ЛАБОРАТОРНАЯ РАБОТА № 4

Декодирование сверточных кодов по максимуму правдоподобия. Алгоритм Витерби

Сверточные коды часто используются как внутренние коды в каскадных схемах кодирования. От эффективности их декодирования в большой степени зависит надежность системы в целом. Поэтому для их декодирования необходимо использовать трудоемкое, но оптимальное в смысле вероятности ошибки правило – декодирование по максимуму правдоподобия. Решающим преимуществом сверточных кодов перед блоковыми кодами является возможность применения весьма эффективной процедуры декодирования по максимуму правдоподобия – алгоритма Витерби. Данная лабораторная работа посвящена изучению алгоритма Витерби, способов его практической реализации и анализу его эффективности

4.1. Декодирование по максимуму правдоподобия

Рассмотрим сначала блоковый код $C = \{\mathbf{x}_m, m = 1,..., M\}$, словами которого являются последовательности $\mathbf{x}_m = (x_{m1},..., x_{mn})$ некоторого дискретного алфавита X. Обозначим через Y множество символов, наблюдаемых на выходе канала. Это множество может быть дискретным или непрерывным. Для определенности вначале мы будем считать алфавит Y дискретным. Предположим, что для передачи некоторого сообщения с номером m использовано кодовое слово $\mathbf{x}_m$ и в результате передачи по каналу принята последовательность $\mathbf{y}$. Задача декодера состоит в принятии решения о том, какое из M кодовых слов было передано.

Правило декодирования задается разбиением множества $Y^n = \{\mathbf{y} = (y_1,..., y_n)\}$ выходных последовательностей канала на непересекающиеся подмножества R_m такие, что при появлении на выходе канала последовательности $\mathbf{y} \in R_m$ принимается решение в пользу кодового слова $\mathbf{x}_m$. Множества R_m называют *решающими областями.*

Формирование решающих областей определяется критерием принятия решений, который в свою очередь зависит от критерия качества системы связи. Для многих задач удовлетворительным критерием качества можно считать вероятность ошибки декодирования P_e. Для блоковых кодов эта вероятность определяется как средняя по всем кодовым словам вероятность принять решение в пользу кодового слова $\mathbf{x}_{m'}$ при условии, что передавалось кодовое слово $\mathbf{x}_{m'}$, $m' \neq m$. Известно, что при равновероятных кодовых словах минимальная вероятность ошибки достигается при *декодировании по критерию максимального правдоподобия* (МП). Правило формирования решающих областей при этом имеет вид

$$R_m = \{\mathbf{y} : p(\mathbf{y} \mid \mathbf{x_m}) \geq p(\mathbf{y} \mid \mathbf{x_{m'}}), m' \neq m\},$$

где $p(\mathbf{y} \mid \mathbf{x})$ представляет собой условную вероятность появления на выходе канала последовательности $\mathbf{y}$ при передаче последовательности $\mathbf{x}$. По

определению, для заданной выходной последовательности $\mathbf{y}$ декодер МП принимает решение в пользу того кодового слова, для которого максимальна условная вероятность появления $\mathbf{y}$ на выходе канала. Вероятность $p(\mathbf{y} \mid \mathbf{x}_m)$ называют функцией правдоподобия кодового слова $\mathbf{x}_m$.

Декодирование по МП требует точного описания модели канала, т.е. правила вычисления вероятностей $p(\mathbf{y} \mid \mathbf{x})$. Канал называют стационарным, если вероятности $p(\mathbf{y} \mid \mathbf{x})$ не зависят от положения последовательностей на оси времени, т.е. однозначно определяются конкретными значениями символов, образующих последовательности $\mathbf{x}$ и $\mathbf{y}$. Канал называется *каналом без памяти,* если для всех n для любой пары последовательностей $\mathbf{x} = (x_1,...,x_n)$, $\mathbf{y} = (y_1,...,y_n)$ имеет место равенство

$$p(\mathbf{y} \mid \mathbf{x}) = \prod_{i=1}^{n} p(y_i \mid x_i). \tag{4.1}$$

Стационарный дискретный канал без памяти называется *дискретным постоянным каналом* (ДПК). Такой канал описывается переходными вероятностями $\{p(y \mid x), x \in X, y \in Y\}$.

