An introduction to linear and cyclic codes

Daniel Augot, Emmanuela Orsini, Emanuele Betti

An introduction to linear and cyclic codes

Daniel Augot[1], Emanuele Betti[2], and Emmanuela Orsini[3]

[1] INRIA Paris-Rocquencourt Daniel.Augot@inria.fr
[2] Department of Mathematics, University of Florence betti@math.unifi.it
[3] Department of Mathematics, University of Milan orsini@posso.dm.unipi.it

Summary. Our purpose is to recall some basic aspects about linear and cyclic codes. We first briefly describe the role of error-correcting codes in communication. To do this we introduce, with examples, the concept of linear codes and their parameters, in particular the Hamming distance.

A fundamental subclass of linear codes is given by cyclic codes, that enjoy a very interesting algebraic structure. In fact, cyclic codes can be viewed as ideals in a residue classes ring of univariate polynomials. BCH codes are the most studied family of cyclic codes, for which some efficient decoding algorithms are known, as the method of Sugiyama.

1 An overview on error correcting codes

We give a brief description of a communication scheme, following the classical paper by Claude Shannon [29]. Suppose that an *information source* A wants to say something to a *destination* B. In our scheme the information is sent through a *channel*. If, for example, A and B are mobile phones, then the channel is the space where electromagnetic waves propagate. The real experience suggests to consider the case in which some interference (*noise*) is present in the channel where the information passes through.

The basic idea of coding theory consists of adding some kind of redundancy to the message m that A wants to send to B. Following Figure 1, A hands the message m to a device called a *transmitter* that uses a *coding procedure* to obtain a longer message m' that contains redundancy. The transmitter sends m' through the channel to another device called a *receiver*. Because of the noise in the channel, it may be that the message m'' obtained after the transmission is different from m'. If the occurred errors are not too many (in a sense that will be clear later), the receiver is able to recover the original message m, using a *decoding procedure*.

To be more precise, the coding procedure is an injective map from the space of the admissible messages to a larger space. The code is the image of

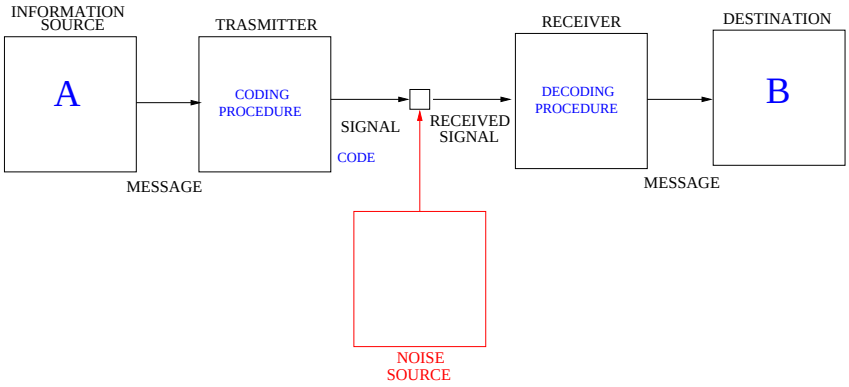

Fig. 1. A communication schema

this map. A common assumption is that this map is a linear function between vector spaces. In the next section we will describe some basic concepts about coding theory using this restriction. The material of this tutorial can be found in [2], [6], [22], [24], [26] and [33].

2 Linear codes

2.1 Basic definitions

Linear codes are widely studied because of their algebraic structure, which makes them easier to describe than non-linear codes.

Let $\mathbb{F}_q = GF(q)$ be the finite field with q elements and $(\mathbb{F}_q)^n$ be the linear space of all n-tuples over $\mathbb{F}_q$ (its elements are row vectors).

Definition 1. *Let $k, n \in \mathbb{N}$ such that $1 \leq k \leq n$. A* **linear code** *C is a k-dimensional vector subspace of $(\mathbb{F}_q)^n$. We say that C is a linear code over $\mathbb{F}_q$ with length n and dimension k. An element of C is called a* **word** *of C.*

From now on we shorten "linear code on $\mathbb{F}_q$ with length n and dimension k" to "$[n,k]_q$ code".

Denoting by "$\cdot$" the usual scalar product, given a vector subspace S of $(\mathbb{F}_q)^n$, we can consider the dual space $S^\perp$.

Definition 2. *If C is an $[n,k]_q$ code, its* **dual code** *$C^\perp$ is the set of vectors orthogonal to all words of C:*

$$C^\perp = \{c' \mid c' \cdot c = 0, \forall c \in C\}.$$

Thus $C^\perp$ is an $[n, n-k]_q$ code.

Definition 3. *If C is an $[n,k]_q$ code, then any matrix G whose rows form a basis for C as a k-dimensional vector space is called a* **generator matrix** *for C. If G has the form $G = \left[I_k \mid A\right]$, where I_k is the $k \times k$ identity matrix, G is called a generator matrix in* **standard form**.

Thanks to this algebraic description, linear codes allow very easy encoding. Given a generator matrix G, the encoding procedure of a message $m \in (\mathbb{F}_q)^k$ into the word $c \in (\mathbb{F}_q)^n$, is just the matrix multiplication $mG = c$. When the generator matrix is in standard form $[I_k \mid A]$, m is encoded in $mG = (m, mA)$. In this case the message m is formed by the first k components of the associated word. Such an encoding is called *systematic*.

We conclude this section with another simple characterization of linear codes.

Definition 4. *A* **parity-check matrix** *for an $[n,k]_q$ code C is a generator matrix $H \in \mathbb{F}^{(n-k)\times n}$ for $C^{\perp}$.*

It is easy to see that C may be expressed as the null space of a parity-check matrix H:

$$\forall x \in (\mathbb{F}_q)^n, \;\; Hx^T = 0 \Leftrightarrow x \in C.$$

2.2 Hamming distance

To motivate the next definitions we describe what could happen during a transmission process.

Example 1. We suppose that the space of messages is $(\mathbb{F}_2)^2$:

$$(0,0) = v_1, \qquad (0,1) = v_2, \qquad (1,0) = v_3, \qquad (1,1) = v_4.$$

Let C be the $[6,2]_2$ code generated by

$$G = \begin{pmatrix} 0\,0\,0\,1\,1\,1 \\ 1\,1\,1\,0\,0\,0 \end{pmatrix}.$$

Then:

$$C = \{(0,0,0,0,0,0), (1,1,1,1,1,1), (0,0,0,1,1,1), (1,1,1,0,0,0)\}.$$

To send $v_2 = (0,1)$ we transmit the word $v_2G = (0,0,0,1,1,1)$; typically, during the transmission the message gets distorted by noise and the receiver has to perform some operations to obtain the transmitted word. Let w be the received vector. Several different situations could come up:

1. $w = (0,0,0,1,1,1)$, then $w \in C$, so the receiver deduces correctly that no errors have occurred and no correction is needed. It concludes that the message was v_2.

