

УДК 621.391

ТУРБОКОД — ОСНОВНЫЕ ХАРАКТЕРИСТИКИ, ОСОБЕННОСТИ ПРИМЕНЕНИЯ И МОДЕЛИРОВАНИЯ

© 2004 В. И. Васильев, Хоанг Тху Ха

Воронежский государственный университет

Рассматривается современная схема кодирования — турбокод. Описаны особенности турбокода. Приведены типы кодеров и оптимальные алгоритмы декодирования. Определены сложности при проектировании турбокода. Приведены результаты моделирования работы турбокода в гауссовском канале.

Турбокод — очень мощная техника исправления ошибок, которая выигрывает у всех известных предыдущих схем кодирования. Он может использоваться в любой системе связи, где существенна экономия энергии или отношение сигнал/шум очень низко. Например, космические коммуникации, мобильные спутниковые коммуникации, микроволновые системы связи, цифровое спутниковое телевидение. Различные турбокоды были приняты в качестве стандартных в W-CDMA (3GPP), CDMA2000 (3GPP2), DVB-RCS, IEEE 802.16, CCSDS [1].

Исторически, кодеры представляли первую попытку достижения более высокой способности исправления ошибки без сложности декодирования. К сожалению, декодирование не давало улучшения работы как ожидалось, даже при том, что мощь исправления ошибки кода увеличилась в целом. Это происходило главным образом из-за использования алгоритма «жесткого входа — жесткого выхода» (Hard Input — Hard Output — HIHO).

Новый класс турбокода использует иной алгоритм — это «мягкий вход — мягкий выход» (Soft Input — Soft Output — SISO). Чем информативнее последовательность для второго кодирующего устройства, тем более «некоррелированный» информационный обмен между декодерами. Это одна из ключевых идей, которая позволяет осуществить непрерывное усовершенствование способности исправления, когда процесс декодирования повторяется, чтобы передать больше информации от выхода одного декодера к входу следующего декодера. Мягкие выхо-

ды обеспечивают вещественное число, которое является мерой вероятности ошибки в декодировании специфического бита, мерой надежности жесткого решения декодера. Эта дополнительная информация очень важна для следующей стадии в повторяющемся процессе декодирования. Для гауссова канала восьмиуровневое квантование, по сравнению с двухуровневым, приводит в результате к улучшению на 2 дБ отношения сигнал/шум при той же вероятности появления ошибочного бита [2].

Есть две важные категории мягких алгоритмов. Первая категория включает максимальное правдоподобие (likelihood) алгоритма, которое минимизирует вероятность ошибки символа, как максимума апостериорной вероятности (Maximum a Posteriori — MAP) алгоритма [3,4]. Вторая категория включает максимальное правдоподобие алгоритма, которое минимизирует вероятность ошибки слова или последовательности. Это алгоритм Viterbi [5] или мягкой выход алгоритма Viterbi (Soft Output Viterbi Algorithm — SOVA) [6].

Повторение декодирования основано на связи между мягкими выходами алгоритма решения, специальным кодированием и информационными методами передачи, что позволило создать новую мощную технику декодирования и привело к появлению турбокода. Определение турбокода дано впервые в [7], где представлен специфический класс параллельной связи двух рекурсивных систематических сверточных кодеров.

Есть несколько различий между поведением турбокодов и сверточных кодов (Con-

volutional Codes — CC). Известно, что выполнение CC улучшается с увеличением длины ограничения и не улучшается значительно с уменьшением степени кодирования. Дело обстоит не так для турбокода — лучшие турбокоды имеют очень маленькую длину ограничения.

1. СТРУКТУРА ТУРБОКОДЕРА

Схема турбокодера состоит из двух одинаковых блоков RSC1 и RSC2, которых называют рекурсивными систематическими кодирующими (Recursive Systematic Codes — RSC) устройствами. Между ними блок чередования INT, который перестраивает информационные биты (рис. 1). Тогда к информационной последовательности (X_k) добавляют две проверочные группы (Y_{1k} и Y_{2k}). Игнорируя, задержки для каждого блока, мы принимаем данные с обеих кодирующих устройств одновременно. Если два кодирующих устройства связаны с переключательным устройством, то степень кодирования будет равна $1/2$ и получим следующую выходную последовательность: $(X_1, Y_{11}), (X_2, Y_{22}), (X_3, Y_{13}), (X_4, Y_{24}), \dots$. Без переключателя степень кодирования кода будет равна $1/3$. Ограничений на количество соединяемых кодеров нет.

