

УДК 681.391

АНАЛИЗ ЭФФЕКТИВНОСТИ ТУРБОДЕКОДИРОВАНИЯ КОДОВ ПРОИЗВЕДЕНИЯ ПРИ РАЗЛИЧНЫХ ВАРИАНТАХ ОБМЕНА ИНФОРМАЦИЕЙ МЕЖДУ ИТЕРАЦИЯМИ

Зайцев Г.В., д.т.н., с.н.с., начальник СКБ ПАО «НПО «Алмаз», e-mail: gennady-zaytsev@yandex.ru;
Лутков А.Н., начальник сектора ПАО «НПО «Алмаз», e-mail: anlutkov@gmail.com.

EFFICIENCY OF PRODUCT-CODES TURBO-DECODING ALGORITHMS WITH DIFFERENT METHODS OF INFORMATION EXCHANGE BETWEEN ITERATIONS

Zaytsev G.V., Lutkov A.N.

We consider Chase-Pandiah turbo decoding algorithm of product codes. Modified versions of information exchange between iterations are suggested. The method is proposed for computational complexity reduction by decreasing the number of recalculated symbols at each iteration. Error probability of information transmission when using these algorithms are compared for different number of recalculated symbols. It is shown that new algorithms outperform standard one if reduced computational complexity is required.

Key words: forward error correction, turbo product code, soft iterative decoding, decoding complexity.

Ключевые слова: помехоустойчивое кодирование, турбокод произведения, мягкое итерационное декодирование, вычислительная сложность декодирования.

Введение

Применение блочных турбокодов является одним из наиболее эффективных методов помехоустойчивого кодирования, особенно при малом размере блока. Использование блочных турбокодов позволяет довольно близко подойти к пропускной способности канала, а сложность декодирования приемлема для многих приложений [1].

В настоящей работе рассматриваются двоичные турбокоды произведения [1]. Для простоты ограничимся кодами, образованными при помощи каскадирования двух линейных блочных кодов C_1 , C_2 . Обозначим длины составляющих кодов символами n_1 , n_2 , числа информационных символов этих кодов – k_1 , k_2 , расстояния Хэмминга – d_1 , d_2 . Структура кодового слова двумерного турбокода произведения, образованного из кодов C_1 , C_2 , показана на рис. 1.

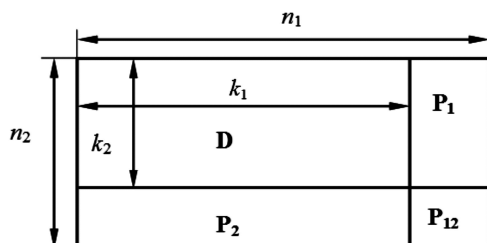


Рис. 1. Структура кодового слова турбокода произведения

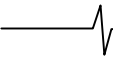
Кодовое слово представлено в виде двумерной матрицы размера $n_2 \times n_1$. Информационные символы составляют подматрицу D размера $k_2 \times k_1$. Остальные символы матрицы являются проверочными. Кодирование осуществляется сначала для строк матрицы D с

В работе рассматривается алгоритм турбодекодирования кодов произведения по методу Чейза-Пандиаха [1]. Предложены модифицированные варианты обмена информацией между итерациями турбодекодера, а также метод снижения вычислительной сложности за счет уменьшения числа переоцениваемых символов на каждом этапе алгоритма. Проводится сравнение эффективности описанных вариантов алгоритма при различном числе переоцениваемых символов. Показано, что предлагаемые варианты алгоритма имеют преимущество по сравнению со стандартными при уменьшении числа переоцениваемых символов.

использованием кода C_1 с образованием проверочных символов подматрицы P_1 , а затем – для столбцов подматриц D и P_1 с использованием кода C_2 с образованием проверочных символов подматриц P_2 и P_{12} .

Нетрудно видеть, что образованный таким образом код имеет длину $n = n_1 \times n_2$, число информационных символов $k = k_1 \times k_2$ и расстояние $d = d_1 \times d_2$. Общепринятое обозначение такого кода – $(n_1, k_1) \times (n_2, k_2)$.

