Check polynomials and parity check matrices for cyclic codes

Let C be a cyclic $[n,k]$-code with the generator polynomial $g(x)$ (of degree $n - k$). By the last theorem $g(x)$ is a factor of $x^n - 1$. Hence

$$x^n - 1 = g(x)h(x)$$

for some $h(x)$ of degree k (where $h(x)$ is called the check polynomial of C).

Theorem Let C be a cyclic code in R_n with a generator polynomial $g(x)$ and a check polynomial $h(x)$. Then an $c(x) \in R_n$ is a codeword of C if $c(x)h(x) \equiv 0$ - this and next congruences are modulo $x^n - 1$.

Proof Note, that $g(x)h(x) = x^n - 1 \equiv 0$

(i) $c(x) \in C \Rightarrow c(x) = a(x)g(x)$ for some $a(x) \in R_n$

$$\Rightarrow c(x)h(x) = a(x)\underbrace{g(x)h(x)}_{\equiv 0} \equiv 0.$$

(ii) $c(x)h(x) \equiv 0$

$$c(x) = q(x)g(x) + r(x),\ deg\ r(x) < n - k = deg\ g(x)$$

$$c(x)h(x) \equiv 0 \Rightarrow r(x)h(x) \equiv 0 \pmod{x^n - 1}$$

Since $deg\ (r(x)h(x)) < n - k + k = n$, we have $r(x)h(x) = 0$ in $F[x]$ and therefore

$$r(x) = 0 \Rightarrow c(x) = q(x)g(x) \in C.$$

POLYNOMIAL REPRESENTATION of DUAL CODES

Since *dim* (⟨h(x)⟩) = *n* - *k* = *dim* ($C^{\perp}$) we might easily be fooled to think that the check polynomial h(*x*) of the code *C* generates the dual code $C^{\perp}$.

Reality is “slightly different":

Theorem Suppose *C* is a cyclic [*n,k*]-code with the check polynomial

$$h(x) = h_0 + h_1 x + \ldots + h_k x^k,$$

then

(i) a parity-check matrix for *C* is

$$H = \begin{pmatrix} h_k & h_{k-1} & \ldots & h_0 & 0 & \ldots & 0 \\ 0 & h_k & \ldots & h_1 & h_0 & \ldots & 0 \\ .. & .. & & & & & \\ 0 & 0 & \ldots & 0 & h_k & \ldots & h_0 \end{pmatrix}$$

(ii) $C^{\perp}$ is the cyclic code generated by the polynomial

$$\bar{h}(x) = h_k + h_{k-1}x + \ldots + h_0 x^k$$

i.e. the reciprocal polynomial of h(*x*).

POLYNOMIAL REPRESENTATION of DUAL CODES

Proof A polynomial $c(x) = c_0 + c_1 x + \dots + c_{n-1}x^{n-1}$ represents a code from *C* if $c(x)h(x) = 0$. For $c(x)h(x)$ to be 0 the coefficients at $x^k, \dots, x^{n-1}$ must be zero, i.e.

$$\begin{array}{c} c_0 h_k + c_1 h_{k-1} + \dots + c_k h_0 = 0 \\ c_1 h_k + c_2 h_{k-1} + \dots + c_{k+1} h_0 = 0 \\ \cdot\cdot \qquad\qquad\qquad\qquad \cdot\cdot \\ c_{n-k-1} h_k + c_{n-k} h_{k-1} + \dots + c_{n-1} h_0 = 0 \end{array}$$

Therefore, any codeword $c_0 c_1 \dots c_{n-1} \in C$ is orthogonal to the word $h_k h_{k-1} \dots h_0 00 \dots 0$ and to its cyclic shifts.

Rows of the matrix *H* are therefore in $C^{\perp}$. Moreover, since $h_k = 1$, these row-vectors are linearly independent. Their number is $n - k = dim\,(C^{\perp})$. Hence *H* is a generator matrix for $C^{\perp}$, i.e. a parity-check matrix for *C*.