На рис. 1 показано получение двухразмерных турбокодов с использованием RSC. Каждый RSC имеет длину кодового ограничения $K=3$ и используется с двумя генераторами кода $G1 = \{1,1,1\}$ и $G2 = \{1,0,1\}$. Может быть достигнута другая степень кодирования, используя или меньше или больше чередования и блоки кодирующего устройства, как показано на рис. 2, для n -пространственных турбокодов.

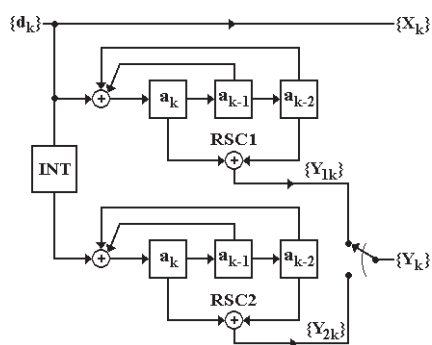


Рис. 1. Схема параллельного соединения двух RSC-кодеров, длина кодового ограничения $K=3$, генераторы кода — $G1 = \{1,1,1\}$ и $G2 = \{1,0,1\}$

Турбокод, представленный в [7], достиг вероятности появления ошибок (Bit Error Rate — BER) 10^{-5} при степени кодирования $1/2$ и бинарной частотной модуляции (Binary Frequency Shift Keying — BPSK) в канале с белым аддитивным гауссовым шумом (Additive White Gaussian Noise — AWGN) с $E_b/N_0 = 0,7$ дБ, что очень близко к пределу Shannon [8].

Структуру на рис. 1, которая используется в [7], называют параллельным каскадным сверточным кодом (Parallel Concatenated Convolutional Code — PCCC), так как один и тот же информационный поток кодируется дважды, параллельно. Другая схема чередует выходные биты одного кодирующего устройства и перекодирует их снова. Эту структуру называют последовательным каскадным сверточным кодом (Serial Concatenated Convolutional Code — SCCC). Она была предложена в [9,10]. Может использоваться любая другая возможная комбинация PCCC и SCCC. Эта структура описана в [11,12], ее называют гибридным каскадным сверточным кодом (Hybrid Concatenated Convolutional Code — HCCC).

PCCC структура имела BER больше чем 10^{-6} при определенных ограничениях системы. Однако, для меньших BER, подобно 10^{-10} для передач данных, структуру PCCC трудно реализовать. Базируясь на аналитических исследованиях [13] и из опыта в осуществлении PCCC в аппаратных средствах ЭВМ и программном обеспечении [14], можно заключить, что изменение в наклоне кривой BER появляется для $BER < 10^{-7}$. Наклон — функция размера чередования и проекта чередования.

Пример SCCC с степенью кодирования $1/3$ показан на рис. 3. Используя “равномерное чередование” (Uniform Interleaver), как показано в [9, 15, 16], получаем большое

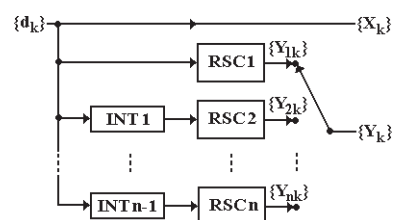


Рис. 2. Структурная схема кодера для n -пространственных турбокодов

различие в чередовании и задержке между структурами SCCC и PCCC. В [15] показано, что SCCC может иметь большую, около 100 %, задержку. В [16] показан выигрыш в 2 дБ для SCCC схемы по сравнению с PCCC схемой для BER 10^{-11} . Выполнение SCCC и PCCC схем для различных проектов чередования было исследовано в [15, 16]. Показано, что PCCC схемы лучше, чем SCCC схемы при низком отношении сигнал/шум (Signal-to-Noise Ratio — SNR). Однако, увеличивая SNR, SCCC схемы выигрывают у PCCC схем. Возможная точка “пересечения” зависит от размера чередования и проекта чередования.



Рис. 3. SCCC структура

HCCC является комбинацией последовательной и параллельной связи и имеет преимущества перед структурами SCCC или PCCC. Структура HCCC показана на рис. 4. Структура HCCC — дальнейшее усовершенствование структуры SCCC.

