УДК 621.397.7

Ш.Ш. Атаджанов, А.А.Турсунова

РАЗРАБОТКА ПОМЕХОУСТОЙЧИВЫХ КОДОВ НА ОСНОВЕ АЛГОРИТМА ИТЕРАТИВНОГО КОДИРОВАНИЯ И ДЕКОДИРОВАНИЯ

В статье рассматривается принципы построение высокоточных итеративных кодов на основе блочных и сверточных кодов. Разрабатывается итеративные алгоритмы помехоустойчивого кодирования и декодирования, позволяющие повысить помехоустойчивости приема сигналов при низких отношениях сигнал-шум (энергии, приходящийся на один бит).

Ключевые слова: итеративный код, итеративный сверхточной код, блочный код, итеративное кодирование, декодирование, кодовое расстояние.

Введение

Впервые в работах К.Шеннона [1] систематически предложены основополагающие аспекты теории информации и кодирования. Шеннон предложил использовать компрессию данных с источника, шифровать данные, затем применять помехоустойчивое кодирование (ПК), дальнейшее развитие которого связано с работами А.Колмогорова, В.Котельникова, Р.Хеминга, П.Элайеса, Р.Галлагера, И.Рида, Г.Соломона, Д.Форни, К.Берру, Д.Маккея и других ученых, которые решили различные проблемы в области передачи информации, разработав технологии, которые позволяют достичь рекордных показателей BER (Bit Error Rate – вероятность ошибки на бит). Совершенствование средств защиты информации на основе ПК особенно важно и актуально.

Важным событием в ПК стала фундаментальная работа 1993 года [2], в которой французские ученые под руководством К.Берру представили термин “турбо” и математический аппарат для работы с параллельной каскадной конкатенацией сверточных кодов (ПККСК или турбокод).

Применение ПК на основе итеративных алгоритмов в цифровых системах передачи данных также позволяет получить энергетический выигрыш кодирования (ЭВК), каждый децибел которого специалистами очень высоко оценивается, поскольку его можно использовать для уменьшения мощности передатчика, повышения скорости передачи данных, уменьшения размеров антенн, повышения дальности связи, экономии полосы пропусканияи улучшения многих других важных свойств систем передачи данных.

Основная часть

Рассмотрим схему кодирования, которая может быть построена следующим образом.

Предположим, что нужно передать девять информационных символов $k = (k_0 , k_1 , ..., k_8)$. Эти символы можно расположить в виде квадратной матрицы, как это показано в табл.1, и добавить к каждой строке и каждому столбцу этой таблицы по проверочному символу (проверка на четность).

Таблица 1

Информационные символы, расположенные в виде квадратной матрицы

k_0	k_1	k_2	$Q_1=m_0+m_1+m_2$
k_3	k_4	k_5	$Q_2=m_3+m_4+m_5$
k_6	k_7	k_8	$Q_3=m_6+m_7+m_8$
$Q_4=m_0+m_3+m_6$	$Q_5=m_1+m_4+m_7$	$Q_6=m_2+m_5+m_8$	$m_0+m_0+m_1+m_1+\ldots.+m_8+m_8$

По строкам и по столбцам этой таблицы будет выполняться правило четности единиц.

Если в процессе передачи по каналу с

помехами в этой таблице произойдет одна ошибка (например в символе k_4), то проверка на четность в соответствующей строке и столбце (в нашем примере – Q_2 и Q_5) не будет выполняться. Координаты ошибки однозначно определяются номерами столбца и строки, в которых не выполняются проверки на четность. Таким образом, этот код, используя различные проверки на четность (по строкам и по столбцам), способен не только обнаруживать, но и исправлять ошибки (если известны координаты ошибки, то ее исправление состоит просто в замене символа на противоположный: если 0, то на 1, если 1 – то на 0).

Описанный метод называется итеративным кодированием (ИКд). ИКд оказывается полезным в случае, когда данные естественным образом формируются в виде массивов, например, в прост-ранстве: сигналы модуляции в цифровом телевидении (ЦТВ) COFDM, QAM, QPSK, MFSK, мультимедийные сигналы и др, а на кабелях: данные на шинах ЭВМ, в памяти, имеющей табличную структуру, и т.д. При этом размер таблицы в принципе не имеет значения (3x3 или 20x20), однако в первом случае будет исправляться одна ошибка на 3x3=9 символов, а во втором – на 20x20=400 символов. В простом коде с проверкой на четность для обнаружения ошибки приходится добавлять к информационной последовательности всего один символ, а для того, чтобы код стал исправлять однократную ошибку, к девяти информационным символам понадобится добавить еще семь проверочных. Таким образом, избыточность этого кода оказывается очень большой, а исправляющая способность сравнительно низкой. Поэтому научные труды и усилия специалистов в области помехоустойчивого кодирования всегда были направлены на поиск таких кодов и методов кодирования и декодирования, которые при минимальной избыточности обеспечивали бы максимальную исправляющую способность.

Итеративный код (ИК). Предположим, что задано сообщение $\{Q_1, Q_2, Q_3,\ldots\}$, образованное последовательностью символов Q_i, выбираемых из конечного алфавита K. В системе используются двоичные сигналы из элементов 0 и 1. Помехоустой-чивость передачи цифровых последовательностей на расстояние может быть обеспечена за счет избыточности кодов.

