Part III

Cyclic codes

CHAPTER 3: CYCLIC CODES, CHANNEL CODING, LIST DECODING

Cyclic codes are very special linear codes. They are of large interest and importance for several reasons:

- They **posses a rich algebraic structure** that can be utilized in a variety of ways.
- They **have extremely concise specifications**.
- Their encodings **can be efficiently implemented** using simple machinery - shift registers.
- Many of the practically very important codes are cyclic.

Channel codes are used to **encode streams of data** (bits). Some of them, as **Concatenated codes** and **Turbo codes**, **reach theoretical Shannon bound concerning efficiency**, and are currently used very often in practice.

List decoding is a new decoding mode capable to deal, in an approximate way, with cases of many errors, and in such a case to perform better than classical **unique decoding**.

Locally decodable codes can be seen as theoretical extreme of coding theory with deep theoretical implications.

IMPORTANT NOTE

In order to specify a non-linear binary code with 2^k codewords of length n one may need to write down

$$2^k$$

codewords of length n.

In order to specify a linear binary code of the dimension k with 2^k codewords of length n it is sufficient to write down

$$k$$

codewords of length n.

In order to specify a binary cyclic code with 2^k codewords of length n it is sufficient to write down

$$1$$

codeword of length n.

BASIC DEFINITION AND EXAMPLES

Definition A code C is cyclic if

(i) **C is a linear code;**
(ii) **any cyclic shift of a codeword is also a codeword, i.e. whenever $a_0, \dots a_{n-1} \in C$, then also $a_{n-1}a_0 \dots a_{n-2} \in C$ and $a_1 a_2 \dots a_{n-1} a_0 \in C$.**

Example

(i) Code $C = \{000, 101, 011, 110\}$ is cyclic.
(ii) Hamming code $Ham(3,2)$: with the generator matrix

$$G = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 \end{bmatrix}$$

is equivalent to a cyclic code.

(iii) The binary linear code $\{0000, 1001, 0110, 1111\}$ is not cyclic, but it is equivalent to a cyclic code.
(iv) Is Hamming code $Ham(2,3)$ with the generator matrix

$$\begin{bmatrix} 1 & 0 & 1 & 1 \\ 0 & 1 & 1 & 2 \end{bmatrix}$$

(a) cyclic?
(b) or at least equivalent to a cyclic code?

FREQUENCY of CYCLIC CODES

Comparing with linear codes, cyclic codes are quite scarce. For example, there are 11 811 linear [7,3] binary codes, but only two of them are cyclic.

Trivial cyclic codes. For any field F and any integer $n \geq 3$ there are always the following cyclic codes of length n over F:

- **No-information code** - code consisting of just one all-zero codeword.
- **Repetition code** - code consisting of all codewords (a, a, ...,a) for $a \in F$.
- **Single-parity-check code** - code consisting of all codewords with parity 0.
- **No-parity code** - code consisting of all codewords of length n

For some cases, for example for $n = 19$ and $F = GF(2)$, the above four trivial cyclic codes are the only cyclic codes.

AN EXAMPLE of a CYCLIC CODE

Is the code with the following generator matrix cyclic?

$$G = \begin{bmatrix} 1 & 0 & 1 & 1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 & 1 & 1 & 0 \\ 0 & 0 & 1 & 0 & 1 & 1 & 1 \end{bmatrix}$$

It is. It has, in addition to the codeword 0000000, the following codewords

$$c_1 = 1011100 \qquad c_2 = 0101110 \qquad c_3 = 0010111$$
$$c_1 + c_2 = 1110010 \qquad c_1 + c_3 = 1001011 \qquad c_2 + c_3 = 0111001$$
$$c_1 + c_2 + c_3 = 1100101$$

and it is cyclic because the right shifts have the following impacts

$$c_1 \to c_2, \qquad c_2 \to c_3, \qquad c_3 \to c_1 + c_3$$
$$c_1 + c_2 \to c_2 + c_3, \qquad c_1 + c_3 \to c_1 + c_2 + c_3, \qquad c_2 + c_3 \to c_1$$
$$c_1 + c_2 + c_3 \to c_1 + c_2$$

POLYNOMIALS over GF(q)

A codeword of a cyclic code is usually denoted by

$$a_0 a_1 \dots a_{n-1}$$

and to each such a codeword the polynomial

$$a_0 + a_1 x + a_2 x^2 + \dots + a_{n-1} x^{n-1}$$

is usually associated – an ingenious idea!!.

NOTATION: $F_q[x]$ will denote the set of all polynomials $f(x)$ over $GF(q)$.

$deg(f(x))$ = the largest m such that x^m has a non-zero coefficient in $f(x)$.

Multiplication of polynomials If $f(x)$, $g(x) \in FQ[x]$, then

$$deg(f(x)g(x)) = deg(f(x)) + deg(g(x)).$$

Division of polynomials For every pair of polynomials $a(x)$, $b(x) \neq 0$ in $F_q[x]$ there exists a unique pair of polynomials $q(x)$, $r(x)$ in $F_q[x]$ such that

$$a(x) = q(x)b(x) + r(x), deg(r(x)) < deg(b(x)).$$

Example Divide $x^3 + x + 1$ by $x^2 + x + 1$ in $F_2[x]$.
Definition Let $f(x)$ be a fixed polynomial in $F_q[x]$. Two polynomials $g(x)$, $h(x)$ are said to be congruent modulo $f(x)$, notation

$$g(x) \equiv h(x) \pmod{f(x)},$$

if $g(x) - h(x)$ is divisible by $f(x)$.

EXAMPLE

If $x^3 + x + 1$ is divided by $x^2 + x = 1$, then

$$x^3 + x + 1 = (x^2 + x + 1)(x - 1) + x$$

and therefore the result of the division is $x - 1$ and the remainder is x.

NOTICE

A **code** C of the words of length **n**
is a set of codewords of length n

$$a_0 a_1 a_2 \dots a_{n-1}$$

or C can be seen as
a set of polynomials of the degree (at most) $n-1$

$$a_0 + a_1 x + a_2 x^2 + \dots + a_{n-1} x^{n-1}$$

RINGS of POLYNOMIALS

For any polynomial $f(x)$, the set of all polynomials in $F_q[x]$ of degree less than $deg(f(x))$, with addition and multiplication modulo $f(x)$, forms a **ring denoted** $F_q[x]/f(x)$.

Example **Calculate** $(x+1)^2$ **in** $F_2[x]/(x^2+x+1)$. It holds

$$(x+1)^2 = x^2 + 2x + 1 \equiv x^2 + 1 \equiv x \pmod{x^2+x+1}.$$

How many elements has $F_q[x]/f(x)$?
Result $|F_q[x]/f(x)| = q^{deg(f(x))}$.

