ЭКОНОМНЫЙ АЛГОРИТМ ДЕКОДИРОВАНИЯ ДВОИЧНЫХ БЛОЧНЫХ ТУРБОКОДОВ

Архипкин А.В.

ООО «Кедах Электроникс Инжиниринг»

В современных системах цифровой связи предъявляются очень высокие требования к достоверности передаваемой информации (BER не хуже 10^{-9}). Для таких систем требуется разработка помехоустойчивых кодеков с высокой исправляющей способностью при высоких информационных скоростях. Одними из самых перспективных с точки зрения практического применения являются турбокоды.

Турбокоды представляют собой сравнительно новый тип кодов для исправления ошибок. Особенное достоинство турбокодов состоит в том, что они допускают итеративную процедуру декодирования, в которой на каждой итерации анализируются данные, принадлежащие простым парциальным кодам [1].

Двумерный блочный турбокод (Block Turbo Code - BTC) может быть представлен в виде прямоугольника, построенного из двух кодов: горизонтальных $C^{-} = (n^{-}; k^{-}; d^{-})$ и вертикальных $C^{\perp} = (n^{\perp}; k^{\perp}; d^{\perp})$. Общая информационная емкость турбокода $k = k^{-} \cdot k^{\perp}$, длительность $n = n^{-} \cdot n^{\perp}$, а минимальный вес Хемминга $d = d^{-} \cdot d^{\perp}$.

Итеративный декодер основан на вычислении апостериорных вероятностей двоичных символов кодовых слов C^{-} и $C^{\perp}$. На рис.1 представлена блок-схема итеративного декодера двоичных блочных турбокодов. Априори символы на входе декодера являются равновероятными. Анализ отдельных порций входных отсчетов на каждой итерации увеличивает вероятность одних и уменьшает вероятность других символов.

Каждая итерация состоит из двух этапов. На каждом из этапов анализируются по отдельности массивы данных, соответствующих кодовым словам, расположенных в строках (столбцах). В результате анализа определяются кодовые добавки, которые суммируются с входными данными и результат подается на следующий этап итеративного декодирования. Процедура повторяется от итерации к итерации, увеличивая вероятность правильного декодирования.

Алгоритм приближенного вычисления кодовых добавок при итеративном декодировании

Парциальный двоичный код $C(n, d, k)$ состоит из набора кодовых слов C_j $(0 \le j \le 2^K - 1)$, элементы которых $C_j(l)$ $(0 \le l \le n-1)$ принадлежат полю GF(2). В случае BPSK манипуляции на приемной стороне при условии идеальной временной и фазовой синхронизации будут иметь место отсчеты $r_l = a_l(-1)^{C(l)} + n(l)$ (индекс j опущен), где $n(l)$ - независимые гауссовские отсчеты с одинаковой дисперсией σ_n^2. Амплитуды a_l полагаются известными.

При сделанных предположениях вероятность

$$\Pr[C(l) \mid r_l] = \frac{\Pr[C(l)]\Pr[r_l \mid C(l)]}{\Pr[r_l]} \equiv \Pr[C(l)] \cdot \exp\left\{-\frac{1}{2\sigma_n^2}\left[r_l - a_l(-1)^{C(l)}\right]^2\right\}.$$

Полагаем, что $C(l)$ принимает равновероятно значения 0 и 1.

Рассмотрим логарифм отношения

$y_l = \ln\frac{\Pr[C(l)=0 \mid r_l]}{\Pr[C(l)=1 \mid r_l]} = \frac{2a_l}{\sigma_n^2} r_l$, соответственно $\Phi_l^{(0)} = \frac{\Pr[C(l)=1 \mid r_l]}{\Pr[C(l)=0 \mid r_l]} = e^{-y_l}$

Введем вместо y_l два других параметра: $q_l = sign(-\ln \Phi_l^{(0)}) = sign(y_l) = sign(r_l)$; $Y_l = |y_l|$; ;

Кроме того, введем параметр $\alpha_l = e^{-Y_l}$.

Параметры Y_l и α_l характеризуют надежность при принятии решений по одному отсчету y_l. Очевидно, что отсчеты с большими значениями Y_l более надежны, чем с малыми.