Если продублировать символ Q, составив кодовую посылку Q, Q, то можно выявить одну ошибку. Однако при дублировании установить место возникновения ошибки не удается, при этом возможно лишь обнаружение, но не исправление ошибки. Для исправления (выявления и локализация) ошибки символ Q должен быть по крайней мере повторен трижды: Q, Q, Q. Для исправления двух ошибок – повторен пять раз и т.д. Для исправления t ошибок символ повторяется $n=2t+1$ раз.

Недостаток n-кратного повторения – слишком большая избыточность при передаче цифровых сигналов. Основная идея возможности коррекции ошибки на основе итеративных методов состоит в том, что используемый кодовый набор должен состоять из слов, отличающихся в $n=2t+1$ числе символов. Здесь важный количественный показатель, характеризующее помехоустойчивость является минимальное кодовое расстояние d.

Основной путь повышения помехоустойчивости – не n-кратное повторение, а расширение сообщения $Q = \{Q_1, Q_2, \ldots\}$ путем введения дополнительных символов m_i, называемых контрольными или проверочными и связанных с информационными символами Q_i той или иной функциональной зависимостью. Один из способов выбора проверочных символов m_j в процессе кодирования заключается в том, что символы m_j для некоторого блока Q определяют лишь по k информационным символам данного блока $m_j=f(Q)$, где f_j – произвольные функции от k аргументов Q_i. В этом случае набор из $n=k+m$ символов называется кодовым словом, а совокупность всех возможных кодовых слов –

итеративным блоковым кодом (ИБК). Кодирование таких кодовых слов называется высокоточным итеративным кодированием (ВИКд).

Информационные символы внутри блока Q нумеруются справа налево:Q= (Q_k, Q_{k-1}, …,Q_1), где Q_iÎ K, i=1, 2, …, k–1. Так как |K|=q, где q–количество кодовых слов ИБК, q^k – число всевозможных сообщений (мощность алфавита K).

В итеративном кодировании информационные символы входят в каждое кодовое слово в неизменном виде, а их положение фиксировано, например, информационные символы расположены в начале кодового слова, а проверочные – приписываются к ним справа. Такие ИБК называются систематическими.

При передаче цифровых сигналов на расстояние по некоторому каналу связи оно может исказиться под воздействием помех. Если позиция, в которой исказилось цифровое последовательность, известна, а не известна лишь величина искажения, то такое искажение называется стиранием. Если не известны ни место, ни величина искажений, их называют ошибками. Ошибки в каналах имеют тенденцию группироваться. Последовательность n искаженных символов, первый и последний символ которой не равны нулю, называется пакетом ошибок длины n.

Возможен вариант, когда все n символов, а не только проверочные, кодового слова W = (w_n, w_{n-1}, …, w_1) является некоторыми функциями от k информационных символов:

$$w_i=F_i(Q),\ i = 1,\ 2,\ \ldots,\ n \qquad (1)$$

Итак, переход Q® W от сообщения Q из k информационных символов Q_iÎ K к кодовому слову W длины n называется итеративным блоковым кодированием. Отношения k/n принято называть скоростью кодирования, n – длиной, а k – размерностью ИБК.

Количество ненулевых компонент кодового слова W называется его весом и обозначается wt (W), а число позиций, в которых различаются два кодовых слова W'=(w_n', w_{n-1}', … , w_1') и W''= (w_n'', w_{n-1}'', … , w_1''), называются кодовым расстоя-нием и обозначается dist (W', W'')

Минимальным кодовым расстоянием (МКР) d ИБК называется минимальное расстояние между собой парой его кодовых слов:

$$d = \min_{i \neq j} \text{dist}\,(W_i, W_j), \qquad (2)$$

где минимум берется по всем парам (W_i, W_j) кодовых слов.

Таким образом, любые два слова ИБК с минимальным расстоянием d различаются по крайней мере в d позициях. Например, dist[(01101), (11011)]=3, dist[(0213),(1233)]=2.

Возможность контролировать ошибки – обнаруживать (без исправления) или исправлять их с помощью ИБК – определяется его минимальным кодовым расстоянием.

Утверждение. ИК с минимальным кодовым расстоянием d позволяет обнаруживать $d-1$ ошибок или одновременно исправлять произвольную конфигурацию из τ стираний и t ошибок при условии, что

$$d \geq 2t+\tau+1 \qquad (3)$$

Таким образом, ИК с минимальным расстоянием d позволяет исправить либо $\tau = d-1$ стираний, либо t = int [0,5(d – 1)] ошибок, если – четное число, то можно не только исправить 0,5(d – 1), но и одновременно обнаружить 0,5d ошибок.

Если в структуру цифровой последовательности не введены избыточные проверочные коды и минимальное кодовое расстояние подобного безызбыточного кода, образованного множеством сообщений Q_i , равно 1, то из формулы (3) следует, что он не способен контролировать ошибки и не является итеративным кодом ($\tau = t = 0$).

