Computer Science Journal of Moldova, vol.**27**, no.1(79), 2019

Error correcting codes from sub-exceeding functions

L. Rabefihavanana, H. Andriatahiny, T. Rabeherimanana

Abstract

In this paper, we present linear systematic error-correcting codes $\mathcal{L}_k$ and $\mathcal{L}_k^+$ which are the results of our research on the sub-exceeding functions.

Given an integer k such that $k \geq 3$, these two codes are respectively $[2k, k]$ and $[3k, k]$ linear codes. The minimum distance of $\mathcal{L}_3$ is 3 and for $k \geq 4$ the minimum distance of $\mathcal{L}_k$ is 4. The code $\mathcal{L}_k^+$, the minimum distances are respectively 5 and 6 for $k = 4$ and $k \geq 5$.

By calculating the complexity of the algorithms, our codes have fast and efficient decoding.

Then, for a short and medium distance data transmission (wifi network, bluetooth, cable, ...), we see that the codes mentioned above present many advantages.

Keywords: Error correction code, encoding, decoding, sub-exceeding function.

MSC 2010: 94B05; 94B35; 11B34.

1 Introduction

New information and communication technologies or NICTs require today a norm increasingly strict in terms of quality of service. The diversity and the increasing volumes of data exchanged/processed also require increasingly fast and reliable systems.

In these constraints related to information processing, we need to take into account the increased sensitivity of technologies in front of

external disruptive sources. It's about especially to protect information against environmental damage during transmission.

The aim of this article is to build new error correcting codes using the results of our two articles entitled: *Parts of a set and sub-exceeding function: coding and decoding* [16] in 2017 and *Encoding of Partition Set Using Sub-exceeding Function* [15] in 2018.

In the last section of this article, we give the decoding algorithm of these codes using Groebner basis.

2 Preliminaries

Let n be a positive integer, $[\![n]\!]$ denotes the set of positive integers less or equal to n and the zero element, i.e.

$$[\![n]\!] = \{0, 1,\ 2,\ ...\ n\}\,.$$

2.1 The necessary ones on the study of sub-exceeding functions

Definition 2.1. (See [16]) Let n be a positive integer and let f be a map from $[n]$ to $[\![n]\!]$. This function f is said sub-exceeding if for all i in $[\![n]\!]$, we have

$$f(i) \leq i.$$

We denote by $\mathcal{F}_n$ the set of all sub-exceeding functions on $[n]$, i.e.

$$\mathcal{F}_n = \{f :\ [\![n]\!] \longrightarrow [\![n]\!] \ \mid\ f(i) \leq i,\ \forall i \in [\![n]\!]\}\,. \tag{1}$$

Remark 2.2. A sub-exceeding function f can be represented by the word of $n+1$ alphabet $f(0)f(1)f(2)...f(n)$. So, we describe f by its images $f = f(0)f(1)f(2)...f(n)$.

Definition 2.3. (See [16]) Let n and k be two integers such that $0 \leq k \leq n$. We define by $\mathcal{H}_n^k$ the subset of $\mathcal{F}_n$ such that

$$\mathcal{H}_n^k = \{f \in \mathcal{F}_n \mid f(i) \leqslant f(i+1) \text{ for all } i \in [\![n]\!] \text{ and } Im(f) = [\![k]\!]\}\ . \tag{2}$$

Here, $\mathcal{H}_n^k$ is the set of all sub-exceeding functions of $\mathcal{F}_n$ with a quasi-increasing sequence of images formed by all elements of $[\![k]\!]$.

Example 2.4. *Take $n = 4$ and $k = 3$. We find that the function $f = 01123$ is really in $\mathcal{H}_4^3$ because $(f(i))_{0\leq i\leq 4}$ is a quasi-increasing sequence formed by all the elements of $[\![3]\!]$. But if we take $f = 01133$, even if the sequence $(f(i))_{0\leq i\leq 4}$ is quasi-increasing, $f = 01133 \notin \mathcal{H}_4^3$ because $Im(f) \neq [\![3]\!]$ (without 2 among the $f(i)$).*

Following Definition 2.3, we denote by $\mathcal{H}_n$ the set defined as follows:

$$\mathcal{H}_n = \bigcup_{k=0}^{n} \mathcal{H}_n^k. \tag{3}$$

Theorem 2.5. *(See [16])*
Let n and k be two integers such that $0 \leq k \leq n$.