2. $w = (0, 0, 0, 1, 0, 1) \notin C$, then the receiver concludes that some errors have occurred. In this case it may "correct" and "detect" the error as follows. It may suppose that the word transmitted was $(0, 0, 0, 1, 1, 1)$, since that is the word that differs in the least number of positions from the received word w.
3. $w = (0, 0, 0, 1, 0, 0) \notin C$. The receiver correctly reaches the conclusion that there were some errors during the transmission, but if it tries to correct as in the previous case, it concludes that the word "nearest" to w is $(0, 0, 0, 0, 0, 0)$. In this case it corrects in a wrong way.
4. $w = (0, 0, 0, 0, 0, 0) \in C$. The receiver deduces incorrectly that no errors have occurred.

From the previous example we understand that when the decoder gets a received vector which is not a word, it has to find the word in C which has been sent by the encoder, *i.e.*, among all words, it has to find the one which has the "highest probability" of being sent. To do this it needs a priori knowledge on the channel, more precisely it needs to know how the noise can modify the transmitted word.

Definition 5. *A* q**-ary symmetric channel** (**SC** *for short*) *is a channel with the following properties:*

a) *the component of a transmitted word* (*an element of* $\mathbb{F}_q$ *that here we name generally "symbol"*) *can be changed by the noise only to another element of* $\mathbb{F}_q$*;*

b) *the probability that a symbol becomes another one is the same for all pairs of symbols;*

c) *the probability that a symbol changes during the transmission does not depend on its position;*

d) *if the* i*-th component is changed, then this fact does not affect the probability of change for the* j*-th components, even if* j *is close to* i*.*

To these channel properties it is usually added a *source property*:

- *all words are equally likely to be transmitted.*

The q-ary SC is a model that rarely can describe real channels. For example the assumption *d*) is not reasonable in practice: if a symbol is corrupted during the transmission there is an high probability that some errors happened in the neighborhood. Despite this fact, the classical approach accepts the assumptions of the SC, since it permits a simpler construction of the theory. The ways of getting around the troubles generated by this "false" assumption in practice are different by case and these are not investigated here. From now we will assume that the channel is a SC, and that the probability that a

symbol changes to another is less than the probability that it is uncorrupted by noise.

In our assumptions, by example 1, it is quite evident that a simple criterion to construct "good" codes would be to try to separate as much as possible the words of code inside $(\mathbb{F}_q)^n$.

Definition 6. *The* **(Hamming) distance** $d_H(u,v)$ *between two vectors* $u, v \in (\mathbb{F}_q)^n$ *is the number of coordinates in which* u *and* v *differ.*

Definition 7. *The* **(Hamming) weight** *of a vector* $u \in (\mathbb{F}_q)^n$ *is the number* $\mathbf{w}(u)$ *of its nonzero coordinates, i.e.* $\mathbf{w}(u) = d_H(u,0)$.

Definition 8. *The* **distance of a code** C *is the smallest distance between distinct words*:

$$d_H(C) = \min\{d_H(c_i, c_j) \mid c_i, c_j \in C, c_i \neq c_j\}.$$

Remark 1. If C is a linear code, the distance $d_H(C)$ is the same as the minimum weight of nonzero words:

$$d_H(C) = \min\{\mathbf{w}(c) \mid c \in C,\ c \neq 0\}.$$

If we know the distance $d = d_H(C)$ of an $[n,k]_q$ code, then we can refer to the code as an $[n,k,d]_q$ code.

Definition 9. *Let* C *be an* $[n,k]_q$ *code and let* A_i *be the number of words of* C *of weight* i. *The sequence* $\{A_i\}_{i=1}^n$ *is called the* **weight distribution** *of* C.

Note that in a linear code $A_0 = 1$ and $\min_{i>0}\{i \mid A_i \neq 0\} = d_H(C)$.

The distance of a code C is important to determine the *error correction capability* of C (that is, the numbers of errors that the code can correct) and its *error detection capability* (that is, the numbers of errors that the code can detect). In fact, we can see the noise as a perturbation that moves a word into some other vector. If the distance between the words is great, there is a low probability that the noise can move a codeword near to another one. To be more precise, we have:

Theorem 1. *Let* C *an* $[n,k,d]_q$ *code, then*

a) C *has detection capability* $\ell = d - 1$
b) C *has correction capability* $t = \lfloor \frac{d-1}{2} \rfloor$.

From now on t denotes the correction capability of the code.

Example 2. The code in the Example 1 has distance $d = 3$. Its detection capability is $\ell = 2$ and its correction capability is $t = 1$.

The following proposition gives an upper bound on the distance of a code in terms of the length and the dimension.

Proposition 1 (Singleton Bound). *For an* $[n,k,d]_q$ *code*

$$d \leq n - k + 1.$$

A code achieving this bound is called *maximum distance separable* (*MDS* for short).

2.3 Decoding linear codes

In the previous section we have seen that the essence of decoding is to guess which word was sent when a vector y is received. This means that y will be decoded as one of the words which is most "likely" to have been sent.

Proposition 2. *If the transmission uses a q-ary SC and the probability that a symbol changes into another one is less then the probability that a symbol is uncorrupted by noise, the word sent with the highest probability is the word "nearest" (in the sense of Hamming distance) to the received vector. If no more then t (the error correction capability) errors have occurred, this word is unique.*

Proof. See [16].

In Example 1 we have informally described this process. We now formally describe the decoding procedure in the linear case. It should be noted that for the remainder C denotes an $[n, k]_q$ code.

Let $c, e, y \in (\mathbb{F}_q)^n$ be the transmitted word, the error, and the received vector, respectively. Then:

$$c + e = y.$$

Given y, our goal is to determine an e of minimal weight such that $y - e$ is in C. Of course, this vector might not be unique, since there may be more than one word nearest to y, but if the weight of e is less then t, then it is unique. By applying the parity-check matrix H to y, we get:

$$Hy^T = H(c + e)^T = He^T = s.$$

Definition 10. *The elements in $(\mathbb{F}_q)^{n-k}$, $s = Hy^T$, are called* **syndromes**. *We say that s is the syndrome corresponding to y.*

Note that the syndrome depends only on the occurred error e and not on the particular transmitted word.

Given a in $(\mathbb{F}_q)^n$, we denote the coset $\{a+c \mid c \in C\}$ by $a+C$. $(\mathbb{F}_q)^n$ can be partitioned into q^{n-k} cosets of size q^k. Two vectors $a, b \in (\mathbb{F}_q)^n$ belong to the same coset if and only if $a - b \in C$. The following fact is just a reformulation of our arguments.

Theorem 2. *Let C be an $[n, k, d]_q$ code. Two vectors $a, b \in (\mathbb{F}_q)^n$ are in the same coset if and only if they have the same syndrome.*

Definition 11. *Let C be an $[n, k, d]_q$ code. For any coset $a+C$ and any vector $v \in a + C$, we say that v is a* **coset leader** *if it is an element of minimum weight in the coset.*

Definition 12. *If s is a syndrome corresponding to an error of weight $\mathbf{w}(s) \leq t$, then we say that s is a* **correctable syndrome**.