Example Addition and multiplication tables for $F_2[x]/(x^2+x+1)$

+	0	1	x	1+x
0	0	1	x	1+x
1	1	0	1+x	x
x	x	1+x	0	1
1+x	1+x	x	1	0

•	0	1	x	1+x
0	0	0	0	0
1	0	1	x	1+x
x	0	x	1+x	1
1+x	0	1+x	1	x

Definition A polynomial $f(x)$ in $F_q[x]$ is said to be **reducible** if $f(x) = a(x)b(x)$, where $a(x), b(x) \in F_q[x]$ and

$$deg(a(x)) < deg(f(x)), \qquad deg(b(x)) < deg(f(x)).$$

If $f(x)$ is not reducible, then it is said to be **irreducible** in $F_q[x]$.
Theorem The ring $F_q[x]/f(x)$ is a field if $f(x)$ is irreducible in $F_q[x]$.

RING (Factor ring) $R_n = F_q[x]/(x^n - 1)$

Computation modulo $x^n - 1$ in the ring $R_n = F_q[x]/(x^n - 1)$

Since $x^n \equiv 1 \pmod{(x^n - 1)}$ we can compute $f(x) \bmod (x^n - 1)$ by replacing, in $f(x)$, x^n by 1, x^{n+1} by x, x^{n+2} by x^2, x^{n+3} by x^3, ...

Replacement of a word

$$w = a_0 a_1 \dots a_{n-1}$$

by a polynomial

$$p(w) = a_0 + a_1 x + \dots + a_{n-1} x^{n-1}$$

is of large importance because

multiplication of $p(w)$ by x in R_n corresponds to a single cyclic shift of w. Indeed,

$$x(a_0 + a_1 x + \dots a_{n-1} x^{n-1}) = a_{n-1} + a_0 x + a_1 x^2 + \dots + a_{n-2} x^{n-1}$$

An ALGEBRAIC CHARACTERIZATION of CYCLIC CODES

Theorem A binary code C of words of length n is cyclic if and only if it satisfies two conditions

(i) $a(x), b(x) \in C \Rightarrow a(x) + b(x) \in C$

(ii) $a(x) \in C, r(x) \in R_n \Rightarrow r(x)a(x) \in C$

Proof

(1) Let C be a cyclic code. C is linear $\Rightarrow$

  (i) holds.

  (ii)

$$\text{If } a(x) \in C, r(x) = r_0 + r_1 x + \dots + r_{n-1} x^{n-1} \text{ then}$$
$$r(x)a(x) = r_0 a(x) + r_1 x a(x) + \dots + r_{n-1} x^{n-1} a(x)$$

  is in C by (i) because summands are cyclic shifts of $a(x)$.

(2) Let (i) and (ii) hold

- Taking $r(x)$ to be a scalar the conditions (i) and (ii) imply linearity of C.
- Taking $r(x) = x$ the conditions (i) and (ii) imply cyclicity of C.

CONSTRUCTION of CYCLIC CODES

Notation For any $f(x) \in R_n$, we can define

$$\langle f(x)\rangle = \{r(x)f(x) \mid r(x) \in R_n\}$$

(with multiplication modulo $x^n - 1$) to be a set of polynomials - a code.

Theorem For any $f(x) \in R_n$, the set $\langle f(x)\rangle$ is a cyclic code (generated by f).

Proof We check conditions (i) and (ii) of the previous theorem.

(i) If $a(x)f(x) \in \langle f(x)\rangle$ and also $b(x)f(x) \in \langle f(x)\rangle$, then

$$a(x)f(x) + b(x)f(x) = (a(x) + b(x))f(x) \in \langle f(x)\rangle$$

(ii) If $a(x)f(x) \in \langle f(x)\rangle$, $r(x) \in R_n$, then

$$r(x)(a(x)f(x)) = (r(x)a(x))f(x) \in \langle f(x)\rangle$$

Example let $C = \langle 1 + x^2\rangle$, $n = 3$, $q = 2$.
In order to determine C we have to compute $r(x)(1 + x^2)$ for all $r(x) \in R_3$.

$$R_3 = \{0, 1, x, 1 + x, x^2, 1 + x^2, x + x^2, 1 + x + x^2\}.$$

Result

$$C = \{0, 1 + x, 1 + x^2, x + x^2\}$$
$$C = \{000, 110, 101, 011\}$$

CHARACTERIZATION THEOREM for CYCLIC CODES

We show that **all cyclic codes C have the form $C = \langle f(x)\rangle$ for some $f(x) \in R_n$.**

Theorem Let C be a non-zero cyclic code in R_n. Then

- there exists a unique monic polynomial $g(x)$ of the smallest degree such that
- $C = \langle g(x)\rangle$
- $g(x)$ is a factor of $x^n - 1$.

Proof

(i) Suppose $g(x)$ and $h(x)$ are two monic polynomials in C of the smallest degree, say **D**.
Then the polynomial $w(x) = g(x) - h(x) \in C$ and it has a smaller degree than **D** and a multiplication by a scalar makes out of $w(x)$ a monic polynomial. Therefore the assumption that $g(x) \neq h(x)$ leads to a contradiction.

(ii) If $a(x) \in C$, then for some $q(x)$ and $r(x)$

$$a(x) = q(x)g(x) + r(x), \qquad (where deg\ r(x) < deg\ g(x)).$$

and therefore

$$r(x) = a(x) - q(x)g(x) \in C.$$

By minimality condition

$$r(x) = 0$$

oand therefore $a(x) \in \langle g(x)\rangle$.

CHARACTERIZATION THEOREM for CYCLIC CODES - continuation

(iii) It has to hold, for some $q(x)$ and $r(x)$

$$x^n - 1 = q(x)g(x) + r(x) \quad \text{with} \quad deg\ r(x) < deg\ g(x)$$

and therefore

$$r(x) \equiv -q(x)g(x) \pmod{x^n - 1} \quad \text{and}$$
$$r(x) \in C \Rightarrow r(x) = 0 \Rightarrow g(x) \quad \text{is therefore a factor of } x^n - 1.$$

GENERATOR POLYNOMIALS - definition

Definition If

$$C = \langle g(x)\rangle,$$

for a cyclic code C, then g is called the generator polynomial for the code C.

HOW TO DESIGN CYCLIC CODES?

The last claim of the previous theorem gives **a recipe to get all cyclic codes of the given length n in GF(q)**

Indeed, all we need to do is to find all factors (**in GF(q)**) of

$$x^n - 1.$$

Problem: Find all binary cyclic codes of length 3.

Solution: Make decomposition

$$x^3 - 1 = \underbrace{(x - 1)(x^2 + x + 1)}_{\text{both factors are irreducible in GF(2)}}$$

Therefore, we have the following generator polynomials and cyclic codes of length 3.