Вычислять функцию правдоподобия по формуле (4.1) не всегда удобно, поскольку для этого необходимо выполнять умножения. Вместо $p(\mathbf{y} \mid \mathbf{x})$ можно вычислять логарифмическую функцию правдоподобия

$$\ln p(\mathbf{y} \mid \mathbf{x}) = \sum_{i=1}^{n} \ln p(y_i \mid x_i). \tag{4.2}$$

Рассмотрим частный случай двоичного симметричного канала (ДСК), который представляет ДПК с двоичными алфавитами $X = Y = \{0,1\}$ и с переходными вероятностями $p(y = 0 \mid x = 1) = p(y = 1 \mid x = 0) = p$, $p(y = 0 \mid x = 0) = p(y = 1 \mid x = 1) = q = 1 - p.$ Такую модель называют двоичным симметричным каналом (ДСК). Для ДСК формула (4.2) принимает вид

$$\ln p(\mathbf{y} \mid \mathbf{x}) = d_H(\mathbf{x}, \mathbf{y}) \ln(p/q) + n \ln q, \tag{4.3}$$

где $d_H(\mathbf{x}, \mathbf{y})$ обозначает расстояние Хэмминга между последовательностями $\mathbf{x}$ и $\mathbf{y}$. Параметр p представляет собой переходную вероятность ДСК. Можно всегда предполагать, что $p < 1/2$. Из формулы (4.3) следует, что для ДСК декодирование по МП эквивалентно декодированию по минимуму расстояния Хэмминга.

Другой важный случай – полунепрерывный канал с двоичным входом и гауссовским шумом. Входные символы выбираются из алфавита $X = \{-1,+1\}$. Условные распределения вероятностей значений y на выходе канала имеют вид

$$f(y \mid x) = \frac{1}{\sqrt{\pi N_0}} \exp\left\{ -\frac{(y - x\sqrt{E})^2}{N_0} \right\},$$

где через E обозначена энергия элементарного сигнала, а $N_0/2$ - дисперсия y, обусловленная действием в канале шума со спектральной плотностью мощности $N_0/2$. При передаче по каналу без памяти последовательности $\mathbf{x}$ длины n получим

$$\ln f(\mathbf{y} \mid \mathbf{x}) = \ln \prod_{i=1}^{n} f(y_i \mid x_i) = \frac{n}{2} \ln(\pi N_0) - \frac{1}{N_0} \sum_{i=1}^{n} (y_i - x_i \sqrt{E})^2 .$$

Таким образом, декодирование по МП эквивалентно поиску кодового слова $\mathbf{x}$, для которого минимально евклидово расстояние

$$d_E(\mathbf{y}, \mathbf{s}) = \sum_{i=1}^{n} (y_i - s_i)^2$$

между последовательностью $\mathbf{y}$ и «сигнальной последовательностью» $\mathbf{s} = \mathbf{x}\sqrt{E}$. Декодирование может быть выполнено еще проще, если принять во внимание, что

$$d_E(\mathbf{y}, \mathbf{s}) = \sum_{i=1}^{n} y_i^{\,2} + \sum_{i=1}^{n} s_i^{\,2} - 2\sqrt{E} \sum_{i=1}^{n} y_i x_i .$$

В правой части только вычитаемое зависит от конкретного кодового слова. Поэтому для декодирования по МП достаточно найти кодовое слово, для которого максимально скалярное произведение

$$(\mathbf{y}, \mathbf{x}) = \sum_{i=1}^{n} y_i x_i .$$

Итак, для декодирования по МП необходимо по принятой последовательности $\mathbf{y}$ для каждого кодового слова вычислить его «метрику». В случае канала без памяти эта метрика является аддитивной, т.е. может быть вычислена как сумма метрик отдельных символов кодового слова. В зависимости от конкретной ситуации, необходимо найти кодовое слово с минимальной метрикой (как в ДСК) либо с максимальной метрикой (как в гауссовском канале). В дальнейшем для определенности будем считать, что задача декодера – найти путь с минимальной метрикой.