1. *For $k = 0$, we always find that $\mathcal{H}_n^0$ is a set of singletons:*

$$\mathcal{H}_n^0 \ = \ \{f \ = \ 000...00_{n+1\text{-terms}}\}.$$

2. *For $k = n$, we also find that $\mathcal{H}_n^n$ is a set of singletons:*

$$\mathcal{H}_n^n \ = \ \{f \ = \ 0123....(n-1)(n)\}.$$

3. *For any integer k such that $0 < k < n$, we can construct all sub-exceeding functions of $\mathcal{H}_n^k$ as follows:*

   (a) *Take all the elements of $\mathcal{H}_{n-1}^{k-1}$ and add the integer k at the end,*

   (b) *Take all the elements of $\mathcal{H}_{n-1}^k$ and add the integer k at the end*

   To better presentation of this construction, we adopt the following writing:

$$\mathcal{H}_n^k \ = \ \left\{\mathcal{H}_{n-1}^{k-1} \curvearrowleft k\right\} \bigcup \left\{\mathcal{H}_{n-1}^k \curvearrowleft k\right\}.$$

   Here, $(*) \curvearrowleft k$ means that we add the integer k at the end of all elements of $(*)$.

Table 1. The iteration table of the elements of $\mathcal{H}_n^k$

$n \quad \diagdown \quad k$	0	1	2	3	4	...
0	0					
1	00	01				
2	000	001 011	012			
3	0000	0001 0011 0111	0012 0112 0122	0123		
4	00000	00001 00011 00111 01111	00012 00112 01112 00122 01122 01222	00123 01123 01223 01233	01234	

From Theorem 2.5, we have this Table 1 which presents all elements of $\mathcal{H}_n^k$ for some integers n ($n = 0,\ 1,\ 2,\ 3$ and 4).

Proposition 2.6. *See [16]*
Let n and k be two integers such that $0 \leqslant k \leqslant n$. So, we have the following relations:

1. *Card* $\mathcal{H}_n^0 = Card\ \mathcal{H}_n^n = 1$,
2. *Card* $\mathcal{H}_n^k = Card\ \mathcal{H}_{n-1}^{k-1} + Card\ \mathcal{H}_{n-1}^k$,
3. *Card* $\mathcal{H}_n^k = \binom{n}{k}$ *and Card* $\mathcal{H}_n = 2^n$.

Proof. From the construction of the elements of $\mathcal{H}_n^k$ in Theorem 2.5, we have directly the result of Proposition 2.6. □

This Proposition 2.6 presents to us the iterative calculus of the cardinal of $\mathcal{H}_n^k$.

Thus, Table 2 below gives the cardinal of $\mathcal{H}_n^k$ for some integers n ($n = 0,\ 1,\ 2,\ 3$ and 4).

Table 2. The cardinal table of $\mathcal{H}_n^k$

$n \quad \diagdown \quad k$	0	1	2	3	4	...
0	1					
1	1	1				
2	1	2	1			
3	1	3	3	1		
4	1	4	6	4	1	
$\vdots$						

Thus constructed, Table 2 is none other than the Pascal triangle.

3 Main result: error-correcting codes from the study on the sub-exceeding function

In this section, we present our linear error-correcting code from sub-exceeding function.

3.1 The error-correcting code constructions

Recall that for a positive integer n, a function f from $[\![n]\!]$ to $[\![n]\!]$ is said to be sub-exceeding if for any integer i in $[\![n]\!]$, we always have the inequality $f(i) \leq i$.

Thus, the sub-exceeding term amounts to saying that the image of an integer i by an application f is always an integer smaller or equal to this one.

Theorem 3.1. *Let k be a positive integer and let f be an application from $[\![k]\!]$ to $\mathbb{F}_2^{k+1}$. Then the application f is a sub-exceeding function if and only if $f(0) = 0$.*

This theorem tells us that all message of k bits on $\mathbb{F}_2$ which begins with 0 is a sub-exceeding function.

Proof. We say that the image of an integer i in $[\![k]\!]$ by the application f is always equal to 0 or 1. Thus, by the condition $f(0) = 0$, we have $f(i) \leq i$ for all i. So, f is a sub-exceeding function. □

Now, let's examine the subset $\mathcal{H}_k$ for the set of sub-exceeding functions in all application from $[\![k]\!]$ in $\mathbb{F}_2^{k+1}$. That is to say the subset $\mathcal{H}_k$ for the set of k bits messages on $\mathbb{F}_2$.

