Системы цифрового телевидения для тех, кто хочет понять: кодирование, исправляющее ошибки

Константин Гласман

Часть 6. Начало в №№ 6…9/2020, № 1/2021

Принципы построения циклических кодов

Циклические коды – это подкласс линейных кодов, которые удовлетворяют дополнительному структурному требованию. В качестве математического аппарата, обеспечивающего поиск хороших циклических кодов, используется теория полей Галуа. Эта теория приводит к алгоритмам кодирования и декодирования, которые эффективны как вычислительные процедуры.

При рассмотрении принципов построения линейных кодов компоненты последовательности из *n* символов (такие последовательности назывались словами) рассматривались как компоненты вектора в *n*-мерном пространстве, а сама последовательность – как вектор в этом пространстве. Множество кодовых слов, получаемое с помощью канального кодера, рассматривалось как подмножество, обладающее определенными свойствами, во множестве всех векторов в n-мерном пространстве. Можно также утверждать, что множество кодовых слов представляет собой линейное подпространство в *n*-мерном пространстве. Каждый символ представляет собой элемент поля Галуа *GF(q)* из *q* элементов (в рассмотренных в предыдущих статьях примерах $q = 2$, но все результаты, относящиеся к кодам с проверками на четность, можно обобщить на недвоичные символы, то есть на поля с большим количеством символов).

Циклическим кодом над полем *GF(q)* называется такой линейный код, у которого при любом циклическом сдвиге какого-либо кодового слова получается другое кодовое слово. Это значит, что если последовательность $\boldsymbol{x'} = (x_1, x_2, \ldots, x_n)$ является кодовым словом, то $\boldsymbol{x''} = (x_2, x_3, \ldots, x_n, x_1)$ также представляет собой кодовое слово.

При описании циклических кодов обычно изменяют обозначения, нумеруя символы не с начала, а с конца, то есть от *(n-1)* до *0*: $\boldsymbol{x} = (x_{n-1}, x_{n-2}, \ldots, x_1, x_0)$. Каждую последовательность из *n* символов, или вектор можно представить многочленом от *z* со степенью не выше *(n-1)* над полем *GF(q)*:

$$x(z) = x_{n-1}z^{n-1} + x_{n-2}z^{n-2} + \ldots + x_1z^1 + x_0. \quad (37)$$

Компоненты вектора отождествляются с коэффициентами многочлена. Множество многочленов обладает структурой, идентичной структуре пространства $GF^n(q)$, и одновременно обладает структурой кольца многочленов по модулю многочлена (x^n-1) над полем *GF(q)*. Это позволяет определить циклический сдвиг как умножение в этом кольце:

$$z*x(z) = R_{(z^n-1)}[x_{n-1}z^n + x_{n-2}z^{n-1} + \ldots + x_1z^2 + x_0z] =$$
$$= R_{(z^n-1)}[x_{n-1}(z^n-1) + x_{n-2}z^{n-1} + \ldots + x_1z^2 + x_0z + x_{n-1}] =$$
$$= x_{n-2}z^{n-1} + \ldots + x_1z^2 + x_0z + x_{n-1}.$$

Выберем в подпространстве кодовых многочленов ненулевой нормированный кодовый многочлен наименьшей степени. Обозначим его степень как *(n-k)*. Этот многочлен называется порождающим многочленом. Обозначим его как *g(z)*. В теории циклических кодов доказывается, что циклический код состоит из всех произведений порождающего многочлена на многочлены степени не выше *(k-1)*. Но циклический код существует только в том случае, если (z^n-1) делится на *g(z)* без остатка. Это значит, что существует такой многочлен *h(z)*, что $z^n - 1 = g(z)*h(z)$. Следовательно

$$R_{(z^n-1)}[g(z)*h(z)] = 0. \quad (38)$$

Многочлен *h(z)* называется проверочным многочленом. Каждое кодовое слово удовлетворяет равенству:

$$R_{(z^n-1)}[x(z)*h(z)] = 0. \quad (39)$$

Информационная последовательность, которая подлежит кодированию, также может быть представлена в виде многочлена, степень которого равна *(k-1)*:

$$u(z) = u_{k-1}z^{k-1} + u_{k-2}z^{k-2} + \ldots + u_1z^1 + u_0.$$

Множество информационных многочленов может быть отображено в кодовые многочлены разными способами. Например, можно умножать информационные многочлены на порождающий многочлен:

$$x(z) = u(z)\,g(z). \quad (40)$$

Такое кодирование является несистематическим. Помимо кодирования по правилу $x(z) = u(z)*g(z)$ существует так называемое систематическое правило кодирования, при котором *k* старших коэффициентов кодового слова устанавливаются равными коэффициентам информационного многочлена: $x(z) = u_{k-1}z^{n-1} + u_{k-2}z^{n-2} + \ldots + u_0z^{n-k} + p_{n-k-1}z^{n-k-1} + \ldots + p_1z + p_0$, а *(n-k)* младших коэффициентов кодового слова $\boldsymbol{p} = (p_{n-k-1}, p_{n-k-2}, \ldots, p_1, p_0)$, которые часто называются проверочными, подбираются такими, чтобы многочлен *x(z)* делился на *g(z)* без остатка, то есть $R_{g(z)}[x(z)] = 0$. Это будет так, если соответствующий проверочный многочлен $p(z) = p_{n-k-1}z^{n-k-1} + \ldots + p_1z + p_0$ рассчитывается как $p(z) = -R_{g(z)}[z^{n-k}u(z)]$. Систематическое правило кодирования дает кодовые слова, более удобные на практике, так как информационные слова в явном виде размещаются в *k* старших разрядах кодовых слов.

Все кодовые многочлены *x(z)* делятся на порождающий многочлен *g(z)* без остатка, то есть $R_{g(z)}[x(z)] = 0$. Это обстоятельство дает ключ к декодированию, позволяющему обнаруживать и исправлять ошибки, возникшие при передаче данных. Слова, которые не делятся без остатка на порождающий многочлен, не являются кодовыми и, следовательно, содержат ошибки.

Передаваемый по каналу связи или записываемый на носитель кодовый блок $\boldsymbol{x} = (x_{n-1}, x_{n-2}, \ldots, x_1, x_0)$ может претерпеть искажения, например, из-за шумов. Это можно в общем виде описать добавлением к кодовому блоку набора ошибки $\boldsymbol{e} = (e_{n-1}, e_{n-2}, \ldots, e_1, e_0)$, которому соответствует многочлен $e(z) = e_{n-1}z^{n-1} + e_{n-2}z^{n-2} + \ldots + e_1z + e_0$. Принятому или воспроизведенному набору символов $\boldsymbol{y} = (y_{n-1}, y_{n-2}, \ldots, y_1, y_0)$ соответствует многочлен $y(z) = x(z) + e(z)$. Найдя остаток от деления принятого многочлена *y(z)* на порождающий *g(z)*, можно понять, были ли на самом деле ошибки. Если $R_{g(z)}[y(z)] = 0$, значит с большой вероятностью можно утверждать, что ошибок не было и принятое слово является кодовым, то есть *y(z)* и является переданным словом *x(z)*. Если остаток не равен нулю, то при передаче были ошибки. Остаток от деления дает многочлен, зависящий только от многочлена ошибки:

$$R_{g(z)}[y(z)] = R_{g(z)}[x(z)] + R_{g(z)}[e(z)] =$$
$$= R_{g(z)}[e(z)] = s(z).$$

Этот многочлен называют синдромным.

Многочлен *s(z)* зависит только от конфигурации ошибок, то есть является синдромом, или описанием ошибок. Если число ошибок не превышает некоторый предел, то между e(z) и *s(z)* существует однозначное соответствие. Этот предел зависит от минимального расстояния кода *d**. Каждому многочлену ошибок, вес которого меньше, чем *d*/2*, соответствует единственный синдромный многочлен.

Таким образом, для исправления ошибок с использованием циклического кода необходимо найти многочлен ошибок *e(z)* с наименьшим числом ненулевых коэффициентов, который отвечает условию:

$s(z) = R_{g(z)}[e(z)]$.

Эту задачу можно решить, например, табличным способом. Для каждого многочлена ошибок с весом, который меньше *d*/2*, вычисляется синдромный многочлен, значения которого табулируются. Получаемая таблица называется таблицей значений синдромов.