4.2. Поиск кратчайшего пути на графе по принципу динамического программирования

Рассмотрим направленный граф, представленный на рис. 4.1. Ребрам графа сопоставлены числа, которые можно интерпретировать как расстояния между узлами графа. Для любого пути можно вычислить его длину как сумму длин составляющих этот путь ребер. Задача состоит в нахождении кратчайшего пути из узла *a* в узел *g*.

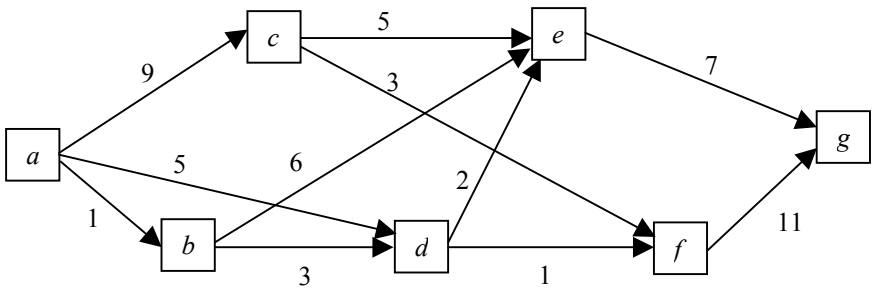

Рис.4.1. Пример направленного графа

Нетрудно перечислить все пути этого графа и найти длину каждого из них. При этом мы найдем, что кратчайшим является путь *abdeg* длины 13. Чтобы

получить этот результат, мы проанализировали 7 путей и выполнили в общей сложности 23 сложения для подсчета их длин. При выборе из 7 путей кратчайшего пути мы выполнили 6 сравнений.

Упростить вычисления позволяет следующее простое наблюдение. Если кратчайший путь из *a* в *g* проходит через некоторую точку, например *d*, то этот путь будет состоять из кратчайшего пути из *a* в *d* и из кратчайшего пути из *d* в *g*. Эта идея принадлежит Р. Беллману и называется принципом динамического программирования.

В нашем примере, используя этот принцип, мы сначала находим, что оптимальный путь из *a* в *d* проходит через *b* и имеет длину 4. Мы получили этот результат, выполнив 1 сложение и 1 сравнение. Затем мы анализируем узел *e* и, затратив 3 сложения и 2 сравнения, определяем, что оптимальный путь из *a* в *e* (путь *abde*) имеет длину 6. На этом шаге мы используем то, что оптимальный путь в *d* и его длина уже известны. Далее в узле *f* выполняем 2 сложения и 1 сравнение и получаем, что кратчайший путь в этот узел имеет длину 5. Наконец, в узле *g,* требуется еще 2 сложения и 1 сравнение, чтобы получить уже известный окончательный результат. Итого, следуя принципу динамического программирования, мы затратили 8 сложений и 6 сравнений.

Если даже на таком простом примере получился выигрыш по сложности вычислений в несколько раз, то нетрудно понять, что в реальных задачах, когда мы рассматриваем решетчатые диаграммы сверточных кодов с сотнями или тысячами узлов на каждом ярусе, выигрыш достигает астрономических значений.

4.3. Алгоритм Витерби

Описанный ниже алгоритм был опубликован А. Витерби в 1967 г. Представление сверточных кодов в виде решетчатой диаграммы и интерпретация задачи декодирования по максимуму правдоподобия как задачи нахождения кратчайшего пути на графе были предложены Г. Д. Форни в 1974 г.

Вернемся к примеру сверточного кода (7,5), рассмотренного в лабораторной работе № 1. Фрагмент решетчатой диаграммы кода показан на рис. 4.2.