In order to show that $C^{\perp}$ is a cyclic code generated by the polynomial

$$\bar{h}(x) = h_k + h_{k-1}x + \dots + h_0 x^k$$

it is sufficient to show that $\bar{h}(x)$ is a factor of $x^n - 1$.

Observe that $\bar{h}(x) = x^k h(x^{-1})$ and since $h(x^{-1})g(x^{-1}) = (x^{-1})^n - 1$

we have that $x^k h(x^{-1}) x^{n-k} g(x^{-1}) = x^n (x^{-n} - 1) = 1 - x^n$

and therefore $\bar{h}(x)$ is indeed a factor of $x^n - 1$.

ENCODING with CYCLIC CODES I

Encoding using a cyclic code can be done by a multiplication of two polynomials - a message polynomial and the generating polynomial for the cyclic code.

Let C be an (n,k)-code over an field F with the generator polynomial
$g(x) = g_0 + \quad g_1 x + \ldots + g_{r-1} x^{r-1}$ of degree $r = n - k$.

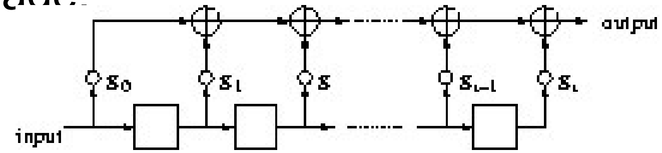
If a message vector m is represented by a polynomial $m(x)$ of degree k and m is encoded by

$$m \Rightarrow c = mG_1,$$

then the following relation between $m(x)$ and $c(x)$ holds

$$c(x) = m(x)g(x).$$

Such an encoding can be realized by the shift register shown in Figure below, where input is the k-bit message to be encoded followed by $n - k$ 0' and the output will be the encoded message.



Shift-register encodings of cyclic codes. Small circles represent multiplication by the corresponding constant, ⊕ nodes represent modular addition, squares are delay elements

ENCODING of CYCLIC CODES II

Another method for encoding of cyclic codes is based on the following (so called systematic) representation of the generator and parity-check matrices for cyclic codes.

Theorem Let C be an (n,k)-code with generator polynomial $g(x)$ and $r = n - k$. For $i = 0,1,\ldots,k - 1$, let $G_{2,i}$ be the length n vector whose polynomial is $G_{2,i}(x) = x^{r+I} - x^{r+I}$ mod $g(x)$. Then the $k * n$ matrix G_2 with row vectors $G_{2,I}$ is a generator matrix for C.

Moreover, if $H_{2,J}$ is the length n vector corresponding to polynomial $H_{2,J}(x) = x^j$ mod $g(x)$, then the $r * n$ matrix H_2 with row vectors $H_{2,J}$ is a parity check matrix for C. If the message vector m is encoded by

$$m \Rightarrow c = mG_2,$$

then the relation between corresponding polynomials is

$$c(x) = x^r m(x) - [x^r m(x)] \bmod g(x).$$

On this basis one can construct the following shift-register encoder for the case of a systematic representation of the generator for a cyclic code:

Shift-register encoder for systematic representation of cyclic codes. Switch A is closed for first k ticks and closed for last r ticks; switch B is down for first k ticks and up for last r ticks.

Hamming codes as cyclic codes

Definition (Again!) Let *r* be a positive integer and let *H* be an $r * (2^r - 1)$ matrix whose columns are distinct non-zero vectors of $V(r,2)$. Then the code having *H* as its parity-check matrix is called binary **Hamming code** denoted by *Ham* (*r*,2).

It can be shown that binary Hamming codes are equivalent to cyclic codes.

Theorem The binary Hamming code *Ham* (*r*,2) is equivalent to a cyclic code.

Definition If p(*x*) is an irreducible polynomial of degree *r* such that *x* is a primitive element of the field *F*[*x*] / p(*x*), then p(*x*) is called a primitive polynomial.

Theorem If p(*x*) is a primitive polynomial over *GF*(2) of degree *r*, then the cyclic code ⟨p(*x*)⟩ is the code *Ham* (*r*,2).