Theorem 3 (Correctable syndrome). *If no more than t errors occurred (i.e. $\mathbf{w}(e) \leq t$) , then there exists only one error e corresponding to the correctable syndrome $s = He$ and e is the unique coset leader of $e + C$.*

We are ready to describe the decoding algorithm. Let y be a received vector. We want to find an error vector e of smallest weight such that $y - e \in C$. This is equivalent to finding a vector e of smallest weight in the coset containing y.
Decoding linear codes:

1. after receiving a vector $y \in (\mathbb{F}_q)^n$, compute the syndrome $s = Hy$;
2. find z, a coset leader of the corresponding coset;
3. the decoded word is $c = y - z$;
4. recover the message m from c (in case of systematic encoding m consists of first k components of c).

Remark 2 (Complexity of decoding linear codes). The procedure described below requires some preliminary operations to construct a matrix (named **standard array**) that contains the 2^n vectors of $(\mathbb{F}_q)^n$ ordered by coset. Then the complexity of the decoding procedure is exponential in terms of memory occupancy.

In [4] and [34] it is shown that the general decoding problem for linear codes and the general problem of finding the distance of a linear code are both NP-complete. This suggests that no algorithm exists that decodes linear codes in a polynomial time.

3 Some bounds on codes

We have seen that the distance d is an important parameter for a code. A fundamental problem in coding theory is, given the length and the number of codewords (dimension if the code is linear), to determine a code with largest distance, or equivalently, to find the largest code of a given length and distance.

The following definition is useful to state some bounds on codes more clearly.

Definition 13. *Let n, d be positive integers with $d \leq n$. Then the number $A_q(n, d)$ denotes the maximum number of codewords in a code over $\mathbb{F}_q$ of length n and distance d. This maximum, when restricted to linear code, is denoted by $B_q(n, d)$.*

Clearly it can be $B_q(n,d) < A_q(n,d)$. Then given n and d, if we look the largest possible code, we have sometimes to use nonlinear codes in practice.

We recall some classical bounds that restrict the existence of codes with given parameters. For any $x \in (\mathbb{F}_q)^n$ and any positive number r, let $B_r(x)$ be the sphere of radius r centered in x, with respect to the Hamming distance. Note that the size of $B_r(x)$ is independent of x and depends only on r, q and n. Let $V_q(n,r)$ denote the number of elements in $B_r(x)$ for any $x \in (\mathbb{F}_q)^n$. For any $y \in B_r(x)$, there are $(q-1)$ possible values for each of the r positions in which x and y differ. So we see that

$$V_q(n,r) = \sum_{i=0}^{r} \binom{n}{i} (q-1)^i.$$

From the fact that the spheres of radius $t = \lfloor \frac{d-1}{2} \rfloor$ about codewords are pairwise disjoint, the *sphere packing bound* (or *Hamming bound*) immediately follows:

$$A_q(n,d) \leq \frac{q^n}{V_q(n,t)}.$$

We rewrite the *Singleton bound* (see Proposition 1)

$$A_q(n,d) \leq q^{n+1-d}.$$

Abbreviating $\gamma = \frac{q-1}{q}$ and assuming $\gamma n < d$ there holds the *Plotkin bound*, which says that

$$A_q(n,d) \leq \frac{d}{d-\gamma n}.$$

The *Elias bound*, as an extensive refinement of the Plotkin bound, states that for every $t \in \mathbb{R}$ with $t < \gamma n$ and $t^2 - 2t\gamma n + d\gamma n > 0$ there holds

$$A_q(n,d) \leq \frac{\gamma n d}{t^2 - 2t\gamma n + d\gamma n} \cdot \frac{q^n}{V_q(n,t)}.$$

We conclude with a lower bound, the *Gilbert–Varshamov bound*

$$A_q(n,d) \geq B_q(n,d) \geq \frac{q^n}{V_q(n,d-1)}.$$

4 Cyclic codes

4.1 An algebraic correspondence

Definition 14. *An $[n,k,d]_q$ linear code C is* **cyclic** *if the cyclic shift of a word is also a word, i.e.*

$$(c_0, \ldots, c_{n-1}) \in C \implies (c_{n-1}, c_0, \ldots, c_{n-2}) \in C.$$

To describe algebraic properties of cyclic codes, we need to introduce a new structure. We consider the univariate polynomial ring $\mathbb{F}_q[x]$ and the ideal $I = \langle x^n - 1 \rangle$. We denote by R the ring $\mathbb{F}_q[x]/I$. We construct a bijective correspondence between the vectors of $(\mathbb{F}_q)^n$ and residue classes of polynomials in R:

$$\mathbf{v} = (v_0, \ldots, v_{n-1}) \longleftrightarrow v_0 + v_1 x + \cdots + v_{n-1} x^{n-1}.$$

We can view linear codes as subsets of the ring R, thanks to the correspondence below. The following theorem points out the algebraic structure of cyclic codes.

Theorem 4. *Let C be an $[n, k, d]_q$ code, then C is cyclic if and only if C is an ideal of R.*

Proof. Multiplying by x modulo $x^n - 1$ corresponds to a cyclic shift:

$$(c_0, c_1, \ldots, c_{n-1}) \to (c_{n-1}, c_0, \ldots, c_{n-2})$$

$$x(c_0 + c_1 x + \cdots + c_{n-1} x^{n-1}) = c_{n-1} + c_0 x + \cdots + c_{n-2} x^{n-2}.$$

Since R is a principal ideal ring, if C is not trivial there exists a unique monic polynomial g that generates C. We call g the *generator polynomial* of C. Note that g divides $x^n - 1$ in $\mathbb{F}_q[x]$. If the dimension of the code C is k, the generator polynomial has degree $n - k$.

A generator matrix can easily be given by using the coefficients of the generator polynomial $g = \sum_{i=0}^{n-k} g_i x^i$:

$$G = \begin{pmatrix} g \\ xg \\ \vdots \\ x^k g \end{pmatrix} = \begin{pmatrix} g_0 & g_1 & \cdots & g_{n-k} & 0 & \cdots & 0 \\ 0 & g_0 & \cdots & g_{n-k-1} & g_{n-k} & 0 & \cdots \\ \vdots & & \ddots & \vdots & & \ddots & \vdots \\ 0 & \cdots & 0 & g_0 & g_1 & \cdots & g_{n-k} \end{pmatrix}.$$

Moreover, a polynomial f in R belongs to the code C if and only if there exists q in R such that $qg = f$.