Generator polynomials	Code in R_3	Code in $V(3,2)$
1	R_3	$V(3,2)$
$x + 1$	$\{0, 1 + x, x + x^2, 1 + x^2\}$	$\{000, 110, 011, 101\}$
$x^2 + x + 1$	$\{0, 1 + x + x^2\}$	$\{000, 111\}$
$x^3 - 1\ (= 0)$	$\{0\}$	$\{000\}$

DESIGN of GENERATOR MATRICES for CYCLIC CODES

Theorem Suppose C is a cyclic code of codewords of length n with the generator polynomial

$$g(x) = g_0 + g_1x + \ldots + g_rx^r.$$

Then $dim\ (C) = n - r$ and a generator matrix G_1 for C is

$$G_1 = \begin{pmatrix} g_0 & g_1 & g_2 & \ldots & g_r & 0 & 0 & 0 & \ldots & 0 \\ 0 & g_0 & g_1 & g_2 & \ldots & g_r & 0 & 0 & \ldots & 0 \\ 0 & 0 & g_0 & g_1 & g_2 & \ldots & g_r & 0 & \ldots & 0 \\ \ldots & \ldots & & & & & & & & \ldots \\ 0 & 0 & \ldots & 0 & 0 & \ldots & 0 & g_0 & \ldots & g_r \end{pmatrix}$$

Proof

(i) All rows of G1 are linearly independent.

(ii) The $n - r$ rows of G represent codewords

$$g(x), xg(x), x^2g(x), \ldots, x^{n-r-1}g(x) \quad (*)$$

(iii) It remains to show that every codeword in C can be expressed as a linear combination of vectors from (*).

Indeed, if $a(x) \in C$, then

$$a(x) = q(x)g(x).$$

Since $deg\ a(x) < n$ we have $deg\ q(x) < n - r$.
Hence

$$q(x)g(x) = (q_0 + q_1x + \ldots + q_{n-r-1}x^{n-r-1})g(x)$$
$$= q_0g(x) + q_1xg(x) + \ldots + q_{n-r-1}x^{n-r-1}g(x).$$

EXAMPLE

The task is to determine all ternary codes of length 4 and generators for them.
Factorization of $x^4 - 1$ over $GF(3)$ has the form

$$x^4 - 1 = (x-1)(x^3+x^2+x+1) = (x-1)(x+1)(x^2+1)$$

Therefore, there are $2^3 = 8$ divisors of $x^4 - 1$ and each generates a cyclic code.

Generator polynomial	Generator matrix
1	I_4
$x-1$	$\begin{bmatrix} -1 & 1 & 0 & 0 \\ 0 & -1 & 1 & 0 \\ 0 & 0 & -1 & 1 \end{bmatrix}$
$x+1$	$\begin{bmatrix} 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 1 \end{bmatrix}$
x^2+1	$\begin{bmatrix} 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \end{bmatrix}$
$(x-1)(x+1) = x^2-1$	$\begin{bmatrix} -1 & 0 & 1 & 0 \\ 0 & -1 & 0 & 1 \end{bmatrix}$
$(x-1)(x^2+1) = x^3-x^2+x-1$	$\begin{bmatrix} -1 & 1 & -1 & 1 \end{bmatrix}$
$(x+1)(x^2+1)$	$\begin{bmatrix} 1 & 1 & 1 & 1 \end{bmatrix}$
$x^4-1=0$	$\begin{bmatrix} 0 & 0 & 0 & 0 \end{bmatrix}$

COMMENTS

The last matrix is not, however, formally a generator matrix - the corresponding code is empty.

On the previous slide "generator polynomials" $x-1$, x^2-1 and x^3-x^2+x+1 are formally not in R_n because only allowable coefficients are $0, 1, 2$.

A good practice is, however, to use also coefficients -2, and -1 as ones that are equal, modulo 3, to 1 and 2 and they can be replace in such a way also in matrices to be formally fully correct.

EXAMPLE - II

In order to determine all binary cyclic codes of length 7, consider decomposition

$$x^7 - 1 = (x-1)(x^3+x+1)(x^3+x^2+1)$$

Since we want to determine binary codes, all computations should be modulo 2 and therefor all minus signs can be replaced by plus signs. Therefore

$$x^7 + 1 = (x+1)(x^3+x+1)(x^3+x^2+1)$$

Therefore generators for 2^3 binary cyclic codes of length 7 are

$$1, \quad a(x) = x+1, \quad b(x) = x^3+x+1, \quad c(x) = x^3+x^2+1$$
$$a(x)b(x), \quad a(x)c(x), \quad b(x)c(x), \quad a(x)b(x)c(x) = x^7+1$$

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. ($h(x)$ 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 and only if $c(x)h(x) \equiv 0$ –(this and next congruences are all 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

Continuation: Since $dim\ (\langle h(x)\rangle) = n - k = dim(C^\perp)$ 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

$$h(x) = h_0 + h_1x + \ldots + h_kx^k,$$

then

(i) a parity-check matrix for C is

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

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

$$\overline{h}(x) = h_k + h_{k-1}x + \ldots + h_0x^k$$

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

POLYNOMIAL REPRESENTATION of DUAL CODES

Proof A polynomial $c(x) = c_0 + c_1x + \ldots + c_{n-1}x^{n-1}$ represents a code from C if $c(x)h(x) = 0$. For $c(x)h(x)$ to be 0 the coefficients at $x^k, \ldots, x^{n-1}$ must be zero, i.e.

$$c_0h_k + c_1h_{k-1} + \ldots + c_kh_0 = 0$$
$$c_1h_k + c_2h_{k-1} + \ldots + c_{k+1}h_0 = 0$$
$$\ldots$$
$$c_{n-k-1}h_k + c_{n-k}h_{k-1} + \ldots + c_{n-1}h_0 = 0$$

Therefore, any codeword $c_0c_1 \ldots c_{n-1} \in C$ is orthogonal to the word $h_kh_{k-1} \ldots h_000 \ldots 0$ 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

$$\overline{h}(x) = h_k + h_{k-1}x + \ldots + h_0x^k$$

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

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

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

and therefore $\overline{h}(x)$ 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 (codeword) polynomial and the generating polynomial for the code.

Let C be a cyclic $[n, k]$-code over a Galois field with the generator polynomial

$$g(x) = g_0 + g_1x + \ldots + g_{r-1}x^{r-1} \text{ of degree } r - 1 = n - k - 1.$$

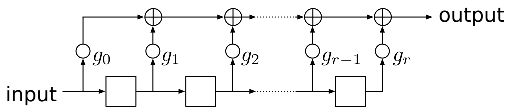
If a message vector m is represented by a polynomial $m(x)$ of the degree k and m is encoded, using the generator matrix G induced by $g(x)$, then

$$m \Rightarrow c = mG,$$

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

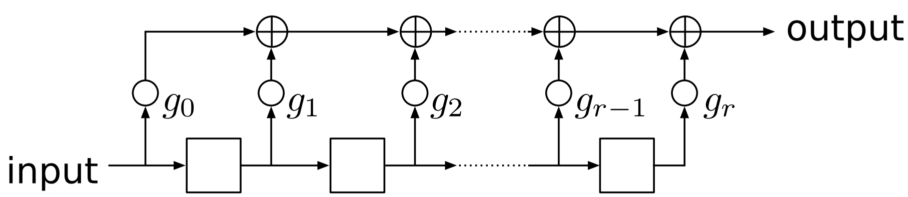
$$c(x) = m(x)g(x).$$

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



Shift-register encodings of cyclic codes. Small circles represent multiplication by the corresponding constant, $\bigoplus$ nodes represent modular additions, squares are shift cells