Referring to Theorem 2.5, we can have all the elements of $\mathcal{H}_k$ (see Table 3). Moreover, from Proposition 2.6, we find

$$\text{Card } (\mathcal{H}_k^0) = 1 \text{ and that } \text{Card}(\mathcal{H}_k^1) = k. \tag{4}$$

Table 3 shows the elements of $\mathcal{H}_k^i$ for each value of $i \in \{0, 1\}$ and some integer k.

Definition 3.2. For a positive integer k, we define by T_k the matrix of $k+1$ rows and k columns such that

$$T_k = \begin{pmatrix} 0 & 0 & 0 & \dots & 0 & 0 & 1 \\ 0 & 0 & 0 & \dots & 0 & 1 & 1 \\ 0 & 0 & 0 & \dots & 1 & 1 & 0 \\ \vdots & \vdots & \vdots & \nearrow & \nearrow & \vdots & \vdots \\ 0 & 0 & 1 & 1 & 0 & \dots & 0 \\ 0 & 1 & 1 & 0 & 0 & \dots & 0 \\ 1 & 1 & 0 & 0 & 0 & \dots & 0 \\ 0 & 1 & 1 & 1 & 1 & \dots & 1 \end{pmatrix}. \tag{5}$$

Here, $T[i, j]$ denotes the element of T_k in the i^{th} row and the j^{th} column and

- for all i in $\{1,\ 2,\ ..,\ k\}$,
  - $T_k[k - i + 1, i] = 1$,
  - $T_k[k + 1, i] = 1$, except $T_k[k + 1, 1] = 0$,
- for all i in $\{2,\ ..,\ k\}$, $T_k[k - i + 2, i] = 1$.

Table 3. The elements of $\mathcal{H}_k^i$

$n \setminus k$	0	1
0	0	
1	00	01
2	000	001
		011
		0001
3	0000	0011
		0111
		00001
4	00000	00011
		00111
		01111
		000001
		000011
5	000000	000111
		001111
		011111
$\vdots$		

In the other cases, $T_k[i,j] = 0$ and j in $\{1,\ 2,\ ..,\ k\}$.

Remark 3.3. The matrix T_k of our definition establishes the relation between the set $\mathcal{H}_k^1$ and the generating matrix of our code that we will see below. (see also [16]).

Proposition 3.4. *Reminding that $\mathcal{H}_k^1$ is the set of sub-exceeding functions f_i of length $k+1$ such that*

$$f_i = 000 \quad ... \quad 0\underbrace{11...1}_{i-times}\ ,\ \textit{with } i \in \{1, ...k\}\,.$$

Then the product $f_i \times T_k$ gives the word g_i such that

$$\left\{\begin{array}{cc} g_1 = & 0111...111 \\ g_2 = & 1011...111 \\ g_3 = & 1101...111 \\ \vdots & \vdots \\ g_{k-1} = & 1111...101 \\ g_k = & 1111...110 \end{array}\right\}. \tag{6}$$

Notation 3.5. *Now let's denote by G_k the matrix*

$$G_k = \begin{pmatrix} g_1 \\ g_2 \\ g_3 \\ \vdots \\ g_k \end{pmatrix} = \begin{pmatrix} 0111...111 \\ 1011...111 \\ 1101...111 \\ \vdots \\ 1111...101 \\ 1111...110 \end{pmatrix}. \tag{7}$$

Example 3.6. *For $k = 3$, we have:*

$$G_3 \left\{\begin{array}{cc} g_1 = & 011 \\ g_2 = & 101 \\ g_3 = & 110 \end{array}\right\}, T_3 = \begin{pmatrix} 0 & 0 & 1 \\ 0 & 1 & 1 \\ 1 & 1 & 0 \\ 0 & 1 & 1 \end{pmatrix} \text{ where } \mathcal{H}_3^1 \left\{\begin{array}{cc} f_1 = & 0001 \\ f_2 = & 0011 \\ f_3 = & 0111 \end{array}\right\}$$

For $k = 4$, we have:

$$G_4 \left\{\begin{array}{cc} g_1 = & 0111 \\ g_2 = & 1011 \\ g_3 = & 1101 \\ g_4 = & 1110 \end{array}\right\}, T_4 = \begin{pmatrix} 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 1 \\ 0 & 1 & 1 & 0 \\ 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 1 \end{pmatrix}, \quad \mathcal{H}_4^1 \left\{\begin{array}{cc} f_1 = & 00001 \\ f_2 = & 00011 \\ f_3 = & 00111 \\ f_4 = & 01111 \end{array}\right\}$$