В нашей республике в качестве формата вещания применяется MPEG-4, для канала связи соответствует АБГШ (аддитивным белым гауссова шумом – AWG), а в качестве вида модуляции используется ортогональная COFDM, QAMи

иногда (для спутникого ЦТВ) применяется BPSK. Поэтому, при разработке математического аппарата, приводящему к определению вероятности ошибки (ВО), математические преобразовании и формулы должны быть написаны таким образом, чтобы эти преобразовании учитывали вышеуказанных параметров ЦТВ. Набор равно энергетических сигналов $s_i(t)$=1, 2, …, M, будет ортонормированным (ортогональным и нормиро-ванным на 1) тогда и только тогда, когда [3]

$$z_{ij} = \frac{1}{E}\int_0^T s_i(t)s_j(t)dt = \begin{cases} 1 & \text{при } i = j \\ 0 & \text{при } i \neq j \end{cases} \quad (4)$$

где z_{ij} – коэффициент взаимной корреляции, E – энергия сигнала

$$E = \int_0^T s_i^2(t)dt \quad (5)$$

Для одинаковых, равноэнергетических ортогональных сиг-налов (РЭОС) вероятность ошибки в кодовом символе P_E, можно оценить сверху, как

$$P_E(M) \le (M-1)Q\left(\sqrt{\frac{E_s}{N_0}}\right), \quad (6)$$

где размер набора кодовых слов M равен 2^k, k – число информационных бит в кодовом слове.

Для определения вероятности появления ошибочного бита можно использовать связь между P_B и P_E, которая:

$$\frac{P_B(k)}{P_E(k)} = \frac{2^{k-1}}{2^k - 1} \quad \text{или} \quad \frac{P_B(M)}{P_E(M)} = \frac{M/2}{M-1} \quad (7)$$

В результате объединении уравнений (6) и (7) вероятность появления ошибочного бита можно оценить следующим образом:

$$P_k(k) \le (2^k - 1)Q\left(\sqrt{\frac{kE_b}{N_0}}\right) \quad \text{или} \quad P_B(M) \le \left(\frac{M}{2}\right)Q\left(\sqrt{\frac{kE_s}{N_0}}\right) \quad (8)$$

Для одинаковых, равноэнергетических биортогональных сигналов (РЭБС) ВО в кодовом слове (символе) можно оценить [10] следующим образом:

$$P_E(M) \le (M-2)Q\left(\sqrt{\frac{E_s}{N_0}}\right) + Q\left(\sqrt{\frac{2E_s}{N_0}}\right) \quad (9)$$

При фиксированном M с ростом E_b/N_0 оценка становится все более точной. Зависимость $P_B(M)$ и $P_E(M)$ можно аппроксимировать следующим образом:

$$P_B(M) \approx \frac{P_E(M)}{2} \quad (10)$$

Биортогональные коды (БОК) значительно снижают P_B по сравнению с ортогональными кодами (ОК) и требуют только половину пропускания ортогональных кодов.

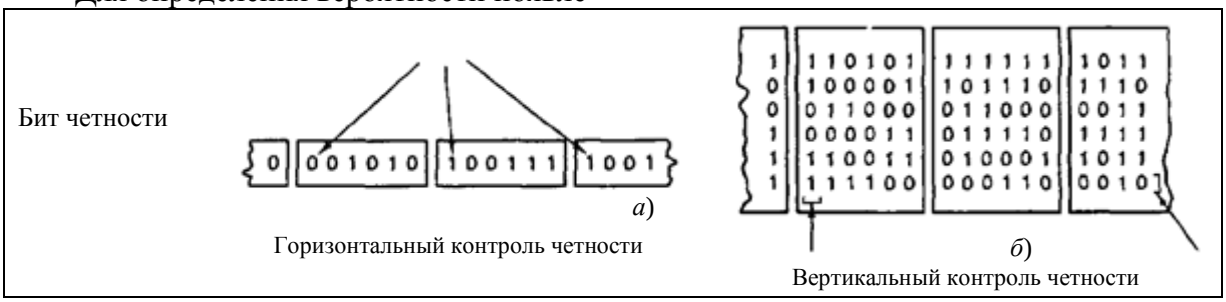

Рис.1. Проверка четности для последовательной (*а*) и параллельной (*б*)структуры итеративного кода

ИК для обнаружения или исправления ошибок используют линейные суммы информационных битов. На рис.1. показано выполнение правило четности единиц по строкам и по столбцам.

На рис.1, *а* приведена последовательная передача данных (первым является крайний справа бит). К каждому блоку добавляется один бит четности (крайний слева бит в каждом блоке), дающий положительную четность. В приемном устройстве производится декодирование, заключающееся в проверке, дают ли нуль суммы принятых битов кодового слова по модулю 2 (положительная четность). Если полученный результат равен 1, то кодовое слово содержит ошибки. Параллельная структура проверка четности состоит из горизонтального и вертикального контроля четности в информационных битах (рис.1, *б*)

Предполагая, что ошибки во всех разрядах равновероятны и появляются независимо, можно записать вероятность

появления j ошибок в блоке из n символов:

$$P(j,n)=\binom{n}{j}p^{j}(1-p)^{n-j} \quad (11)$$

здесь p – вероятность получения канального символа с ошибкой, а через

$$\binom{n}{j}=\frac{n!}{j!(n-j)!} \quad (12)$$

обозначается число различных способов выбора из n бит j ошибочных. Для итеративного кода с одним битом четности вероятность необнаруженной ошибки P_{nd} в блоке из n бит вычисляется следующим образом:

$$P_{nd}=\sum_{j=1}^{n/2(\text{при } n=\text{четное})\,/\,(n-1)/2(\text{при } n=\text{нечетное})}\binom{n}{2j}p^{2j}(1-p)^{n-2j} \quad (13)$$