Турбокоды произведения имеют эффективный алгоритм мягкого итерационного декодирования [1], реализующий принцип турбодекодирования. Согласно данному алгоритму поочередно выполняется мягкое декодирование строк с использованием кода C_1 и столбцов с использованием кода C_2 . Однократное декодирование кодов C_1 и C_2 называется итерацией алгоритма, а каждое декодирование кода C_1 или кода C_2 – полуитерацией, или элементарным декодированием. В базовом варианте, называемом алгоритмом Чейза-Пандиаха, на каждой полуитерации используется алгоритм мягкого декодирования на основе второго модифицированного алгоритма Чейза, однако возможны и другие варианты [1]. Детали используемого алгоритма мягкого декодирования в данной работе не существенны, тем не менее, следует указать, что авторы используют базовый вариант.



Обозначим действительную матрицу на входе k -го элементарного декодера, осуществляющего мягкое декодирование строк (или столбцов), символом \mathbf{R}_k^{in} , а действительную матрицу на выходе этого декодера – \mathbf{R}_k^{out} , $k = 1, 2, \dots$. Начальная входная матрица \mathbf{R}_1^{in} , для первой полуитерации принимается равной матрице \mathbf{R}_0 , составленной из чисел, полученных из канала связи на выходе демодулятора, формирующего мягкие решения:

$$\mathbf{R}_1^{in} = \mathbf{R}_0. \quad (1)$$

Знаки чисел матриц \mathbf{R}_k^{in} , \mathbf{R}_k^{out} представляют собой текущую оценку переданных двоичных символов, а их модули – надежность такой оценки.

Тогда согласно рассматриваемому алгоритму, операции, осуществляемые в элементарном декодере, состоят в следующем. На выходе каждого элементарного декодера вычисляется так называемая внешняя информация \mathbf{W}_{k+1} (extrinsic information), являющаяся дополнительной к входной и получаемая при декодировании за счет избыточности:

$$\mathbf{W}_{k+1} = \mathbf{R}_k^{out} - \mathbf{R}_k^{in}, \quad k = 1, 2, \dots \quad (2)$$

Для унификации индексации будем считать, что $\mathbf{W}_1 = 0$.

Затем внешняя информация (2) умножается на весовой коэффициент α_{k+1} и складывается с информацией, полученной из канала связи:

$$\mathbf{R}_{k+1}^{in} = \mathbf{R}_0 + \alpha_{k+1} \mathbf{W}_{k+1}, \quad k = 1, 2, \dots \quad (3)$$

Матрица (3) является входной матрицей для следующего элементарного декодера. Коэффициенты α_k возрастают от малых значений до единицы по мере увеличения k , что обусловлено тем фактом, что на первых итерациях дисперсия внешней информации относительно велика [1], и коэффициенты выполняют демпфирующую роль. Выбор конкретных значений этих коэффициентов производится для каждого кода отдельно, исходя из минимума вероятности ошибки декодирования.

На рис. 2 показана структурная схема одной полуитерации описанного алгоритма в том виде, как она обычно приводится в литературе [1]. Схема реализует приведенные выше формулы.

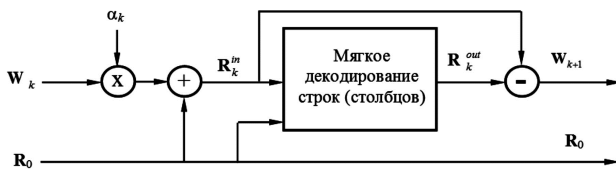


Рис. 2. Структурная схема элементарного декодера

В блоке мягкого декодирования строк (столбцов) производится переоценка всех элементов текущей матрицы \mathbf{R}_k^{in} на основе второго алгоритма Чейза. В случае невозможности такой переоценки для некоторого элемента r_{ij}^{in} матрицы \mathbf{R}_k^{in} , соответствующий элемент r_{ij}^{out} выходной матрицы \mathbf{R}_k^{out} определяется по формуле

$$r_{ij}^{out} = \beta_k r_{ij}^{in}, \quad (4)$$

где i – номер строки, j – номер столбца, β_k – положительный коэффициент для k -той полуитерации, подби-

раемый для данного кода, исходя из минимума вероятности ошибки декодирования.

Следует заметить, что если алгоритм мягкого декодирования строк и столбцов, а также формула (2) для получения внешней информации, хорошо обоснованы теоретически [1], то формула (3) для формирования входной информации следующего элементарного декодера, является эмпирической.

В связи с этим целесообразно исследовать различные варианты комбинирования матриц \mathbf{R}_0 , \mathbf{R}_k^{in} , \mathbf{R}_k^{out} для получения входной матрицы \mathbf{R}_{k+1}^{in} следующего элементарного декодера. В данной работе описываются результаты такого исследования.