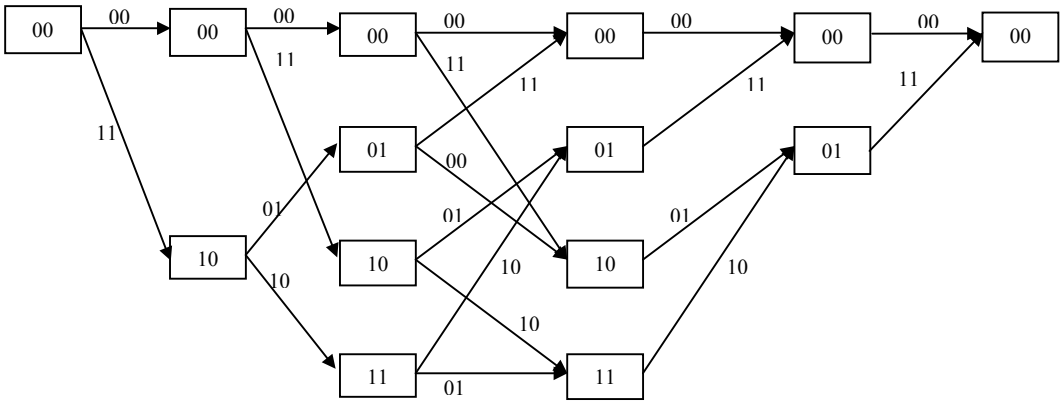

Рис. 4.2. Решетчатая диаграмма кода (5,7)

Эта решетчатая диаграмма описывает множество кодовых слов усеченного сверточного кода, получаемого при подаче на вход кодера некоторой

последовательности информационных символов и вслед за ней – последовательности из ν нулей, где ν – длина кодового ограничения кода. В данном примере вслед за 3 информационными символами подается 2 нуля, в результате, получается, по сути, линейный блоковый код длины 10 с 3 информационными символами. Рассмотрение усеченных сверточных кодов удобно с точки зрения описания алгоритма. Ниже будет показано, каким образом алгоритм Витерби может быть использован для декодирования бесконечно длинных последовательностей.

Для каждого узла решетки мы называем предшествующими узлами те узлы предыдущего яруса, которые связаны ребрами с данным узлом. Путь, ведущий в данный узел на ярусе с номером t, задается последовательностью из t информационных символов, при подаче которой на вход кодера содержимое регистре кодера совпадет с номером узла.

При описании алгоритма мы предполагаем без потери общности, что задачей декодера является отыскание пути с **минимальной** метрикой. Метрикой узла мы называем метрику оптимального пути, ведущего в узел, метрикой ребра – сумму метрик соответствующих кодовых символов. Через L обозначим число ярусов решетчатой диаграммы, описывающей усеченный сверточный код.

Алгоритм Витерби

1. Инициализация. Номер яруса $t = 0$. Метрика нулевого узла приравнивается нулю, за этим узлом закрепляется «пустой» путь.
2. Для ярусов с номерами $t = 1,\ldots, L$ для каждого из узлов на ярусе t выполняются следующие вычисления:
   a. Находим метрику каждого из путей, ведущих в узел, как сумму метрик предшествующих узлов и ребер, связывающих узлы-предшественники с данным узлом.
   b. Находим путь с минимальной метрикой и эту метрику приписываем данному узлу.
   c. Путь, ведущий в узел, вычисляется дописыванием к пути, ведущему в выбранный предшествующий узел, информационного символа, соответствующего переходу из узла-предшественника в данный узел.
3. Путь, соответствующий единственному узлу на ярусе L, выдается получателю как результат декодирования.

Пути, сходящиеся в одном узле решетки, называют конкурирующими, а выбранный из их числа наилучший путь называют выжившим. Алгоритм Витерби на каждом ярусе решетки из каждого узла решетки «смотрит назад» на узлы предыдущего яруса. Находит метрики конкурирующих путей и выбирает путь с наименьшей метрикой. В результате, к последнему ярусу, выживает единственный путь, обладающий минимальной метрикой из всех возможных путей в решетке.

На рис. 4.3. приведен пример работы декодера Витерби в ДСК. Рассматривается усеченный код (7,5), решетка которого показана на рис.4.2.