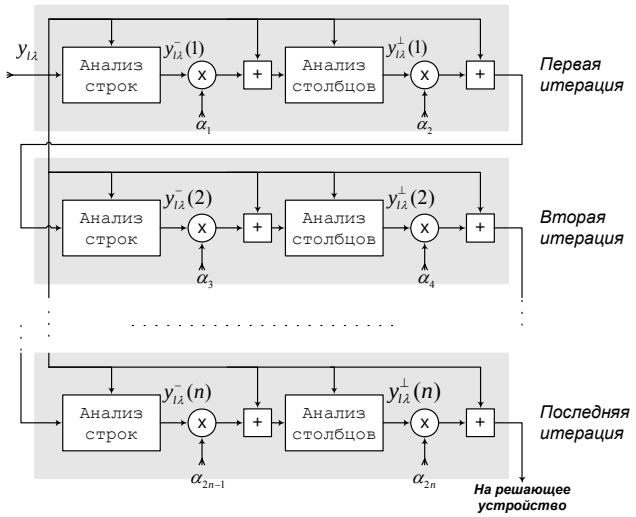

Рис.1. Блок-схема итеративного декодера двоичных блочных турбокодов

Перейдем теперь непосредственно к анализу массива данных, соответствующего кодовому слову. Массив состоит из отсчетов $r_l\ (0 \le l \le n-1)$, обозначим его через вектор R. Можно вычислить апостериорные вероятности $\Pr[C(l)=1 \mid R]$, $\Pr[C(l)=0 \mid R]$ и их отношение $\Phi_l^{(1)}$. Тогда величина кодовой добавки на первом этапе первой итерации y_l^c определяются как

$$y_l^c = -\ln \Phi_l^{(1)} - (-\ln \Phi_l^{(0)}) = -\ln \Phi_l^{(1)} - y_l \qquad (1)$$

На последующих этапах в качестве y_l во второй части равенства всегда используется отсчет на входе декодера, который вычитается из логарифма отношения апостериорных вероятностей.

Если информационные двоичные символы априори независимы и равновероятны, то равновероятными являются также кодовые слова C_j. Поэтому отношение апостериорных вероятностей

$$\Phi_l^{(1)} = \frac{\sum\limits_{j \in J_l} \Pr[C_j \mid R]}{\sum\limits_{j \in \bar{J}_l} \Pr[C_j \mid R]} = \frac{\sum\limits_{j \in J_l} \prod\limits_{m=0}^{n-1} \Pr[r_m \mid C_j(m)]}{\sum\limits_{j \in \bar{J}_l} \prod\limits_{m=0}^{n-1} \Pr[r_m \mid C_j(m)]}, \qquad (2)$$

где J_l - подмножество кодовых слов, у которых $C_j(l)=1$, а $\bar{J}_l$ - дополнение до полного множества $(C_j(l)=0)$.

Техническая реализация алгоритма (2) в общем случае вряд ли возможна. Существенное упрощение можно получить с помощью аппроксимации алгоритма (2), дающей приближенную оценку отношений апостериорных вероятностей. Для выражения (2) была предложена аппроксимация [2]:

$$\Phi_l^{(1)} = \frac{\max\limits_{j \in J_l} \exp\{-M_j\}}{\max\limits_{j \in \bar{J}_l} \exp\{-M_j\}} = \frac{\exp\{-\min\limits_{j \in J_l} M_j\}}{\exp\{-\min\limits_{j \in \bar{J}_l} M_j\}} \qquad (3)$$

где величина $M_j = \frac{1}{2\sigma_n^2} \sum\limits_{l=0}^{n-1} [r_l - a_l(-1)^{C_j(l)}]^2$ названа метрикой j-го кодового слова.

Из этого выражения и выражения (1) получаем

$$y_l^c \cong \min_{j \in J_l} M_j - \min_{j \in \bar{J}_l} M_j - y_l. \qquad (4)$$

Нахождение минимумов метрик эквивалентно оптимальному декодированию кодового слова в целом. Строгий поиск минимумов задача сложная, так как объемы подмножеств J_l и $\bar{J}_l$ равны 2^{k-1} кодовых слов. Поэтому для решения этой задачи предлагается использовать алгоритм Чейза [2, 4], суть которого описана в [3].