При проведении анализа авторы используют класс блочных турбокодов, наиболее часто используемых на практике и обсуждаемых в литературе, для которых в качестве составляющих кодов используются расширенные коды Хэмминга [1]. Фирма АНА (США) выпускает несколько микросхем-кодеков для работы с кодами этого класса. В приводимых ниже численных примерах в качестве кодов C_1 , C_2 используется расширенный (16,11)-код Хэмминга с расстоянием $d = 4$, результирующий код обозначается $(16,11) \times (16,11) = (16,11)^2$ и имеет параметры $n = 256$, $k = 121$, $d = 16$. Малая длина рассматриваемого кода обусловлена практической задачей, в которой происходит обмен одиночными сообщениями малого размера.

Вторая задача, рассматриваемая в данной работе и весьма актуальная для систем реального времени, – это снижение вычислительной сложности алгоритма декодирования. В стандартном алгоритме при мягком декодировании строк (столбцов) переоцениваются все символы строки (столбца), при этом вычисления для каждого символа независимы. В работе предлагается метод снижения вычислительной сложности, которое достигается за счет уменьшения числа переоцениваемых символов на каждой полуитерации алгоритма. Проводится сравнение эффективности описанных вариантов алгоритма при различном числе переоцениваемых символов. Показано, что при уменьшении числа переоцениваемых символов вероятность ошибки в предлагаемых вариантах алгоритма увеличивается незначительно и более медленно, чем в стандартном алгоритме.

Варианты обмена информацией между полуитерациями декодера

При исследовании различных вариантов комбинирования матриц \mathbf{R}_0 , \mathbf{R}_k^{in} , \mathbf{R}_k^{out} для получения входной матрицы \mathbf{R}_{k+1}^{in} следующего элементарного декодера структурную схему рис. 2 удобно представить в виде, изображенном на рис. 3.

На рисунке на первое место вынесен блок мягкого декодирования строк (столбцов), а комбинирование массивов \mathbf{R}_0 , \mathbf{R}_k^{in} , \mathbf{R}_k^{out} реализуется во втором блоке. Менее детальное изображение второго блока по сравнению со схемой рис. 2 позволяет использовать такую структурную схему для различных вариантов вычисления

входной информации для очередной полуитерации.

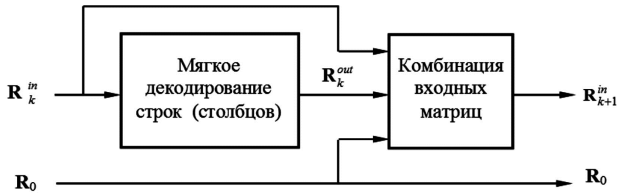


Рис. 3. Альтернативная структурная схема элементарного декодера

В работе рассматриваются следующие варианты операций между элементарными декодерами.

1. Для вычисления дополнительной информации, передаваемой следующему элементарному декодеру, используется одно из следующих выражений

$$W_{k+1} = R_k^{out} - R_k^{in} \quad (5)$$

$$V_{k+1} = R_k^{out} - R_0 \quad (6)$$

По формуле (5) рассчитывается внешняя информация, полученная на k -ой полуитерации стандартного алгоритма. Эта формула является классической и приведена практически во всех работах по рассматриваемой тематике. В формуле (6) вычисляется вся внешняя информация, полученная за все полуитерации до текущей.

2. Для вычисления выходной информации второго блока, являющейся входной матрицей для следующего элементарного декодера, используются следующие выражения:

$$R_{k+1}^{in} = R_0 + \alpha_{k+1} W_{k+1} \quad (7)$$

$$R_{k+1}^{in} = R_k^{in} + \alpha_{k+1} W_{k+1} \quad (8)$$

$$R_{k+1}^{in} = R_0 + \alpha_{k+1} R_k^{out} \quad (9)$$

Первый вариант (7) является классическим, если внешняя информация вычисляется по формуле (5). Второй вариант (8) при использовании формулы (5) сводится к вычислению линейной комбинации входной и выходной информации текущей полуитерации и не требует сохранения матрицы R_0 во время всего процесса декодирования:

$$R_{k+1}^{in} = (1 - \alpha_{k+1}) R_k^{in} + \alpha_{k+1} R_k^{out}$$

В третьем варианте (9) усиливается влияние входных данных R_0 , поскольку при расчете матрицы R_k^{out}

входные данные уже учтены. При этом матрица внешней информации не используется.