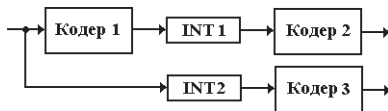


Рис. 4. HCCC структура

В настоящее время турбокоды развиваются по двум альтернативным направлениям — это “сверточные” турбокоды (Turbo Convolutional Codes — TCC), появившиеся исторически первыми [7], и турбокоды произведения (Turbo Product Codes — TPC) [17]. Оба класса кодов характеризуются итеративными алгоритмами декодирования и мягкими решениями на входе и выходе каждой итерации. В работе [18] показана эффективность TCC при малых степенях кодирования (менее 2/3). Это определяет выбор TCC в каналах с низкой энергетикой.

2. ТУРБОДЕКОДЕР

Оптимальный алгоритм декодирования для турбокода — это максимальное правдоподобие, примененное к структуре решетчатого турбокода. Однако, из-за чередования, которое включается в кодер турбоко-

да, решетчатый турбокод будет иметь очень большое количество состояний. Этот факт делает получение максимального правдоподобия в процессе декодирования практически невозможным для больших размеров чередования. Более практический подход — это повторение декодирования, где максимальное правдоподобие применено к элементарным кодерам TCC или TPC. Эта повторяющаяся техника — очень эффективный способ турбодекодирования — позволяет достигать теоретического предела.

Есть две проблемы для процесса декодирования: как закрыть повторяющийся процесс, чтобы выполнение декодирования было оптимальным и как передать информацию от одного декодера к другому. Однако, из-за повторяющегося процесса, некоторые параметры могут расти до бесконечности, если нет ограничений в алгоритме декодирования. Ограничения же, наложенные на эти параметры, могут дать отрицательные эффекты для малых BER. Другая причина ограничения BER — плохой проект чередования. Для очень коррелированных последовательностей BER уменьшается до некоторого предельного. Этот эффект BER называют “этажом ошибки” турбокода.

На рис. 5. показаны три кривые. Кривая “этажа ошибки” типична для плохого проекта чередования или усечений в алгоритме декодирования. Кривая результата моделирования также представляет изменение в наклоне, но для малых BER, когда теоретический предел достигнут.

Мягкий выход решения декодера дает вещественное число, которое является ме-

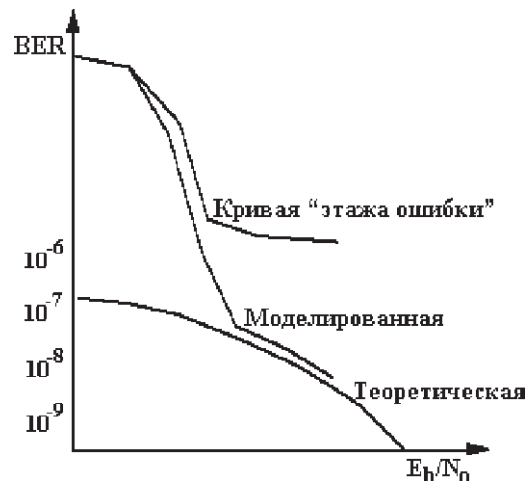


Рис. 5. Типичные BER кривые для турбокода

рой вероятности правильного решения. Это число называют апостериорной вероятностью (A Posteriori Probability — APP). Двумя типами мягкого решения алгоритма декодирования являются SOVA [6] и MAP [3,4] алгоритмы. По оценке, сложность алгоритма MAP в два раза больше, чем Viterbi алгоритм. Однако, алгоритм MAP дает лучший результат для низких SNR из-за более точной оценки APP. Если BER мало, существует незначительное различие в производительности между алгоритмами MAP и SOVA. При высоких значениях BER и низких значениях E_b/N_0 алгоритм MAP превосходит алгоритм SOVA на 0,5 дБ и более [14]. Это может оказаться очень важным для турбокодов.

В канале AWGN декодирование для систематических СС, как показано в [14], алгоритм MAP основывается на той же идее, что и алгоритм Viterbi — обработка блоков кодовых битов в двух направлениях. Эта обработка дает метрики состояний и метрики ветвей блока. После этого можно начинать расчет APP и MAP для каждого бита данных в блоке.