Далее представим подход к построению алгоритмов оценки кодовых добавок, дающий дополнительную экономию в числе требуемых для декодирования вычислительных операций. Поделив числитель и знаменатель выражения (2) на $\prod_m \Pr[r_m \mid 0]$, получим

$$\Phi_l^{(1)} = \frac{\sum_{j\in J_l}\prod_m \alpha_m^{C_j(m)\oplus q_m}}{\sum_{j\in \bar{J}_l}\prod_m \alpha_m^{C_j(m)\oplus q_m}}. \qquad (5)$$

Замечаем, что

$\prod_m \alpha_m^{C_j(m)\oplus q_m} = \alpha_{m_1}\cdot\ldots\cdot\alpha_{m_t}$, где $m_1,\ldots,m_t$ - номера позиций, в которых $C_j(m)\oplus q_m = 1$, т.е. номера ошибочных позиций, в которых q_m отличается от $C_j(m)$. Откуда следует, что позиции $m_1,\ldots,m_t$ удовлетворяют условию $h(m_1)\oplus\ldots\oplus h(m_t) = S$, где $h(m)$ - вектор-столбцы проверочной матрицы H кода C, а S - вектор-синдром с элементами $S_i = \sum_l q_l h_i(l)$ $(0 \le i \le n-k-1)$.

Обозначим через $J_l^t(S)$ подмножество различных наборов из t номеров позиций $(m_1,\ldots,m_t; m_i \ne m_j)$, один из которых равен l, удовлетворяющих условию

$$\sum_{i=1}^{t} h(m_i) = S. \qquad (6)$$

Через $\bar{J}_l^r(S)$ обозначим аналогичное подмножество, но с номерами $m_i \ne l$. Из (5) видно, что если $q_l = 0$, то набор подмножеств $J_l^t(S)$ образует множество J_l в числителе (5), а совокупность $\bar{J}_l^r(S)$ - множество $\bar{J}_l$ в знаменателе (5).

При $q_l = 1$ числитель и знаменатель меняются местами. Из чего следует, что логарифм апостериорных вероятностей

$$\ln\Phi_l^{(1)} = (-1)^{q_l}\left\{\ln\sum_t\sum_{J_l^t(S)}\alpha_{m_1}\cdot\ldots\cdot\alpha_{m_t} - \ln\sum_\tau\sum_{\bar{J}_l^\tau(S)}\alpha_{\nu_1}\cdot\ldots\cdot\alpha_{\nu_\tau}\right\}.$$

Значения t и τ сверху ограничены длиной кода n. Нижняя граница по t равна 1, т.к. по крайней мере одна позиция с номером l должна содержаться в подмножествах $J_l^t(S)$. Нижняя граница по τ равна нулю, причем $\tau = 0$ имеет место только при $S = 0$, т.е. последовательность q_l совпадает с одним из кодовых слов C_j. В этом случае $\prod_m \alpha_l^{C_j(m)\oplus q_m} = 1$, а $\ln\prod_m = 0$.

Так как $\alpha_l = e^{-Y_l} < 1$, то их произведение быстро убывает с ростом t и τ. Исходя из этого, делаем первое приближение, аналогичное переходу от формулы (2) к (3):

$$\ln\Phi_l^{(1)} \cong (-1)^{q_l}\left\{-\min_t\min_{J_l^t(S)}(Y_{m_1}+\ldots+Y_{m_t}) + \min_\tau\min_{\bar{J}_l^\tau(S)}(Y_{\nu_1}+\ldots+Y_{\nu_1})\right\} \qquad (7)$$

А кодовая добавка

$$y_l^c \cong (-1)^{q_l}\left\{-\min_t\min_{J_l^t(S)}(Y_{m_1}+\ldots+Y_{m_t}) - \min_\tau\min_{\bar{J}_l^\tau(S)}(Y_{\nu_1}+\ldots+Y_{\nu_1})\right\} - y_l. \qquad (8)$$

Поиск минимумов в (7) остается задачей такой же сложной в реализации, как и декодирование кодового слова в целом. Так же как и в работе [6] для дальнейших упрощений привлечен алгоритм Чейза.

Описанная процедура итеративного декодирования применима при строгом вычислении апостериорных вероятностей. При приближенном анализе, как показали исследования [2], кодовые добавки ($y_{l\lambda}^-$ и $y_{l\lambda}^\perp$) имеют высокие дисперсии, что вызывает необходимость их ослабления путем домножения на коэффи-

циент меньше 1. Оптимизация итеративного декодера осуществляется моделированием, путем подбора коэффициентов $\alpha_1, \alpha_2, \ldots \alpha_{2n}$, обеспечивающих минимум вероятности ошибочного декодирования.

На основании рассмотренного синтезирован алгоритм декодирования блочных турбокодов с парциальными расширенными кодами Хемминга.