3. Оптимальные значения коэффициентов α_k и β_k в формулах (4)-(9) зависят от конкретного кода и деталей реализации алгоритма. Коэффициенты для исходного алгоритма, используемые ниже, взяты из рекомендаций, приведенных в литературе [1]. Коэффициенты для модифицированных вариантов алгоритма выбраны эмпирическим путем.

Конкретные варианты алгоритма декодирования, анализируемые в данной работе, представлены в табл. 1. В первом столбце приведено используемое далее обозначение вариантов алгоритма заглавными латинскими буквами. Стандартный алгоритм обозначен буквой А. Второй и третий столбцы содержат описанные формулы (5)-(9), задающие вариант алгоритма. Из сравнения этих формул нетрудно видеть, что вычислительная сложность модифицированных алгоритмов не выше, чем стандартного, а для алгоритма D даже ниже.

В столбце 4 приведены выражения для входной матрицы элементарного декодера через матрицу на входе турбодекодера R_0 и матрицы внешней информации W_k , которые обычно используются в стандартных алгоритмах турбодекодирования кодов произведения. Приведенные формулы нетрудно доказать методом математической индукции. Эти формулы наглядно демонстрируют отличие предлагаемых вариантов алгоритма от стандартного. Оно заключается в том, что в новых вариантах для формирования входа очередного элементарного декодера используется не только внешняя информация текущей итерации, но и линейная комбинация внешней информации всех предыдущих итераций. Поэтому такие алгоритмы в данной работе называются алгоритмами с накоплением внешней информации.

Основными характеристиками алгоритмов декодирования являются зависимости вероятности символьных P_S и блочных P_B ошибок от отношения E_b / N_0 энергии передаваемого бита информации E_b к спектральной плотности мощности шума N_0 . Для описанных выше турбодекодеров эти характеристики получены с помощью имитационной модели методом статистического моделирования.

Таблица 1. Варианты алгоритма декодирования

Обозначение алгоритма	Вычисление внешней информации	Вычисление выходного массива	
		по формулам (7)-(9)	выражение через R_0 и W_k
1	2	3	4
A	$W_{k+1} = R_k^{out} - R_k^{in}$	$R_{k+1}^{in} = R_0 + \alpha_{k+1} W_{k+1}$	
B	$W_{k+1} = R_k^{out} - R_k^{in}$	$R_{k+1}^{in} = R_k^{in} + \alpha_{k+1} W_{k+1}$	$R_{k+1}^{in} = R_0 + \sum_{i=2}^{k+1} \alpha_i W_i$
C	$V_{k+1} = R_k^{out} - R_0$	$R_{k+1}^{in} = R_0 + \alpha_{k+1} V_{k+1}$	$R_{k+1}^{in} = R_0 + \sum_{i=2}^{k+1} (\prod_{j=i}^{k+1} \alpha_j) W_i$
D	Не рассчитывается	$R_{k+1}^{in} = R_0 + \alpha_{k+1} R_k^{out}$	$R_{k+1}^{in} = [1 + \alpha_{k+1} \sum_{i=2}^{k+1} (\prod_{j=i}^{k+1} \alpha_j)] R_0 + \sum_{i=2}^{k+1} (\prod_{j=i}^{k+1} \alpha_j) W_i$

Таблица 2. Параметры алгоритмов турбодекодирования

Обозначение алгоритма	Коэффициенты α_{2k-1}	Коэффициенты β_{2k-1}	Вероятность ошибки для $E_b / N_0 = 3$ дБ *		Среднее число итераций
			P_S	P_B	
1	2	3	4	5	6
A	0.2, 0.3, 0.5, 0.7, 0.9, 1.0, 1.0, 1.0	0.2, 0.4, 0.6, 0.8, 1.0, 1.0, 1.0, 1.0	$2,18 \cdot 10^{-4}$	$1,23 \cdot 10^{-3}$	1,76
B	0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8	$\beta \equiv 1$	$5,64 \cdot 10^{-4}$	$7,55 \cdot 10^{-3}$	2,11
C	0.4, 0.4, 0.4, 0.8, 0.95, 1.0, 1.0, 1.0	$\beta \equiv 1$	$1,29 \cdot 10^{-4}$	$1,69 \cdot 10^{-3}$	1,38
D	0.4, 0.4, 0.4, 0.8, 0.95, 1.0, 1.0, 1.0	$\beta \equiv 1$	$1,60 \cdot 10^{-4}$	$2,36 \cdot 10^{-3}$	1,5

* Числовые данные приведены для $N_{iter} = 8$.