Since the generator polynomial is a divisor of $x^n - 1$ and is unique, the *parity-check* polynomial of C is well defined as the polynomial $h(x)$ in R such that $h(x) = (x^n - 1)/g(x)$. The parity-check polynomial provides a simple way to check if an $f(x)$ in R belongs to C, since

$$f(x) \in C \Leftrightarrow f(x) = q(x)g(x) \Leftrightarrow f(x)h(x) = q(x)(g(x)h(x)) = 0 \text{ in } R.$$

Proposition 3. *Let $h(x), g(x)$ be, respectively, the parity-check and the generator polynomial of the cyclic code C. The dual code $C^\perp$ is cyclic with generator polynomial*

$$g^\perp(x) = x^{\deg(h)} h(x^{-1}).$$

Proof. The generator matrix obtained by $g^{\perp}(x)$ has the form:

$$H = \begin{pmatrix} & & h_k & \dots & h_1 & h_0 \\ & h_k & \dots & h_1 & h_0 & \\ & & \vdots & & & \\ h_k & \dots & h_1 & h_0 & & \end{pmatrix}.$$

Given c in R, the i-th component of $H \cdot c^T$ is $x^i h(x)c(x)$, which vanishes if and only if $c \in C$.

4.2 Encoding and decoding with cyclic codes

The properties of cyclic codes suggest a very simple method to encode a message. Let C be an $[n, k, d]_q$ cyclic code with generator polynomial g, then C is capable of encoding q–ary messages of length k and requires $n - k$ redundancy symbols.

Let $m = (m_0, \dots, m_{k-1})$ be a message to encode, we consider its polynomial representation $m(x)$ in R. To obtain an associated word it is sufficient to multiply $m(x)$ by the generator polynomial $g(x)$:

$$c(x) = m(x)g(x) \in C.$$

Even if this way to encode is the simpler, another procedure is used to obtain a systematic encoding, which again exploits some properties of the polynomial ring.

Given the message $m(x)$, multiply it by x^{n-k} and divide the result by g, obtaining:

$$m(x)x^{n-k} = q(x)g(x) + r(x)$$

where $\deg(r(x)) < \deg(g(x)) = n - k$. So the remainder can be thought of as an $(n-k)$-vector. Joining the k-vector m with the $(n-k)$-vector r we obtain an n-vector c, which is the encoded word, i.e.:

$$c(x) = m(x)x^{n-k} + r(x).$$

This way, in the absence of errors the decoding is immediate: the message is formed by the last k components of the received word.

On the other hand, the receiver does not know if no errors have occurred during transmission, but it is sufficient to check if the remainder of the division of the received polynomial by g is equal to zero to state that it is most likely that no errors have occurred.

It is not hard to prove that if an error e occurred during the transmission, the remainder of the division by g in the procedure below gives exactly the syndrome associated to e, and then we can find e in the same way as described for linear codes.

Other decoding procedures exist for particular cyclic codes, such as the BCH codes, which work faster than the procedure above. (See Section 7).

4.3 Zeros of cyclic codes

Cyclic codes of length n over $\mathbb{F}_q$ are generated by divisors of $x^n - 1$. Let

$$x^n - 1 = \prod_{j=1}^{r} f_j, \qquad f_j \text{ irreducible over } \mathbb{F}_q.$$

Then to any cyclic code of length n over $\mathbb{F}_q$ there corresponds a subset of $\{f_j\}_{j=1}^r$. A very interesting case [4] is when $\mathrm{GCD}(n, q) = 1$. Let $\mathbb{F} = \mathbb{F}_{q^m}$ be the splitting field of $x^n - 1$ over $\mathbb{F}_q$ and let α be a primitive n-th root of unity over $\mathbb{F}_q$. We have:

$$x^n - 1 = \prod_{i=0}^{n-1} (x - \alpha^i).$$

In this case the generator polynomial of C has powers of α as roots. We remember that, given $g \in \mathbb{F}_q[x]$, if $g(\alpha^i) = 0$ then $g(\alpha^{qi}) = 0$.

Definition 15. *Let C be an $[n, k, d]_q$ cyclic code with generator polynomial g_C, with GCD$(n, q) = 1$. The set:*

$$S_{C,\alpha} = S_C = \{i_1, \ldots, i_{n-k} \mid g_C(\alpha^{i_j}) = 0,\ j = 1, \ldots, n-k\}$$

is called the **complete defining set of C.**

We can collect the integers modulo n into *q-cyclotomic classes C_i*:

$$\{0, \ldots, n-1\} = \bigcup C_i, \quad C_i = \{i, qi, \ldots, q^r i\},$$

where r is the smallest positive integer such that $i \equiv iq^r (\mod n)$. So the complete defining set of a cyclic code is collection of q-cyclotomic classes.

From now on we fix a primitive n-th root of unity α and we write $S_{C,\alpha} = S_C$. A cyclic code is defined by its complete defining set, since

$$C = \{c \in R \mid c(\alpha^i) = 0,\ i \in S_C\} \iff g_C = \prod_{i \in S_C} (x - \alpha^i).$$

By this fact it follows that

$$H = \begin{pmatrix} 1 & \alpha^{i_1} & \alpha^{2i_1} & \cdots & \alpha^{(n-1)i_1} \\ 1 & \alpha^{i_2} & \alpha^{2i_2} & \cdots & \alpha^{(n-1)i_2} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & \alpha^{i_{n-k}} & \alpha^{2i_{n-k}} & \cdots & \alpha^{(n-1)i_{n-k}} \end{pmatrix}$$

[4] In [10] is shown that if there exists a family of "good" codes $\{C_m\}_m$ over $\mathbb{F}_q$ of lengths m with $\mathrm{GCD}(m, q) \neq 1$, there exists a family $\{C'_n\}_n$ with $\mathrm{GCD}(n, q) = 1$ with the same properties.

is a parity-check (defined over $\mathbb{F}_{q^m}$) matrix for C, since

$$Hc^T = \begin{pmatrix} c(\alpha^{i_1}) \\ c(\alpha^{i_2}) \\ \vdots \\ c(\alpha^{i_{n-k}}) \end{pmatrix} = \underline{0} \;\Leftrightarrow\; c \in C.$$

Remark 3. H maybe defined over $\mathbb{F}_{q^m}$, but C is its nullspace over $\mathbb{F}_q$.

Remark 4. We note that, as S_C is partitioned into cyclotomic classes, there are some subsets S'_C of S_C any of them sufficient to specify the code unambiguosly and we call any such S'_C a **defining set**.

5 Some examples of cyclic codes

5.1 Hamming and simplex codes

Definition 16. *A code which attains the Hamming bound (see Section 3) is called a* **perfect** *code.*

In other words, a code is said to be perfect if for every possible vector v in $(\mathbb{F}_q)^n$ there is a unique word $c \in C$ such that $d_H(v, c) \leq t$.