Исходя из вероятности наличия j ошибок в блоке из n символов, записанной в (6), можно записать вероятность ошибки сообщения для итеративного кода, который может исправить модели ошибок, состоящие из t или менее ошибочных битов:

$$P_{M}=\sum_{j=t+1}^{n}\binom{n}{j}p^{j}(1-p)^{n-2j} \quad (14)$$

Кодирование итеративных кодов (ВИКд). Математическое описание итеративного кодера. Итеративный сверточный код (ИСК). Для того чтобы задать структуру итеративного кодера, необходимо указать, какие разряды регистра сдвига связаны с каждым из сумматоров по модулю 2. Связи j-го сумматора по модулю 2 описываются путем задания j-й порождающей последовательности. В основе ИСК лежит последовательность символов, не разделяющихся на отдельные кодовые комбинации. Обозначая информационные символы через a_i, а корректирующие через b_i, получаем основное выражение последовательности символов сверхточного кода:

$$a_1b_1a_2b_2a_3b_3\ldots a_kb_ka_{k+1}b_{k+1}\ldots \quad (15)$$

Информационные символы определяются передаваемым сообщением, входной информационной последовательностью, а корректирующие формируются по следующему правилу:

$$b_i = a_{k-S} + a_{k+S+1} \ (\text{mod}2), \quad (16)$$

где s – произвольное целое число, называемое шагом кода.

Видно, что при ошибочном приеме некоторого корректирующего символа b_i соотношение (16) в принятой последовательности не будет выполнено для $i=k$. В случае же ошибочного приема информационного символа a_i соотношение (16) не будет выполняться при двух значениях k, а именно при $k_1 = i - s - 1$ и при $k_2 = i + s$. В принятой кодовой последовательности для каждого b_k проверяется соотношение (16). Если оно оказалось не выполненным при двух значениях k ($k = k_1$ и $k = k_2$), то при этом информативный элемент a_{k_1+S+1} должен быть заменен на противоположный.

$$k_2 - k_1 = 2s+1. \quad (17)$$

Очевидно, что избыточность такого кода равна 1/2. Он позволяет исправлять все ошибочно принятые символы, кроме некоторых неудачных сочетаний. Так, если s=0, он обеспечивает правильное декодирование, когда между двумя ошибочно принятыми символами имеется не менее трех правильно принятых символов.

Для выполнения данного условия в представленном методе реализована схема итеративного кодирования внешним кодом (рис.2 и рис.3).

В уравнении (18) представлена форма итеративных кодов через параметры n, k, t и некоторое положительное число m> 2.

$$(n, k) = (2^m - 1, 2^m - 1 - 2t) \quad (18)$$

здесь $n–k=2t$ – число контрольных символов, t – количество ошибочных битов в символе, которые может исправить итеративный код. При этом генерирующий полином для итеративного кода имеет следующий вид:

$$g(X) = g_0 + g_1X + g_2X^2 + \ldots + g_{2t-1}X^{2t-1} + X^{2t}. \quad (19)$$

$$X^{n-k}m(X)=q(X)g(X) + p(X). \quad (20)$$

здесь q(X) и p(X) – это частное и остаток от полиномиального деления. Как и в случае двоичного кодирования, остаток

будет четным. Уравнение (20) можно переписать следующим образом:

$$p(X)=X^{n-k}m(X)\bmod g(X) \qquad (21)$$

Основной результирующий полином кодового слова U(X) можно переписать следующим образом:

$$U(X) = p(X) + Xn\text{-}km(X) \qquad (22)$$

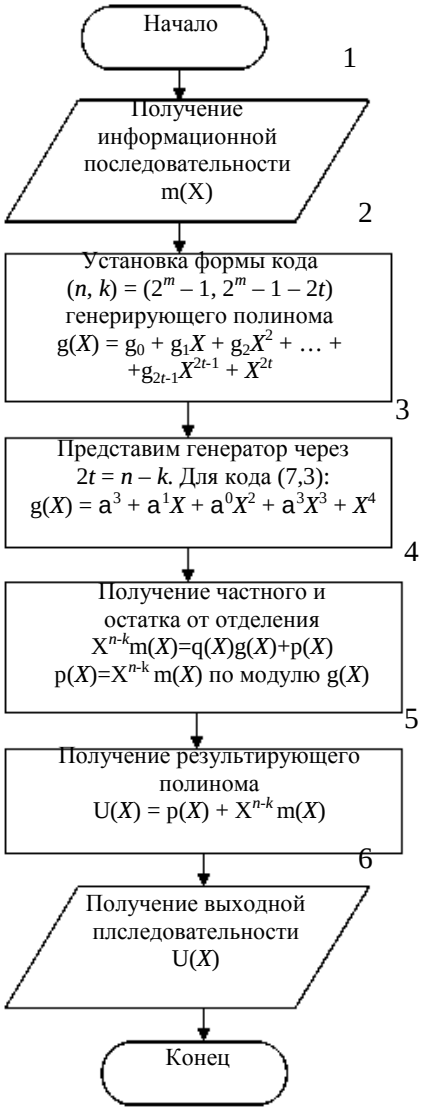

Рис.2. Алгоритм работы кодера ИБК.