3.2 The linear systematic code $\mathcal{L}_k$

Theorem 3.7. *Let k be a positive integer and let ψ be the linear application from $\mathbb{F}_2^k$ to $\mathbb{F}_2^{2k}$ such that*

$$\begin{array}{rccc} \psi & : & \mathbb{F}_2^k & \longrightarrow & \mathbb{F}_2^{2k} \\ & & m & \longmapsto & \psi(m) = m \times G_{\mathcal{L}_k}, \end{array} \tag{8}$$

where m is the message of k bits such that $m = m_1 m_2 ... m_k$ and $G_{\mathcal{L}_k}$ is the generator matrix such that $G_{\mathcal{L}_k} = \begin{pmatrix} I_k & G_k \end{pmatrix}$, where G_k is the matrix defined in the equation (6).

Thus, the application ψ forms a systematic $[2k, k]$-linear error-correcting code denoted by $\mathcal{L}_k$. The minimum distance of $\mathcal{L}_3$ is 3, and for $k \geq 4$ the minimum distance of $\mathcal{L}_k$ is 4.

Proof. First, since $G_{\mathcal{L}_k} = \begin{pmatrix} I_k & G_k \end{pmatrix}$ is a matrix of k rows and $2k$ columns whose rows are linearly independent vectors, so the application ψ is injective from $\mathbb{F}_2^k$ to $\mathbb{F}_2^{2k}$. Thus, $\psi(\mathbb{F}_2^k)$ is a vector space over $\mathbb{F}_2$ of dimension k. Then ψ forms a systematic linear error-correcting code of dimension k and length $2k$.

Now, let m be the message such as $m = m_1\ m_2 ... m_k$ and note by c its image by the application ψ.

$$c = \psi(m) = m \times G_{\mathcal{L}_k}.$$

Since ψ is a systematic code, a codeword c of length $2k$ can be separated into two vectors c_1 and c_2. That is to say, $c = c_1\ c_2$. Here, the vector c_1 is the original message ($c_1 = m$), and c_2 is the vector (control bits) such that $c_2 = m \times G_k$.

So, for any integer i in $\{1, 2, ..., k\}$, we have

$$c_2[i] = \sum_{j=1, j\neq i}^{k} m_j.$$

So, two cases are possible:

- If the weight of m is even

$$\begin{cases} \text{and that } m_i = 0 & \Rightarrow \quad c_2[i] = \sum_{j=1, j\neq i}^{k} m_j = 0. \\ \text{and if } m_i = 1 & \Rightarrow \quad c_2[i] = \sum_{j=1, j\neq i}^{k} m_j = 1. \end{cases}$$

  In this case, the code word c is: $c = m\ m$.
  (Ex: for $k = 6$, if $m = 011101$, we have $c = 011101\ 011101$)

- If the weight of m is odd

$$\begin{cases} \text{and that } \ m_i = 0 & \Rightarrow \quad c_2[i] = \sum_{j=1, j\neq i}^{k} m_j = 1. \\ \text{and if } \ m_i = 1 & \Rightarrow \quad c_2[i] = \sum_{j=1, j\neq i}^{k} m_j = 0. \end{cases}$$

In this case, the code word c is: $c = m\ \overline{m}$, where $\overline{m}$ is the opposite of m.
(Ex: for $k = 7$, if $m = 0000111$, we have $c = 0000111\ 1111000$)

Now,

1. Take $k = 3$,

$$\begin{array}{rcl} \text{if } w(m) = 1 & \longrightarrow & w(c) = 3, \\ \text{if } w(m) = 2 & \longrightarrow & w(c) = 4, \\ \text{if } w(m) = 3 & \longrightarrow & w(c) = 3. \end{array} \tag{9}$$

Thus, the minimum distance for the code $\mathcal{L}_3$ is 3.

2. for $k \geq 4$,

$$\begin{array}{rcl} \text{if } w(m) = 1 & \longrightarrow & w(c) = k, \\ \text{if } w(m) = 2 & \longrightarrow & w(c) = 4, \\ \text{if } w(m) = 3 & \longrightarrow & w(c) = k, \\ & \vdots & \\ \text{if } w(m) = p \text{ (even)} & \longrightarrow & w(c) = 2p, \\ \text{if } w(m) = q \text{ (odd)} & \longrightarrow & w(c) = k. \end{array} \tag{10}$$

Thus, the minimum distance for the code $\mathcal{L}_k$ is 4.