Расчет начинается с расчета отношения значений APP, называемое отношением правдоподобия, или с их логарифма, называемого логарифмическим отношением правдоподобия (Log-Likelihood Ratio — LLR) $L(\hat{d}_k)$ для декодированного бита \hat{d}_k по формуле:

$$L(\hat{d}_k) = \log \left[\frac{\sum \lambda_k^{1,m}}{\sum_m \lambda_k^{0,m}} \right]. \quad (1)$$

Здесь $\lambda_k^{i,m} = \mathbf{P}\{d_k = i, S_k = m | R_1^N\}$ — вероятность того, что переданные данные $d_k = i$ и текущее состояние RSC декодера $S_k = m$, при условии, что принята кодовая последовательность R_1^N , получаемая в момент $k=1$ в течение некоторого времени N .

Причем $R_1^N = \{R_1^{k-1}, R_k, R_{k+1}^N\}$, т.е. на декодер демодулятора подавалось по одному блоку из N бит за такт. Тогда:

$$\begin{aligned} \lambda_k^{i,m} &= \mathbf{P}\{d_k = i, S_k = m | R_1^{k-1}, R_k, R_{k+1}^N\} = \\ &= \mathbf{P}\{R_1^{k-1} | d_k = i, S_k = m, R_k^N\} \times \\ &\quad \times \mathbf{P}\{R_{k+1}^N | d_k = i, S_k = m, R_k\} \times \\ &\quad \times \mathbf{P}\{d_k = i, S_k = m, R_k\} / \mathbf{P}\{R_1^N\}. \end{aligned} \quad (2)$$

В (2) множитель $\alpha_k^m = \mathbf{P}\{R_1^{k-1} | d_k = i, S_k = m, R_k^N\}$ называется прямой метрикой состояния для момента времени k и состояния m . Если $R_k = \{x_k, y_k\}$, x_k и y_k являются независимыми нормальными случайными величинами, то $d_k = i$ и R_k^N можно не учитывать. Причем, $S_k = m$ подразумевает, что события до момента k не влияют на изменение после момента k . Тогда можем написать, что $\alpha_k^m = \mathbf{P}\{R_1^{k-1} | S_k = m\}$.

Аналогично, второй множитель $\beta_{k+1}^{f(i,m)} = \mathbf{P}\{R_{k+1}^N | S_k = f(i, m)\}$ называется обратной метрикой состояния в момент $k+1$ и состоянии $f(i, m)$, определяемое входом i и состоянием m . Третий множитель $\delta_k^{i,m} = \mathbf{P}\{d_k = i, S_k = m, R_k\}$ — метрика ветви в момент k и состоянии m .

После преобразований получим [14]:

$$\alpha_k^m = \sum_{j=0}^1 \alpha_{k-1}^{b(j,m)} \delta_{k-1}^{j,b(j,m)} \quad \text{и} \quad \beta_k^m = \sum_{j=0}^1 \beta_{k+1}^{f(j,m)} \delta_k^{j,m}. \quad (3)$$

Для $\delta_k^{i,m}$ в канале AWGN с нулевым средним и дисперсией σ^2 получим

$$\delta_k^{i,m} = A_k \pi_k^i \exp \left[\frac{1}{\sigma^2} (x_k X_k^i + y_k Y_k^{i,m}) \right]. \quad (4)$$

Здесь A_k — постоянная, π_k^i обозначает $\mathbf{P}\{d_k = i\}$ — априорная вероятность d_k . Тогда, если подставим (3, 4) в (1):

$$\begin{aligned} L(\hat{d}_k) &= \log \left[\pi_k \times \exp \left(\frac{2x_k}{\sigma^2} \right) \times \pi_k^e \right] = \\ &= L_a(d_k) + L_c(x_k) + L_e(\hat{d}_k). \end{aligned} \quad (5)$$

Здесь $\pi_k = \pi_k^1 / \pi_k^0$ — входное отношение априорных вероятностей и $L_a(d_k)$ — это априорное LLR бита данных d_k ; $L_c(x_k)$ член LLR получается в результате канальных измерений, произведенных в приемнике; π_k^e — внешнее выходное правдоподобие и $L_e(\hat{d}_k)$ — внешний LLR. Мягкий выход декодера $L(\hat{d})$ является вещественным числом, обещающим в итоге как само принятие жесткого решения, так и его надежность. Его знак задает жесткое решение — $d=1$, если $L(\hat{d}) > 0$ и $d=0$, если $L(\hat{d}) < 0$.