Let C be an $[n, n-r, d]_q$ code with parity-check matrix $H \in (\mathbb{F}_q)^{r\times n}$. We denote by $\{H_i\}_{i=1}^n$ the set of columns of H. We observe that if two columns H_i, H_j belongs to the same line in $(\mathbb{F}_q)^r$ (i.e. $H_j = \lambda H_i$), then the vector

$$c = (0, \ldots, 0, \underset{i}{-\lambda}, 0, \ldots, 0, \underset{j}{1}, 0, \ldots, 0)$$

belongs to C, since $Hc^T = 0$. Then $d(C) \leq 2$. On other hand, if we construct a parity-check matrix H such that the columns H_i belong to different lines, the corresponding linear code has distance at least 3.

Definition 17. *An* **Hamming Code** *is a linear code for which the set of columns of $H \in (\mathbb{F}_q)^{n\times r}$ contains exactly one element different from zero of every line in $(\mathbb{F}_q)^r$.*

By the definition above, given two columns H_i, H_j of H, there exists a third column H_k of H, and $\lambda \in \mathbb{F}_q$ such that $H_k = \lambda(H_i + H_j)$. This fact implies that

$$c = (0, \ldots, 0, \underset{i}{-\lambda}, 0, \ldots, 0, \underset{j}{-\lambda}, 0, \ldots, 0, \underset{k}{1}, 0, \ldots, 0)$$

is a word, and hence the minimum distance of a Hamming code is 3. In the vector space $(\mathbb{F}_q)^r$ there are $n = \frac{q^r-1}{q-1}$ distinct lines, each with $q-1$ elements different from zero. Hence:

Proposition 4. *An $[n, k, d]_q$ code is a Hamming code, if and only if $n = \frac{q^r-1}{q-1}$, $k = n - r$, $d = 3$, for some $r \in \mathbb{N}^*$.*

On the other hand, a direct computation shows that:

Proposition 5. *The Hamming codes are perfect codes.*

Example 3. Let C be the $[7, 4, 3]_2$ code with parity-check matrix:

$$H = \begin{pmatrix} 1\,0\,1\,0\,1\,0\,1 \\ 0\,1\,1\,0\,0\,1\,1 \\ 0\,0\,0\,1\,1\,1\,1 \end{pmatrix}.$$

Then C is an $[7, 4, 3]$ Hamming code. Note that the columns of H are exactly the non-zero vectors of $\mathbb{F}_2^3$.

The following theorem states that Hamming codes are cyclic.

Theorem 5. *Let $n = (q^r - 1)/(q - 1)$. If $GCD(n, q - 1) = 1$, then the cyclic code over $\mathbb{F}_q$ of length n with defining set $\{1\}$ is a $[n, n - r, 3]$ Hamming code.*

Proof. By Proposition 4 it is sufficient to show that the distance of C is equal to 3. The Hamming bound applied to C ensures that the distance cannot be greater than 3; we show that it can not be 2 (it is obvious that it is not one). Let α be a primitive n-th root of unity over $\mathbb{F}_q$ such that $c(\alpha) = 0$ for c in C. If c is a word of weight 2 with nonzero coefficients c_i and c_j $(i < j)$, then $c_i\alpha^i + c_j\alpha^j = 0$. Then $\alpha^{j-i} = -c_i/c_j$. Since $-c_i/c_j \in \mathbb{F}_q^*$, $\alpha^{(j-i)(q-1)} = 1$. Now $\mathrm{GCD}(n, q - 1) = 1$ implies that $\alpha^{j-i} = 1$, but this is a contradiction since $0 < j - i < n$ and the order of α is n.

Example 4. The Hamming code of Example 3 can be viewed as the $[7, 4, 3]_2$ cyclic code with generator polynomial $g = x^3 + x + 1$.

We have seen that the dual code of a cyclic code is cyclic itself. This means in particular that the dual of a Hamming code is cyclic.

Definition 18. *The dual of a Hamming code is called a* **simplex code**.

The simplex code has the following property:

Proposition 6. *A simplex code is a $[(q^r - 1)/(q - 1), r, q^{r-1}]$ constant weight code over $\mathbb{F}_q$.*

5.2 Quadratic residue codes

Let n be an odd prime. We denote by $\mathcal{Q}_n \subset \{1, \ldots, n - 1\}$ the set of *quadratic residues modulo n*, *i.e.*:

$$\mathcal{Q}_n = \{k \mid k \equiv x^2 \mod (n) \text{ for some } x \in \mathbb{Z}\}.$$

If q is a quadratic residue modulo n, it is easy to see that $\mathcal{Q}_n$ is a collection of q-cyclotomic classes with cardinality $(n-1)/2$. Then we can give the following definition.

Definition 19. *Let n be a positive integer relatively prime to q and let α be a primitive n-th root of unity. Suppose that n is an odd prime and q is a quadratic residue modulo n. The $[n, (n-1)/2+1]_q$ cyclic code with complete defining set $\mathcal{Q}_n$ is called* **quadratic residue code**.

Example 5. The $[23, 12, 7]_2$ quadratic residue code is the **perfect binary Golay code**.

6 BCH codes

Theorem 6 (BCH bound). *Let C be an $[n, k, d]_q$ cyclic code with defining set $S_C = \{i_1, \ldots, i_{n-k}\}$ and let $(n, q) = 1$. Suppose there are $\delta - 1$ consecutive numbers in S_C, say $\{m_0, m_0+1, \ldots, m_0+\delta-2\} \subset S_C$. Then*

$$d \geq \delta.$$

Definition 20. *Let $S = (m_0, m_0+1, \ldots, m_0+\delta-2)$ be such that*

$$0 \leq m_0 \leq \cdots \leq m_0 + \delta - 2 \leq n-1$$

If C is the $[n, k, d]_q$ cyclic code with defining set S, we say that C is a **BCH code** *of* **designed distance** *δ. The BCH code is called* **narrow sense** *if $m_0 = 1$ and it is called* **primitive** *if $n = q^m - 1$.*

Example 6. We consider the polynomial $x^7 - 1$ over $\mathbb{F}_2$:

$$\begin{array}{ccccc} & & f_0 & & f_1 & & f_3 \\ & & \shortparallel & & \shortparallel & & \shortparallel \\ x^7 - 1 & = & (x+1) & & (x^3+x^2+1) & & (x^3+x+1) \end{array}$$

Let C be the cyclic code generated by $g = f_0 \cdot f_1$. Then $S_C = \{0, 1, 2, 4\}$ with respect to a primitive n-th root of unity α s.t. $f_1(\alpha) = 0$. C is a $[7, 3, d]_2$ code with $S_C = \{0, 1, 2, 4\}$ and so it is a BCH code of designed distance $\delta = 4$. The BCH bound ensures that the minimum distance is at least 4. On the other hand, the generator polynomial

$$g(x) = x^4 + x^2 + x + 1$$

has weight 4 and we finally can state that $d = 4$.