□

Example 3.8. *For $k = 3$, from the main theorem (3.7), we have*

$$\mathcal{L}_3 = \left\{ \begin{array}{c} 000000 \\ 001110 \\ 010101 \\ 100011 \\ 011011 \\ 101101 \\ 110110 \\ 111000 \end{array} \right\}, \quad G_{\mathcal{L}_3} = \begin{pmatrix} 1 & 0 & 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 & 1 & 0 \end{pmatrix}.$$

Initial messages:

$$\mathbb{F}_2^3 = \begin{pmatrix} 000 & 010 & 011 & 110 \\ 001 & 100 & 101 & 111 \end{pmatrix}.$$

Example 3.9. *For $k = 4$, from the main theorem (3.7), we have*

$$G_{\mathcal{L}_4} = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 1 & 1 & 1 \\ 0 & 1 & 0 & 0 & 1 & 0 & 1 & 1 \\ 0 & 0 & 1 & 0 & 1 & 1 & 0 & 1 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 & 0 \end{pmatrix} \tag{11}$$

and

$$\mathcal{L}_4 = \begin{array}{ccc} 00000000 & 00110011 & 01111000 \\ 00011110 & 01010101 & 10110100 \\ 00101101 & 10011001 & 11010010 \\ 01001011 & 01100110 & 11100001 \\ 10000111 & 10101010 & 11111111 \\ & 11001100 & \end{array}$$

3.3 The linear systematic code $\mathcal{L}_k^+$

Theorem 3.10. *Let k be an integer such that $k \geq 4$, and let's take the system $\{e'_1, e'_2, ..., e'_k\}$, where $e'_i = \psi(g_i)$ (see the equation 6). So, we can build a $[3k, k]$-linear systematic code with generator matrix*

$$G_{\mathcal{L}_k^+} = \begin{pmatrix} & e'_1 \\ I_k & \vdots \\ & e'_k \end{pmatrix} \tag{12}$$

which is denoted by $\mathcal{L}_k^+$.

The code $\mathcal{L}_4^+$ has minimum distance 5, and for $k \geq 5$, the code $\mathcal{L}_k^+$ has minimum distance 6. The generating matrix G of this code has the form

$$G_{\mathcal{L}_k^+} = \begin{pmatrix} I_k & G_k & I_k \end{pmatrix}. \tag{13}$$

Proof. Since $\mathcal{L}_k$ is a sub-space over $\mathbb{F}_2$, any linear combination between the code words $e'_1, e'_2, ..., e'_k$ gives a code in $\mathcal{L}_k$ of weight equal to 4. Then, for a message m in $\mathbb{F}_2^k$, the code word c generated by the matrix $G_{\mathcal{L}_k^+}$ (ie $c = m \times G_{\mathcal{L}_k^+}$) has a weight:

1. For $k = 4$,
$$\begin{array}{lll} \text{if } w(m) = 1 & \longrightarrow & w(c') = 5, \\ \text{if } w(m) = 2 & \longrightarrow & w(c') = 6, \\ \text{if } w(m) = 3 & \longrightarrow & w(c') = 7, \\ \text{if } w(m) = 4 & \longrightarrow & w(c') = 12. \end{array} \tag{14}$$
So we have a code 2-corrector $\mathcal{L}_4^+$ with a minimal distance $d = 5$.

2. For $k \geq 5$,
$$\begin{array}{lll} \text{if } w(m) = 1 & \longrightarrow & w(c') \geq 6, \\ \text{if } w(m) = 2 & \longrightarrow & w(c') = 6, \\ \text{if } w(m) = 3 & \longrightarrow & w(c') \geq 7, \\ & \vdots & \\ \text{if } w(m) = k & \longrightarrow & w(c') \geq k + 4. \end{array} \tag{15}$$
So we have a code 2 -corrector $\mathcal{L}_k^+$ with a minimal distance $d = 6$.

