|d) + L(d). \quad (14)$$

Выражение (14) характеризует данные полученные детектором сигнала. Далее, оценка детектированных данных должна поступать на декодер помехоустойчивого кода, работающий по алгоритму мягкого декодирования. Мягкий выход декодера формируется из суммы логарифмического отношения функций правдоподобия бита данных d на выходе детектора ($L'(\hat{d})$) и внешней информации $L_e(\hat{d})$, получаемой в результате процедур декодирования:

$$L(\hat{d}) = L'(\hat{d}) + L_e(\hat{d}). \quad (15)$$

Заключительное выражение для расчета значений мягкого выхода декодера, с учетом всей доступной информации до декодирования и собранной информацией в процессе декодирования, имеет вид:

$$L(\hat{d}) = L_c(x) + L(d) + L_e(\hat{d}), \quad (16)$$

где $L_c(x)$ — канальное измерение величины x , полученное приемником, $L(d)$ — априорное значение данных, имеющееся до декодирования и $L_e(\hat{d})$ внешнее значение LLR, получаемое в результате декодирования. Следует отметить, что все три компоненты суммы выражения (16) являются статистически независимыми, а операция суммирования определяет вклад каждого значения в заключительное решение. На рис. 6 представлен вид функции правдоподобия. Мягкий выход декодера $L(\hat{d})$ является вещественным числом, обеспечивающим в итоге надежность принятия жесткого решения, причем знак логарифма определяет область жесткого решения, а величина значения $L(\hat{d})$ устанавливает достоверность данного решения.

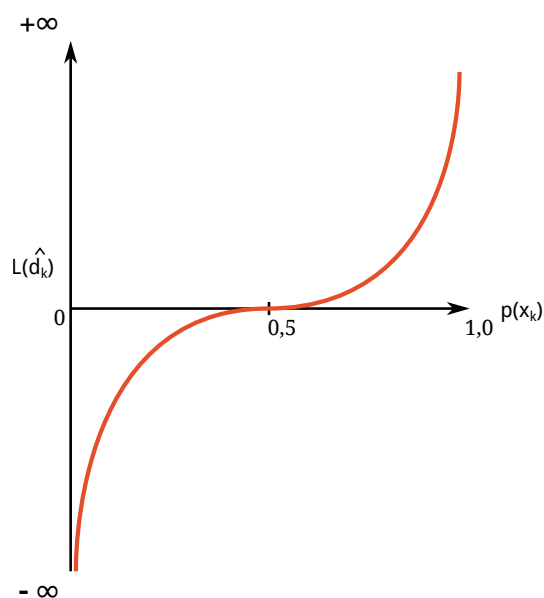


Рисунок 6 — Вид функции правдоподобия

При декодировании турбокодов используется некоторое заданное количество итераций обработки данных (Q). В задачу каждой итерации входит получение априорных данных о декодируемой информации, вынесение мягких решений и передача оценки результата декодирования на последующую итерацию. Подобная циклическая схема порождает турбообработку данных, в том смысле, что результаты предыдущей итерации декодирования являются внешней априорной информацией для следующего шага декодирования. За счет цикличной, многоитерационной обработки кодовых слов, становится возможным увеличить достоверность обрабатываемой информации и уменьшить вероятность ошибочного декодирования.

На рис. 7 изображен принцип итерационной обработки данных турбодекодером с мягкими входами и мягкими выходами (SISO — *Soft Inputs, Soft Outputs*). На первой

итерации декодирования исходя из канальных измерений детектора формируется массив значений канальных оценок принятых символов $L_c(x_k)$, вычисленных из логарифмического отношения значений l_1 и l_2 для рассматриваемого x_k (рис. 5). При инициализации первой итерации априорная информация о декодируемом символе отсутствует, т.к. декодер 2 не производил декодирования. Т.о. априорная информация на входе декодера 1 $L_e^2(d_k) = 0$. Входы x_k^1 и y_k^1 , представляют собой детектированные значения сигнала информационного и проверочного бита кодера 1 (см. схему рис. 3) сформированные по методу мягких оценок. Информационный вход первого декодера x_k^1 , также передается на перемежитель для формирования перемеженной последовательности, которая будет использована в качестве информационного входа кодера 2 (см. пример кодирования в разделе 1). После выполнения декодирования на выходе декодера 1 (рис. 7) может быть сформирован мягкий выход из суммы значений логарифма отношения правдоподобий детектора $L_c(x_k)$, априорной информации $L(d_k)$ и собственной информации декодера 1 $L_e^1(d)$. Т.к. на первой итерации $L(d_k) = L_e^2(d_k) = 0$, то выход декодера 1 представляет собой только сумму логарифмов канальных оценок и результатов декодирования данного декодера $L_e^1(d_k)$.

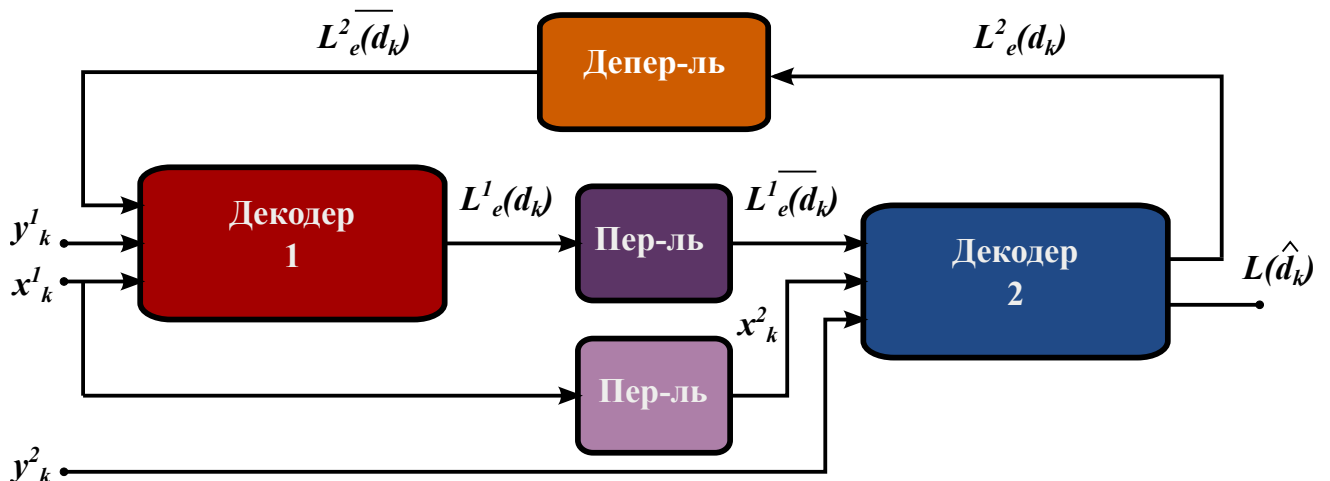


Рисунок 7 — Итерационная обработка данных в формате мягких решений

Условимся, что мягких решений полученных декодером 1 недостаточно, чтобы вынести решение о декодируемом символе с заданным уровнем надежности. Продолжим декодирование на декодере 2 и используем проверочные биты кодера 2 (см. схему рис. 3), а также ставшую теперь доступной априорную информацию $L(d_k)$, полученную в ходе декодирования первого набора кодовых слов, т.е. для декодера 2 $L(d_k) = L_e^1(d_k)$. Следует отметить, что результаты декодирования декодера 1 $L_e^1(d_k)$ необходимо подать на перемежитель для правильного позиционирования (соответствия) с перемеженными битами систематического входа второго кодера. Т.о. вход декодера 2 включает в себя перемеженную информационную последовательность x_k^2 , проверочные биты второго кодера y_k^2 и априорную информацию $L_e^1(d_k)$ декодера 1 для декодируемых x_k^2 .

В результате декодирования кодовых комбинаций второго кодера, декодер 2 формирует собственные мягкие оценки в виде суммы канальной оценки декодируемого бита $L_c(x_k)$,

априорной информации от декодера 1 ($L(d_k) = L_e^1(d_k)$), и собственных вычислений $L_e^2(d_k)$. Используя суммарную оценку, декодер 2 может рассчитать логарифмическое отношение правдоподобия (LLR) для каждого декодируемого бита, и, учитывая величину и знак отношения (см. график 6), вынести окончательное жесткое решение.