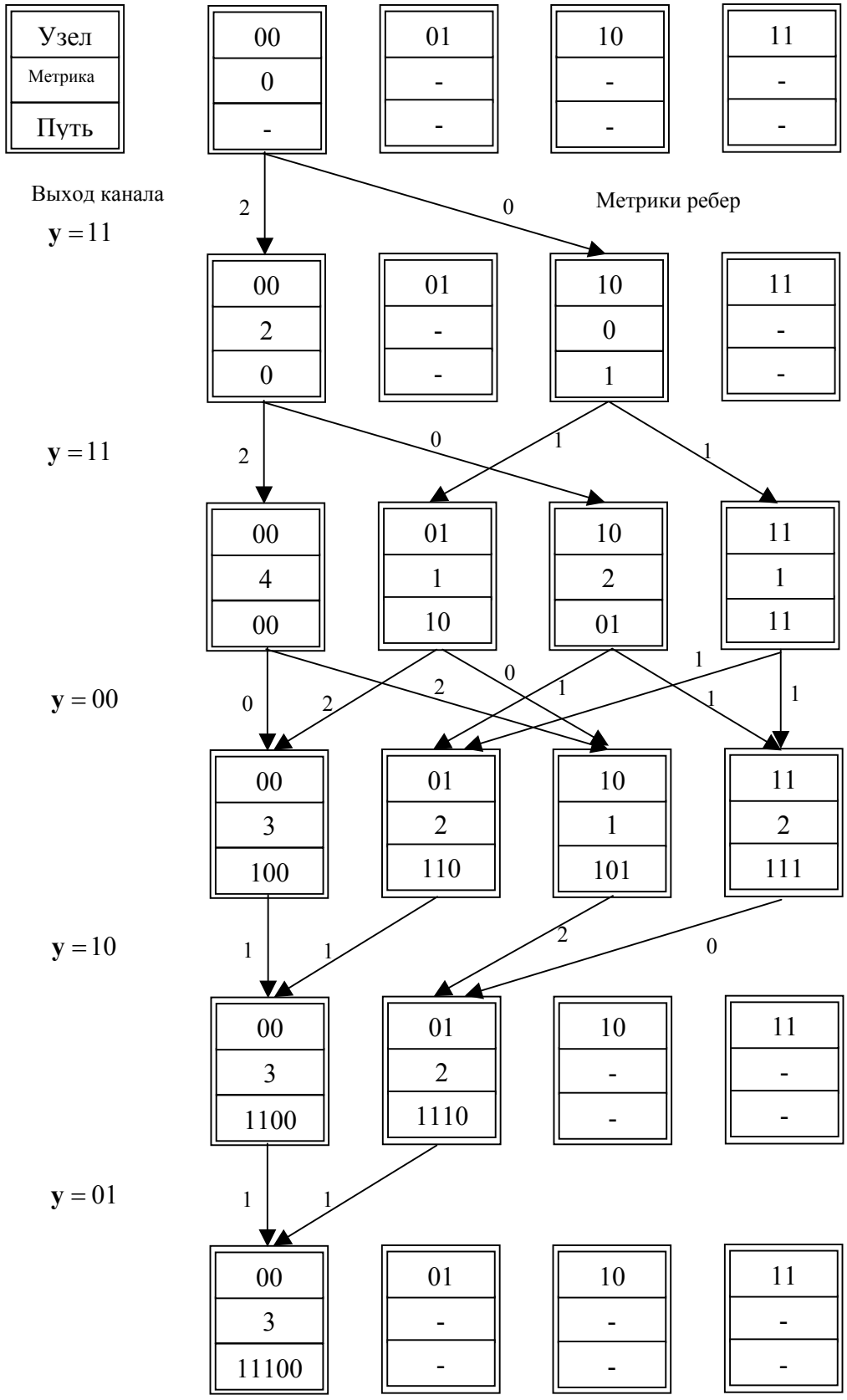

Рис.4.3. Пример декодирования в ДСК по алгоритму Витерби

4.4. Реализация алгоритма Витерби

Описание, приведенное в предыдущем разделе, иллюстрирует основную идею алгоритма Витерби. При практической реализации возникает ряд проблем, самые важные проблемы мы обсудим в данном разделе. К их числу мы относим

- начало работы декодера;
- декодирование бесконечных последовательностей;
- организация памяти для хранения метрик путей;
- организация памяти для хранения выживших путей.