6.1 On the optimality of BCH codes

Definition 21. *Given n and d two integers, a code is said to be* optimal *if it has maximal size in the class of codes with length n and distance d.*

Theorem 7. *Narrow sense primitive binary BCH codes of fixed minimum distance are optimal when the length is large, but the relative distance*

$$d/n \to 0.$$

In other words, consider such an $[n, k, d]_2$ BCH code, with $t = \lfloor \frac{d-1}{2} \rfloor$, then $k \geq n - mt$. Then there does not exist a $t+1$ correcting code with the same length and dimension.

Proof. Let t be fixed, and let $n = 2^m - 1$ go to infinity. Then

$$\begin{aligned} \mathrm{V}_2(n, t+1) &= \sum_{i \leq t+1} \binom{n}{i} > \binom{n}{t+1} \\ &= \frac{n!}{(t+1)!(n-t-1)!} \\ &= O\left(\frac{1}{(t+1)!} \cdot n^{t+1}\right) \\ &\sim \frac{1}{(t+1)!} 2^{m(t+1)} \gg 2^{mt} > 2^{n-k}. \end{aligned}$$

This means that the Hamming bound is exceeded for the parameters n, k and $t+1$, which implies that a $t+1$ error correcting code does not exist.

A precise evaluation of the length n such that an $[n, k, n - mt]$ BCH code is optimal is given in [3], p. 299.

We now define a subclass of BCH codes that are always optimal.

Definition 22. *A* **Reed Solomon** *code over $\mathbb{F}_q$ is a BCH code with length $n = q - 1$.*

Note that if $n = q - 1$ then $x^n - 1$ splits into linear factors. If the designed distance is d, then the generator polynomial of a RS code has the form $g(x) = (x - \alpha^{i_0})(x - \alpha^{i_0+1}) \cdots (x - \alpha^{i_0+d-1})$ and $k = n - d + 1$. It follows that RS codes are MDS codes.

7 Decoding BCH codes

There are several algorithms for decoding BCH codes. In this section we briefly discuss the method, first developed in 1975 by Sugiyama et al. ([32]), that uses the extended Euclidean algorithm to solve the key equation. Note that the Berlekamp–Massey algorithm ([2],[23]) is preferable in practice.

Let C be a BCH code of length n over $\mathbb{F}_q$, with designed distance $\delta = 2t+1$ (where t is the error correction capability of the code), and let α be a primitive n-th root of unity in $\mathbb{F}_{q^m}$. We consider a word $c(x) = c_0 + \cdots + c_{n-1}x^{n-1}$ and we assume that the received word is $v(x) = v_0 + \cdots + v_{n-1}x^{n-1}$. Then the error vector can be represented by the *error polynomial*

$$e(x) = v(x) - c(x) = e_0 + e_1 x + \cdots + e_{n-1} x^{n-1}.$$

If the weight of e is $\mu \leq t$, let

$$L = \{l \mid e_l \neq 0, 0 \leq l \leq n-1\}$$

be the set of the *error positions*, and $\{\alpha^l \mid l \in L\}$ the set of the *error locators*. Then the *classical error locator polynomial* is defined by

$$\sigma(x) = \prod_{l \in L} (1 - x\alpha^l),$$

i.e. the univariate polynomial which has as zeros the reciprocal of the error locations. The error locations can also be obtained by the *plain error locator polynomial*, that is

$$L_e(x) = \prod_{l \in L} (x - \alpha^l).$$

The *error evaluator polynomial* is defined by

$$\omega(x) = \sum_{l \in L} e_l \alpha^l \prod_{i \in L \setminus \{l\}} (1 - x\alpha^i).$$

The importance of finding the two polynomials $\sigma(x)$ and $\omega(x)$ is clear to correct the errors: an error is in the positions l if and only if $\sigma(\alpha^{-l}) = 0$ and in this case the value of the error is:

$$e_l = -\alpha^l \frac{\omega(\alpha^{-l})}{\sigma'(\alpha^{-l})}, \tag{1}$$

in fact, since the derivative $\sigma'(x) = \sum_{l \in L} -\alpha^l \prod_{i \neq l} (1 - x\alpha^i)$, so $\sigma'(\alpha^{-l}) = -\alpha^l \prod_{i \neq l} (1 - \alpha^{i-l})$ and $\sigma'(\alpha^{-l}) \neq 0$. The goal of decoding can be reduced to determine the error locator polynomial and apply an exahustive search of the roots to obtain the error positions. We will need the following lemma later on.

Lemma 1. *The polynomials $\sigma(x)$ and $\omega(x)$ are relatively prime.*

Proof. It is an obvious consequence of the fact that no zero of $\sigma(x)$ is a zero of $\omega(x)$.

We are now ready to describe the decoding algorithm.

The first step: the key equation

At the first step we calculate the syndrome of the received vector $v(x)$:

$$Hv^T = \begin{pmatrix} 1 & \alpha & \alpha^2 & \cdots & \alpha^{n-1} \\ 1 & \alpha^2 & \alpha^4 & \cdots & \alpha^{2(n-1)} \\ \vdots & \vdots & \vdots & \cdots & \vdots \\ 1 & \alpha^{\delta-1} & \alpha^{2(\delta-1)} & \cdots & \alpha^{(\delta-1)(n-1)} \end{pmatrix} \begin{pmatrix} e_0 \\ e_1 \\ \vdots \\ e_{n-1} \end{pmatrix} = \begin{pmatrix} e(\alpha) \\ e(\alpha^2) \\ \vdots \\ e(\alpha^{\delta-1}) \end{pmatrix} = \begin{pmatrix} S_1 \\ S_2 \\ \vdots \\ S_{2t} \end{pmatrix}$$

We define the *syndrome polynomial*:

$$S(x) = S_1 + S_2 x + \cdots + S_{2t} x^{2t-1},$$

where $S_i = e(\alpha^i) = \sum_{l \in L} e_l \alpha^{il}$, $i = 1, \ldots, 2t$. The following theorem establishes a relation among $\sigma(x), \omega(x)$ and $S(x)$.

Theorem 8 (The key equation). *The polynomials $\sigma(x)$ and $\omega(x)$ satisfy:*

$$\sigma(x)S(x) \equiv \omega(x) \pmod{x^{2t}} \qquad (\textbf{key equation})$$

If there exist two polynomials $\sigma_1(x)$, $\omega_1(x)$, such that $deg(\omega_1(x)) < deg(\sigma_1(x)) \leq t$ and that satisfy the key equation, then there is a polynomial $\lambda(x)$ such that $\sigma_1(x) = \lambda(x)\sigma(x)$ and $\omega_1(x) = \lambda(x)\omega(x)$.