На рис. 6 показана схема декодера с обратной связью. Следует отметить, что выход DEC1 имеет структуру чередования INT, аналогичную структуре, использованной в передаче информации обратно на другой декодер является то, что декодер никогда не сле-

дует заполнять собственными данными (иначе искажения на входе и выходе будут сильно коррелировать). Член π_k^e в (5) можно считать фактором коррекции (вследствие кодирования), который меняет входные априорные сведения о битах данных. В турбокоде такие корректировочные члены проходят из одного декодера в другой, чтобы улучшить отношение правдоподобий для каждого информационного бита и, таким образом, минимизировать вероятность появления ошибок декодирования. Внешнее правдоподобие π_k^e , получаемое из конкретной итерации, заменяет априорное правдоподобие π_{k+1}^e на следующую итерацию.

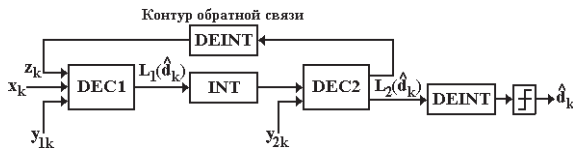


Рис. 6. Структурная схема турбодекодера с обратной связью

Рассмотрим случай использования СС с M -символьным алфавитом в качестве базового. Поскольку число M должно являться степенью 2, тогда число бит, составляющих один символ кодового алфавита $L = \log_2 M$. Из работы [19] суть величин α , β и δ сохраняется, в (2) число i принадлежит множеству $\Omega = \{0, 1, \dots, M - 1\}$, а длина информационной части блока N должна быть кратна L . Основные отличия от бинарного случая проявляются при чередовании и при вычислении и обработке внешней (апостериорной) информации. Процесс чередования символов M -позиционного кода проводится в два этапа — перестановка символов и перестановка бит внутри символа. Процесс вычисления надо выполнить по каждому $i \in \Omega$.

3. ПРАКТИЧЕСКАЯ РЕАЛИЗАЦИЯ ТУРБОКОДОВ

Реализация турбокодов зависит от размера чередования, проекта чередования, учредительных кодеров и числа повторений декодера. При больших размерах чередования могут быть достигнуты меньшие значения BER. С увеличением размера чередования увеличиваются кодированные и декодированные задержки. Чтобы достигнуть лучшей работы, от турбокода требуются

большие задержки. Однако, даже для маленьких задержек, которые подразумевают маленькие размеры чередования, их работа все еще лучше, чем обычных кодеров подобной сложности. Полная задержка бита может быть весьма большой. Однако, так как скорость передачи данных высока, соответствующая задержка времени может быть сделана очень маленькой.

Схема турбокодера состоит из двух устройств СС и устройства для хранения адресов чередования. Схема турбодекодера включает два мягких выхода декодера и устройства запоминания (RAM) функций чередования/дечередования. Мягкие выходы декодера могут быть или SOVA декодеры или MAP декодеры. Декодеры MAP дают на $0,5 \div 1,0$ дБ больше выигрыш, чем SOVA и приблизительно в два раза более сложны.

Для выполнения цифрового обработчика сигналов (Digital Signal Processing — DSP) мы должны оценить число инструкций, необходимых для декодирования бит. В работе [19] есть оценка сложности арифметикологических устройств (Arithmetic-Logic Devices — ALD), определяется число операций ALD в виде сложений и умножений, а также необходимая емкость RAM для M -позиционного турбодекодера со степенью кодирования $1/3$. Схема реализации 4-позиционных турбодекодеров существенно проще классических с точки зрения требуемого числа арифметических устройств и емкости RAM. Если увеличение числа символов в алфавите до 8 и более следует считать целесообразным для данной схемы построения кода, то при незначительном увеличении помехоустойчивости сложность декодера существенно возрастает.

Рассмотрим моделирование турбокода в среде MATLAB. Первое — изменяется число итерации от 1 до 10. При этом мы используем генератор кода $G1 = \{1, 1, 1\}$ и $G2 = \{1, 0, 1\}$, степень кодирования $r = 1/2$, число битов в одном кадре $N = 1500$ (т.е. 1498 информационных битов и 2 бита “хвоста”). Второе — рассмотрение разницы между двумя степенями кодирования $1/2$ и $1/3$. Здесь также используются генераторы кода и число битов в кадре, как в первом случае и учитываются результаты 1-ой, 5-ой и 10-ой итераций. Третье — моделирование с раз-

ным числом битов в кадре: $N = 50$, $N = 500$, $N = 1500$, $N = 2500$. В этом случае учитываются только результаты 10-ой итерации. Генератор кода и степень кодирования оказываются аналогичными с первым случаем. Во всех трех случаях моделирования используется кодер на рис. 1.