Как будет показано далее на числовом примере декодирования, по завершению полной первой итерации декодирования можно вынести жесткое решение. Однако, если необходимо выполнить заданное количество итераций декодирования, то результаты декодирования декодера 2 $L_e^2(d_k)$ сохраняются и используются в качестве априорной информации на следующей итерации декодирования. Для переупорядочивания порядка следования оценок декодируемых бит значения $L_e^2(d_k)$ поступают на депережежитель и присваиваются значениям априорной информации для декодера 1. Т.е. на начало второй итерации декодирования декодер 1 обладает канальными оценками декодируемых символов $L_c(x_k)$ и априорной информацией с декодера 2 (по результатам первой итерации), т.е. $L(d_k) = L_e^2(d_k)$. После данной процедуры начинается очередная итерация декодирования.

На последующих итерациях происходит уточнение информации и, наконец, на последней заданной итерации декодирования выносится окончательное жесткое решение. Как уже было сказано ранее, мягкий выход декодера $L(\hat{d}_k)$ представляет собой вещественное число, указывающее на достоверность принятия жесткого решения, а знак LLR определяет область жесткого решения. Необходимо подчеркнуть, что основная парадигма итеративного декодирования заключается в непрерывном обмене результатами декодирования между составляющими декодерами. Результаты декодирования по завершению второй полной итерации представляют собой уточненные оценки, скорректированные за счет наличия априорных данных с первой итерации и т.д. Благодаря «разгонному» циклу декодирования, на каждом из этапов которого происходит уточнение (корректировка) значений декодируемых символов, рассматриваемый метод помехоустойчивого кодирования приобрел название «турбодекодирования», поскольку алгоритмика и цели турбодекодирования во многом схожи с процессами происходящими в блоке турбореактивного двигателя.

Как указано ранее, применение сверточных турбокодов и их декодирование проводится по принципам минимизации вероятности ошибки и учета значений априорной вероятности данных. Благодаря формированию каждым составным декодером (см. рис. 7) собственных апостериорных оценок (вероятностей) значений информационных бит, осуществляется многопороговая схема принятия решения о наиболее вероятном значении декодируемого символа, основанная на вычислении ряда метрик вероятностей перехода по решетчатой диаграмме сверточного кода.

Кратко рассмотрим процедуры вероятностного декодирования сверточных турбокодов и особое внимание уделим численному примеру декодирования, позволяющему наглядно продемонстрировать эффективность метода итеративного мягкого декодирования составных кодов.

4. Алгоритм вероятностного декодирования сверточных турбокодов

Необходимо отметить, что в методе расчета и определения метрик алгоритма MAP широко используется принцип прямой и обратной рекурсии. Это вызывает некоторое осложнение понимания теории и методов вероятностного декодирования турбокода, однако позволяет достаточно эффективно строить и реализовывать алгоритмы для применения машинных и компьютерных вычислений.

В основу главной задачи решаемой с помощью алгоритма MAP входит вычисление апостериорных вероятностей всех возможных переходов принятого кодового символа по решетке сверточного декодера. В соответствии с выражением (13), расчет начинается с логарифмического отношения функций правдоподобия апостериорных вероятностей того, что декодируемый бит равен $d_k = 1$ или $d_k = 0$:

$$L(d_k) = \ln \left(\frac{\sum_m p_k^{1,m}}{\sum_m p_k^{0,m}} \right), \quad (17)$$

где $p_k^{i,m}$ — вероятность того, что $d_k = [0 : 1]$ в состоянии кодера $S_k = m$ при принятой кодовой последовательности C . Применяя теорему Байеса (см. приложение ??) можно представить $p_k^{i,m}$ в полном виде:

$$p_k^{i,m} = \frac{P(C_1^{k-1} |_{d_k=i, S_k=m, C_k^N}) P(C_{k+1}^N |_{d_k=i, S_k=m, C_k}) P(d_k = i, S_k = m, C_k)}{P(R_1^N)}, \quad (18)$$

где C_1^N — значение принятой последовательности бит ($C_k^N = (C_k, C_{k+1}^N)$). Каждый из членов совокупного вероятностного уравнения (18) представляет собой метрику состояний прямого (α_k^m) и обратного прохода (β_k^m) по кодовой решетке и метрику ветвей ($\delta_k^{i,m}$) кодовой решетки сверточного кода. Области значения метрик MAP кодовой решетки изображены на рис. 8 (а).

Рассмотрим подробнее принципы расчета метрик кодовой решетки. Последний член произведения (18) — это метрика ветви, которая выходит из состояния m в момент времени k :

$$\delta_k^{i,m} = P(d_k = i, S_k = m, C_k). \quad (19)$$

Метрику ветви можно представить в форме произведения вероятностей $P(x_k)$ и $P(y_k)$, учитывая значение принятого кодового слова, состоящего из битов данных x_k и проверочных бит y_k , а также априорной вероятности $P(d_k = i)$ появления бита d_k :

$$\delta_k^{i,m} = P(x_k | d_k = i, S_k = m) P(y_k | d_k = i, S_k = m) \frac{\pi_k^i}{2^m}, \quad (20)$$

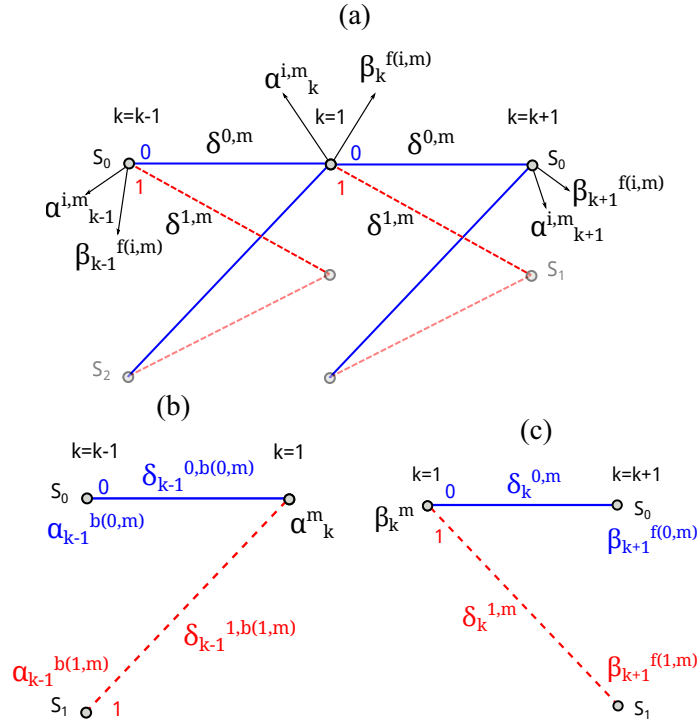


Рисунок 8 — Фрагменты решетки для расчета метрик состояний и ветвей

где π_k^i — априорная вероятность принятого бита d_k , а 2^m количество состояний кодера. Для канала АБГШ, с нулевым средним и дисперсией помехи σ^2 , вероятностные члены $P(x_k)$ и $P(y_k)$ можно заменить на эквивалентную функцию плотности вероятности:

$$\delta_k^{i,m} = \frac{\pi_k^i}{2^m \sqrt{2\pi\sigma}} \exp\left(-\frac{1}{2}\left(\frac{x_k - u_k^i}{\sigma}\right)^2\right) dx_k \frac{\pi_k^i}{\sqrt{2\pi\sigma}} \exp\left(-\frac{1}{2}\left(\frac{y_k - v_k^{i,m}}{\sigma}\right)^2\right) dy_k \quad (21)$$

Упростив выражение и используя замену некоторых членов получим уравнение для расчета метрик ветвей кодовой решетки:

$$\delta_k^{i,m} = A_k \pi_k^i \exp\left(\frac{1}{\sigma^2}(x_k u_k^i + y_k v_k^{i,m})\right), \quad (22)$$

где u_k^i и v_k^i — каналные значения информационного и проверочного бита соответственно, π_k^i — априорная вероятность принятого бита d_k , а постоянная A_k включает в себя дифференциалы dx_k и dy_k уравнения (21).