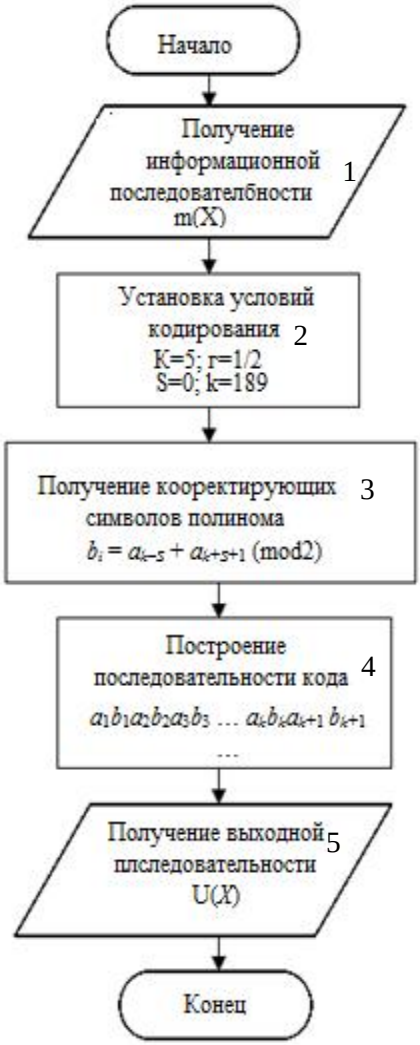

Рис.3. Алгоритм кодирования ИСК.

Декодирование итеративных кодов.

1.Использование MAP-алгоритм (Maximum Aposteriory Probability – максимум апостериорной вероятности). Декодирование ИК – это итеративный процесс, в ходе которого два декодера ИСК с мягким выходом обмениваются значениями оценок внешних вероятностей [2, 4, 5]. Обычно достаточно 8-10 итераций для того, чтобы изменения оценок декодированных символов стали незначительными, дальнейшее итерирование декодера практически не приводит к уменьшению вероятности ошибки. Одним из способов снижения вероятности ошибки является использование высокоточного итеративного декодирования (ВИДк).

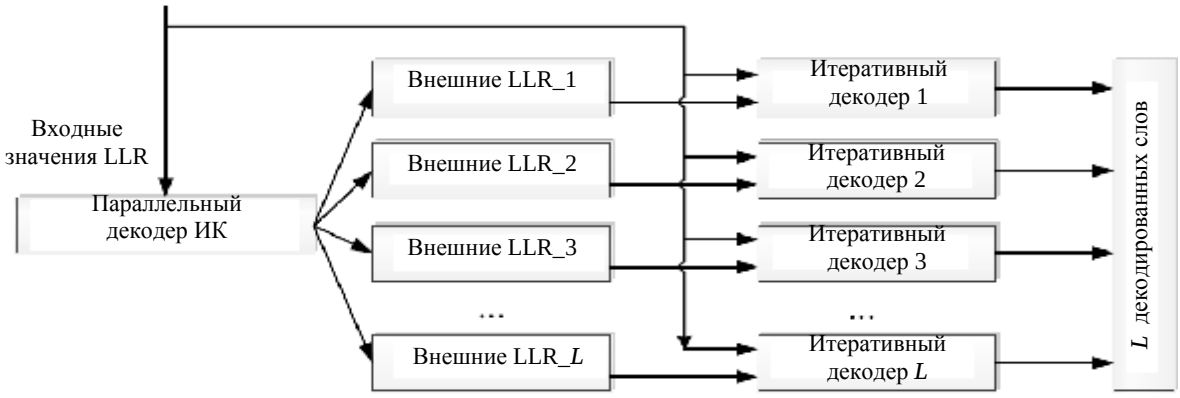

Рис.4. Схема параллельного декодирования ИК, основанный на ВИДк.

В работе описывается новый метод параллельного декодирования ИК, основанный на высокоточном итеративном декодере (ВИДк) ИСК с мягким выходом. Каждый мягкий выход является последовательностью априорных вероятностей, которые подаются на вход независимых ВИДк (рис.4). Предложенный алгоритм обеспечивает сходимость разных декодеров к различным кодовым словам, из которых затем выбирается подходящее.

Список декодированных слов с мягкими решениями может быть сгенерирован с использованием первого, второго или обоих ИСК.

Предлагаемый алгоритм строит на участках решетки (окнах) мягкие списки размером L, которые в дальнейшем используются при получении априорных вероятностей для всего информационного слова. Для получения такого результата в предложенном алгоритме используется списочный декодер Витерби и MAP-декодер (Maximum Aposteriory Probability – максимум апостериорной вероятности). При этом для оконного алгоритма Витерби вводят понятие суффикса, так как в текущем окне не известны начальные и конечные состояния пути (рис.5). Известно, что если длина суффикса N_{suff} равна 4-5 длинам кодового ограничения сверточного кода, то выжившие пути в конце суффикса, полученные с помощью алгоритма Витерби, с большой вероятностью имеют общий корень в конце окна (N_{win}).

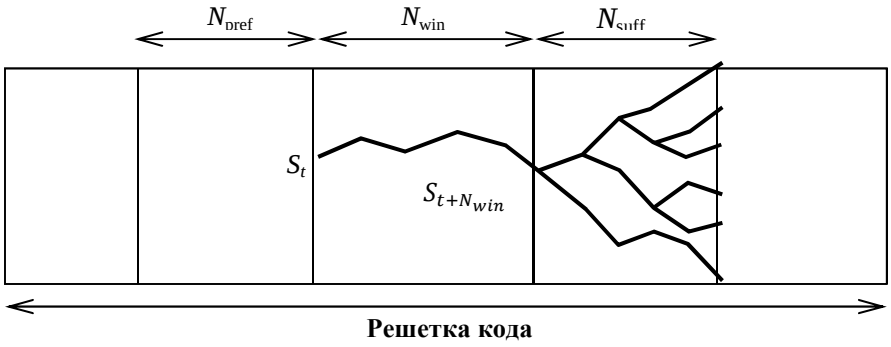

Рис.5. Введение понятия суффикса для определения оптимальных путей.