Рассмотрим турбокод в канале AWGN с BPSK модуляцией. Схема декодера на рис. 7. При передаче производится операция преобразования сигнала из параллельной формы в последовательную, а при приеме из последовательной в параллельную форму. Принятый сигнал: систематический — x_k и избыточный — y_k . Причем $x_k = (2X_k - 1) + p_k$ и $y_k = (2Y_k - 1) + q_k$, где p_k и q_k являются двумя случайными статистически независимыми переменными с одинаковой дисперсией σ^2 , определяющей распределение помех. Избыточная информация y_k разуплотняется и пересылается на декодер DEC1 как y_{1k} , если $Y_k = Y_{1k}$, и на декодер DEC2 как y_{2k} , если $Y_k = Y_{2k}$.

Моделирование в среде MATLAB осуществлялось на IBM PC с процессором Pentium-4 с тактовой частотой 2.4ГГц и 256Мб RAM. Полученные результаты не полностью отражают все возможности турбокода, минимальное время процесса моделирования для каждого случая длится порядка 7 ч. Для получения $BER < 10^{-5}$ или $N > 2000$ понадобится значительно больше времени (15 ч и более). Здесь значения BER рассматриваются как отношение между неправильными декодированными битами данных и общим количеством переданных битов данных. Условием завершения процесса

моделирования для каждого значения E_b/N_0 являлось число кадров, содержащих ошибки. В нашем случае, это число равно 15. Для числа 30 время моделирования возрастает до 14 ч, хотя результат улучшается. В преобразовании полученного сигнала в форму LLR требуются некоторые статистические параметры канала. Для канала AWGN, E_b/N_0 должно быть известно.

Рис. 8 показывает, что результат декодирования улучшается при увеличении числа итераций. После первой итерации, BER постепенно снижается. До десятой итерации достигается $BER \approx 10^{-5}$ при $E_b/N_0 = 1.8$ дБ. Необходимо подчеркнуть, что после каждой последующей итерации улучшается результат (т.е. значение BER), но с одиннадцатой итерации увеличиваются сложность проектирования и задержка декодера.

Рис. 9 показывает результаты по разным степеням кодирования. Как и с любым кодером, результат улучшается при снижении степени кодирования. Однако это снижение одновременно приводит к увеличению избыточных битов.

Рис. 10 показывает изменение результатов по разным числам битов в кадре. Результат улучшается при увеличении числа битов в кадре N . Это происходит из-за большего размера чередования. В тех случаях, когда N мало (например, для $N = 50$), мы можем использовать меньше числа итерации.

Турбокод имеет некоторые особенности, которые должны быть учтены. Некоторые особенности могут ухудшить результат декодирования. Одна из них — это кривая