Вернемся к выражению 18. Первый множитель данного выражения представляет собой метрику состояния m в момент k при прямом проходе по кодовой решетке сверточного кодера:

$$\alpha_k^{i,m} = P(C_1^{k-1} | d_k=i, S_k=m, C_k^N). \quad (23)$$

Метрика состояний прямого прохода $\alpha_k^{i,m}$ представляет собой сумму всех возможных переходов из момента времени $k - 1$ в рассматриваемый k и может быть представлена как сумма вероятностей этих переходов:

$$\alpha_k^m = \sum_{m'} \sum_{j=0}^1 P(d_{k-1} = j, S_{k-1} = m', C_1^{k-1} |_{S_k=m}) \quad (24)$$

В результате, раскрыв данное выражение с помощью теоремы Байеса, получим:

$$\alpha_k^m = \sum_{j=0}^1 P(C_1^{k-2}, S_{k-1} = b(j, m)) P(d_{k-1} = j, S_{k-1} = b(j, m), C_{k-1}), \quad (25)$$

где функция $b(j, m)$ есть состояние, связанное по переходу предыдущей ветви, соответствующей входу j и исходящее обратно по времени из состояния m (см. рис. 8 (b)). Первый множитель выражения (25) представляет собой значение метрики прямого прохода в предыдущий момент времени ($k - 1$), а второй множитель метрику ветви исходящей из состояния с данной метрикой. Это значит, что метрика состояния m в момент времени k является суммой произведения значений метрик состояний в момент $k - 1$ и метрик ветвей, ведущих в данное состояние m . Подставив соответствующие обозначения метрик, получим выражение для расчета метрики состояний прямого прохода в момент k :

$$\alpha_k^m = \sum_{j=0}^1 \alpha_{k-1}^{b(j,m)} \delta_{k-1}^{j,b(j,m)}. \quad (26)$$

Аналогичным образом вычисляется и второй множитель уравнения (18), который представляет собой метрику состояния m в момент k при обратном проходе по кодовой решетке и обозначается:

$$\beta_{k+1}^{f(i,m)} = P(C_{k+1}^N |_{d_k=i, S_k=m, C_k}), \quad (27)$$

где функция $f(i, m)$ определяет следующее состояние кодера в зависимости от нового входа $i[0 : 1]$ и текущего состояния кодера m . Таким же образом, как и метрику состояний прямого прохода $\alpha_k^{i,m}$, метрику состояний обратного прохода $\beta_k^{i,m}$ можно представить в виде суммы вероятностей всех переходов в момент $k + 1$:

$$\beta_k^m = \sum_{m'} \sum_{j=0}^1 P(d_k = j, S_{k+1} = m', C_{k+1}^N |_{S_k=m}). \quad (28)$$

Вновь как и ранее, раскрывая данное выражение с помощью теоремы Байеса и после замены соответствующих членов, в результате получим уравнение для расчета метрики состояний при обратном проходе по кодовой решетке:

$$\beta_k^m = \sum_{j=0}^1 \beta_{k+1}^{f(j,m)} \delta_k^{j,m}. \quad (29)$$

Т.е. при обратном направлении прохода по кодовой решетке, метрика состояния m в момент времени k зависит от суммы произведения значений метрик состояний в момент $k + 1$ и метрик ветвей, ведущих в данное состояние m (см. рис. 8 (с)).

Используя выражение логарифмического отношения функций правдоподобия для $d_k = [0 : 1]$ (17) и указанные выражения метрик правдоподобия, запишем уравнение для вычисления мягкого выхода на итерации декодирования:

$$L_e(d_k) = \ln \left(\frac{\sum_m \alpha_k^m \delta_k^{1,m} \beta_{k+1}^{f(1,m)}}{\sum_m \alpha_k^m \delta_k^{0,m} \beta_{k+1}^{f(0,m)}} \right), \quad (30)$$

где делимое представляет собой функцию правдоподобия бита $d_k = 1$, а делитель функцию правдоподобия бита $d_k = 0$. Как уже было указано ранее, мягкий выход представляет собой вещественное число, устанавливающее надежность данного решения, а знак логарифма определяет область принятия жесткого решения (см. рис. 6).

5. Пример итеративного декодирования сверточных турбокодов

Рассмотрим применение методов мягкого итеративного декодирования с использованием алгоритма MAP на примере сверточного турбодекодера. В качестве исходного кодера воспользуемся схемой рис. 1, ранее рассмотренной при исследовании методов сверточного турбокодирования. В качестве опорной схемы декодирования используем соответствующий кодеру составной итеративный декодер, изображенный на рис. 7. Пусть в качестве исходной информационной последовательности выступает двоичный вектор (31), используемый в разделе 1, данной работы:

$$[1 \ 1 \ 0 \ 0] \quad (31)$$

Как и ранее, условимся, что кодирование начинается в нулевом состоянии, т. е. оба кодера обнулены. Для формирования входной последовательности на втором коdere используется перемежитель размерностью 2×2 , который читает информационный блок по 2 бита по строкам, а выдает кодеру по столбцам. Т. о., перемеженная последовательность имеет вид:

$$[1 \ 0 \ 1 \ 0] \quad (32)$$

Исходя из диаграммы состояний рис. 2 (а) для каждого из составных кодеров турбокода, процесс кодирования представлен ранее в табл. 1. Результирующая выходная кодовая комбинация имеет вид:

$$[1 \ 1 \ 1 \ 1 \ 0 \ 1 \ 0 \ 0 \ 0 \ 0 \ 1 \ 1] \quad (33)$$

Для упрощения понимания примера, условимся, что передача информации между приемником и передатчиком осуществляется посредством биполярных импульсов $[+1; -1]$,

Таблица 2 — Порядок следования значений кодовых слов составного турбокодера

Назначение	Уровень			
Информационный бит, x_k^1	1.2	1.1	0.3	-0.6
Проверочный бит, y_k^1	0.9	0.5	-0.2	0.5
Проверочный бит, y_k^2	0.7	1.5	-0.8	0.9

потенциально кодирующих биты «1» и «0», соответственно. Сформированная модулятором передатчика последовательность принимает вид:

$$[+1 \ +1 \ +1 \ +1 \ -1 \ +1 \ -1 \ -1 \ -1 \ -1 \ +1 \ +1] \quad (34)$$

Выделенный приемником сигнал $r(t)$ является суммой исходного дискретного сообщения $s_i(t)$ и помехи в виде АБГШ $n(t)$:

$$r(t) = s_i(t) + n(t).$$

Шум канала передачи данных имеет нулевое среднее и дисперсию $\sigma^2 = 1$. Априорные вероятности принятия бита равного «1» или «0» одинаковы и равны $\pi^1 = \pi^2 = 1/2$. Пусть в результате воздействия помехи исходный биполярный сигнал (34) примет значения:

$$[1.2 \ 0.9 \ 0.7 \ 1.1 \ 0.5 \ 1.5 \ 0.3 \ -0.2 \ -0.8 \ -0.6 \ 0.5 \ 0.9] \quad (35)$$

Перед непосредственным декодированием необходимо определить принадлежность каждого бита принятой кодовой последовательности к составному кодеру сверточного турбокода. Исходя из общей кодовой скорости $r = 1/3$ турбокодера на рис. 3, кодовое слово состоит из одного систематического бита (информационного) и двух проверочных бит, с первого и второго кодера, соответственно. В табл. 2 представлена принадлежность бит к кодерам из принятого сигнального вектора. При параллельной схеме декодирования вначале будут декодироваться систематические и проверочные биты первого кодера, а затем систематические и проверочные биты второго кодера и т. д.

В соответствии с алгоритмом проведения декодирования MAP, процесс декодирования начинается с расчета первого сегмента ($k = 1$) кодовой решетки декодера. Исходя из условий, рассчитаем метрику ветвей $\delta_k^{i,m}$ на каждом участке k кодовой решетки:

$$\delta_k^{i,m} = \frac{1}{2} e^{x_k u_k^i + y_k v_k^{i,m}}, \quad (36)$$

где x_k и y_k представляют собой каналные значения информационного и соответствующего ему проверочного бита (см. табл. 2), а $u_k^{i,m}$ и $v_k^{i,m}$ обозначают уровни выходного сигнала, формируемые кодером исходя из значений выходных бит ($[0 : -1]$ и $[1 : +1]$). Основываясь на решетке переходов кодера, приведенной ранее на рис. 2 (b), метрики ветвей для первого перехода равны:

Таблица 3 — Расчет метрик ветвей декодера D_1

Переход	$\delta^{0,m=S_0}$	$\delta^{1,m=S_0}$	$\delta^{0,m=S_1}$	$\delta^{1,m=S_1}$	$\delta^{0,m=S_2}$	$\delta^{1,m=S_2}$	$\delta^{0,m=S_3}$	$\delta^{1,m=S_3}$
k=1–2	0.061	4.083	0.370	0.674	0.370	0.674	0.061	4.083
k=2–3	0.100	2.476	0.274	0.911	0.274	0.911	0.100	2.476
k=3–4	0.452	0.552	0.303	0.842	0.303	0.842	0.452	0.552
k=4–5	0.552	0.452	1.500	0.166	1.500	0.166	0.553	0.452

$$\delta_{k=1}^{0,m=S_0} = \frac{1}{2} \exp [1.2(-1) + 0.9(-1)] = 0.061$$

$$\delta_{k=1}^{1,m=S_0} = \frac{1}{2} \exp [1.2(+1) + 0.9(+1)] = 4.083$$

$$\delta_{k=1}^{0,m=S_1} = \frac{1}{2} \exp [1.2(-1) + 0.9(+1)] = 0.370$$

$$\delta_{k=1}^{1,m=S_1} = \frac{1}{2} \exp [1.2(+1) + 0.9(-1)] = 0.674$$

$$\delta_{k=1}^{0,m=S_2} = \frac{1}{2} \exp [1.2(-1) + 0.9(+1)] = 0.370$$

$$\delta_{k=1}^{1,m=S_2} = \frac{1}{2} \exp [1.2(+1) + 0.9(-1)] = 0.674$$

$$\delta_{k=1}^{0,m=S_3} = \frac{1}{2} \exp [1.2(-1) + 0.9(-1)] = 0.061$$

$$\delta_{k=1}^{1,m=S_3} = \frac{1}{2} \exp [1.2(+1) + 0.9(+1)] = 4.083$$

Полученные значения изображены на первом переходе ($k = 1$) кодовой решетки на рис. 9 (а). Аналогичным образом рассчитываются метрики ветвей для последующих переходов $k = 2 \dots 5$. Расчитанные значения метрик ветвей декодера для каждого перехода приведены в табл. 3 и обозначены на соответствующих переходах рис. 9.

На следующем этапе алгоритма декодирования MAP, необходимо произвести расчет метрик прямого прохода по решетке декодера. В соответствии с ранее рассмотренным алгоритмом декодирования, выражение для расчета метрики прямого прохода:

$$\alpha_{k+1}^m = \sum_{j=0}^1 \delta_k^{j,b(j,m)} \alpha_k^{b(j,m)} \quad (37)$$

В следствии того, что кодирование информационной последовательности начинается в исходном S_0 (обнуленном состоянии), метрика прямого прохода на первом шаге ($k = 1$)

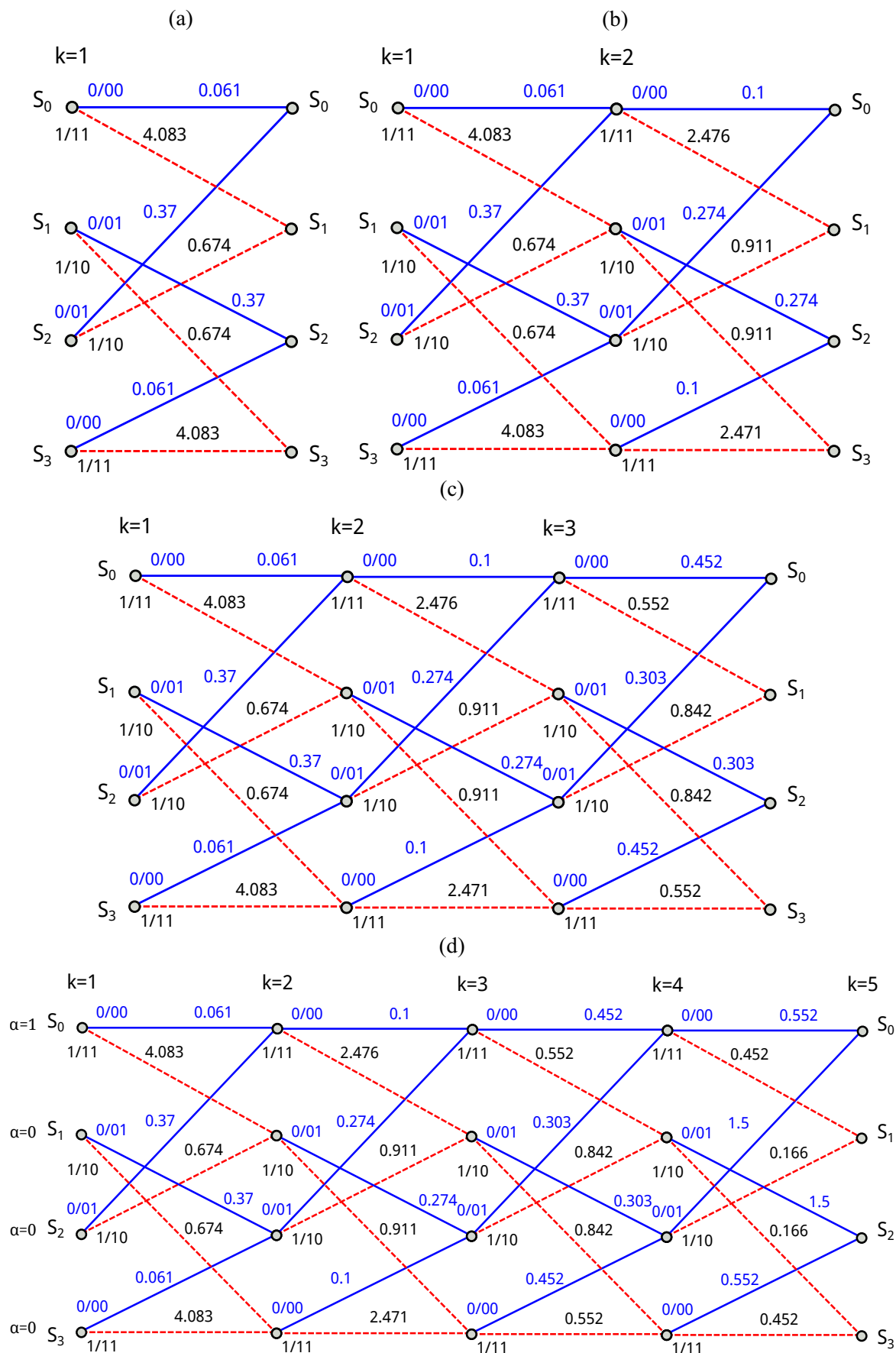
Рисунок 9 – Метрики ветвей δ решетки сверточного декодера D_1

Таблица 4 — Расчет метрик прямого прохода декодера D_1

Момент	$\alpha^{m=S_0}$	$\alpha^{m=S_1}$	$\alpha^{m=S_2}$	$\alpha^{m=S_3}$
k=1	1	0	0	0
k=2	0.061	4.083	0	0
k=3	0.0061	0.151	1.118	3.719
k=4	0.341	0.944	1.726	2.18
k=5	2.77	0.44	2.619	1.14

для S_0 равна 1, а для всех остальных состояний нулю. Далее, для расчета метрики прямого прохода в последующих состояниях необходимо учитывать значение метрик ветвей входящих в рассматриваемый узел решетки и значение метрики прямого прохода на предыдущем шаге. Основываясь на выражении (37), решетке декодера на рис. 9 (d) и значениях метрик ветвей δ полученных на предыдущем этапе декодирования, метрики прямого прохода для момента $k = 2$ равны:

$$\alpha_{k=2}^{m=S_0} = 0.061(1) + 0.37(0) = 0.061 \quad (38)$$

$$\alpha_{k=2}^{m=S_1} = 4.083(1) + 0.674(0) = 4.083 \quad (39)$$

$$\alpha_{k=2}^{m=S_2} = 0.37(0) + 0.061(0) = 0 \quad (40)$$

$$\alpha_{k=2}^{m=S_3} = 0.674(0) + 4.083(0) = 0 \quad (41)$$

Т.к. метрики прямого прохода для состояний S_1, S_2 и S_3 ($k = 1$) равны нулю, то для момента времени $k = 2$ имеют значения только ветви исходящие из состояния S_0 ($k = 1$), поскольку остальные члены суммы будут равны нулю. Аналогично, для момента времени $k = 3$ метрики прямого прохода учитывают значения метрик ветвей входящих в заданный узел и значения метрик прямого прохода для $k = 2$. Расчет метрик прямого прохода для $k = 3$ представлен в выражениях (42–45), а для оставшихся моментов времени в табл. 4. На рис. 10 изображена решетка декодера и рассчитанные значения метрик прямого прохода для каждого узла решетки.

