

УДК 681.32

СИСТЕМЫ ОБНАРУЖЕНИЯ И ИСПРАВЛЕНИЯ ОШИБОК, АДАПТИВНЫЕ К ШУМУ В КАНАЛЕ

Сорокин С.С.

Предлагается способ генерации пирамидальных кодов и построения кодеков, позволяющих легко адаптироваться к устойчивым изменениям характера и количества ошибок в единицу времени.

1 Введение. Замечено, что шум в реальных каналах не является чисто гауссовским или чисто импульсным, и характеристики канала связи часто изменяются во времени. Например [1], в высокочастотных каналах шум является смесью гауссовского и импульсного шумов. Кроме того, характеристики затухания в высокочастотных каналах могут изменяться в течение дня. Следовательно, обнаружение или исправление ошибок необходимо проводить в различные периоды времени различными кодами или, по крайней мере, кодами различной длины, но с одинаковыми значениями t_1 и t_2 . «Поэтому при проектировании очень важно использовать системы обнаружения или исправления ошибок, которые можно адаптировать к действующему в канале шуму по результатам его измерения» [1].

Свойствами, необходимыми для оперативной и лёгкой адаптации системы обнаружения или исправления ошибок, обладают пирамидальные коды [2]. Рассмотрим основные понятия, поясняющие пирамидальное кодирование.

Пирамидальный код: код $P = (b_0, b_1, \dots, b_s)$, состоящий из $t_1 + 1$ блоков, где b_0 представляет собой информационную часть всего пирамидального кода, а совокупность остальных блоков служит контрольной частью пирамидального кода. Каждая соседняя пара блоков представляет собой *квазилинейный код* $C_i = (b_{i-1}, b_i)$, $i = 1, \dots, t_1$, где b_{i-1} соответствует информационной части i -го квазилинейного кода, а b_i - его контрольной части. Каждый код $C_i(b_{i-1}, b_i)$ может исправлять $t_{i-1} + 1$ ошибок, в случае если блок b_i , был принят без ошибок и число ошибок соответствующего информационного блока не превысило $t_{li} = t_{i-1} + 1$.

Квазилинейный код: код, состоящий из информационного блока, содержащего k_i двоичных символов и контрольного блока, содержащего r_i двоичных символов. Здесь в отличие от линейного кода контрольный блок кодирует только информационную часть кода. Основная идея построения квазилинейного кода с исправлением $1, 2, \dots, t_1$ ошибок состоит в том, что из

общего числа $K = 2^k$ комбинаций k -разрядной информационной части кода для кодирования информации выбирается $K_1 = k + C_k^2 + \dots + C_k^{t_1}$ комбинаций так, чтобы при возникновении ошибок из заданного класса одна комбинация не превращалась в другую. Используемые в данном коде комбинации называются разрешенными, а остальные $K - K_1$ комбинаций - запрещенными.

Таким образом, для представления K_1 ошибок множества исправляемых ошибок E^1 символами контрольного блока квазилинейного кода необходимо иметь не менее чем $r \geq \log_2 K_1$ контрольных символов.

Если кроме исправления ошибок множества E^1 пирамидальный код должен и обнаруживать не более чем t_2 ошибок, то в кодировочном пространстве V_r размерности 2^r должны быть предусмотрены некоторое мно-

жество точек для представления $K_2 \leq \sum_{i=t_1+1}^{t_2} C_k^i$ обнаруживаемых ошибок.

Здесь, знак нестрогого равенства обоснован тем, что между v_j векторами (точками) подмножества $V_r^2 \subset V_r$, отображающими множество обнаруживаемых ошибок E^2 , не требуется отношения независимости, т.е. некоторые векторы из подмножества V_r^2 могут быть равны друг другу. Заметим, что векторы подмножества V_r^2 могут занимать свободные точки в пространстве V_r , либо для их размещения потребуется увеличение r -размерности кодировочного пространства V_r .

Определение других понятий, используемых далее, дано в работе [2].

2 Построения кодировочной матрицы квазилинейного кода. Основной алгоритм, который назовем *Cod_t1_t2*, использует две процедуры *Sind_test* и *Sind_plus*.

2.1 Процедура *Sind_test*

Процедура *Sind_test(u, j, t1, t2, Sind, ind)* используется для определения значения переменной *ind* по следующему правилу.

Правило 1. Если для вектора u в индикаторе синдромов *Sind* имеются свободные компоненты, такие, у которых $\sigma = 0$ для самого вектора u и его линейных комбинаций с каждой из j предшествующих векторов-строк и с их сочетаниями по 2, 3, ..., по $(t1-1)$ строк, а также имеются компоненты со значениями $\sigma \neq 3$, для линейных комбинаций вектора u с сочетаниями из j строк по $t1, t1+1, \dots, t2-1$ строк, то $ind = 0$, в противном случае $ind = 1$, что соответствует случаю, когда вектор u не может быть использован в качестве кода очередной строки, т.е. кода очередного информационного символа.