Начало работы декодера.

Из приведенной на рис. 4.1. решетчатой диаграммы видно, что, пока номер яруса не превысил кодового ограничения кода ν, структура решетки отличается от структуры решетки на остальных ярусах. Эта нерегулярность усложняет как программную, так и аппаратную реализацию декодирования.

Эта проблема имеет очень простое решение. Предположим, что декодирование производится по критерию минимальной метрики. В начале работы декодера всем состояниям, кроме нулевого, приписываются одинаковые и достаточно большие значения метрик. Нулевому узлу приписывается нулевая метрика. Далее выполняются точно такие же действия, как при декодировании на ярусах с номерами большими ν. Нетрудно убедиться, что при таком назначении начальных метрик на ярусе с номером ν декодер получит правильные значения метрик всех узлов. «Несуществующие» пути решетки будут отвергнуты декодером, поскольку их метрики будут больше метрик путей, исходящих из нулевого узла.

Применение алгоритма Витерби для декодирования бесконечных последовательностей.

Из описания алгоритма следует, что окончательное решение выдается декодеру только после обработки всей принятой выходной последовательности канала. Понятно, что, если длина усечения кода велика или если используется неусеченный сверточный код, алгоритм не может быть использован из-за ограничений на задержку декодирования и сложность реализации.

Введем параметр T, представляющий собой максимальную задержку декодирования в числе информационных символов. Величина T существенно превышает кодовое ограничение кода ν**.** На ярусе с номером T память путей декодера будет заполнена информационными последовательностями, соответствующими 2^{ν} узлам решетки. Эти последовательности различны, но их число существенно меньше числа 2^{T} различных последовательностей длины T. На предыдущих шагах декодирования производилось сравнение метрик конкурирующих путей, и отбрасывались пути с худшими значениями метрик. Если величина T достаточно велика, то среди путей, начавшихся T ярусов назад, сохранилась лишь небольшая доля путей, и все они имеют достаточно хорошие значения метрик. Наиболее вероятна ситуация, когда на ярусе с номером $t, t \geq T$, все пути имеют одинаковый бит, соответствующий ярусу с номером $t-T$. Этот бит можно выдать получателю, стереть из памяти путей, произвести сдвиг всех путей в регистрах памяти на одну ячейку и тем самым освободить место для очередного информационного символа.

В принципе, при сильном шуме в канале, возможно, что старшие биты в ячейках памяти путей различны. На этот случай нужно разработать стратегию

формирования решений об информационных символах. На практике применяют одну из следующих стратегий:

- выбрать путь с наилучшей метрикой и выдать первый символ этого пути;
- подсчитать количество путей с различными значениями первого символа и принять решение по большинству;
- выдавать получателю первый символ одного и того же пути.

Нетрудно упорядочить эти стратегии по надежности и по сложности реализации, но можно заранее сказать, что при больших T разница между ними невелика. В практических схемах считается, что достаточно выбрать величину T приблизительно в 5 - 6 раз больше длины кодового ограничения кода.

Вычисление и хранение метрик путей.

Основная область применения декодера Витерби – каналы с мягкими решениями, т.е. каналы, в которых на значения входе кодера – вещественные числа. Для дальнейшей обработки они должны быть представлены в цифровой форме. От точности представления зависит сложность аналого-цифрового преобразования, сложность устройств, выполняющих арифметические операции над метриками ребер и путей, объем памяти для хранения метрик.

В 70-е годы, когда разрабатывались первые реализации декодера Витерби, было принято считать, что для представления метрик символов достаточно 3 бит. По мере развития микроэлектроники ограничения на сложность арифметических устройств становились менее жесткими. Тем не менее, увеличение точности свыше 8 бит (для каналов с двоичным входным алфавитом) действительно не имеет смысла.