EXAMPLE



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

The input (message) is given by a polynomial $m^{k-1}x^{k-1} + \dots m^2x^2 + m_1x + m_0$ and therefore the input to the shift register is the word

$$m_{k-1}m_{k-2}\dots m_2m_1m_0 \longrightarrow\longrightarrow\longrightarrow$$

MULTIPLICATION of POLYNOMIALS by SHIFT-REGISTERS

Let us compute

$$(m_0 + m1x + \dots m_{k-1}x^{k-1}) \times (g_0 + g_1x + g_2x^2 \dots g_{r-1}x^{r-1})$$
$$=$$
$$m_0g_0$$
$$+$$
$$(m_0g_1 + m_1g_0)x$$
$$+$$
$$(m_0g_2 + m_1g_1 + m_2g_0)x^2$$
$$+$$
$$(m_0g_3 + m_1g_2 + m_2g_1 + m_3g_0)x^3$$
$$+$$
$$\vdots$$

HAMMING CODES as CYCLIC CODES I

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

It can be shown:

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 $\langle p(x)\rangle$ is the code $Ham\ (r, 2)$.

HAMMING CODES as CYCLIC CODES II

Hamming $ham\ (3, 2)$ code has generator polynomial $x^3 + x = 1$.

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

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

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

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

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

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

$$(a_0, a_1, \dots, a_{r-1})^\top$$

and consider the binary $r \times (2^r - 1)$ matrix

$$H = [1 \ \alpha \ \alpha^2 \dots \alpha^{2^r-2}].$$

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

$$C = \{f_0 f_1 \dots f_{n-1} \in V(n, 2) | f_0 + f_1\alpha + \dots + f_{n-1}\alpha^{n-1} = 0\} \quad (1)$$
$$= \{f(x) \in R_n | f(\alpha) = 0 \text{ in } F_2[x]/p(x)\} \quad (2)$$

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.

EXAMPLES of CYCLIC CODES

EXAMPLES of CYCLIC CODES

GOLAY CODES - DESCRIPTION

Golay codes G_{24} and G_{23} were used by spacecraft Voyager I and Voyager II to transmit color pictures of Jupiter and Saturn. Generator matrix for G_{24} has the form

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

G_{24} is $(24, 12, 8)$-code and the weights of all codewords are multiples of 4. G_{23} is obtained from G_{24} by deleting last symbols of each codeword of G_{24}. G_{23} is $(23, 12, 7)$-code. It is a perfect code.

GOLAY CODE II

Golay code G_{23} is a $(23, 12, 7)$-code and can be defined also as the cyclic code generated by the codeword

11000111010100000000000

This code can be constructed via factorization of $x^{23} - 1$.
In his search for perfect codes Golay observed that

$$\sum_{j=0}^{3} \binom{23}{j} = 2^{23-12} = 2^{11}$$

Observe that an $(n, M, 2t + 1)$-code is perfect if

$$M \sum_{i=0}^{t} \binom{n}{i} (q-1)^i = q^n.$$

Golay code G_{24} was used in NASA Deep Space Missions - in spacecraft Voyager 1 and Voyager 2. It was also used in the US-government standards for automatic link establishment in High Frequency radio systems.

Golay codes are named to honour Marcel J. E. Golay - from 1949.

GOLAY CODES - III

Golay [24, 12, 8] code is called also **extended binary Golay code**.

Golay [23, 12, 7] code is called also **perfect binary Golay code**.

It is the linear code generated by the polynomial

$$x^{11} + x^{10} + x^6 + x^5 + x^4 + x^2 + 1/(x^{23} - 1)$$

POLYNOMIAL CODES

A Polynomial code, with codewords of length n, generated by a (generator) polynomial $g(x)$ of degree $m < n$ over a GF(q) is the code whose codewords are represented exactly by those polynomials of degree less than n that are divisible by $g(x)$.

Example For the binary polynomial code with $n = 5$ and $m = 2$ generated by the polynomial $g(x) = x^2 + x + 1$ all codewords are of the form:

$$a(x)g(x)$$

where

$$a(x) \in \{0, 1, x, x + 1, x^2, x^2 + 1, x^2 + x, x^2 + x + 1\}$$

what results in the code with codewords

$$00000, 00111, 01110, 01001,$$

$$11100, 11011, 10010, 10101.$$

BCH CODES and REED-SOLOMON CODES

BCH codes and **Reed-Solomon codes** belong to the most important codes for applications.

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

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

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

for some l, where

$$\omega \text{ is the primitive } n\text{-th root of unity.}$$

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

Applications of BCH codes: satelite communications, compact disc players,disk drives, two-dimensional bar codes,...

Comments: For BCH codes there exist effcient variations of syndrome decoding. A Reed-Solomon code is a special primitive BCH code.

[1] BHC stands for Bose and Ray-Chaudhuri and Hocquenghem who discovered these codes in 1959.

REED-SOLOMON CODES - basic idea behind - I

A message of k symbols can be encoded by viewing these symbols as coefficients of a polynomial of degree $k - 1$ over a finite field of order N, evaluating this polynomial at more than k distinct points and sending the outcomes to the receiver.

Having more than k points of the polynomial allows to determine exactly, through the Lagrangian interpolation, the original polynomial (message).

Variations of Reed-Solomon codes are obtained by specifying ways distinct points are generated and error-correction is performed.

Reed-Solomon codes found many important applications from deep-space travel to consumer electronics.

They are very useful especially in those applications where one can expect that errors occur in bursts - such as ones caused by solar energy.

SOLOMON CODES - BASIC IDEAS II.

Reed-Solomon (RS) codes were discovered in 1960 and since that time they have been applied in CD-ROOMs, wireless communications, space communications, DVD, digital TV.

RS encoding is relatively straightforward, effcient decodings are recent developments.

There several mathematical nontrivial descriptions of RS codes. However the basic idea behind is quite simple.

RS-codes work with groups of bits called symbols.

If a k-symbol message is to be sent, then $n = k + 2s$ symbols are transmitted in order to guarantee a proper decoding of not more than s symbols corruptions.

Example: If $k = 223, s = 16, n = 235$, then up to 16 corrupted symbols can be corrected.

Number of bits in symbols and parameters k and s depend on applications.

A CD-ROOM can correct a burst of up to 4000 consecutive bit-errors.

BASICS of ENCODING and DECODING

If symbols have j bits they are considered as elemnets of GF(2^j)

To a k symbols message $M = (m_0, m_1, \dots, m_{k-1})$ we associate a $k-1$ degree **(message) polynomial**

$$P_M = m_0 + m_1x + m_2x^2 + \dots + m_{k-1}x^{k-1}$$

P_M is uniquely detrmined given any k of its points.

Encoding To encode the message M so that s corruptions of symbols can be corrected we compute and send $n = k + 2s$ values of p_M at poits $x_1, \dots, x_n$, properly chosen in advance.