Proof. Interchanging summations and the sum formula for a geometric series, we get

$$\begin{aligned} S(x) &= \sum_{j=1}^{2t} e(\alpha^j) x^{j-1} = \sum_{j=1}^{2t} \sum_{l \in L} e_l \alpha^{jl} x^{j-1} \\ &= \sum_{l \in L} e_l \alpha^l \sum_{j=1}^{2t} (\alpha^l x)^{j-1} = \sum_{l \in L} e_l \alpha^l \frac{1 - (\alpha^l x)^{2t}}{1 - \alpha^l x}. \end{aligned}$$

Thus

$$\sigma(x)S(x) = \prod_{i \in L} (1 - \alpha^i x) S(x) = \sum_{l \in L} e_l \alpha^l (1 - (\alpha^l x)^{2t}) \prod_{i \neq l \in L} (1 - \alpha^i x).$$

and then

$$\sigma(x)S(x) \equiv \sum_{l \in L} e_l \alpha^l \prod_{i \neq l \in L} (1 - \alpha^i x) \equiv \omega(x) \pmod{x^{2t}}.$$

Suppose we have another pair $(\sigma_1(x), \omega_1(x))$ such that

$$\sigma_1(x)S(x) \equiv \omega_1(x) \pmod{x^{2t}}$$

and $\deg(\omega_1(x)) < \deg(\sigma_1(x)) \leq t$. Then

$$\sigma(x)\omega_1(x) \equiv \sigma_1(x)\omega(x) \pmod{x^{2t}}$$

and the degrees of $\sigma(x)\omega_1(x)$ and $\sigma_1(x)\omega(x)$ are strictly smaller than $2t$. Since $GCD(\sigma(x), \omega(x)) = 1$ by Lemma 1, there exists a polynomial $\lambda(x)$ *s.t.* $\sigma_1(x) = \lambda(x)\sigma(x)$ and $\omega_1(x) = \lambda(x)\omega(x)$.

The second step: the extended Euclidean algorithm

Once we have the syndrome polynomial $S(x)$, the second step of the decoding algorithm consists of finding $\sigma(x)$ and $\omega(x)$, using the key equation.

Theorem 9 (Bezout's Identity). *Let $\mathcal{K}$ be a field and $f(x), g(x) \in \mathcal{K}[x]$. Let us denote $d(x) = \gcd(f(x), g(x))$. Then there are $u(x), v(x) \in \mathcal{K}[x] \setminus \{0\}$, such that:*

$$f(x)u(x) + g(x)v(x) = d(x).$$

It is well known that is possible to find the greatest common divisor $d(x)$ and the polynomials $u(x)$ and $v(x)$ in Bezout's identity using the Extended Euclidean Algorithm (EEA). Suppose that $\deg(f(x)) > \deg(g(x))$, then let:

$$\begin{array}{lll} u_{-1} = 1 & v_{-1} = 0 & d_{-1} = f(x) \\ u_0 = 0 & v_0 = 1 & d_0 = g(x) \end{array}$$

The first step of the Euclidean algorithm is:

$$d_1(x) = d_{-1}(x) - q_1(x)d_0(x) = f(x) - q_1(x)g(x),$$

so that

$$u_1(x) = 1, v_1(x) = -q_1(x) \qquad \text{and}$$
$$\deg(d_1) < \deg(d_0) \quad \text{and} \quad \deg(v_1) < \deg(d_{-1}) - \deg(d_0).$$

From the j-th step, we get:

$$\begin{aligned} d_j(x) &= d_{j-2}(x) - q_j(x)d_{j-1}(x) \\ &= u_{j-2}(x)f(x) + v_{j-2}(x)g(x) - q_j(x)[u_{j-1}(x)f(x) + v_{j-1}(x)g(x)] \\ &= [-q_j(x)u_{j-1}(x) + u_{j-2}(x)]f(x) + [-q_j(x)v_{j-1}(x)f(x) + v_{j-2}(x)]g(x). \end{aligned}$$

This means:

$$u_j(x) = -q_j(x)u_{j-1}(x) + u_{j-2}(x) \quad \text{and} \quad v_j(x) = -q_j(x)v_{j-1}(x) + v_{j-2}(x)$$

with $\deg(d_j) \leq \deg(d_{j-1})$, $\deg(u_j) = \sum_{i=2}^{j} \deg(q_i)$, $\deg(v_j) = \sum_{i=2}^{j} \deg(q_i)$ and $\deg(v_j) = \deg(f) - \deg(d_{j-1})$. The algorithm proceeds by dividing the previous remainder by the current remainder until this becomes zero.

STEP 1	$d_{-1}(x) = q_1(x)d_0(x) + d_1(x),$	$\deg(d_1) < \deg(d_0)$
STEP 2	$d_0(x) = q_2(x)d_1(x) + d_2(x),$	$\deg(d_2) < \deg(d_1)$
$\vdots$	$\vdots$	
STEP j	$d_{j-2}(x) = q_j(x)d_{j-1}(x) + d_j(x),$	$\deg(d_j) < \deg(d_{j-1})$
$\vdots$	$\vdots$	
STEP k	$d_{k-1}(x) = q_{k+1}(x)d_k(x)$	

We conclude that the $\mathrm{GCD}(f(x), g(x)) = \mathrm{GCD}(d_{-1}(x), d_0(x)) = d_k(x)$.

We would like to be able to find $\omega(x)$ and $\sigma(x)$ using the Euclidean algorithm. First we observe that $\deg(\sigma(x)) \leq t$ and $\deg(\omega(x)) \leq t-1$. For this reason we apply the EEA to the known polynomials $f(x) = x^{2t}$ and $g(x) = S(x)$, until we find a $d_{k-1}(x)$ such that:

$$\deg(d_{k-1}(x)) \geq t \quad \text{and} \quad \deg(d_k(x)) \leq t-1.$$

In this way we obtain a polynomial $d_k(x)$ such that:

$$d_k(x) = x^{2t} u_k(x) + S(x) v_k(x), \tag{2}$$

with $deg(v_k(x)) = deg(x^{2t}) - deg(d_{k-1}(x)) \leq 2t - t = t$.

Theorem 10. *Let $d_k(x)$ and $v_k(x)$ as in (2). Then the polynomials $v_k(x)$ and $d_k(x)$ are scalar multiplies of $\sigma(x)$ and $\omega(x)$, respectively, i.e:*

$$\sigma(x) = \lambda v_k(x) \quad \omega(x) = \lambda d_k(x),$$

for some scalar $\lambda \in \mathbb{F}_q$.