□

Example 3.11. *The code $\mathcal{L}_4^+$.*
Now take the four (4) vectors in $\mathcal{L}_4$ which are:

$$\begin{array}{l} e'_1 = 01111000 \\ e'_2 = 10110100 \\ e'_3 = 11010010 \\ e'_4 = 11100001 \end{array} . \tag{16}$$

The code $\mathcal{L}_4^+$ is as follows:

$$G_{\mathcal{L}_4^+} = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 & 1 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 1 & 1 & 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 1 \end{pmatrix} . \tag{17}$$

$$\mathcal{L}_4^+ = \begin{array}{c} 0000\ 0000\ 0000 \\ 0001\ 1110\ 0001 \\ 0010\ 1101\ 0010 \\ 0100\ 1011\ 0100 \\ 1000\ 0111\ 1000 \end{array} \quad \begin{array}{c} 0011\ 0011\ 0011 \\ 0101\ 0101\ 0101 \\ 1001\ 1001\ 1001 \\ 0110\ 0110\ 0110 \\ 1010\ 1010\ 1010 \\ 1100\ 1100\ 1100 \end{array} \quad \begin{array}{c} 0111\ 1000\ 0111 \\ 1011\ 0100\ 1011 \\ 1101\ 0010\ 1101 \\ 1110\ 0001\ 1110 \\ 1111\ 1111\ 1111 \end{array}$$

Example 3.12. *The code $\mathcal{L}_5^+$.*
Now take the five (5) vectors in $\mathcal{L}_5$ which are:

$$\begin{array}{l} e'_1 = 01111\ 10000 \\ e'_2 = 10111\ 01000 \\ e'_3 = 11011\ 00100\ . \\ e'_4 = 11101\ 00010 \\ e'_4 = 11110\ 00001 \end{array} \tag{18}$$

The code $\mathcal{L}_5^+$ is thus as follows:

$$G_{\mathcal{L}_5^+} = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 1 & 0 & 1 & 1 & 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 & 1 & 0 & 1 & 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 1 & 1 & 1 & 0 & 1 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix}. \tag{19}$$

The code words of $\mathcal{L}_5^+$ are:

$$\begin{array}{ccccc} & 00011\ 00011\ 00011 & 11100\ 00011\ 11100 & \\ & 00101\ 00101\ 00101 & 11010\ 00101\ 11010 & \\ 00000\ 00000\ 00000 & 00110\ 00110\ 00110 & 11001\ 00110\ 11001 & 01111\ 01111\ 01111 \\ 00001\ 11110\ 00001 & 01001\ 01001\ 01001 & 10110\ 01001\ 10110 & 10111\ 10111\ 10111 \\ 00010\ 11101\ 00010 & 01010\ 01010\ 01010 & 10101\ 01010\ 10101 & 11011\ 11011\ 11011 \\ 00100\ 11011\ 00100 & 01100\ 01100\ 01100 & 10011\ 01100\ 10011 & 11101\ 11101\ 11101 \\ 01000\ 10111\ 01000 & 10001\ 10001\ 10001 & 01110\ 10001\ 01110 & 11110\ 11110\ 11110 \\ 10000\ 01111\ 10000 & 10010\ 10010\ 10010 & 01101\ 10010\ 01101 & 11111\ 00000\ 11111 \\ & 10100\ 10100\ 10100 & 01011\ 10100\ 01011 & \\ & 11000\ 11000\ 11000 & 00111\ 11000\ 00111 & \end{array} \tag{20}$$

4 Decoding for the error correcting codes $\mathcal{L}_k$ and $\mathcal{L}_k^+$

After considering the parameters necessary for the study of these codes, we present here the appropriate decoding algorithms.

4.1 The dual codes of $\mathcal{L}_k$ and $\mathcal{L}_k^+$

Theorem 4.1. *(See [9], [12], [17])*
If C is an $[n,k]$ code over $\mathbb{F}_2$, then the dual code $C^\perp$ is given by all words $u \in \mathbb{F}_2^n$ such that $< u\ ,\ c >= 0$ for each $c \in C$, where $<\ ,\ >$ denotes the ordinary inner product. The dual code $C^\perp$ is an $[n, n-k]$ code. If $G = (I_k \mid M)$ is a generator matrix for C, then $H = \left(M^T \mid I_{n-k}\right)$ is the generator matrix for $C^\perp$.