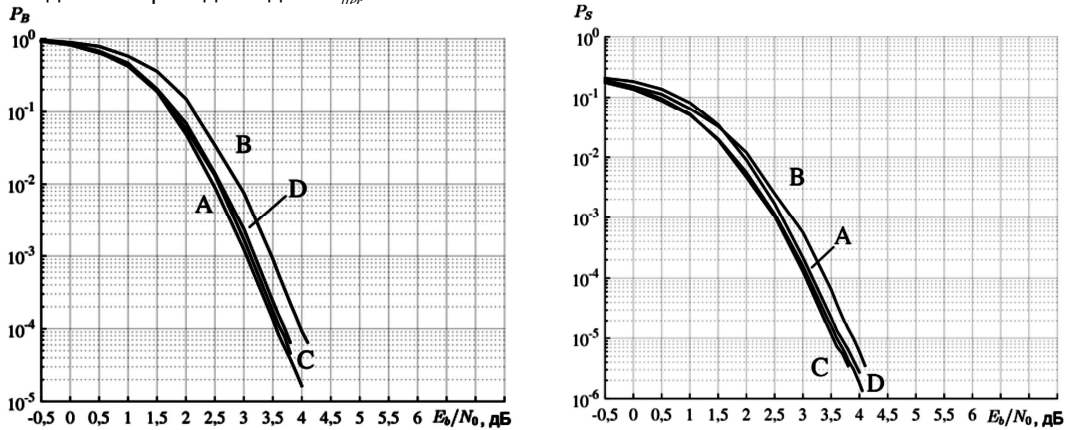


Рис. 4. Зависимости $P_B(E_b / N_0)$ и $P_S(E_b / N_0)$ турбодекодеров при $N_{iter} = 8$

Модель реализована на языке C++ в среде Microsoft Visual C++. Характеристики рассчитаны для описанного ранее кода $(16, 11)^2$, полученного каскадированием расширенных кодов Хэмминга. В модели имитируется канал связи с аддитивным белым гауссовским шумом, а при передаче используется двоичная фазовая модуляция. Максимальное число итераций турбодекодера N_{iter} принято равным 8. Расчет статистики символьных ошибок и ошибок передачи сообщений производится по информационной части кодовой матрицы. Для каждого обмена формируется отдельная матрица информации с равномерным распределением символов ± 1 . Приведенные ниже результаты соответствуют диапазону вероятностей ошибки на символ $P_S > 10^{-5} - 10^{-6}$ в соответствии с требованиями прикладной задачи, вызвавшей настоящее исследование.

Моделирование останавливается при 100 ошибочных приемах сообщений. При доверительной вероятности 0,8 (0,9; 0,95; 0,99) это число соответствует доверительному интервалу 13 % (соответственно, 16 %; 20 %; 26 %) [2].

В табл. 2 приведены параметры алгоритмов декодирования, позволяющие вместе с формулами табл. 1 однозначно определить все операции алгоритма. В столбцах 2 и 3 таблицы даны используемые коэффициенты для нечетных значений индексов α_{2k-1} , β_{2k-1} , $k = 1, 2, \dots$. Коэффициенты для четных значений определяются условием их постоянства в пределах итерации, т.е. коэффициент для четного значения индекса равен предыдущему нечетному: $\alpha_{2k} = \alpha_{2k-1}$, $k = 1, 2, \dots$.

В столбцах 4 и 5 даны для сравнения численные значения вероятности символьных P_S и блоковых P_B

ошибок при отношении $E_b / N_0 = 3$ дБ, а на рис. 4 – вероятностные характеристики рассматриваемых турбодекодеров.

Согласно рис. 4 зависимости $P_B(E_b / N_0)$ и $P_S(E_b / N_0)$ имеют примерно одинаковый характер. Отличия в отношении E_b / N_0 по уровню вероятности $P_B = 10^{-4}$ или $P_S = 10^{-5}$ исходного и модифицированных алгоритмов не превышает 0,1-0,2 дБ для алгоритмов C, D и 0,5 дБ для алгоритма B. Такие отличия в отношении E_b / N_0 во многих приложениях являются допустимыми. Отметим, что в некоторых случаях алгоритмы C и D превосходят стандартный алгоритм A. Проигрыш в отношении E_b / N_0 будем называть далее энергетическими потерями и обозначать $\Delta(E_b / N_0)$. Преимущества модифицированных алгоритмов проявляются при сокращении объема вычислений по методу следующего раздела.