Моделирование

При моделировании предлагаемого декодера в зеркальном однолучевом канале с белым гауссовским шумом в качестве парциальных были использованы расширенные коды Хемминга (32,26). Эффективность декодера блочного турбокода BTC(32,26)x(32,26) со скоростью $R_{BTC}=26^2/32^2\approx0{,}66$ была сопоставлена с близким по скорости декодером Витерби (K=7), $R_{Viterbi}=2/3\approx0{,}66$. Результаты моделирования представлены на рис.2.

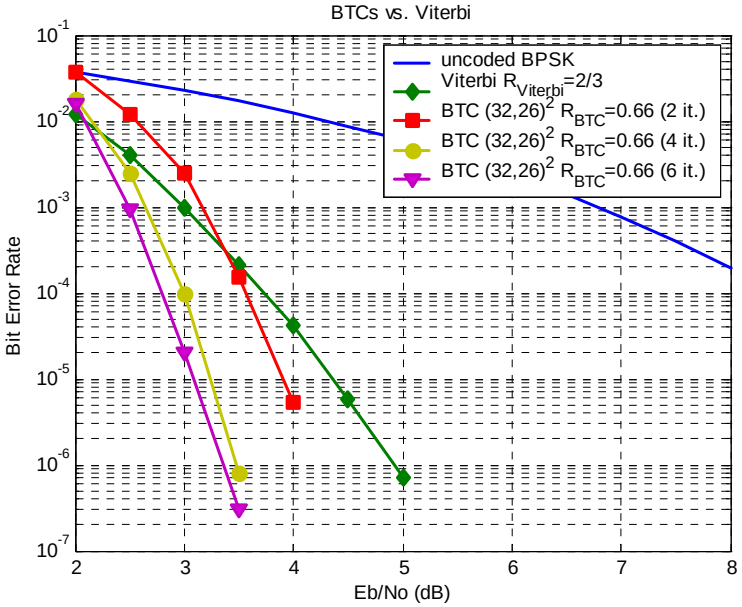

Рис.2. Результаты моделирования декодеров BTC и Viterbi

Из рис.2 следует, что выигрыш кода $BTC(32,26)^2$ по энергетике при 6 итерациях составляет около 1,5дБ по сравнению с близким по скорости сверточным кодом. Оценка аппаратных затрат на реализацию исследуемых декодеров показала, что для реализации декодера BTC на ПЛИС потребуется примерно на 45-50% меньше аппаратных ресурсов, чем для декодера Витерби.

Полученные результаты позволяют считать турбокодирование одним из наиболее эффективных и экономичных на сегодняшний день видов помехоустойчивого кодирования. Кроме того, турбокоды являются более гибкими по сравнению с другими кодами, т.к. позволяют варьировать количество итераций в зависимости от допустимой задержки на декодирование.

В настоящее время теория турбокодирования и технология микроэлектронной реализации турбокодеков развивается высокими темпами. Создание российской платформы, реализующей этот вид помехоустойчивого кодирования, является важным шагом на пути к созданию конкурентоспособных отечественных систем радиосвязи.

Литература

[1] C. Berrou, A. Glavieux, P. Thitimajshima: “Near Shannon limit error-correcting coding and decoding: Turbo codes (1)”, IEEE Int. Conf. Communications, ICC’93, vol.2/3, pp.1064-1070, May 1993.

[2] R. Pyndiah, “Near-Optimum Decoding of Product Codes: Block Turbo Codes”, IEEE Trans. On Comm., Vol.46, No8, August 1998.

[3] D. Chase, “A Class of Algorithms for Decoding Block Codes With Channel Measurement Information”, IEEE Trans. On Information Theory, Vol.IT-18, No1, January 1972.

[4] S. Hirst, B. Honary, G. Markarian, “Fast Chase Algorithm With an Application in Turbo Decoding”, IEEE Trans. On Comm., Vol.49, No10, October 2001.

COST-EFFECTIVE BINARY BLOCK TURBO DECODING ALGORITHM

Ahkhipkin A.

«Kedah Electronics Engineering»

Modern digital communication systems require high level of transmit information reliability. Such systems need forward error correction algorithms which support hign data rates and good correction capability. One of the most prospective is turbo code sheme.

Turbo codes are relatively new forward error correction type. The main advantage is that they allow iterative decoding procedure. In this scheme decoder analyzes simple partial codes during each iteration.

In this paper cost-effective binary block turbo decoding algorithm based on eHamming partial codes is presented. The simulation showed that block turbo code decoder has better efficiency that Viterbi decoder for convolutional codes with the same code rate. Moreover, its FPGA-based implementation needs less hardware resources.