Таблица значений синдромов циклического кодирования

Многочлен ошибок e(z)	Синдромный многочлен s(z)
1	$R_{g(z)}[1]$
z	$R_{g(z)}[z]$
z^2	$R_{g(z)}[z^2]$
...	...
1+z	$R_{g(z)}[1+z]$
$1+z^2$	$R_{g(z)}[1+z^2]$
...	...

Итак, был введен ряд многочленов, которые приведены ниже в систематизированном виде с указанием их степеней:

информационный многочлен	*u(z)*	*deg [u(z)] = k - 1,*
кодовый многочлен	*x(z)*	*deg [u(z)] = n - 1,*
порождающий многочлен	*g(z)*	*deg [u(z)] = n - k,*
принятый многочлен	*y(z)*	*deg [u(z)] = n - 1,*
проверочный многочлен	*h(z)*	*deg [u(z)] = k,*
многочлен ошибок	*e(z)*	*deg [u(z)] = n - 1,*
синдромный многочлен	*s(z)*	*deg [u(z)] = n – k - 1.*

Если набор из *k* символов образует блок информации $\boldsymbol{u} = (u_{k-1}, u_{k-2}, ..., u_1, u_0)$, кодируемой с целью обнаружения и исправления ошибок с помощью циклического кода, то кодирование означает формирование блока $\boldsymbol{x} = (x_{n-1}, x_{n-2}, ..., x_1, x_0) = (u_{k-1}, u_{k-2}, ..., u_0, p_{n-k-1}, ..., p_1, p_0)$, который приобрел некоторую избыточность в виде дополнительных проверочных символов $\boldsymbol{p} = (p_{n-k-1}, ..., p_1, p_0)$. Эта избыточность имеет строго дозированную величину в соответствии с заданной степенью помехозащищенности.

Декодирование принятого или воспроизведенного набора символов $\boldsymbol{y} = (y_{n-1}, y_{n-2}, ..., y_1, y_0)$ предполагает следующие действия:

- нахождение синдромного многочлена $s(z) = R_{g(z)}[y(z)]$;
- нахождение многочлена ошибок *e(z)* в таблице значений синдромов на основе вычисленного синдромного многочлена (ошибки нет, если *s(z) = 0*);
- определение оценки переданного кодового многочлена путем вычисления *x(z) = y(z) - e(z)*;
- определение переданного блока информации $u = (u_{k-1}, u_{k-2}, ..., u_1, u_0)$ по *k* старшим коэффициентам восстановленного кодового многочлена *x(z)*.

Изложенный способ определяет принципиальную возможность исправления ошибок на базе канального кодирования с помощью циклических кодов. Для многих циклических кодов разработано большое число эффективных схем декодирования.

Циклические коды: кодирование и декодирование

Циклический код существует только в том случае, если (z^n-1) делится на порождающий многочлен *g(z)* без остатка, что следует из выражения (39). Каждый многочлен *g(z)*, который делит многочлен (z^n-1), порождает циклический код. Естественный подход к получению порождающего многочлена заключается в разложении многочлена (z^n-1) на простые множители:

$z^n-1 = f_1(z)^* f_2(z)^* \dots f_s(z)$,

где *s* – число простых множителей.

Порождающий многочлен *g(z)* можно получить в виде произведения некоторого подмножества этих простых множителей. Но как выбрать это подмножество? Если все простые множители различны, то существует 2^s вариантов построения порождающего многочлена. Если исключить из этих вариантов тривиальные случаи *g(z) = 1* (когда в g(z) не включается ни одного множителя) и $g(z) = (z^n-1)$ (когда в порождающий многочлен входят все множители), то остается 2^s-2 варианта, что при больших значениях *n* составит большое число. Какие варианты дают коды с большим минимальным расстоянием? Для ответа на этот вопрос в теории циклических кодов прослеживается взаимосвязь простых многочленов с их корнями в расширении поля. Но сначала надо ввести понятие минимального многочлена.

Пусть *GF(q)* – поле, *GF(Q)* – расширение этого поля. Пусть β – элемент поля *GF(Q)*. Простой многочлен *f(z)* наименьшей степени над *GF(q)*, для которого элемент β является корнем (это значит, что *f(β) = 0*), называется минимальным многочленом элемента β над полем GF(q). Минимальный многочлен всегда существует и является единственным. Если минимальный многочлен элемента β над полем *GF(q)* равен *f(z)* и β является корнем *g(z)*, то *f(z)* делит *g(z)*.