$$\alpha_{k=3}^{m=S_0} = 0.1(0.061) + 0.274(0) = 0.0061 \quad (42)$$

$$\alpha_{k=3}^{m=S_1} = 2.476(0.061) + 0.911(0) = 0.151 \quad (43)$$

$$\alpha_{k=3}^{m=S_2} = 0.274(4.083) + 0.1(0) = 1.118 \quad (44)$$

$$\alpha_{k=3}^{m=S_3} = 0.911(4.083) + 2.476(0) = 3.719 \quad (45)$$

Используя значения метрик прямого прохода α_k^m , становится возможным вычислить метрики обратного прохода β_k^m по решетке декодера:

Таблица 5 — Расчет метрик обратного прохода декодера D_1

Момент	$\beta^{m=S_0}$	$\beta^{m=S_1}$	$\beta^{m=S_2}$	$\beta^{m=S_3}$
k=5	1	0	0	0
k=4	0.552	0	1.5	0
k=3	0.249	0.454	0.167	0.678
k=2	1.149	0.663	0.481	1.695
k=1	2.777	1.32	0.871	6.95

$$\beta_k^m = \sum_{j=0}^1 \delta_k^{j,m} \beta_{k+1}^{f(j,m)} \quad (46)$$

Процесс вычисления метрик обратного прохода начинается с конца ($k = 5$) кодовой решетки рис. 9 (d). Согласно диаграмме состояний (рис. 2) первого кодера составного турбокодера, изображенного на рис. 3, процесс кодирования исходной последовательности завершается в обнуленном состоянии кодера (S_0 при $R_1^1 = 0, R_2^1 = 0$, табл. 1). Исходя из данного обстоятельства, метрика обратного прохода $\beta_{k=5}^{m=S_0}$ в момент времени $k = 5$ для состояния S_0 равна 1, а для всех остальных состояний кодовой решетки нулю. Необходимо подчеркнуть, что хотя в данном примере не использовалось принудительное завершение (обнуление) кодеров составного турбокодера, подобранный в примере информационный вектор привел к обнулению внешнего кодера. Данное положение несколько упрощает декодирование для внешнего декодера. Однако, согласно табл. 1, внутренний кодер не перешел в нулевое состояние по окончанию процедур кодирования, что необходимо учитывать далее, при декодировании кодовых слов внутреннего кодера.

Т. о., используя метрику обратного прохода для исходного состояния и значения метрик ветвей в обратном направлении, рассчитаем β_k^m для всех узлов решетки. Для момента времени $k = 4$, β_k^m составляет:

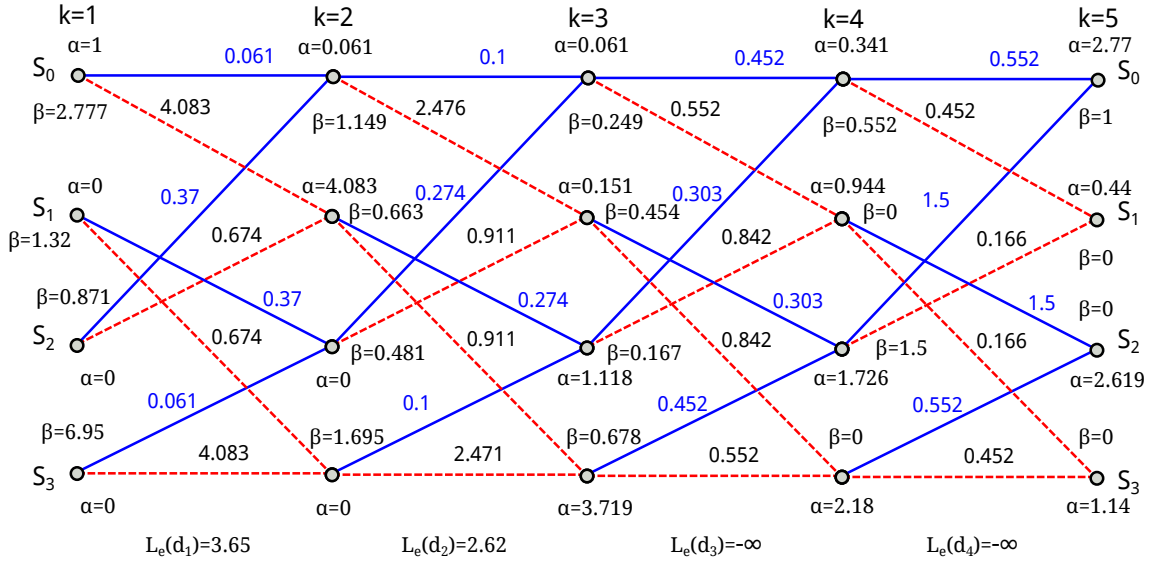
$$\beta_{k=4}^{m=S_0} = 0.552(1) + 0.452(0) = 0.552 \quad (47)$$

$$\beta_{k=4}^{m=S_1} = 1.5(0) + 0.166(0) = 0 \quad (48)$$

$$\beta_{k=4}^{m=S_2} = 1.5(1) + 0.166(0) = 1.5 \quad (49)$$

$$\beta_{k=4}^{m=S_4} = 0.552(0) + 0.452(0) = 0 \quad (50)$$

Рассчитанные значения β_k^m для оставшихся узлов решетки приведены в табл. 5, а на рис. 10 изображена решетка декодера с нанесенными значениями метрик обратного прохода для каждого узла решетки.

Рисунок 10 — Метрики прямого α и обратного β прохода декодера D_1

Обладая значениями трех метрик для каждого состояния решетки декодера, можно произвести расчет логарифмического отношения функций правдоподобия $L_e(d_k)$ по декодируемым символам:

$$L_e(d_k) = \ln \left(\frac{\sum_m \alpha_k^m \delta_k^{1,m} \beta_{k+1}^{f(1,m)}}{\sum_m \alpha_k^m \delta_k^{0,m} \beta_{k+1}^{f(0,m)}} \right). \quad (51)$$

Рассчитаем отношение правдоподобия для первого участка кодовой решетки $k = 1$, учитывая значения метрик ветвей (табл. 3), метрик прямого прохода для данных узлов (табл. 4) и значения метрик обратного направления для узлов $k + 1$ (табл. 5):

$$\begin{aligned} L_e(d_1) &= \ln \left(\frac{(1)(4.083)(0.663) + (0)(0.674)(1.695) + (0)(0.674)(0.663) + (0)(4.083)(1.695)}{(1)(0.061)(1.149) + (0)(0.37)(0.481) + (0)(0.37)(1.149) + (0)(0.061)(0.481)} \right) = \\ &= \ln \left(\frac{(1)(4.083)(0.663)}{(1)(0.061)(1.149)} \right) = 3.65 \end{aligned}$$

Опуская для краткости нулевые слагаемые и промежуточные вычисления произведений, значение второго и последующих декодируемых бит:

$$\begin{aligned} L_e(d_2) &= \ln \left(\frac{(0.061)(2.476)(0.454) + (4.083)(0.911)(0.678)}{(0.061)(0.1)(0.249) + (4.083)(0.274)(0.167)} \right) = 2.62 \\ L_e(d_3) &= \ln \left(\frac{0}{2.78} \right) = -\infty \\ L_e(d_4) &= \ln \left(\frac{0}{0.188} \right) = -\infty \end{aligned}$$

Таблица 6 — Значения входов декодера D_2

Назначение	Уровень			
Информационный бит, x_k^2	1.2	0.3	1.1	-0.6
Проверочный бит, y_k^2	0.7	-0.8	1.5	0.9
Априорная информация, $L_e^{12}(d_k)$	3.65	$-\infty$	2.62	$-\infty$