Hamming codes as cyclic codes

Example Polynomial $x^3 + x + 1$ is irreducible over $GF(2)$ and x is primitive element of the field $F_2[x] / (x^3 + x + 1)$.

$$F_2[x] / (x^3 + x + 1) =$$

$$\{0, x, x^2, x^3 = x + 1, x^4 = x^2 + x, x^5 = x^2 + x + 1, x^6 = x^2 + 1\}$$

The parity-check matrix for a cyclic version of *Ham* (*3*,*2*)

$$H = \begin{pmatrix} 1 & 0 & 0 & 1 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 & 1 & 1 & 0 \\ 0 & 0 & 1 & 0 & 1 & 1 & 1 \end{pmatrix}$$

PROOF of THEOREM

The binary Hamming code *Ham* (r,2) is equivalent to a cyclic code.

It is known from algebra that if p(x) is an irreducible polynomial of degree r, then the ring $F_2[x]$ / p(x) is a field of order 2^r.

In addition, every finite field has a primitive element. Therefore, there exists an element α of $F_2[x]$ / p(x) such that

$$F_2[x] / p(x) = \{0, 1, \alpha, \alpha^2, \ldots, \alpha^{2r-2}\}.$$

Let us identify an element $a_0 + a_1 + \ldots a_{r-1}x^{r-1}$ of $F_2[x]$ / p(x) with the column vector

$$(a_0, a_1, \ldots, a_{r-1})^T$$

and consider the binary r * (2^r -1) matrix

$$H = [\, 1 \;\; \alpha \;\; \alpha^2 \; \ldots \; \alpha^{2^\wedge r-2} \,].$$

Let now C be the binary linear code having H as a parity check matrix.

Since the columns of H are all distinct non-zero vectors of $V(r,2)$, C = *Ham* (r,2).

Putting $n = 2^r - 1$ we get

$$C = \{f_0 f_1 \ldots f_{n-1} \in V(n, 2) \mid f_0 + f_1 \alpha + \ldots + f_{n-1} \alpha^{n-1} = 0 \qquad (2)$$

$$= \{f(x) \in R_n \mid f(\alpha) = 0 \text{ in } F_2[x] / p(x)\} \qquad (3)$$

If f(x) $\in C$ and r(x) $\in R_n$, then r(x)f(x) $\in C$ because

$$r(\alpha)f(\alpha) = r(\alpha) \bullet 0 = 0$$

and therefore, by one of the previous theorems, this version of *Ham* (r,2) is cyclic.

BCH codes and Reed-Solomon codes

To the most important cyclic codes for applications belong BCH codes and Reed-Solomon codes.

Definition A polynomial p is said to be *minimal* for a complex number x in Z_q if $p(x) = 0$ and p is irreducible over Z_q.

Definition A cyclic code of codewords of length n over Z_q, $q = p^r$, p is a prime, is called BCH code[1] of distance d if its generator $g(x)$ is the least common multiple of the minimal polynomials for

$$\omega^{l}, \omega^{l+1}, \dots, \omega^{l+d-2}$$

for some l, where

ω is the primitive n-th root of unity.

If $n = q^m - 1$ for some m, then the BCH code is called primitive.

Definition A Reed-Solomon code is a primitive BCH code with $n = q - 1$.

Properties:

- Reed-Solomon codes are self-dual.

[1] BHC stands for Bose and Ray-Chaudhuri and Hocquenghem who discovered these codes.

CONVOLUTION CODES

Very often it is important to encode an infinite stream or several streams of data – say bits.

Convolution codes, with simple encoding and decoding, are quite a simple generalization of linear codes and have encodings as cyclic codes.