Decoding Let $y_1, y_2, \dots, y_n$ be evaluations received (with at most s corruptions of symbols. Try to find a subset of at least $k + s$ points from $((x_1, y_1), \dots, (x_n, y_n$ such that a degree $k-1$ polynomial passes through these points. Such a subset has to exist since we start with $k + 2s$ points and at most s are corrupted. Once we have such a subset we know that it matches the evaluation of $p_M(x)$ in at least k distinct x-values.

Since k points uniquely determines a degree $k-1$ polynomial we can construct the polynomial $p_M(x)$ and to get the correct decoding.

REED-SOLOMON CODES - TECHNICALITIES

Reed-Solomon codes RSC(k, q), for $k \leq q$. are codes generator matrix of which has rows labeled by polynomials X^i, $0 \leq i \leq k-1$, columns are labelled by elements $0, 1, \dots, q-1$ and the element in a row labeled by a polynomial p and in a column labeled by an element u is $p(u)$.

Each RSC(k, q) code is $[q, k, q-k+1]$ code

Example Generator matrix for RSC$(3, 5)$ code is

$$\begin{pmatrix} 1 & 1 & 1 & 1 & 1 \\ 0 & 1 & 2 & 3 & 4 \\ 0 & 1 & 4 & 4 & 1 \end{pmatrix}$$

An interesting property of Reed-Solomon codes:

$$\text{RSC}(k, q)^\perp = \text{RSC}(q-k, q).$$

Reed-Solomon codes were used in digital television, satellite communication, wireless communication, bar-codes, compact discs, DVD,...

REED-SOLOMON CODES - HISTORY and APPLICATIONS

- Reed-Solomon (RS) codes are non-binary cyclic codes.
- They were invented by Irving S. Reed and Gustave Solomon in 1960.
- Efficient decoding algorithm for them was invented by Elwyn Berlekamp and James Massey in 1969.
- Using Reed-Solomon codes one can show that it is sufficient to inject $2e$ additional symbols into a message in order to be able to correct e errors.
- Reed-Solomon codes can be decoded efficiently using so-called list decoding method (described next).
- In 1977 RS codes have been implemented in Voyager space program
- The first commercial application of RS codes in mass-consumer products was in 1982.

CHANNEL (STREAMs) CODING

Channel coding is concerned with an efficient encoding of the streams of data and sending them, at the highest possible rate, over a given communication channel and then obtaining the original data reliably, at the receiver side, by decoding the received data efficiently.

Shannon's channel coding theorem says that over many common channels there exist data coding schemes that are able to transmit data reliably at all rates smaller than a certain threshold, called nowadays the **Shannon channel capacity** of a given channel.

Moreover, the probability of a decoding error can be made to decrease exponentially as the block length N of the coding scheme goes to infinity.

However, the complexity of a "naive" optimum decoding scheme increases exponentially with N - therefore such an optimum decoder rapidly becomes infeasible.

As already mentioned, a breakthrough came when D. Forney, in his PhD thesis in 1972, showed that concatenated codes could be used to achieve exponentially decreasing error probabilities at all data rates less than the Shannon channel capacity, with decoding complexity increasing only polynomially with the code block length.

CHANNEL CAPACITY

Channel capacity is the tightest upper bound on the rate of information that can be reliably transmitted over a communication channel.

By the **noisy-channel Shannon coding theorem**, the channel capacity of a given channel is the limiting information rate (in units of information per unit time) that can be achieved with arbitrary small error probability.

CHANNEL CAPAVITY - FORMAL DEFINITION

Let X and Y be raandom variables representing the input and output of the channel.

Let $P_{Y|X}(y|x)$ be the conditional distribution function of Y given X, which is an inherent fixed probability of the communication channel.

The joint distribution $P_{X,Y}(x,y)$ is then defined by

$$P_{X,Y}(x,y) = P_{Y|X}(y|x)P_X(x),$$

where $P_X(x)$ is the marginal distribution.

The channel capacity is then defined by

$$C = \sup_{P_X(x)} I(X,Y)$$

where

$$I(X,Y) = \sum_{y \in Y} \sum_{x \in X} P_{X,Y}(x,y) \log \left(\frac{P_{X,Y}(x,y)}{P_X(x)P_Y(y)} \right)$$

is the mutual distribution - a measure of variables mutual distribution.

CHANNEL (STREAMS) CODING I.

The task of channel coding is to encode streams of data in such a way that if they are sent over a noisy channel errors can be detected and/or corrected by the receiver.

In case no receiver-to-sender communication is allowed, we speak about **forward error correction**.

An important parameter of a channel code is **code rate**

$$r = \frac{k}{n}$$

in case k bits are encoded by n bits.

The code rate express the amount of redundancy in the code - the lower is the rate, the more redundant is the code.

CHANNEL (STREAM) CODING II

Design of a channel code is always a tradeoff between **energy efficiency** and **bandwidth efficiency**.

Codes with lower code rate can usually correct more errors. Consequently, the communication system can operate

- **with a lower transmit power;**
- **transmit over longer distances;**
- **tolerate more interference from the environment;**
- **use smaller antennas;**
- **transmit at a higher data rate.**

These properties make codes with lower code rate energy efficient.

On the other hand such codes require larger bandwidth and decoding is usually of higher complexity.

The selection of the code rate involves a tradeoff between energy efficiency and bandwidth efficiency.

Central problem of channel encoding: encoding is usually easy, but decoding is usually hard.

CONVOLUTION CODES

Our first example of channel codes are convolution codes.

Convolution codes have 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 $k \times n$ 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, denoted 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 denoted CC_2

ENCODING of FINITE POLYNOMIALS

An (n,k) convolution code with a k x n generator matrix G can be used 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

$$C = I \cdot G$$

EXAMPLES

EXAMPLE 1

$$\begin{aligned}(x^3 + x + 1) \cdot G_1 &= (x^3 + x + 1) \cdot (x^2 + 1, x^2 + x + 1) \\ &= (x^5 + x^2 + x + 1, x^5 + x^4 + 1)\end{aligned}$$

EXAMPLE 2

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

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

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

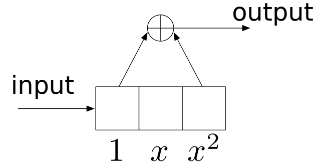
The first multiplication can be done by the first shift register from the next figure; second multiplication can be performed by the second shift register on the next slide and it holds

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

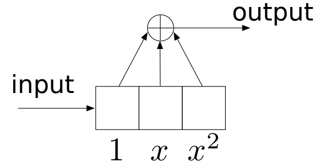
That is the output streams C_0 and C_1 are obtained by convolving the input stream with polynomials of G_1.

ENCODING

The **first shift register**



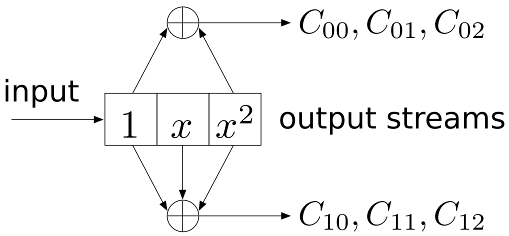
will multiply the input stream by $x^2 + 1$ and the second shift register



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 decoding of convolution codes so called

Viterbi algorithm

Is used.