2.2 Процедура *Sind_plus*

Процедура *Sind_plus* используется для заполнения компонент индикатора синдромов *Sind* значениями $\sigma \in \{1,3\}$ по следующему правилу.

Правило 2. Присвоить значение $\sigma = 3$ компоненте с адресом u , а также компонентами, которые соответствуют линейным комбинациям вектора u с каждой из j предшествующих векторов-строк и с их сочетаниями по 2,3, ... $(t1-1)$ строк. Если $t2 > t1$, то присвоить значения $\sigma = 1$ компонентам соответствующим линейным комбинациям вектора u с сочетаниями из j по $t1, (t1+1), \dots, (t2-1)$ строк.

Другими словами, процедура *Sind_plus* присваивает значение 3 тем компонентам вектора *Sind*, номера которых, соответствуют синдромам корректируемых ошибок, а значение 1 - тем компонентам, номера которых соответствуют синдромам обнаруживаемых ошибок

2.3 Синтез кодировочной матрицы G

Исходными данными для построения кодировочной матрицы первой степени пирамидального кода являются: $t1, t2, k, w$ (параметры $t1, t2$ и k описаны выше, а $w \geq t1+1$ - минимальный вес контрольного блока, обеспечивающий защиту от ложной коррекции первого информационного блока). Для последующих ступеней кода параметры $t1, t2$ и w уменьшаются на единицу, а параметр k принимает значение, равное длине предыдущего контрольного блока.

Процесс синтеза кодировочной матрицы начинается с присваивания текущему вектору v и первой строке матрицы G начального значения, состоящего из w единиц в младших (правых) разрядах. Далее, значение вектора v увеличивается на единицу. Для каждого значения очередного вектора проверяется выполнение условий

$$W(v) \geq w, \text{ Sind}[v] = 0,$$

где $W(v)$ -вес Хемминга (число ненулевых компонент вектора v), $\text{Sind}[v]$ - значение v - ой компоненты вектора *Sind*.

Эффективность такого способа построения вытекает из удобства представления и обработки матрицы P на ЭВМ в виде последовательности векторов-строк, которые в данном случае представляются и обрабатываются как машинные слова (двоичные векторы) достаточно большой длины. Например, для кода, имеющего 484 информационных разряда и $d_{\min} = 7$, число контрольных разрядов, и следовательно, длина машинного слова должна быть не более 30 [1].

3 Построение диагностической матрицы (дешифратора). Для реализации процедуры декодирования необходимо иметь диагностическую матрицу $D = \|d_{ij}\|$ размерности $2^r \times k$, где $d_{ij} = 1$, если необходимо скорректировать j -ый информационный разряд; i - номер строки, в которой

размещается корректор информационной части кодового слова. Здесь i -десятичное представление синдрома s_i .

Процедура построения диагностической матрицы состоит в следующем:

а) моделируются ошибки $e_i \in E^1$; для каждой ошибки вычисляется код равный её синдрому s_i ; строке матрицы, имеющей номер s_i , присваивается значение вектора e_i .

б) строкам матрицы, соответствующим ошибкам множества E^2 , присваивается значение некоторого произвольного вектора длины k , имеющего вес больший, чем w .

4 Использование особенностей структуры шифратора и дешифратора. Из содержания пунктов 3 и 4 следует вывод, что матрицы P и D получаются ступенчатыми. Верхняя строка матрицы P содержит w единиц, последующие строки расширяются влево, при этом, некоторые из них сохраняют свою ширину, получая значение двоичного вектора большего, чем предыдущие строки. Подобную структуру имеют и диагностические матрицы.

Отмеченные особенности позволяют выполнять шифратор (кодировочная матрица) и дешифратор (модификация диагностической матрицы) в виде (k, r) - и (r, k) -полюсников, соответственно. При этом значения k и r будут использоваться полностью при самом низком уровне шумов. При увеличении интенсивности шумов следует использовать не все входы и выходы рассмотренных устройств, сокращая их число “сверху”, т.е. не используя старшие разряды информационной и кодировочной частей кодового слова, и уменьшая таким образом n -длину кодового слова и вероятность недопустимого его искажения. Информацию о необходимости изменения длины кодового слова можно получить при изменении частоты запросов на повторную передачу слова с обнаруженной, но неисправленной ошибкой.

Библиография

1 Питерсон У. Коды, исправляющие ошибки / У.Питерсон, Э.Уэлдон: Пер. с англ. — М.: Мир, 1976. — 594 с.

2 Сорокин С.С. Пирамидальное кодирование и его эффективность в системах диагностики / С.С.Сорокин // Автоматизация проектирования дискретных систем: Материалы III междунар. конф., 10-12 ноября 1999 г. — Минск, 1999 — С. 36 – 41.

Статья поступила 25.09.00 г.