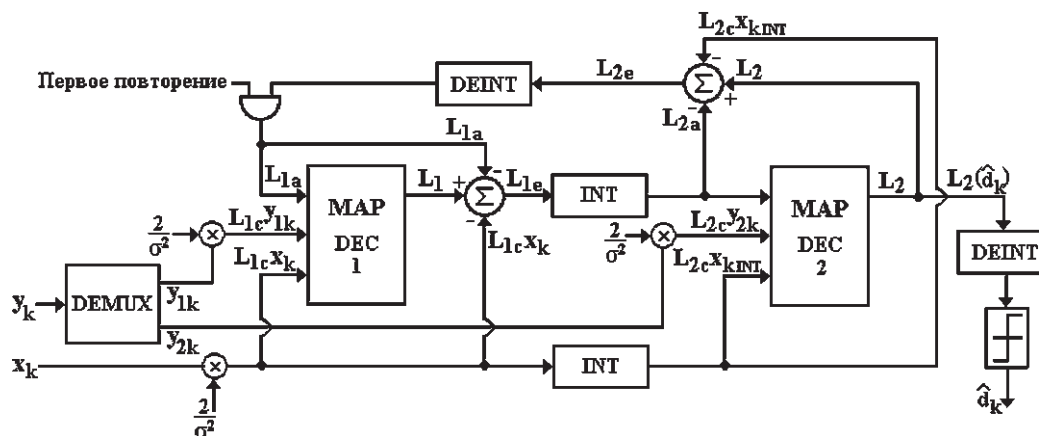


Рис. 7. Структурная схема турбодекодера для моделирования

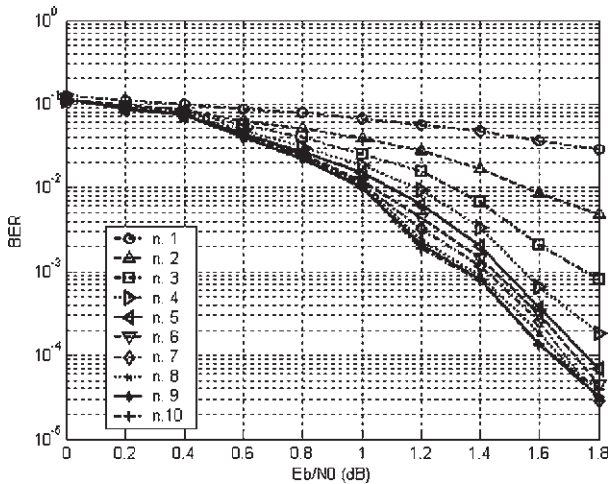


Рис. 8. Кривые BER как функции E_b/N_0 и количества итераций. $G1 = \{1,1,1\}$, $G2 = \{1,0,1\}$, $r = 1/2$, $N = 1500$

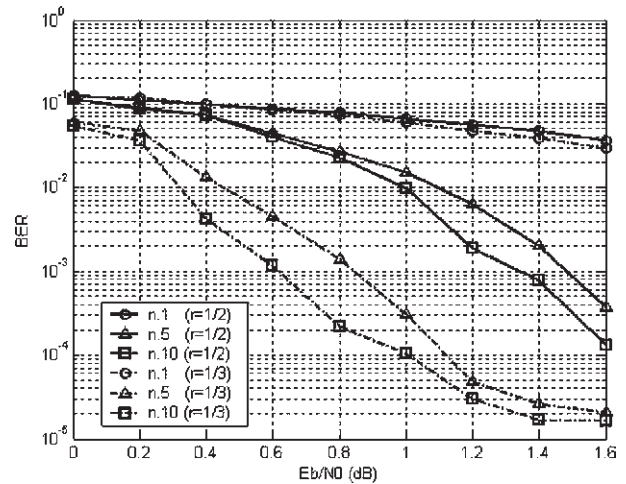


Рис. 9. Кривые BER как функции E_b/N_0 и степени кодирования. $G1 = \{1,1,1\}$, $G2 = \{1,0,1\}$, $r = 1/2$ и $r = 1/3$, $N = 1500$, 1-ая, 5-ая и 10-ая итерация

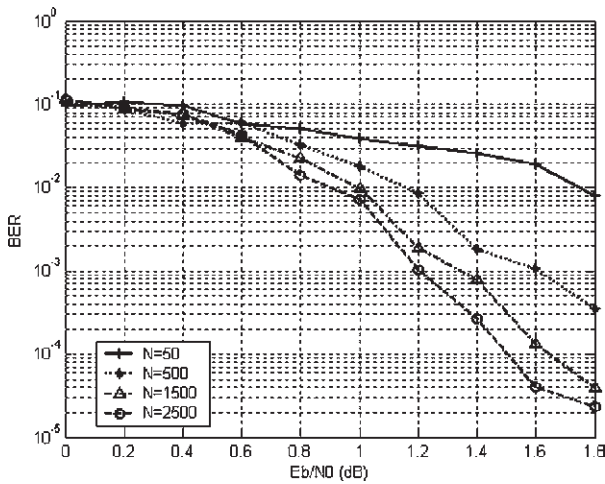


Рис. 10. Кривые BER как функции E_b/N_0 и количества битов в кадре. $G1 = \{1,1,1\}$, $G2 = \{1,0,1\}$, $r = 1/2$, 10-ая итерация

BER, которая резко уменьшается с увеличивающимся E_b/N_0 (например, $BER < 10^{-5}$) и начинает сглаживаться при высоком E_b/N_0 (рис. 9). Эта область является “этажом ошибки”. Другая проблема с турбокодерами — сложность проектирования. Один из способов уменьшить сложность состоит в том, чтобы остановить итерацию, как только кадр будет полностью исправлен.

ЗАКЛЮЧЕНИЕ

Статья представляет собой попытку обобщения результатов реализации и моделирования в среде MATLAB алгоритмов самого мощного помехоустойчивого кода, называемого турбокодом. Существует много алго-

ритмов и их модификаций. Оптимальным алгоритмом является MAP алгоритм для бинарного и M -ого случаев. Чем больше число битов в символьном алфавите, тем сложнее проектирование.