Таблица 7 — Расчет метрик ветвей декодера D_2

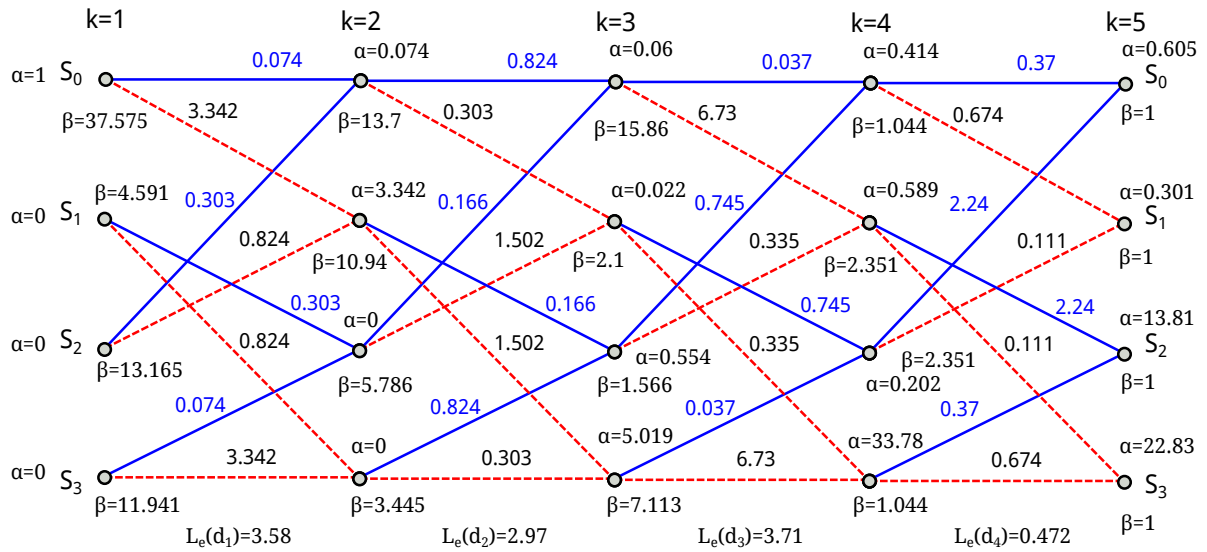
Переход	$\delta^{0,m=S_0}$	$\delta^{1,m=S_0}$	$\delta^{0,m=S_1}$	$\delta^{1,m=S_1}$	$\delta^{0,m=S_2}$	$\delta^{1,m=S_2}$	$\delta^{0,m=S_3}$	$\delta^{1,m=S_3}$
k=1-2	0.074	3.342	0.303	0.824	0.303	0.824	0.074	3.342
k=2-3	0.824	0.303	0.166	1.502	0.166	1.502	0.824	0.303
k=3-4	0.037	6.73	0.745	0.335	0.745	0.335	0.037	6.73
k=4-5	0.37	0.674	2.24	0.111	2.24	0.111	0.37	0.674

Полученные значения отмечены на соответствующих участках кодовой решетки рис. 10. Необходимо отметить, что при машинном вычислении, наличие бесконечных величин, полученных для бит d_3 и d_4 , в принципе не требует дальнейшей обработки, поскольку задает максимально вместимое отрицательное число системы, на которой проводится вычисление. При машинном компьютерном расчете, например в распространенных прикладных пакетах инженерных систем расчета, значение « $-\infty$ » представляет собой число $[-9 \cdot 10^{-19}]$, которое в данном случае, однозначно определяет результат декодирования. Однако в целях демонстрации и анализа методов итеративного декодирования продолжим процедуры обработки, для увеличения надежности результатов декодирования для бит d_1 и d_2 .

Вновь обратимся к схеме декодера на рис. 7. На данном этапе декодирование кодовых слов внешнего декодера D_1 завершено, и результаты декодирования можно передать на внутренний декодер D_2 в формате априорной информации $L_e^{12}(d_k)$. Следует особенно подчеркнуть необходимость перемежения априорной информации для соответствия исходной (перемеженной) информационной последовательности (32) внутреннего декодера D_2 . В табл. 6 представлены значения всех входов декодера D_2 .

Алгоритм вычисления метрик ветвей декодера D_2 , полностью аналогичен рассмотренному выше методу расчета метрик ветвей декодера D_1 . Результаты расчета приведены в табл. 7 и отмечены на соответствующих переходах кодовой решетки на рис. 11.

Используя массив метрик ветвей, как и в предыдущем случае для декодера D_1 , вычисляются метрики состояний прямого прохода по кодовой решетке α_k^m . Значения метрик состояний при прямом проходе для декодера D_2 приведены в табл. 8 и отмечены на узлах кодовой решетки.

Рисунок 11 — Метрики ветвей δ и состояний α, β декодера D_2 Таблица 8 — Расчет метрик прямого прохода декодера D_2

Момент	$\alpha^{m=S_0}$	$\alpha^{m=S_1}$	$\alpha^{m=S_2}$	$\alpha^{m=S_3}$
k=1	1	0	0	0
k=2	0.074	3.342	0	0
k=3	0.06	0.022	0.554	5.019
k=4	0.414	0.589	0.202	33.78
k=5	0.605	0.301	13.81	22.83

Как было указано ранее, при кодировании информационной последовательности не происходило принудительного обнуления составных кодеров. В связи с этим, приемник не имеет возможности вычислить точное значение обратной метрики состояний для декодера D_2 в конечный момент времени ($k = 5$). Исходя из данного условия, предполагается что все конечные состояния кодера равновероятны, т.е. с одинаковой вероятностью кодер может находиться в любом из 2^{M_c} состояний. В зависимости от реализаций, обратная метрика для конечного момента времени (в данном случае $k = 5$) может быть приравнена к прямой метрике состояний в конечный момент времени, т.е. $\beta_N^m = \alpha_n^m$. Однако, существует и более простой способ определения обратной метрики состояний, который при равновероятности состояний кодера, определяет обратную метрику для конечного момента времени равной 1, т.е. $\beta_N^m = 1$.

Необходимо отметить, что оба указанных метода вносят определенную погрешность в результат декодирования, что подчеркивает актуальность поиска методов быстрого общего обнуления внутренних кодеров составных турбокодеров. При условии, что оба кодера турбокодера были бы обнулены и закончили кодирование в состоянии S_0 , определение

Таблица 9 – Расчет метрик обратного прохода декодера D_2

Момент	$\beta^{m=S_0}$	$\beta^{m=S_1}$	$\beta^{m=S_2}$	$\beta^{m=S_3}$
k=5	1	1	1	1
k=4	1.044	2.351	2.351	1.044
k=3	15.86	2.1	1.565	7.113
k=2	13.7	10.94	5.786	3.445
k=1	37.575	4.591	13.165	11.941

значений метрик состояний при обратном проходе по кодовой решетке не составило бы труда, поскольку декодеру априори известно конечное состояние (S_0) любого составного кодера (D_1 , D_2).

Используя предположение, что метрики состояний для $k = 5$ при обратном проходе по кодовой решетке равны 1, результаты расчета для всей решетки приведены в табл. 9, а также отмечены на узлах кодовой решетки рис. 11.

Наконец, рассчитаем отношение правдоподобия для всех участков кодовой решетки, учитывая значения метрик ветвей (табл. 7), метрик прямого прохода для данных узлов (табл. 8) и значения метрик обратного направления для узлов $k + 1$ (табл. 9). Опустив некоторые нулевые сомножители и произведя предварительные расчеты произведений, получим:

$$L_e(d_1) = \ln \left(\frac{(1)(3.342)(10.94)}{(1)(0.074)(13.7)} \right) = 3.58$$

$$L_e(d_2) = \ln \left(\frac{(0.074)(0.303)(2.1) + (3.342)(1.502)(7.113)}{(0.074)(0.824)(15.86) + (3.342)(0.166)(1.566)} \right) = 2.97$$

$$L_e(d_3) = \ln \left(\frac{0.949 + 0.007 + 0.436 + 35.26}{0.002 + 0.038 + 0.43 + 0.43} \right) = 3.77$$

$$L_e(d_4) = \ln \left(\frac{23.13}{14.42} \right) = 0.472$$

После декодирования кодовых слов декодера D_2 существует два сценария развития дальнейших действий. В первом случае, в соответствии со схемой декодера на рис. 7, полученные приращения надежности $L_e(d_k)$ могут быть переданы далее на новую итерацию декодирования внешнему декодеру в виде априорной информации $L(d_i)$. Будет повторен весь рассмотренный цикл декодирования для обоих декодеров. Наконец, по завершению заданного количества (Q) итераций декодирования, декодеру турбокода необходимо вынести окончательное жесткое решение.

Предположим что, завершение декодирования наступает в конце первой итерации ($Q = 1$), после декодирования кодовых слов декодерами D_1 и D_2 . В этом случае для расчета мягкого

выхода итерации декодирования необходимо воспользоваться заключительным выражением (16). Для вычисления мягкого выхода турбодекодера, согласно выражению (16), следует учесть вклад априорной информации от внешнего декодера D_1 (табл. 6), и вклад собственных результатов декодирования декодера D_2 :

$$L(\hat{d}_1) = 9.63$$

$$L(\hat{d}_2) = 9.43$$

$$L(\hat{d}_3) = -\infty$$

$$L(\hat{d}_4) = -\infty$$

Основываясь на принципе принятия жестких решений, согласно выражению (12), результат декодирования принятого кодового вектора после выполнения полной первой итерации декодирования равен $[1\ 1\ 0\ 0]$, и полностью совпадает с исходным информационным вектором (31) введенным в кодер, что свидетельствует об отсутствии ошибок декодирования.