BIAGWN CHANNELS

Binary Input Additive Gaussian White Noise (BIAGWN) channel, is a continuous channel.

BIAGWN channel with a standard deviation $\sigma \geq 0$ can be seen as a mapping

$$X = \{-1, 1\} \rightarrow R,$$

where R is the set of reals.

The noise of BIAGWN is modeled by continuous Gaussian probability distribution function:

Given $(x, y) \in \{-1, 1\} \times R$, the noise $y - x$ is distributed according to the Gaussian distribution of zero mean and standard derivation σ

$$Pr(y|x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(y-x)^2}{2\sigma^2}}$$

SHANNON CHANNEL CAPACITY

For every combination of bandwidth (W), channel type , signal power (S) and received noise power (N), there is a theoretical upper bound, called **channel capacity** or **Shannon capacity**, on the data transmission rate R for which error-free data transmission is possible.

For BIAGWN channels, that well capture deep space channels, this limit is (by so-called Shannon-Hartley theorem):

$$R < W \log\left(1 + \frac{S}{N}\right) \quad \{\text{bits per second}\}$$

Shannon capacity sets a limit to the energy efficiency of the code.

Till 1993 channel code designers were unable to develop codes with performance close to Shannon capacity limit, that is so called Shannon capacity approaching codes, and practical codes required about twice as much energy as theoretical minimum predicted.

Therefore, there was a big need for better codes with performance (arbitrarily) close to Shannon capacity limits.

Concatenated codes and Turbo codes have such a Shannon capacity approaching property.

CONCATENATED CODES - I

The basic idea of concatenated codes is extremely simple. Input is first encoded by one code C_1 and C_1-output is then encoded by second code C_2. To decode, at first C_2 decoding and then C_1 decoding are used.

In 1972 Forney showed that concatenated codes could be used to achieve exponentially decreasing error probabilities at all data rates less than channel capacity in such a way that decoding complexity increases only polynomially with the code block length.

In 1965 concatenated codes were considered as infeasible. However, already in 1970s technology has advanced sufficiently and they became standardize by NASA for space applications.

CONCATENATED CODES - II

Let $C_{in} : A^k \rightarrow A^n$ be an $[n, k, d]$ code over alphabet A.

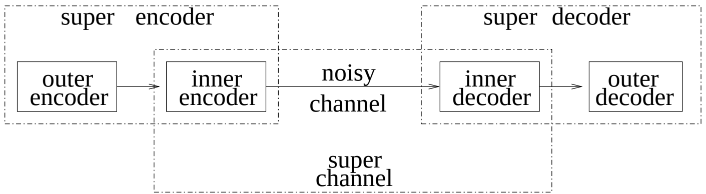
Let $C_{out} : B^K \rightarrow B^N$ be an $[N, K, D]$ code over alphabet B with $|B| = |A|^k$ symbols.

Concatenation of C_{out} (as outer code) with C_{in} (as inner code), denoted $C_{out} \circ C_{in}$ is the $[nN, kK, dD]$ code

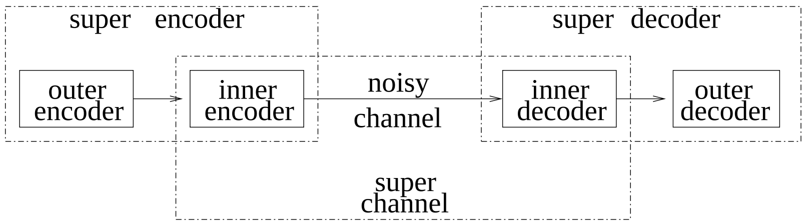
$$C_{out} \circ C_{in} : A^{kK} \rightarrow A^{nN}$$

that maps an input message $m = (m_1, m_2, \ldots, m_K)$ to a codeword $(C_{in}(m_1'), C_{in}(m_2'), \ldots, C_{in}(m_N'))$, where

$$(m_1', m_2', \ldots, m_N') = C_{out}(m_1, m_2, \ldots, m_K)$$

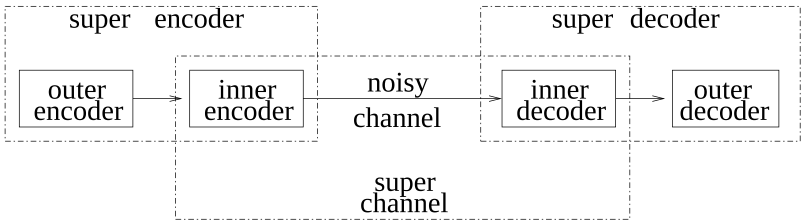


CONCATENATED CODES - III



Of the key importance is the fact that if C_{in} is decoded using the *maximum-likelihood principle* (thus showing an exponentially decreasing error probability with increasing length) and C_{out} is a code with length $N = 2^n r$ that can be decoded in polynomial time in N, then the concatenated code can be decoded in polynomial time with respect to $n2^{nr}$ and has exponentially decreasing error probability even if C_{in} has exponential decoding complexity.

ANOTHER VIEW of CONCATENATED CODES



- **Outer code:** - (n_2, k_2) code over GF(2^{k_1});
- **Inner code**: - (n_1, k_1) binary code
- **Inner decoder** - (n_1, k_1) code
- **Outer decoder** - (n_2, k_2) code

- **length** of such a concatenated code is $n_1 n_2$
- **dimension** of such a concatenated code is $k_1 k_2$
- if **minimal distances** of both codes are d_1 and d_2, then resulting concatenated code has minimal distance $\geq d_1 d_2$.

APPLICATIONS

- Concatenated codes started to be used for deep space communication starting with Voyager program in 1977 and stayed so until the invention of Turbo codes and LDPC codes.
- Concatenated codes are used also on Compact Disc.
- The best concatenated codes for many applications were based on outer Reed-Solomon codes and inner Viterbi-decoded short constant length convolution codes.

EXAMPLE

When the primary antenna failed to deploy on the Galileo mission to Jupiter in 1977, heroic engineering effort was undertaken to design the most powerful concatenated code conceived up to that time, and to program it into the spacecraft computer.

The inner code was a 2^{14} convolution code, decoded by the Viterbi algorithm.

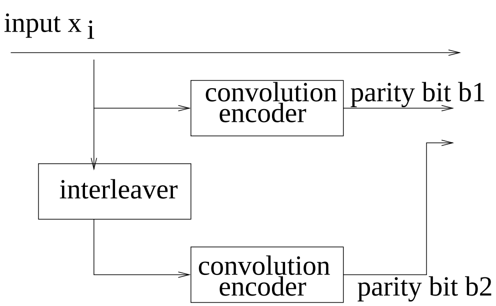
The outer code consisted of multiple Reed-Solomon codes of varying length.

The system achieved a coding gain of more than 10dB at decoding error probabilities of the order 10^{-7}. Original antenna was supposed to send 100,000 bits per second. With a small antenna only 10 b/s could be sent. After all reparations and new codings it was possible to send up to 1000 b/s.