Теперь общий подход к созданию циклического кода может быть сформулирован следующим образом. Циклический код строится по порождающему многочлену, который задается своими корнями в поле Галуа $GF(q^m)$. Поле Галуа $GF(q^m)$ является расширением поля GF(q), над которым создается циклический код.

Пусть $\beta_1, \beta_2, \dots, \beta_r$ из поля Галуа $GF(q^m)$ являются корнями порождающего многочлена g(z). Обозначим через $f_1(z), f_2(z), \dots, f_r(z)$ минимальные многочлены элементов $\beta_1, \beta_2, \dots, \beta_r$ из поля Галуа $GF(q^m)$. Порождающий многочлен циклического кода находится как наименьшее общее кратное (НОК) произведения:

$$g(z) = HOK[f_1(z), f_2(z), \dots, f_r(z)], \qquad (41)$$

где $f_1(z), f_2(z), \dots, f_r(z)$ – минимальные многочлены элементов $\beta_1, \beta_2, \dots, \beta_r$, то есть корней порождающего многочлена g(z).

Описанный общий принцип построения циклического кода позволяет указать способ исправления ошибок с использованием расширения поля. Пусть x(z) – кодовое слово, полученное на выходе канального кодера. Если на кодовое слово в канале воздействуют ошибки e(z), то многочлен, описывающий принятое слово, можно записать в виде:

$y(z) = x(z) + e(z)$.

Можно вычислить значения этого многочлена на элементах расширенного поля Галуа $GF(q^m)$ в точках, кото-

рые являются корнями порождающего многочлена $g(z)$, т.е. в точках $\beta_1, \beta_2, \ldots, \beta_r$. Такое вычисление дает компоненты синдрома

$S_j = y(\beta_j), \quad j = 1, 2, \ldots, r.$

Кодовый многочлен в этих точках равен нулю $x(\beta_1) = x(\beta_2) = \ldots x(\beta_r) = 0$, так как он представляет собой произведение информационного многочлена на порождающий многочлен в соответствии с выражением (40). Поэтому компоненты синдрома зависят только от конфигурации ошибок:

$$S_j = y(\beta_j) = x(\beta_j) + e(\beta_j) = e(\beta_j), \quad j = 1, 2, \ldots, r.$$

Используя выражение для многочлена ошибок в форме $e(z) = e_{n-1}z^{n-1} + e_{n-2}z^{n-2} + \ldots + e_1 z + e_0$, находим:

$$S_j = e_{n-1}\beta_j^{n-1} + e_{n-2}\beta_j^{n-2} + \ldots + e_1\beta_j + e_0 = \sum_{l=0}^{n-1} e_l \beta_j^l, \quad j = 1, 2, \ldots, r.$$

Элементы S_j не являются коэффициентами синдромного многочлена, но они дают эквивалентную информацию. Итак, получена система r уравнений, которая содержит только величины, определяемые ошибками. Если эти уравнения можно решить относительно величин e_i, то станет возможным рассчитать многочлен ошибок.

Коды Боуза-Чоудхури-Хоквингема

Коды Боуза-Чоудхури-Хоквингема (БЧХ) являются подклассом циклических кодов. Они определяются следующим образом. Пусть заданы числа q и m, пусть β – элемент поля Галуа $GF(q^m)$, порядок которого равен n. Для любого положительного числа t и любого целого числа j_0 код БЧХ является циклическим кодом с длиной n и порождающим многочленом

$$g(z) = НОК [f_{j_0}(z), f_{j_0+1}, \ldots, f_{j_0+2t-1}], \qquad (42)$$

где $f_{j_0}(z), f_{j_0+1}(z), \ldots, f_{j_0+2t-1}(z)$ – минимальные многочлены элементов $\beta^{j_0}, \beta^{j_0+1}, \ldots, \beta^{j_0+2t-1}$.