Вероятностные характеристики алгоритмов турбодекодирования при снижении вычислительной сложности

Для снижения вычислительной сложности алгоритмов турбодекодирования в настоящей работе предлагается на каждой полуитерации алгоритма производить переоценку не всех символов, а некоторого сниженного их количества в строке и столбце до значения n_e , $n_e < \min(n_1, n_2)$. При этом для переоценки выбираются наименее надежные символы. Объем вычислений при такой переоценке снижается в n_1 / n_e раз для строки и в n_2 / n_e раз для столбца.

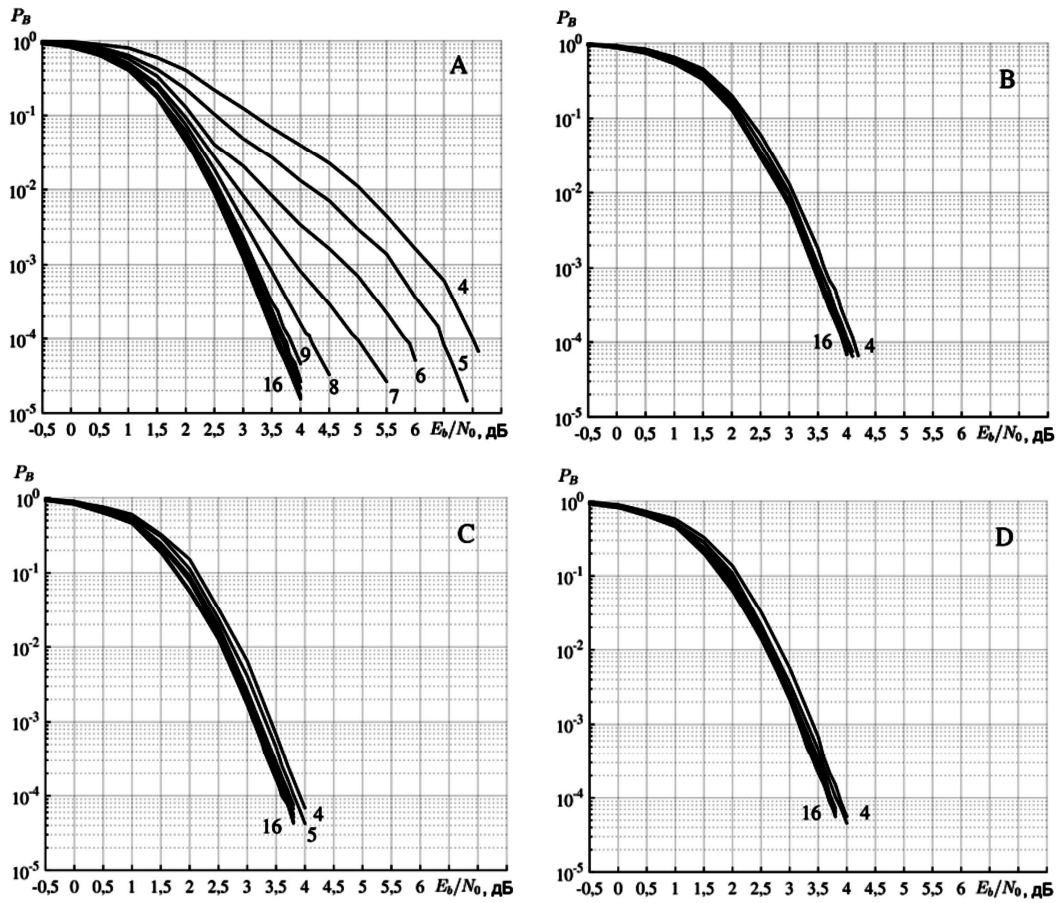


Рис. 5. Зависимости $P_B(E_b / N_0)$ турбодекодеров для $n_e = 4...16$ при $N_{iter} = 8$