Nowadays when so called iterative decoding is used concatenation of even very simple codes can yield superb performance.

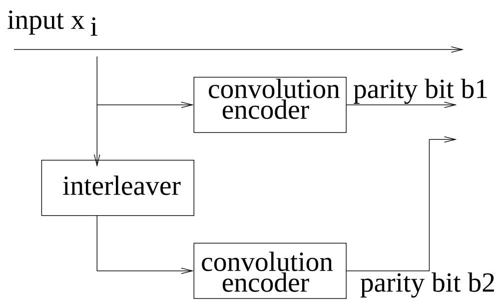
TURBO CODES

Channel coding was revolutionized by invention of Turbo codes. Turbo codes were introduced by Berrou, Glavieux and Thitimajshima in 1993. Turbo codes are specified by special encodings.
A Turbo code can be seen as formed from the parallel composition of two (convolution) codes separated by an interleaver (that permutes blocks of data in a fixed (pseudo)-random way).
A Turbo encoder is formed from the parallel composition of two (convolution) encoders separated by an interleaver.

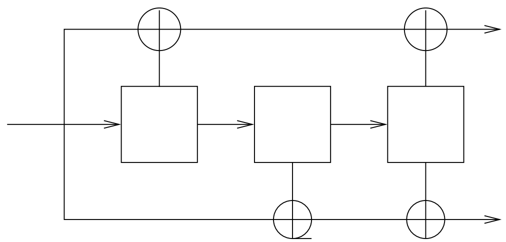


EXAMPLES of TURBO and CONVOLUTION ENCODERS

A Turbo encoder



and a convolution encoder



ADVANTAGES of INTERLEAVING

In case when a word

$$cenaje200kc$$

is transmitted and symbols 7-10 are lost:

$$cenaje....c$$

then very important information is definitely lost.

However, if the input word is permuted according to permutation

$$3, 8, 7, 9, 10, 1, 2, 6, 4, 11, 5$$

then the input will be actually

$$n020kceeacj$$

and if the same four positions are damaged the output will be

$$n020kc....j$$

However, after the inverse permutation the output actually is

$$c.n.j.200k.$$

which is easy to decode correctly!!!!

DECODING and PERFORMANCE of TURBO CODES

- A soft-in-soft-out decoding is used - the decoder gets from the analog/digital demodulator a soft value of each bit - probability that it is 1 and produces only a soft-value for each bit.
- The overall decoder uses decoders for outputs of two encoders that also provide only soft values for bits and by exchanging information produced by two decoders and from the original input bit, the main decoder tries to increase, by an iterative process, likelihood for values of decoded bits and to produce finally hard outcome - a bit 1 or 0.
- Turbo codes performance can be very close to theoretical Shannon limit.
- This was, for example the case for UMTS (the third Generation Universal Mobile Telecommunication System) Turbo code having a less than 1.2-fold overhead. in this case the interleaver worked with block of 40 bits.
- Turbo codes were incorporated into standards used by NASA for deep space communications, digital video broadcasting and both third generation cellular standards.
- Literature: M.C. Valenti and J.Sun: Turbo codes - tutorial, Handbook of RF and Wireless Technologies, 2004 - reachable by Google.

REACHING SHANNON LIMIT

- Though Shannon developed his capacity bound already in 1940, till recently code designers were unable to come with codes with performance close to theoretical limit.
- In 1990 the gap between theoretical bound and practical implementations was still at best about 3dB

  A decibel is a relative measure. If E is the actual energy and E_{ref} is the theoretical lower bound, then the relative energy increase in decibels is

  $$10 \log_{10} \frac{E}{E_{ref}}$$

  Since $\log_{10} 2 = 0.3$ a two-fold relative energy increase equals $3dB$.
- For code rate $\frac{1}{2}$ the relative increase in energy consumption is about 4.8 dB for convolution codes and 0.98 for Turbo codes.

TURBO CODES - SUMMARY

- Turbo codes encoding devices are usually built from two (usually identical) recursive systematic convolution encoders , linked together by nonuniform interleaver (permutation) devices.
- Soft decoding is an iterative process in which each component decoder takes advantage of the work of other at the previous step, with the aid of the original concept of intrinsic information.
- For sufficiently large size of interleavers , the correcting performance of turbo codes, as shown by simulations, appears to be close to the theoretical shannon limit.
- Permutations performed by interleaver can often by specified by simple polynomials that make one-to-one mapping of some sets $\{0, 1, \dots, q-1\}$.

WHY ARE TURBO CODES SO GOOD?

- Turbo codes are linear codes.
- A "good" linear code is one that has mostly high-weight codewords.
- High-weight codewords are desirable because they are more distinct and the decoder can more easily distinguish among them.
- A big advantage of Turbo encoders is that they reduce the number of low-weight codewords because their output is the sum of the weights of the input and two parity output bits.
- A turbo code can be seen as a refinement of concatenated codes plus an iterative algorithm for decoding.

UNIQUE versus LIST DECODING

In the **unique decoding** model of error-correction, considered so far, the task is to find, for a received (corrupted) message w_c, the closest codeword w (in the code used) to w_c.

This error-correction task/model is not sufficiently good in case when the number of errors can be large.

In the **list decoding** model the task is for a received (corrupted) message w_c and a given ϵ to output (list of) all codewords with the distance at most ε from w_c.

List decoding is considered to be successful in case the outputted list contains the codeword that was sent.

It has turned out that for a variety of important codes, including the Reed-Solomon codes, there are efficient algorithms for list decoding that allow to correct a large variety of errors.

List decoding seems to be a stronger error-correcting mode than unique decoding.

LIST DECODING - INTUITIONS BEHIND

For a polynomial-time list decoding algorithm to exist we need that any Hamming ball of a radius pn around a received word (where p is the fraction of errors in terms of the block length n) has a small number of codewords.

This is because the list size itself is a lower bound for the running time of the algorithm. Hence it is required that the list size has to be polynomial in the block length of the code.

A combinatorial consequence of the above requirement is that it implies an upper bound on the rate of the code. List decoding promises to meet this bound.

EFFICIENCY of LIST DECODING - SUMMARY

With list decoding the error-correction performance can double.

It has been shown, non-constructively, for any rate R, that such codes of the rate R exist that can be list decoded up to a fraction of errors approaching $1 - R$.

The quantity $1 - R$ is referred to as the **list decoding capacity**.

For Reed-Solomon codes there is a list decoding up to $1 - \sqrt{2R}$ errors.

LIST DECODING - MATHEMATICAL FORMULATION

Let C be a q-nary linear $[n, k, d]$ error correcting code.

For a given q-nary input word w of length n and a given error bound ε the task is to output a list of codewords of C whose Hamming distance from w is at most ε

We are, naturally, interested only in polynomial, in n, algorithms able to do that.

(p, L)-list decodability: Let C be a q-nary code of codewords of length n; $0 \leq p \leq 1$ and $L > 1$ an integer.

If for every q-nary word w of length n the number of codewords of C withing Hamming distance pn from w is at most L, then the code C is said to be (p, L)-list-decodable.