Рассмотренный способ вероятностного декодирования сверточного турбокода, основанный на алгоритме MAP, помимо достоинств, обладает одним существенным недостатком в случае использования его для большого числа декодируемых кодовых векторов. С увеличением количества декодируемой информации возрастает число операций умножения и деления над метриками кодовой решетки для каждого бита. Среднее количество операций умножения для вычисления значений метрик состояний для обоих направлений прохода и метрик ветвей зависит от размеров памяти кодера, и, с увеличением ее объема, так же возрастает. В связи с конвейерной обработкой данных также возрастает и время декодирования, что может оказать существенное влияние на быстродействие телекоммуникационного приложения в целом. С другой стороны, для снижения вычислительной нагрузки на систему обработки и передачи данных при декодировании сверточного турбокода могут быть использованы методы приближенного вычисления метрик решетки кодера, позволяющие ускорить процесс вычисления (декодирования) за счет частичного снижения точности результата декодирования.

Часть III

Указания к выполнению курсовой работы

6. Цели курсовой работы

Задания курсовой работы преследуют цели углубленного изучения принципов реализации, методов и алгоритмов взаимодействия цифровых систем обработки и передачи информации, основанных на применении сверточных турбокодов и циклическом итеративном декодировании. В качестве модели канала передачи данных используется непрерывный канал с помехой в виде АБГШ, что позволяет приблизить условия имитационного моделирования и результаты компьютерных расчетов к условиям функционирования распространенных наземных и спутниковых систем связи и обработки информации.

7. Задачи курсовой работы

Для достижения поставленных в рамках курсовой работы целей, требуется решить ряд задач.

7.1. Кодирование исходного символа

В соответствии с исходными данными (см. табл. 16) необходимо произвести избыточное кодирование заданного текстового символа представленного в табл. 10. Для всех вариантов длина текстового символа в двоичной форме равна $L = 8$ бит.

В качестве кодирующей схемы следует использовать кодер сверточного турбокода изображенный на рис. 3. Двоичная комбинация подается на вход кодера в порядке следования разрядов — начиная с младшего (крайнего правого) и заканчивая старшим разрядом (крайним левым). Процесс кодирования начинается в нулевом состоянии (кодеры обнулены).

Для формирования входной последовательности на втором кодере необходимо использовать перемежитель, который читает информационные символы по 4 бита по строкам, а передает кодеру по столбцам:

$$[x_7 \ x_6 \ x_5 \ x_4 \ x_3 \ x_2 \ x_1 \ x_0] = \begin{bmatrix} x_0 & x_1 & x_2 & x_3 \\ x_4 & x_5 & x_6 & x_7 \end{bmatrix} = [x_0 \ x_4 \ x_1 \ x_5 \ x_2 \ x_6 \ x_3 \ x_7]$$

Руководствуясь диаграммой состояний для каждого кодера (рис. 2 (а)) следует закодировать информационную последовательность, учитывая порядок следования бит на первый и второй кодер, соответственно.

Если по исходному заданию необходимо произвести обнуление указанного кодера, то исходную информационную последовательность следует дополнить обнуляющими битами, в

результате кодирования которых кодер перейдет в исходное состояние ($S_0 = 00$, т. е. регистры кодера обнулены). Подбор обнуляющей комбинации входит в индивидуальное задание данной части курсовой работы. Необходимо отметить, что обнуление одного из составных кодеров приведет к увеличению длины (L) исходной информационной последовательности на число обнуляющих бит. Например, если для обнуления кодера потребовалось 3 дополнительных бита, то длина исходной информационной последовательности составит $L = 11$ бит.

Результаты кодирования, состояния регистров и исходные информационные символы должны быть отражены в форме табл. 1, а полученная кодовая комбинация имеет вид $x = (x_0^{(0)}, x_0^{(1)}, x_0^{(2)}, \dots, x_{L-1}^{(0)}, x_{L-1}^{(1)}, x_{L-1}^{(2)})$.

7.2. Передача и прием данных

Завершив процедуру помехоустойчивого кодирования исходных данных следует заполнить первый и второй столбцы табл. 11.

По условиям курсовой работы кодовая комбинация подвергается потенциальному кодированию биполярными импульсами со значениями сигнала $[+1, -1]$, соответствующими исходным битам $[1, 0]$. Полученные сигналы передаются по непрерывному каналу с воздействием помехи в виде аддитивного белого гауссова шума (АБГШ) (см. раздел 3).

Значения принятого сигнала определяются с помощью разработанной на кафедре ОПДС программной модели канала передачи данных с помехой АБГШ ([ссылка на программу модели на сайте кафедры](#)). Программная модель канала передачи данных требует указания длины и значения кодовой комбинации в качестве аргумента программы. Например, если в результате кодирования получена кодовая комбинация 01101101 длиной 8 бит, вызов программы будет выглядеть следующим образом:

Листинг 1 — Программная модель канала АБГШ

```
> awgn 8 01101101
```

В результате исполнения программы (т. е. имитации передачи кодовой комбинации по каналу АБГШ) будут получены значения сигналов на приемнике декодера:

Листинг 2 — Результат передачи данных

```
Coder:
01101101

Modulator:
-1.0 1.0 1.0 -1.0 1.0 1.0 -1.0 1.0

Receiver:
-0.422 0.578 1.068 -1.397 0.942 1.948 -1.211 1.436
```

Полученные данные необходимо внести в табл. 11 и представить на проверку преподавателю.

Таблица 10 — Набор символов в двоичном виде

Номер	Символ	Код
1	A	01000001
2	B	01000010
3	C	01000011
4	D	01000100
5	E	01000101
6	F	01000110
7	G	01000111
8	H	01001000
9	I	01001001
10	J	01001010
11	K	01001011
12	L	01001100
13	M	01001101
14	N	01001110
15	O	01001111
16	P	01010000
17	Q	01010001
18	R	01010010
19	S	01010011
20	T	01010100
21	U	01010101
22	V	01010110
23	W	01010111
24	X	01011000
25	Y	01011001
26	Z	01011010

Таблица 11 — Результаты передачи кодовой комбинации по каналу АБГШ

Бит	Переданный сигнал	Принятый сигнал
0	-1	-0.994
1	+1	1.211
0	-1	-0.719
1	+1	0.624
1	+1	1.007
...

Таблица 12 — Порядок следования значений кодовых слов составного турбокодера

Назначение	Уровень		
Информационный бит, x_k^1			...
Проверочный бит, y_k^1			...
Проверочный бит, y_k^2			...

Таблица 13 — Расчет метрик ветвей декодера D_1

Переход	$\delta_{0,m=S_0}$	$\delta_{1,m=S_0}$	$\delta_{0,m=S_1}$	$\delta_{1,m=S_1}$	$\delta_{0,m=S_2}$	$\delta_{1,m=S_2}$	$\delta_{0,m=S_3}$	$\delta_{1,m=S_3}$
k=1–2								...
k=2–3								...
k=3–4								...
k=4–5								...
...								...

7.3. Итеративное декодирование

Дальнейшие этапы и процедуры декодирования полностью основываются на материале раздела 4 и примера декодирования приведенного в разделе 5.

В соответствии с примером раздела 5, в первую очередь необходимо определить принадлежность каждого бита принятой кодовой последовательности к составному декодеру сверточного турбокода. Следует напомнить, что при кодовой скорости $r = 1/3$ турбокодера (рис. 3), кодовое слово состоит из одного информационного бита и двух проверочных. Полученные данные следует внести в табл. 12.

Далее, в соответствии с алгоритмом, процесс декодирования начинается с расчета первого сегмента ($k = 1$) кодовой решетки декодера. Исходя из условий, рассчитывается метрика ветвей $\delta_k^{i,m}$ для каждого участка k кодовой решетки. Расчитанные данные записываются в форме таблицы 13 и обозначаются на решетке декодера (см. рис 9).

На следующем этапе следует расчет метрик прямого прохода α_{k+1}^m по решетке декодера в соответствии с выражением (37). Результаты расчета вносятся в таблицу 14.

Используя значения метрик прямого прохода α_k^m , необходимо вычислить метрику обратного прохода β_k^m (см. выражение (46)). Значения β_k^m для всех узлов решетки записываются в табл. 15, и отмечаются на рисунке решетки декодера (см. рис. 10).