Таким образом, суффикс позволяет найти состояние в конце окна, в то время как начальные состояния равновероятны. Пусть t – номер начальной секции окна. Тогда оконный списочный алгоритм Витерби выглядит следующим образом.

1. Выполним параллельный списочный алгоритм Витерби [5, 6] на участке решетки от секции t до $t+N_{\text{win}}+N_{\text{suff}}$. В результате работы этого алгоритма получим для каждого состояния окна и суффикса L лучших путей.

2. Найдем путь с наибольшей конечной метрикой в секции $t+N_{win}+N_{suff}$. Пусть S_t– его начальное состояние в окне, а S_t+N_{win}– конечное.

3. Из состояния S_t+N_{win} выполним обратный проход по решетке в окне для оставшегося $L-1$ пути.

В результате работы алгоритма получаются L путей в окне, которые могут начинаться в произвольном состоянии решетки, но заканчиваются в состоянии $S_{t+N_{win}}$.

Поскольку в МАР-алгоритме выполняется два прохода по решетке для нахождения прямых и обратных метрик, в оконном варианте помимо суффикса вводят префикс, который служит для более корректного вычисления метрик в окне (рис.5), т.е. в оконном МАР [7] алгоритме расчет метрик начинается в секции $t-N_{pref}$ и заканчивается в секции $t+N_{win}+N_{suff}$, а начальные и конечные состояния равновероятны.

Используя полученные пути в окне и оконный МАР-алгоритм, можно получить список мягких решений, выполнив следующие шаги.

1. Найдем с помощью списочного оконного алгоритма Витерби L путей в решетке.

2. Определим первый элемент списка как результат работы алгоритма МАР в окне.

3. Обозначим как Γ_l все ребра, которые принадлежат l-му пути. Для нахождения l-го элемента списка:

- исключим в окне все ребра, которые принадлежат $l-1$ лучшему пути, но не принадлежат оставшимся $L-1$ путям, т.е. исключим все ребра из множества

$$\left(\bigcup_{i=1}^{l-1}\Gamma_i - \bigcup_{i=1}^{l-1}(\Gamma_l \cap \Gamma_i)\right)$$

- выполним МАР-алгоритм в окне с исключенными ребрами. Выходные надежности алгоритма и будут искомым элементом списка.

Удаление из решетки ребра лучших путей, которые будут учтены в соответствующих элементах списка, позволит рассмотреть менее вероятные решения. Это обеспечит схождение последующих процессов списочного высокоточного итеративного декодирования к другим кодовым словам, среди которых, возможно, будет правильное.

Стоит отметить, что помимо МАР-алгоритма можно использовать его подоптимальные варианты, такие как Max-Log-MAP или Scaled-Max-Log-MAP [6].

2. Алгоритм декодирования ИБК и ИСК. Рассмотрим операции декодирования кода Рида-Соломона. В данном случае принятый полином поврежденного кодового слова r(X) представляется в виде суммы полинома переданного кодового слова и полинома модели ошибки, как показано ниже:

$$r(X) = U(X) + e(X) \qquad (23)$$

Вычисление синдрома. Если r является членом набора, то синдром S имеет значение равное 0. Любое ненулевое значение S означает наличие ошибок, синдром S состоит из $n–k$ символов, $\{S_i\}$ ($i = 1, ..., n–k$):

$$U(X) = m(X)g(X) \qquad (24)$$

Из этой структуры можно видеть, что каждый правильный полином кодового слова U(X) является кратным полиномиальному генератору g(X).

Следовательно, корни g(X) также должны быть корнями U(X). Вычисления символов синдрома можно записать следующим образом:

$$S_i = r(X)\big|_{X=a^i} = r(a^i)\ i = 1, \ ...,\ n-k \qquad (25)$$

Локализация ошибки. Допустим, в кодовом слове имеется v ошибок, расположенных на позициях X^{j1}, X^{j2}, ..., X^{jv}. Тогда полином ошибок можно записать следующим образом:

$$e(X) = e_{j1}X^{j1} + e_{j2}X^{j2} + \ldots + e_{jv}X^{jv} \qquad (26)$$

Если вычислен ненулевой вектор синдрома, это означает, что была принята ошибка. Далее нужно узнать расположение ошибки. Полином локатора ошибок можно определить следующим образом:

$$\sigma(X)=(1 + \beta_1X)(1 + \beta_2X) \ldots (1 + \beta_vX) = 1 + \sigma_1X + +\sigma_2X^2 + \ldots + \sigma_vX^v. \qquad (27)$$

Корнями σ(X) будут $1/\beta_1$, $1/\beta_2$, …, $1/\beta_v$. Величины, обратные корням, будут представлять номера расположений моделей ошибки e(X).

Перфорация кода состоит в систематическом удалении из процесса передачи в канал некоторых битов (символов) с выхода низкоскоростного кодера. Матрица перфорации P задает правило удаления выходных символов. Матрица P есть k x n_p двоичная матрица, элементы которой p_{ij} указывают, что соответствующий выходной двоичный символ будет передан ($p_{ij} = 1$) или нет ($p_{ij} = 0$).

Для реализации данного алгоритма построена схема итеративного декодирования ИБК и ИСК (рис.6 и рис.7).