Example 4.2. *For the code $\mathcal{L}_4$ and $\mathcal{L}_4^+$ the generator matrix is respectively*

$$G_{\mathcal{L}_4} = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 1 & 1 & 1 \\ 0 & 1 & 0 & 0 & 1 & 0 & 1 & 1 \\ 0 & 0 & 1 & 0 & 1 & 1 & 0 & 1 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 & 0 \end{pmatrix}$$

and

$$G_{\mathcal{L}_4^+} = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 & 1 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 1 & 1 & 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 1 \end{pmatrix}.$$

So the dual code $\mathcal{L}_4^\perp$ and $(\mathcal{L}_4^+)^\perp$ have respectively his generator matrix:

$$H_{\mathcal{L}_4^\perp} = \begin{pmatrix} 0 & 1 & 1 & 1 & 1 & 0 & 0 & 0 \\ 1 & 0 & 1 & 1 & 0 & 1 & 0 & 0 \\ 1 & 1 & 0 & 1 & 0 & 0 & 1 & 0 \\ 1 & 1 & 1 & 0 & 0 & 0 & 0 & 1 \end{pmatrix}$$

and

$$H_{(\mathcal{L}_4^+)^\perp} = \begin{pmatrix} 0 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix}. \tag{21}$$

As the columns of $H_{\mathcal{L}_4^\perp}$ (or $H_{(\mathcal{L}_4^+)^\perp}$) are pairwise distinct, so for a codeword c that contains exactly one error, the decoding will be easy by looking at the $H \times {}^t c$ syndrome.

Definition 4.3. Let c be an element of linear code C such that $c = m_1\ m_2\ ...\ m_n$, where $m_i \in \mathbb{F}_2$ for all i. We define the monomial X^c of $\mathbb{F}_2[X_1X_2...X_n]$ by

$$X^c = X_1^{m_1} X_2^{m_2} ... X_n^{m_n}. \tag{22}$$

Example 4.4. *Take $c = 101101$ which is a codeword of $\mathcal{L}_3$. In $\mathbb{F}_2[X_1X_2...X_6]$, the monomial X^c was*

$$X^c = X_1^1 X_2^0 X_3^1 X_4^1 X_5^0 X_6^1 = X_1X_3X_4X_6.$$

Theorem 4.5. *(see [4])*
Let C be an $[n,k]$-linear systematic code over $\mathbb{F}_2$. Define by I_C the binomial ideal of $\mathbb{F}_2[X_1X_2...X_n]$ associated with C such that

$$I_C = \langle X^c - X^{c'} \mid c - c' \in C \rangle + \langle X_i^2 - 1 \mid 1 \leq i \leq n \rangle. \tag{23}$$

Theorem 4.6 (Groebner basis of the Binomial ideal I_C)**.** *(See [12])*
Take the lexicographic order on $\mathbb{F}_2[X_1X_2...X_n]$, i.e. $X_1 \succ ... \succ X_n$. An $[n,k]$ linear systematic code C of generator matrix $(I_k \mid M)$ has the reduced Groebner basis

$$\mathcal{B} = \{X_i - X^{m_i} \mid 1 \leq i \leq k\} \cup \left\{X_i^2 - 1 \mid k+1 \leq i \leq n\right\}. \tag{24}$$

Here m_i is the i^{th} line of the matrix M.

Example 4.7. *For the code* $(\mathcal{L}_4^+)$*, the corresponding binomial ideal of this code in* $\mathbb{F}_2[X_1X_2...X_n]$ *has the reduced Groebner basis given by the elements*

$$\begin{array}{ll} b_1 = X_1 - X_6X_7X_8X_9 & b_5 = X_5^2 - 1 \\ b_2 = X_2 - X_5X_7X_8X_{10} & b_6 = X_6^2 - 1 \\ b_3 = X_3 - X_5X_6X_8X_{11} & b_7 = X_7^2 - 1 \\ b_4 = X_4 - X_5X_6X_7X_{12} & b_8 = X_8^2 - 1 \\ b_9 = X_9^2 - 1 & b_{10} = X_{10}^2 - 1 \\ b_{11} = X_{11}^2 - 1 & b_{12} = X_{12}^2 - 1 \end{array}, \tag{25}$$

where

$$G_{\mathcal{L}_4^+} = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 & 1 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 1 & 1 & 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 1 \end{pmatrix}. \tag{26}$$