We can determine λ by $\sigma(0) = 1$, *i.e.* $\lambda = v_k(0)^{-1}$. So we have:

$$\sigma(x) = \frac{v_k(x)}{v_k(0)} \quad \omega(x) = \frac{d_k(x)}{v_k(0)}.$$

The third step: determining the error values

In the last step we have to calculate the error values. In the binary case it is immediate. Otherwise we can use the relations

$$e_l = -\alpha^l \frac{\omega(\alpha^{-l})}{\sigma'(\alpha^{-l})}, \qquad l = 1, \ldots, \mu.$$

8 On the asymptotic properties of cyclic codes

There is a longstanding question which is to know whether the class of cyclic codes is *asymptotically good*. Let us recall that a sequence of linear binary $[n_i, k_i, d_i]_2$ codes C_i is asymptotically good if

$$\liminf \frac{k_i}{n_i} > 0, \quad \text{and} \quad \liminf \frac{d_i}{n_i} > 0.$$

The first explicit construction of an asymptotically good sequence of codes is due to Justesen [17], but the codes are not cyclic. Although it is known that the class of BCH codes is not asymptotically good [8, 20], (see [22] for

a proof), we do not know if there is a family of asymptotically good cyclic codes. Still on the negative side, Catagnoli [9] has shown that, if the length n_i goes to infinity while having a fixed set of prime factors, then there is no asymptotically good family of codes C_i of length n_i. Other negative results are in [5]. Known partial positive results are due to Kasami [18], for *quasi-cyclic codes*[5]. Bazzi-Mitter [1] have shown that there exists an asymptotically good family of linear codes which are very close to cyclic codes. Also Willems and Martínez-Pérez [21] have shown that there exists an asymptotically good family of cyclic codes, provided there exists an asymptotically good family of linear codes C_i with special properties on their lengths n_i. So, although some progress has been achieved, the question is still open.

References

1. L. M. J. Bazzi and S. K. Mitter, "Some randomized code constructions from group actions". *IEEE Trans. on Inf. Th.*, vol 52, p. 3210–3219, 2006.
2. E. R. Berlekamp, *Algebraic Coding Theory*, New York McGraw-Hill, 1968.
3. E. R. Berlekamp, *Algebraic Coding Theory (Revised Edition)*, Aegean Park Press, 1984.
4. E. R. Berlekamp, R. J. McEliece and H. C. A. Van Tilborg, "On the inherent intractability of certain coding problems", *IEEE Trans.Inf. Theory*, vol. 24, p. 384-386, 1978.
5. S. D. Berman, "Semisimple cyclic and abelian codes. II", *Cybernetics* 3, no. 3, p. 17-23, 1967.
6. R.E. Blahut, *Theory and Practice of error Control Codes*, Addison-Wesley Publishing Company, 1983.
7. R. C. Bose and D. K. Ray-Chaudhuri, "On a class of error correcting binary group codes", *Inform. Control*, vol. 3, p. 68-79, 1960.
8. P. Camion, "A proof of some properties of Reed-Muller codes by means of the normal basis theorem", In R. C. Bose and T. A. Dowling, editors, *Combinatorial mathematics and its applications*, University of North Carolina at Chapel Hill.
9. G. Castagnoli. "On the asymptotic badness of cyclic codes with block-lengths composed from a fixed set of prime factors", *Lectures Notes in Computer Science*, n. 357, p. 164-168, Berlin / Heidelberg, 1989. Springer.
10. G. Castagnoli, J. L. Massey, P. A. Schoeller, and N. von Seeman, "On repeated-root cyclic codes", *IEEE Trans. on Inf. Theory*, vol. 37, p. 337-342, 1991.
11. R. T. Chien, "Cyclic Decoding Procedure for the Bose-Chaudhuri-Hocquenghem Codes", *IEEE Trans. on Inf. Th.*, vol. 10, p. 357-363, 1964.
12. P. Fitzpatrick, "On the Key Equation", *IEEE Trans. on Inf. Th.*, vol. 41, p. 1290-1302, 1995.
13. G. D. Forney, Jr, "On decoding BCH codes", *IEEE Trans. on Inf. Th.*, vol. 11, p. 549-557, 1965.
14. R. W. Hamming, "Error detecting and error correcting codes", *Bell Systems Technical Journal*, vol. 29, p. 147-160, 1950.

[5] Recall that a code is quasi-cyclic code if it is invariant by a power of the cyclic shift.

15. A. Hocquenghem "Codes correcteurs d'erreurs", *Chiffres*, vol. 2, p. 147-156, 1959.
16. D. G. Hoffman *et al. Coding Theory: The Essential*. Marcel Dekker Inc. New York, 1991.
17. J. Justesen. "A class of constructive asymptotically good algebraic codes". *IEEE Trans. on Inf. Th.*, vol 18, p. 652-656, 1972.
18. T. Kasami. "A Gilbert-Varshamov bound for quasi-cycle codes of rate 1/2". *IEEE Trans. on Inf. Th.*, vol. 20, n.5, p. 679–679, 1974.
19. S. Lin, *An Introduction to Error-Correcting Codes*, Englewood Cliff, NJ: Prentice Hall, 1970.
20. Shu Lin and E. J. Weldon, Jr., "Long BCH codes are bad", *Inform. Control*, vol. 11, n. 4, p.445–451, October 1967.
21. Martínez-Pérez and W. Willems, "Is the class of cyclic codes asymptotically good?". *IEEE Trans. on Inf. Th.*, vol 52, p. 696-700, 2006.
22. F. J. MacWilliams, N. J. A. Sloane, *The Theory of Error-Correcting Codes*, North Holland, 1977.
23. J. L. Massey, "Shift-Register synthesis and BCH Decoding", *IEEE Trans. on Inf. Th.*, vol. 15, p. 122-127, 1969.
24. W. W. Peterson, E. J. Weldon, *Error-Correcting Codes*, (2nd Edition), MIT Press, Massachusetts, 1972.
25. V. Pless, *Introduction to the theory of Error-Correcting codes*. UIUC, John Wiley, 1982.
26. V. S. Pless and W. Huffman *Handbook of coding theory*, North-Holland, Amsterdam, 1998.
27. E. Prange, "Cyclic Error-Correcting Codes in Two Symbols", *Air Force Cambridge Research Center*, Cambridge, MA, Tech. Rep. AFCRC-TN-57-103, 1957.
28. I. S. Reed and G. Solomon, "Polynomial codes over certain finite fields", *J.SIAM*, vol. 8, p. 300-304, 1960.
29. C. E. Shannon, "A mathematical theory of communication", *Bell Systems Technical Journal*, vol. 27, p. 379–423, p. 623-656, 1948.
30. R. Singleton, "Maximum distance of q-nary codes", *IEEE Trans. on Inf. Th.*, vol. 10, p. 116-118, 1964.
31. H. Stichtenoth. "Transitive and self-dual codes attaining the Tsfasman-Vlăduţ-Zink bound". *IEEE Trans. on Inf. Th.*, vol. 52, n.5, p. 2218–2224, 2006.
32. Y. Sugiyama, M. Kasahara, S. Hirasawa,T. Namekawa, "A Method for Solving Key Equation for Decoding Goppa Codes", *Inform. Contr.*, vol. 27, n.1, p.87-89, 1975.
33. J.H. Van Lint, *Introduction to Coding Theory*, Springer Verlag, 1999.
34. A. Vardy, " Algorithmic complexity in coding theory and the minimum distance problem", *STOC '97: Proceedings of the twenty-ninth annual ACM symposium on Theory of computing*, p. 92–109, 1997.