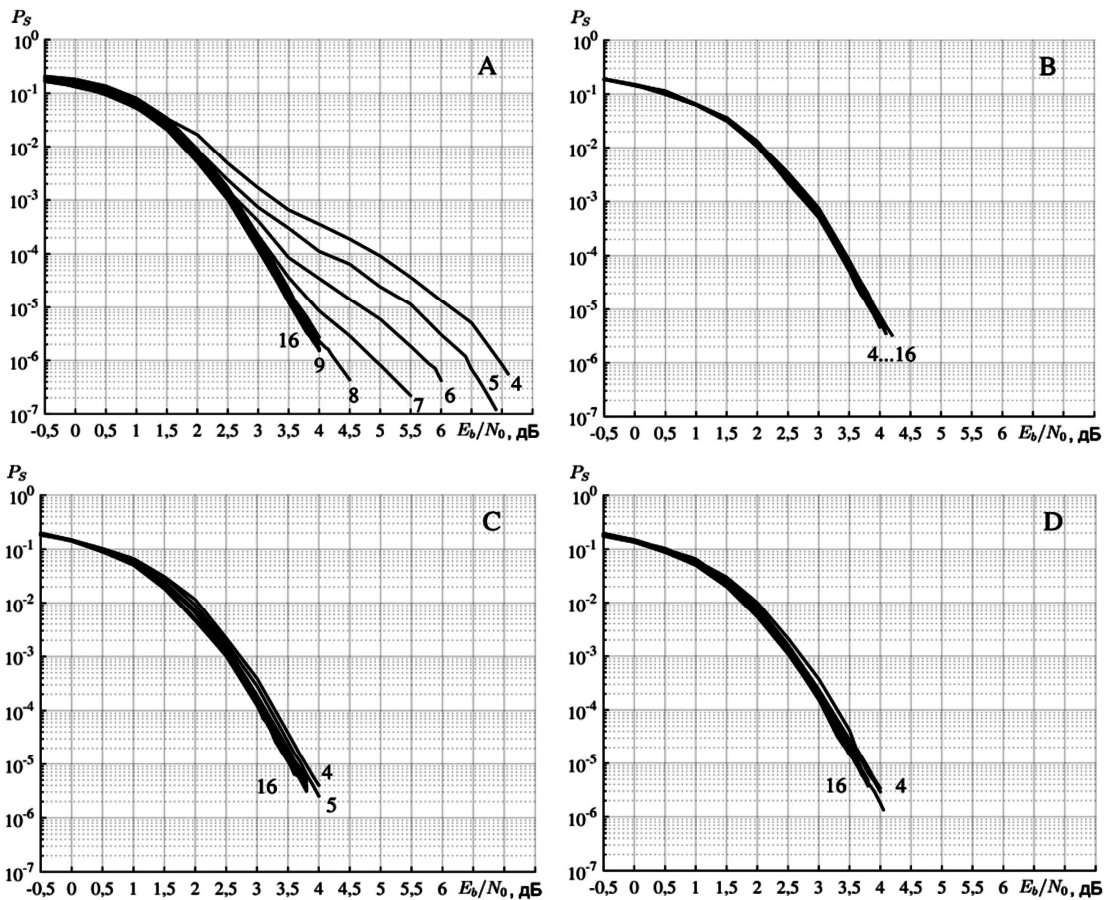


Рис. 6. Зависимости $P_S(E_b / N_0)$ турбодекодеров для $n_e = 4...16$ при $N_{iter} = 8$

Рассмотрим влияние на вероятностные характеристики алгоритмов турбодекодирования числа переоцениваемых символов n_e . На рис. 5 и рис. 6 для алгоритмов табл. 1 представлены семейства вероятностных характеристик P_S и P_B при различных значениях параметра n_e для кода $(16,11)^2$.

Из анализа рис. 5 можно сделать следующие выводы. Исходный алгоритм A имеет значительную чувствительность к уменьшению числа n_e . Разница в отношении E_b / N_0 для вероятности $P_B = 10^{-4}$ превышает 3 дБ при уменьшении числа n_e в пределах от 16 до 4. Модифицированные алгоритмы B , C и D обладают значительно меньшей чувствительностью к вариации числа n_e . Разница в отношении E_b / N_0 для вероятности $P_B = 10^{-4}$ при уменьшении числа n_e в пределах от 16 до 4 не превышает 0,25 дБ и 0,3 дБ для алгоритмов B и D соответственно, и 0,4 дБ для алгоритма C .

Аналогичные соотношения имеют место и для вероятности символьных ошибок. Подобное поведение вероятностных характеристик модифицированных алгоритмов показывает, что новые алгоритмы являются более предпочтительными при необходимости сократить вычислительную сложность предлагаемым методом.