На завершающем этапе декодирования, для декодера D_1 следует произвести расчет логарифмического отношения функций правдоподобия $L_e(d_k)$ по декодируемым символам

Таблица 14 – Расчет метрик прямого прохода декодера D_1

Момент	$\alpha^{m=S_0}$	$\alpha^{m=S_1}$	$\alpha^{m=S_2}$	$\alpha^{m=S_3}$
k=1				...
k=2				...
k=3				...
k=4				...
k=5				...
...				...

Таблица 15 – Расчет метрик обратного прохода декодера D_1

Момент	$\beta^{m=S_0}$	$\beta^{m=S_1}$	$\beta^{m=S_2}$	$\beta^{m=S_3}$
k=5				
k=4				
k=3				
k=2				
k=1				
...				...

согласно выражению (51):

$$\begin{aligned}L_e(d_1) &= \dots \\L_e(d_2) &= \dots \\L_e(d_3) &= \dots \\L_e(d_4) &= \dots \\L_e(d_5) &= \dots \\L_e(d_6) &= \dots \\&\dots\end{aligned}\tag{52}$$

Т. о. декодирование кодовой комбинации на декодере D_1 завершается. Далее, используя информационные биты x_k^1 и проверочные биты y_k^2 , производится декодирование комбинации на декодере D_2 . Все этапы декодирования выполняются полностью аналогично процедурам декодирования рассмотренным выше.

После завершения полной итерации декодирования (на декодере D_1 и декодере D_2), выполняется расчет мягкого выхода. В соответствии с выражением (16) необходимо учесть вклад априорной информации от внешнего декодера D_1 (табл. 6), вклад собственных результатов декодирования декодера D_2 и значение канального измерения сигнала x ($(L_e(x))$), полученное приемником. Основываясь на принципе принятия жестких решений (выражение (12)), необходимо записать результат декодирования принятого кодового вектора после выполнения полной итерации декодирования.

Используя табл. 10, следует перевести декодированную последовательность в символьную форму (текст) и убедиться в отсутствии ошибок декодирования. В случае наличия ошибки декодирования, следует указать позицию ошибки, символы полученные в результате декодирования и исходные символы передаваемые кодеру на приемнике.

8. Форма отчета

9. Варианты заданий

Таблица 16 — Исходные данные

Вариант	Передаваемые символы	Обнуляемый кодер
1	AZ	1
2	BY	2
3	CX	2
4	DW	2
5	EV	1
6	FU	1
7	GT	2
8	HS	1
9	IR	1
10	JQ	2
11	KP	2
12	LO	1
13	MN	2
14	AB	2
15	CD	1
16	EF	1
17	GH	1
18	IJ	2
19	KL	1
20	MN	1
21	OP	—
22	QR	1
23	ST	2
24	UV	2
25	WX	1
26	YZ	2
27	ZZ	1
28	YY	2
29	XX	1
30	WW	—
31	VV	—
32	UU	1
33	TT	2
34	SS	1
35	RR	2
36	QQ	1

Таблица 16 – Исходные данные

Вариант	Передаваемые символы	Обнуляемый кодер
37	PP	2
38	OO	2
39	NN	1
40	MM	2
41	LL	–
42	KK	1
43	JJ	1
44	II	1
45	HH	1
46	GG	2
47	FF	2
48	EE	2
49	DD	1
50	CC	1
51	BB	2
52	AA	1
53	ZY	2
54	YX	–
55	XW	1
56	WV	2
57	VU	1
58	UT	2
59	TS	1
60	SR	2
61	RQ	2
62	QP	1
63	PO	–
64	ON	1
65	NM	1
66	ML	2
67	LK	2
68	KJ	2
69	JI	2
70	IH	–
71	HG	1
72	GF	1
73	FE	1
74	ED	1

Таблица 16 – Исходные данные

Вариант	Передаваемые символы	Обнуляемый кодер
75	DC	2
76	CB	1
77	BA	2
78	AC	2
79	BD	1
80	EG	–
81	FH	1
82	IK	1
83	JL	2
84	MO	2
85	NP	2
86	QS	1
87	RT	2
88	SU	1
89	TV	1
90	WY	–
91	XZ	1
92	AM	2
93	BN	2
94	CO	1
95	DP	1
96	EQ	–
97	FR	1
98	GS	2
99	HT	1
00	IU	–

Часть IV

Приложение

Определение вероятности принятия сигнала соответствующего «0» или «1», базируется на основе проверки гипотез по теореме Байеса. Теорема Байеса позволяет получить условную вероятность $P(A|B)$ из условной вероятности $P(B|A)$ следующим образом:

$$P(A|B) = \frac{P(B|A)P(A)}{P(B)}. \quad (53)$$

Существует несколько форм записи теоремы Байеса в приложении к теории связи. В дискретной форме теорему Байеса можно рассматривать как описание эксперимента, в котором задействованы значение принятого сигнала и априорные значения о классах сигнала, к одному из которых может принадлежать принятый сигнал:

$$P(s_i|z_j) = \frac{P(z_j|s_i)P(s_i)}{P(z_j)}, \quad (54)$$

где s_i — есть i значение сигнала из заранее обозначенного набора M , а z_j — принятый (зарегистрированный) сигнал. Член $P(z_j)$ представляет собой вероятность принятого сигнала со значением z_j в пространстве всех возможных значений из набора M :

$$P(z_j) = \sum_{i=1}^M P(z_j|s_i)P(s_i). \quad (55)$$

Вероятность появления i значения сигнала считается априорной вероятностью $P(s_i)$. При регистрации конкретного значения z_j из плотности условной вероятности $P(z_j|S_i)$ находится правдоподобие принадлежности принятого сигнала z_j к известному набору сигналов. После регистрации вычисляется апостериорная вероятность $P(s_i|z_j)$, уточняющая априорную вероятность.

Однако, на практике, интерес представляет смешанная форма теоремы Байеса, поскольку значения принятого сигнала принадлежат к непрерывного диапазону значений, из-за воздействия на телекоммуникационный канал помех типа АБГШ. В этой форме, теорема Байеса содержит плотность вероятности $p(z)$ с непрерывными значениями, а не дискретными:

$$P(s_i|z) = \frac{P(z|s_i)P(s_i)}{p(z)}. \quad (56)$$

Плотность вероятности $p(z)$ определяется как

$$p(z) = \sum_{i=1}^M p(z|s_i)P(s_i), \quad (57)$$

где $p(z|s_i)$ — плотность условной вероятности принятого сигнала z , принадлежащего к значению исходного сигнала s_i .

Принцип принятия решения, используемый при декодировании турбокодов, основывается на применении гипотез утверждающих, что при регистрируемом значении z , принят сигнал s_1 , если апостериорная вероятность $P(s_1|z)$ больше, чем апостериорная вероятность $P(s_2|z)$, иначе верно обратное утверждение:

$$P(s_1|z) \underset{s_2}{\overset{s_1}{\geq}} P(s_2|z). \quad (58)$$

Используя форму теоремы Байеса (56) для непрерывного значения z , получим выражение для принятия решений, основанное на плотности вероятности (правдоподобия):

$$P(z|s_1)P(s_1) \underset{s_2}{\overset{s_1}{\geq}} P(z|s_2)P(s_2). \quad (59)$$

Перенеся условные вероятности в левую часть выражения, получим отношение функций правдоподобия:

$$\frac{P(z|s_1)}{P(z|s_2)} \underset{s_2}{\overset{s_1}{\geq}} \frac{P(s_2)}{P(s_1)}. \quad (60)$$

Выражение (60) является критерием отношения функций правдоподобия. Т.е. принятие решения о регистрируемом символе, основывается на сравнении принятого сигнала и значения допустимого порога. Поскольку сравнение основывается на выборе сигнала с максимальной апостериорной вероятностью, то такой критерий называется критерием максимума апостериорной вероятности (МАР).

На практике, значение априорных вероятностей принимается равновероятными. При равновероятных априорных вероятностях принятого значения сигнала, критерий принятия решения называется критерием максимального правдоподобия. При этом выражение (60) принимает форму:

$$\frac{P(z|s_1)}{P(z|s_2)} \underset{s_2}{\overset{s_1}{\geq}} 1. \quad (61)$$

В итерационном процессе декодирования турбокода, на первой итерации априорные значения вероятностей для внешнего декодера равновероятны (см. рис. 7). Однако, для внутреннего декодера, значение априорной вероятности (до декодирования) принимает значение апостериорной вероятности (после декодирования), полученное внешним декодером в ходе текущей итерации. Процесс обмена внешней информацией выполняется до окончания заданного числа циклов декодирования.