Таким образом, в качестве корней порождающего многочлена задаются 2t последовательных степени произвольного элемента β расширенного поля Галуа $GF(q^m)$. Длина кодового слова над полем Галуа GF(q) равна порядку элемента β, то есть наименьшему числу n, для которого $\beta^n=1$. Число j_0, равное начальному значению показателя степени элемента β, часто выбирают равным единице, что во многих случаях приводит к порождающему многочлену наименьшей степени. Если требуется большая длина кода, то выбирается элемент поля с наибольшим порядком, то есть примитивный элемент. Длина кода в этом случае будет равна $n = (q^m-1)$. Число t определяет конструктивное число исправляемых ошибок, то есть число исправляемых ошибок, задаваемое при построении кода. Оно связано с конструктивным минимальным расстоянием кода d соотношением $d = 2t+1$. Истинное минимальное расстояние d* может быть больше, чем конструктивное. Последовательные степени примитивного элемента α, минимальные элементы которых используются при построении порождающего многочлена, можно записать следующим образом: $\alpha^1, \alpha^2, \ldots, \alpha^{2t} = \alpha^1, \alpha^2, \ldots, \alpha^{d-1}$.

При сделанных допущениях алгоритм построения кода БЧХ оказывается следующим:

- задаются числа q и m;
- строится поле Галуа $GF(q^m)$ с использованием примитивного многочлена степени m;
- находятся минимальные многочлены $f_j(z)$, j = 1, 2,… , 2t для степеней примитивного элемента $\alpha^1, \alpha^2, \ldots, \alpha^{2t}$, где t – число ошибок, которые необходимо исправлять;
- находится порождающий многочлен кода и определяется длина информационного слова k.

$g(z) = НОК [f_1(z), f_2(z), \ldots f_{2t}(z)]$.

В качестве примера построим код БЧХ над полем GF(2), исправляющий 2 ошибки, задавая $q = 2$, $m = 4$, $t = 2$. Длина кода при заданных параметрах будет равна $n = (q^m-1) = 15$.

Представление поля $GF(2^4)$, построенного с использованием примитивного многочлена $p(z) = z^4+z+1$, дано в таблице.

Элементы поля GF(24) и минимальные многочлены

№	Степенные обозначения	Многочленные обозначения	Двоичные обозначения	Целочисленные обозначения	Минимальные многочлены
1	0	0	0000	0	
2	α^0	1	0001	1	z+1
3	α^1	z	0010	2	z^4+z+1
4	α^2	z^2	0100	4	z^4+z+1
5	α^3	z^3	1000	8	$z^4+z^3+z^2+z+1$
6	α^4	z+1	0011	3	z^4+z+1
7	α^5	z^2+z	0110	6	z^2+z+1
8	α^6	z^3+z^2	1100	12	$z^4+z^3+z^2+z+1$
9	α^7	z^3+z+1	1011	11	z^4+z^3+1
10	α^8	z^2+1	0101	5	z^4+z+1
11	α^9	z^3+z	1010	10	$z^4+z^3+z^2+z+1$
12	α^{10}	z^2+z+1	0111	7	z^2+z+1
13	α^{11}	z^3+z^2+z	1110	14	z^4+z^3+1
14	α^{12}	z^3+z^2+z+1	1111	15	$z^4+z^3+z^2+z+1$
15	α^{13}	z^3+z^2+1	1101	13	z^4+z^3+1
16	α^{14}	z^3+1	1001	9	z^4+z^3+1

Минимальные многочлены для степеней примитивного элемента $\alpha^1, \alpha^2, \alpha^3, \alpha^4$:

$$f_1(z) = z^4+z+1,$$
$$f_2(z) = z^4+z+1,$$
$$f_3(z) = z^4+z^3+z^2+z+1,$$
$$f_4(z) = z^4+z+1.$$

Из списка минимальных многочленов, приведенных в таблице, видно, что минимальные многочлены для четных степеней примитивного элемента α равны многочленам для меньших степеней. Это несколько упрощает нахождение порождающего многочлена кода:

$$g(z) = НОК [(z^4+z+1), (z^4+z+1), (z^4+z^3+z^2+z+1), (z^4+z+1)] = (z^4+z+1)*(z^4+z^3+z^2+z+1) = z^8+z^7+z^6+ z^4+1.$$

Степень порождающего многочлена равна 8. В общем виде эта степень записывается как (n-k) = 8, следовательно длина информационного слова k = 7. Итак, построен (15, 7)-код БЧХ, который имеет длину 15 символов и исправляет 2 ошибки.

Продолжение следует