Развитие турбокодеров для AWGN и каналов с замиранием показали большой потенциал этой техники кодирования для очень низких E_b/N_0 . Выполнение турбокодеров зависит от проекта чередования. Большой размер чередования означает задержку расшифровки, дает более низкую BER.

Из-за их превосходной работы, ожидается, что турбокодеры станут стандартом, кодирующим технику этого тысячелетия.

СПИСОК ЛИТЕРАТУРЫ

1. <http://www.rd.francetelecom.fr/en/brevets/tc.php>
2. Heller J.A., Jacobs I.W. Viterbi Decoding for Satellite and Space Communication // IEEE Trans. Commun. Technol., Oct. 1971, V. COM19, № 5, P. 835—848.
3. Bahl L., Cocke J., Jelinek F., Raviv J. Optimal Decoding of Linear Codes For Minimizing Symbol Error Rate // IEEE Transactions on Information Theory, Mar. 1974, V. IT-20, P. 248—287.
4. Pietrobon S., Barbulescu A. A Simplification of The Modified Bahl Decoding Algorithm for Systematic Convolutional Codes // Int. Symp. on Inform. Theory and Its Applications, Sydney, Australia, Nov. 1994, Revised 4 Jan. 1996, P. 1073—1077.
5. Viterbi J. Error Bounds for Convolutional Codes and an Asymptotically Optimum Decoding Algorithm // IEEE Transactions on Information Theory, Apr. 1967, V. IT-13, P. 260—269.

6. Hagenauer J., Hoehner P. A Viterbi Algorithm with Soft-Decision Outputs and Its Applications // Proc. Globecom '89, Dallas, USA, P. 1680—1686.
7. Berrou C., Glavieux A., Thitimajshima P. Near Shannon limit error-correcting coding and decoding: turbo-codes // Proc. of IEEE ICC'93, Geneva, Switzerland, May 1993, V. 2, P. 1064—1070.
8. Shannon E. A Mathematical Theory and Communication // Bell System Technical Journal, Jun.—Oct. 1948, V. 27, P. 379—423 (pt. I) — 623—656 (pt. II).
9. Benedetto S., Montorsi G. Iterative decoding of serially concatenated convolutional codes // IEE Electron. Lett., Jun. 1996, V. 32, P. 1186—1188.
10. Couleaud J.Y., Pietrobon S.S. Serial Concatenated Systematic Convolutional Codes for Space Communication // A letter submitted to the IEEE Transactions on Communications, 6 Oct. 1995.
11. Divsalar D., Pollara F. Hybrid concatenated codes and iterative decoding // JPL TDA Progress Report 42-130, Aug. 1997.
12. Divsalar D., Pollara F. Serial and hybrid concatenated codes with applications // Proc. Int. Symp. on Turbo Codes and Related Topics, Brest, France, Sep. 1997, P. 80—87.
13. Benedetto S., Montorsi G. Unveiling turbo codes: some results on parallel concatenated coding schemes // IEEE Trans, on Inform. Theory, Mar. 1996, V. 42, № 2, P. 409—428.
14. Pietrobon S.S. Implementation and performance of a turbo/MAP decoder // Int. J. of Satellite Communications, Jan. — Feb. 1998, V. 16, P. 23—46.
15. Mark S. Ho C. Serial and Parallel Concatenated Turbo Codes // Ph. D. Dissertation, Uni of South Australia, Nov. 2002.
16. Barbulescu S.A., Buetefer J. Practical delay reduction schemes in serial and Parallel Concatenated Schemes (Turbo Codes) // Symp. on Inform. Theory and Its Applic., Yuzawa, Japan, Nov. — Dec. 1999, V. 2, P. 599—602.
17. Lodge J., Young R., Hoehner P., Hagenauer J. Separable MAP “Filters” for the Decoding of Product and Concatenated Codes // IEEE ICC'93, May 1993, P. 1740—1745.
18. Hagenauer J., Offer E., Papke L. Iterative decoding of binary block and convolutional codes // IEEE Transactions on Information Theory, Mar. 1996, V. 42, P. 429—445.
19. Щербаков Я. Ю. Применение турбо-кодов с M-позиционным алфавитом в системах связи с низкой энергетикой // Телекоммуникации, 2003, № 12, С. 30—35.