На основании указанных уравнений восстанавливаем принятый полином, выдавая в итоге предполагаемое переданное кодовое слово и в конечном счете декодированное сообщение.

$$\hat{U}(X) = r(X) + \hat{e}(X) = U(X) + e(X) + \hat{e}(X) \quad (28)$$

1. Полученная входная последовательность вначале проходит этап кодирования кодом ИБК (рис.2), далее следует алгоритм перемежения (депере-межения), осуществляющий псевдослучайную перестановку символов внешнего кода и соответственно восстановление исходного порядка символов на этапе декодирования.

2. Преобразованная последовательность кодируется внутренним ИСК, особенностью которого является включение в информационную последовательность проверочных символов (рис. 3).

3. Затем следует этап перфорирования информационной последовательности. Данная процедура определяет длину кода и соответственно устанавливает нужную скорость кодирования путем исключения из последовательности ряда элементов.

4. Деперфорация кода представляет собой восстановление исходной (до этапа перфорирования кода) последовательности путем анализа и сравнения информации с элементами матрицы перфорирования, определяющей порядок исключения элементов из кода.

5. Итеративное декодирование осуществляется в обратном порядке по отношению к процедуре кодирования. Декодирование ИСК основано на определении показателя синдрома ошибки путем анализа значений проверочных символов (рис.6).

6. После соответсвующего этапа деперемежения и декодера ИБК получаем искомую информационную последовательность (рис.7).

Рис.6. Алгоритм декодирования ИСК
Рис.7. Алгоритм декодирования ИБК

На основе данных алгоритмов, представленных в блок-схемах (рис.2,3 и рис.6, 7) отдельных моду-лей построена система кодер-декодер. Основными являются модули каскадного кодера и декодера, осуществляющие непосредственное преобразование сигнала. На основе алгоритма согласования скорости кодирования построены модули перфорации и деперфорации кода. Модель позволяет формировать входную информационную последовательность, ошибки в имитированном канале связи, а также проводить сравнение исходной и полученной на выходе декодера последовательности.

В работе проведена оценка эффективности реализованной функциональной модели системы кодер/декодер.