Следует отметить, что нечувствительность модифицированных алгоритмов к уменьшению числа n_e достигается путем аккуратного выбора коэффициентов $\alpha_k, \beta_k, k = 1, 2, \dots$. В стандартном алгоритме указанной нечувствительности достичь не удастся. Такое отличие объясняется тем, что в модифицированных алгоритмах используется накопление внешней информации от всех полуитераций до текущей, а в стандартном алгоритме учитывается только внешняя информация, полученная на предыдущей полуитерации.

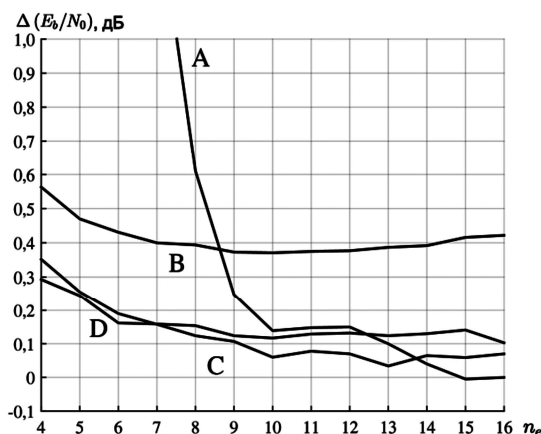


Рис. 7. Энергетические потери при вариации числа n_e для вероятности $P_B = 10^{-4}$

Рассмотрим энергетические потери при снижении числа n_e . На рис. 7 представлены типичные зависимости энергетических потерь $\Delta(E_b / N_0)$ от числа n_e по сравнению со стандартным алгоритмом A с номинальными параметрами $N_{iter} = 8, n_e = 16$. Зависимости приведены для фиксированной вероятности $P_B = 10^{-4}$. Для других значений вероятностей P_B и P_S графики энер-

гетических потерь имеют аналогичный вид.

Из рис. 7 следует, что при числе $n_e \leq 12$ предлагаемые алгоритмы B, C, D имеют меньшие энергетические потери, чем стандартный алгоритм, и разница тем больше, чем меньше число n_e . Причем, модифицированные алгоритмы B, C и D дают приемлемые потери даже при снижении числа n_e в 4 раза. В табл. 3 приведены значения энергетических потерь алгоритмов B, C и D при вероятности $P_B = 10^{-4}$ для нескольких характерных точек на рис. 7.

Таблица 3. Энергетические потери

№	$\Delta(E_b / N_0)$, дБ	Алгоритм	n_e
1	0,1	C	9
2	0,15	D	9
3	0,2	C	6
4	0,2	D	6
4	0,25	D	5
5	0,3	D	4
6	0,35	C	4
7	0,4	B	7
8	0,6	B	4

Модифицированные алгоритмы C и D способны работать с энергетическими потерями не более 0,3-0,35 дБ, а алгоритм B с энергетическими потерями не более 0,6 дБ при уменьшении числа n_e до 4. Численное значение снижения вычислительной сложности модифицированных алгоритмов, по сравнению со стандартным алгоритмом, зависит от способа реализации и методики оценки этого параметра.

Заключение

В работе обосновываются два предложения по модификации алгоритма турбодекодирования блочных кодов производства. Первое из них заключается в изменении способа обмена информацией между элементарными декодерами. Предложенные варианты обмена приводят к накоплению внешней информации с некоторыми весами за все выполненные итерации, в то время как в стандартном алгоритме такого накопления нет.

Второе предложение сводится к сокращению числа переоцениваемых символов на каждой итерации для снижения вычислительной сложности алгоритма.

Оба изменения целесообразно использовать совместно, так как характеристики алгоритмов с накоплением внешней информации имеют слабую чувствительность к числу переоцениваемых символов в отличие от стандартного алгоритма. Это свойство открывает пути к повышению скорости передачи информации при работе в реальном масштабе времени.

Литература

1. Channel coding in communication networks: from theory to turbo codes/edited by Alain Glavieux. 1sted. ISTE Ltd, 2007. См. также приведенную в монографии литературу.
2. Зайцев Г.В. Потери в отношении сигнал/шум в системе связи, использующей двоичную дифференциальную фазовую модуляцию и ортогональное кодирование, Цифровая обработка сигналов, 2014. № 2. С. 15-21.