Theorem let $q \geq 2$, $0 \leq p \leq 1 - 1/q$ and $\varepsilon \geq 0$ then for large enough block length n if the code rate $R \leq 1 - H_q(p) - \varepsilon$, then there exists a $(p, O(1/\varepsilon))$-list decodable code. $[H_q(p) = p \log_q(q-1) - p \log_q p - (1-p)\log_q(1-p)$ is q-ary entropy function.$]$
Moreover, if $R > 1 - H_q(p) + \varepsilon$, then every (p, L)-list-decodable code has $L = q^{\Omega(n)}$

LIST DECODING POTENTIAL

- The concept of list decoding was proposed by Peter Elias in 1950s.
- In 2006 Guruswami and Atri Rudra gave explicit codes that achieve list decoding capacity.
- Their codes are called folded Reed-Solomon codes and they are actually nothing but plain Reed-Solomon codes but viewed as codes over a larger alphabet by careful bundling of codeword symbols.
- List decoding can be seen as formalizing the notion of error-correction when the number of errors is potentially very large. In such a case the received word can actually be closer to other codewords than the transmitted one.
- Algorithms developed for list decoding of several code families found interesting applications in computational complexity theory and in cryptography (for example in construction of hard-core predicates, extractors and pseudo-random generators).

APPLICATIONS in COMPLEXITY THEORY

Surprisingly, list-decoding found interesting applications in computational complexity theory. For example, in

- designing of hard core predicates from one-way permutations;
- predicting witnesses for **NP**-problems;
- designing randomness extractors and pseudorandom generators.

APPENDIX

APPLICATIONS of REED-SOLOMON CODES

- Reed-Solomon codes have been widely used in mass storage systems to correct the burst errors caused by media defects.
- Special types of Reed-Solomon codes have been used to overcome unreliable nature of data transmission over erasure channels.
- Several bar-code systems use Reed-Solomon codes to allow correct reading even if a portion of a bar code is damaged.
- Reed-Solomon codes were used to encode pictures sent by the Voyager spacecraft.
- Modern versions of concatenated Reed-Solomon/Viterbi decoder convolution coding were and are used on the Mars Pathfinder, Galileo, Mars exploration Rover and Cassini missions, where they performed within about 1-1.5dB of the ultimate limit imposed by the shannon capacity.

FUTURE of CODING DEVELOPMENTS

The following reasons are behind increasing needs to develop new and new codes, new and new encoding and decoding methods:

- Needs for miniaturization, higher quality and better efficiency as well as energy savings of many important information storing and processing devices.
- New channels are used, new types of errors start to be possible.
- New computation tools are developed - for example special types of paralelization,....

APPENDIX

LOCALLY DECODABLE CODES -I

Classical error-correcting codes allow one to encode an n-bit message w into an N-bit codeword $C(w)$, in such a way that w can still be recovered even if $C(w)$ gets corrupted in a number of bits.

The disadvantage of the classical error-correcting codes is that one needs to consider all, or at least most of, the (corrupted) codeword to recover anything about w.

On the other hand so-called **locally decodable codes** allow reconstruction of any arbitrary bit w_i, from looking only at k randomly chosen bits of $C(w)$, where k is as small as 3.

Locally decodable codes have a variety of applications in

LOCALLY DECODABLE CODES -II

Locally decodable codes have another remarkable property:

A message can be encoded in such a way that should a small enough fraction of its symbols die in the transit, we could, with high probability, to recover the original bit anywhere in the message we choose.

Moreover, this can be done by picking at random only three bits of the received message and combining them in a right way.

GROUPS

A **group** G is a set of elements and an operation, call it *, with the following properties:

- G is closed under *; that is if $a, b \in G$, so is $a * b$.
- The operation * is associative, hat is $a * (b * c) = (a * b) * c$, for any $a, b, c \in G$.
- G has an identity e element such that $e * a = a * e = a$ for any $a \in G$.
- Every element $a \in G$ has an inverse $a^{-1} \in G$, such that $a * a^{-1} = a^{-1} * a = e$.

A group G is called an **Abelian group** if the operation $*$ is commutative, that is $a * b = b * a$ for any $a, b \in G$.

Example Which of the following sets is an (Abelian) group:

- The set of real numbers with operation $*$ being: (a) addition; (b) multiplication.
- The set of matrices of degree n and operation: (a) addition; (b) multiplication.
- What happens if we consider only matrices with determinants not equal zero?

RINGS and FIELDS

A **ring** R is a set with two operations $+$ (addition) and $\cdot$ (multiplication) , having the following properties:

- R is closed under $+$ and $\cdot$.
- R is an Abelian group under $+$ (with a unity element for addition called **zero**).
- The associative law for multiplication holds.
- R has an identity element 1 for multiplication
- The distributive law holds: $a \cdot (b + c) = a \cdot b + a \cdot c$ for all $a, b, c \in R$.

A ring is called a **commutative ring** if multiplication is commutative.

A **field** F is a set with two operations $+$ (addition) and $\cdot$ (multiplication) , with the following properties:

- F is a commutative ring.
- Non-zero elements of F form an Abelian group under multiplication.

A non-zero element g is a **primitive element** of a field F if all non-zero elements of F are powers of g.

FINITE FIELDS

Finite fields are very well understood.

Theorem If p is a prime, then the integers mod p, $GF(p)$, constitute a field. Every finite field F contains a subfield that is $GF(p)$, up to relabeling, for some prime p and $p \cdot \alpha = 0$ for every $\alpha \in F$.

If a field F contains the prime field $GF(p)$, then p is called the **characteristic** of F.

Theorem (1) Every finite field F has p^m elements for some prime p and some m.
(2) For any prime p and any integer m there is a unique (up to isomorphism) field of p^m elements $GF(p^m)$.
(3) If $f(x)$ is an irreducible polynomial of degree m in $F_p[x]$, then the set of polynomials in $F_p[x]$ with additions and multiplications modulo $f(x)$ is a field with p^m elements.

FINITE FIELDS $GF(p^k), k > 1$

There are two important ways GF(4), the Galois field of four elements, is realized.
1. It is easy to verify that such a field is the set

$$\text{GF}(4) = \{0, 1, \omega, \omega^2\}$$

with operations $+$ and $\cdot$ satisfying laws

- $0 + x = x$ for all x;
- $x + x = 0$ for all x;
- $1 \cdot x = x$ for all x;
- $\omega + 1 = \omega^2$

2. Let $\mathbf{Z}_2[x]$ be the set of polynomials whose coefficients are integers mod 2. GF(4) is also $\mathbf{Z}_2[x] \pmod{x^2 + x + 1}$ therefore the set of polynomials

$$0, 1, x, x + 1$$

where addition and multiplication are $\pmod{x^2 + x + 1}$.

3. Let p be a prime and $\mathbf{Z}_p[x]$ be the set of polynomials with coefficients mod p. If $p(x)$ is a irreducible polynomial mod p of degree n, then $\mathbf{Z}_p[x] \pmod{p(x)}$ is a GF(p^n) with p^n elements.