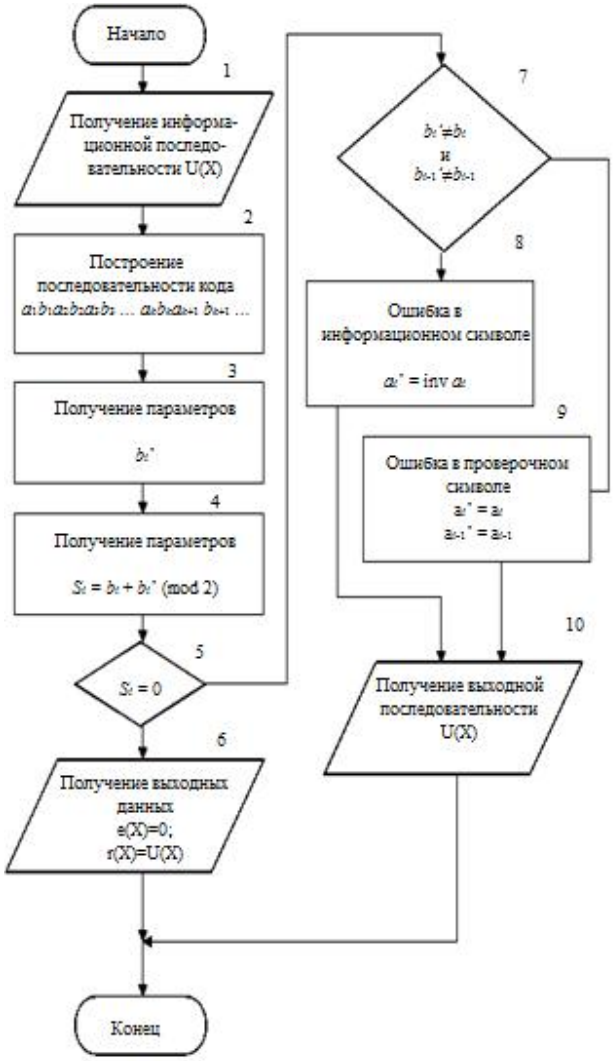

Рис.6. Алгоритм декодирования ИСК

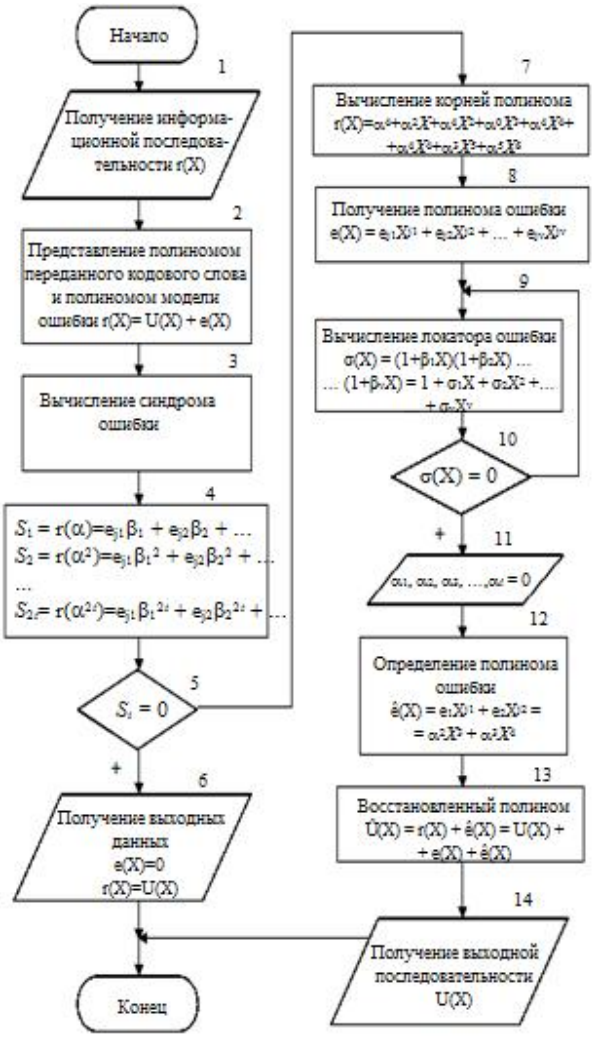

Рис.7. Алгоритм декодирования ИБК

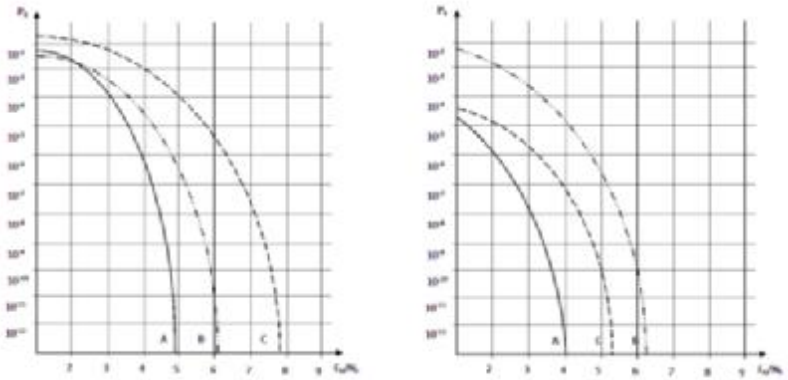

Рис.8. Зависимость вероятности битовой (*а*) и блочной (*б*) ошибки в канале связи от величины соотношения сигнал-шум:A – зависимость для каскадного кодера; B – зависимость для недвоичногокода; C – зависимость для итеративного сверточного кода

Заключение

Анализ оценки эффективности разработанного алгоритма показал, что по сравнению со сверточными кодерами при использовании итеративных алгоритмов значения вероятностей битовойи блочной ошибки снижены (рис.8), а также алгоритм позволяет получить энергетический выигрыш по сравнению с сверточными кодами. Полученный алгоритм может работать с более высокими скоростями кодирования, отличными от r=1/2, что позволяет сократить избыточность кода и увеличить информативность выходной последова-тельности, не снизив при этом помехоустойчивости системы. Применение ПЛИС, построенного на основе данного алгоритма в качестве цифровых модулей позволяет сократить временной цикл разработки устройств, исключает необходимость изготовления инженерных образцов, тем самым снижая затраты на проектирование. Изменение и перепроектирование структуры ПЛИС также несет в себе меньшие затраты, чем моделей СБИС.

Список использованной литературы

1. Shannon C.E. A Mathematical Theory of Communication / C.E. Shannon // Reprinted from The Bell System Technical Journal. – 1948. – V. 27. – P. 379–423, 623–656.

2. BerrouC. Near Shannon Limit Error-Correcting Coding and Decoding: Turbo-Codes / C. Berrou, A. Glavieux, P. Thitimajshima // IEEE Transactions on Information Theory. – 1996. – V. 44. – № 10. – P. 1064-1070.

3. Бернард Скляр. Цифровая связь. Вильямс. Москва-Санкт-Петербург-Киев. 2003. стр. 335-339.

4. Козлов А. В. Декодирование LDPC-кодов в дискретном канале flash-памяти//ИУС.2007. №5(30), стр.31-35.

5. Белоголовый А. В., Крук Е. А. Многопороговое декодирование кодов с низкой плотностью проверок на четность // ИУС. 2005. № 1(14).стр. 25-31.

6. Claussen H., Karimi H. R., Mulgrew B. Improved max-log-map turbo decoding by maximization of mutual information transfer // EURASIP J. on Applied Signal Processing. 2005. P. 820-827.

7. Пирогов А.А. Алгоритм работы сверточного канального кодека сети абонентского доступа / А.А. Пирогов, Н.В. Астахов, О.Ю. Макаров // Вестник Воронежского государственного технического университета. – 2011. Т.7. № 2. стр. 178-180.

Атаджанов Шерзод Шухратович
Начальник отдела по учебно-методической работе с филиалами Ташкентского университета информационных технологий имени Мухаммада ал-Хоразмий (ТУИТ), соискатель кафедры «Системы телерадиовещания»
Тел.: +998 (90) 903-32-61
Эл.почта: umb@tuit.uz; tatuumb@mail.ru
Турсунова Азиза Ахмаджановна
Ассистент кафедры «Системы энергообеспечения» ТУИТ
Тел.: +998 (71) 238-64-95
Эл.почта: umb@tuit.uz
AtadjanovSh.Sh., TursunovaA.A.

Development of Error-Correcting Codes based on Iterative Encoding and Decoding

This paper examines the construction of iterative codes based on block and convolutional codes. This article is devoted to the algorithm of error-correcting coding based on iterative methods, allowing to increase the noise immunity of signal reception at low relations signal to noise ratio(energy per bit).

Keywords: iterative code, iterative convolutional code, block code, iterative coding, decoding, the code distance.