Итак, будем считать, что метрика одного символа записана в виде 1 байта. Сколько бит нужно для представления метрик путей?

Если длина пути составляет N символов канала, то максимальное значение метрики путей может достигнуть величины $N2^8$ и для хранения такой метрики нужно отвести $(8+\log_2 N)$ двоичных ячеек памяти. При $N \to \infty$ память любого объема неизбежно переполнится.

В решении этой проблемы определяющую роль играет следующее наблюдение. Пусть ν - кодовое ограничение кода, n - число кодовых символов, соответствующих ребру, μ_0 - максимально значение метрики символа. Тогда разница между максимально и минимальной метрикой путей, выживших на данном ярусе, не превышает величины $\nu n \mu_0$. Этот факт следует из того, что для любого узла решетки существует путь, ведущий в этот узел и отличающийся от пути с наименьшей метрикой не больше чем в νn символах.

Если на некотором ярусе из всех значений метрик вычесть минимальное, это не повлияет на выбор путей в будущем, и, значит, вероятность ошибки декодирования не изменится. Эта операция (ее называют *нормализацией* метрик) может выполняться на каждом ярусе или один раз после обработки нескольких ярусов. В результате число бит для хранения метрик каждого пути приблизительно равно $(8+\log_2 \nu n)$ и при реальных значения параметров кодера не превышает 16 бит.

Организация памяти для хранения путей.

Каждый из выживших путей записывается в виде двоичной последовательности длины T. Заметим, что декодер Витерби, обрабатывая очередной узел решетчатой диаграммы, вычисляет новый путь, ведущий в

данный узел, но не может занести его в ту же ячейку памяти, в которой хранился предыдущий путь. Дело в том, что предыдущий путь может понадобиться при обработке других узлов текущего яруса. Значит, нужно хранить два массива путей: «старые» и «новые». После обработки очередного яруса массивы меняются названиями. Таким образом, объем памяти для хранения путей составляет $2T2^{\nu} \approx 10\nu 2^{\nu}$. Именно эта память на практике определяет основные затраты на реализацию декодера.

Существует альтернативный вариант реализации декодера, проигрывающий по скорости работы, но требующий вдвое меньше памяти для хранения путей. Этот вариант называют *декодированием с возвращениями* (back-trace decoder). Декодер с возвращениями на каждом ярусе работы после обработки узла заносит в память один бит, указывающий, какой из двух конкурирующих путей выбран.

Для выработки решения и выдачи получателю декодер, используя «обратный кодер» из текущего узла двигается назад по решетке и вычисляет путь в данный узел. Первый бит пути выдается получателю.

4.5. Порядок выполнения работы

1. Получить задание в виде схемы кодера и тестовой последовательности
2. Выполнить декодирование тестовой последовательности с помощью алгоритма Витерби.
3. Используя готовое программное обеспечение, методом моделирования исследовать эффективность кодов, для которых выполнялись расчеты в лабораторной работе №3.
4. Оформить отчет.

4.6. Контрольные вопросы

1. Вывести формулу (4.3).
2. Показать, что в ДСК декодирование по МП эквивалентно декодированию по минимуму расстояния Хэмминга.
3. Предположим, что для передачи двоичных символов по каналу с белым аддитивным гауссовским шумом применяются двоичные сигналы одинаковой энергии. Показать, что в этом случае декодирование блокового кода по МП эквивалентно декодированию по максимуму скалярного произведения принятой последовательности и кодового слова.
4. Является ли декодер с ограниченной задержкой МП-декодером?
5. Подсчитайте максимальную разность между мнимальной и максимальной метрикой на одном ярусе для декодера Витерби кода с генераторами (7,5) при декодировании в ДСК.
6. Минимизирует ли МП-декодер число ошибочных бит, выдаваемых получателю?
7. Выпишите асимптотические (по длине кодового ограничения) оценки вычислительной сложности декодера с двойной памятью путей и декодера с возвращениями.
8. Нарисуйте блок-схемы двух вариантов реализации алгоритма Витерби с учетом ограничений на задержку, разрядность памяти метрик и т.п.