An *(n,k)* convolution code (CC) is defined by an *k x n generator matrix, entries of which are polynomials over* F$_2$

For example,

$$G_1 = [x^2 + 1, x^2 + x + 1]$$

is the generator matrix for a (2,1) convolution code CC$_1$ and

$$G_2 = \begin{pmatrix} 1+x & 0 & x+1 \\ 0 & 1 & x \end{pmatrix}$$

is the generator matrix for a (3,2) convolution code CC$_2$

ENCODING of FINITE POLYNOMIALS

An (n,k) convolution code with a k x n generator matrix G can be usd to encode a k-tuple of plain-polynomials (polynomial input information)

$$I=(I_0(x),\ I_1(X),\ldots,I_{k-1}(x))$$

to get an n-tuple of crypto-polynomials

$$C=(C_0(x),\ C_1(x),\ldots,C_{n-1}(x))$$

As follows

$$C= I\ .\ G$$

EXAMPLES

EXAMPLE 1

$$(x^3 + x + 1).G_1 = (x^3 + x + 1)\ .\ (x^2 + 1,\ x^2 + x + 1]$$

$$= (x^5 + x^2 + x\ \ + 1,\ x^5 + x^4 + 1)$$

EXAMPLE 2

$$(x^2 + x, x^3 + 1).G_2 = (x^2 + x, x^3 + 1).\begin{pmatrix} 1\,0\,x + 1 \\ 0\,1\quad x \end{pmatrix}$$

ENCODING of INFINITE INPUT STREAMS

The way infinite streams are encoded using convolution codes will be Illustrated on the code CC_1.

An input stream $I = (I_0, I_1, I_2, \ldots)$ is mapped into the output stream $C = (C_{00}, C_{10}, C_{01}, C_{11} \ldots)$ defined by

$$C_0(x) = C_{00} + C_{01}x + \ldots = (x^2 + 1)\, I(x)$$

and

$$C_1(x) = C_{10} + C_{11}x + \ldots = (x^2 + x + 1)\, I(x).$$

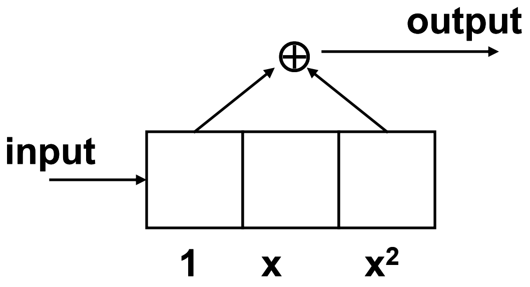
The first multiplication can be done by the first shift register from the next figure; second multiplication can be performed by the second shift register on the next slide and it holds

$$C_{0i} = I_i + I_{i+2}, \qquad C_{1i} = I_i + I_{i-1} + I_{i-2}.$$

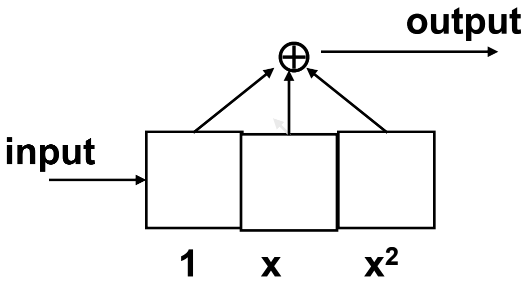
That is the output streams C_0 and C_1 are obtained by convolving the input stream with polynomials of G_1,

ENCODING

The first shift register



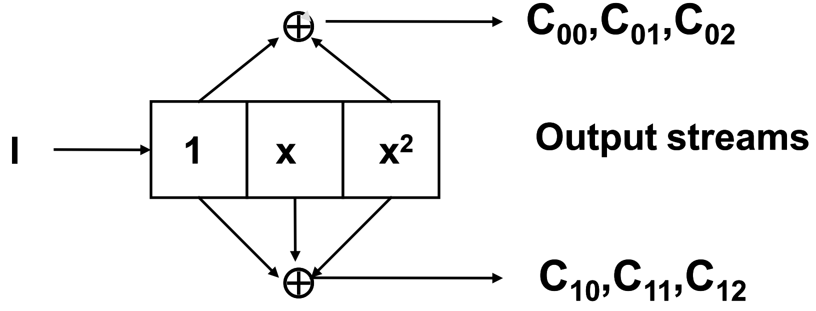
will multiply the input stream by x^2+1 and the second shift register



will multiply the input stream by x^2+x+1.

ENCODING and DECODING

The following shift-register will therefore be an encoder for the code CC_1



For encoding of convolution codes so called

Viterbi algorithm

Is used.