4.2 Error-correction of the code $\mathcal{L}_k$

1. The ideal case is that no error was produced during transmission. We can use two methods to detect the presence of errors:
   - We make the product of the control matrix H with the received code and we have to find a null vector, which means that there was no error during the transmission.
   - Now the second method: as our code $\mathcal{L}_k$ is a systematic code, the received code word can be split into two, i.e. $c = m_1\ m_2$, where m_1 is the message sent and m_2 is the control code. So, if the weight of m_1 is even and $m_1 = m_2$ or if the weight of m_1 is odd and $m_2 = \overline{m_1}$, in both cases the code has no error during the transmission. Otherwise there are errors.
2. The other case is that errors occur during transmission.
   Suppose that one error was produced. So, we find out here how to fix it. The only error must be in m_1 or m_2. Moreover, if the real message m sent is of even (odd) weight, the word $m_1 + m_2$ is of weight 1 (resp $(k-1)$). As a result, the decoding is as follows:

- If the weight of $m_1 + m_2$ is 1, it remains to find the only bit that distinguishes m_1 from m_2 and fix it for the weight of m_1 to be even.
- If the weight of $m_1 + m_2$ is $k-1$, it remains to find the only bit for that $m_2 = \overline{m_1}$ and fix it for the weight of m_1 to be odd.

Algorithm of Decoding for the code $\mathcal{L}_k$

Input r (*received word*)

Output c (*corrected word*)

Begin

Determine m_1 and m_2 such that $r = m_1 m_2$;
Calculate $w_1 = w(m_1)$, $w_2 = w(m_2)$
and $w_{1,2} = w(m_1 + m_2)$;

- If $w_{1,2} = 0$ or $w_{1,2} = k$, so $c = r$;
- If $w_{1,2} = 1$
  - and if w_1 is even, so $c = m_1\ m_1$;
  - and if w_1 is odd, so $c = m_2\ m_2$;
- If $w_{1,2} = k - 1$
  - and if w_1 is odd, so $c = m_1\ \overline{m_1}$;
  - and if w_1 is even, so $c = \overline{m_2}\ m_2$;
- Else print("The message contains more than one error, we can not correct them")

End

By simple calculation, we find that the complexity of this algorithm is linear, i.e. $\mathcal{O}(n)$.

Remark 4.8. For $k = 4$, the parameters of the code $\mathcal{L}_4$ coincide with those of the extended Hamming code $H(8, 4, 4)$. But by comparing the decoding algorithm presented in [14] on page 88 to 92 (decoding by the butterfly operator) with our algorithm, the complexity of our decoding is interesting.

Remark 4.9. The code Hadamard$[4, 2, 2]$ have also the same parameters as our code $\mathcal{L}_2$.

4.3 Error correction for the code $\mathcal{L}_k^+$

We try to give here the correction steps for a codeword that contains at most 2 errors.
An immediate consequence of the study of the reduced Groebner basis of the Binomial ideal I_C is a decoding algorithm for linear codes. This algorithm was given in slightly different form in [4].

Theorem 4.10. *(See [12])*
Let C be an [n,k] code over $\mathbb{F}_2$, and let $\mathcal{B}$ be the reduced Groebner basis for C given in (24)*. Suppose the code C is t-error-correcting. The following algorithm gives a decoder D for the code C. Given a received word $c \in \mathbb{F}_2^n$, if the word given by $rem(X^c - 1,\ \mathcal{B})$ has at most t nonzero entries, then form $D(c) = (X^c - 1) - rem(X^c - 1,\ \mathcal{B})$. This gives the codeword that is closest to the received word.*

Remark 4.11. In other ways, for the linear systematic code $\mathcal{L}_k^+$, we can also use the parity check matrix $H_{\mathcal{L}_k^+}$ for the decoding. By the form of this parity check matrix, we have:

- all the columns of $H_{\mathcal{L}_k^+}$ (see (21)) are different from each other,
- all additions of two columns of $H_{\mathcal{L}_k^+}$ are also pairwise distinct.

Then, for a received word c which contains at most two errors,

- if one error was presented at the i^{th} position of c, thus $H \times {}^t c = h_i$.
- If c contains two errors at the i^{th} and j^{th} position $(i < j)$, thus $H \times {}^t c = h_i + h_j$.

The calculation of $H_{\mathcal{L}_k^+} \times {}^t c$ specifies the positions of errors. So, we find that the number of operations for the decoding of the code $\mathcal{L}_k^+$ denoted N_{op} is $2k(6k + 1)$ i.e.

$$N_{op}(\mathcal{L}_k^+) = \frac{2n(2n+1)}{3}, \text{ where } n = 3k.$$

Then, we find that the complexity of this algorithm is quadratic, i.e. $\mathcal{O}(n^2)$.

4.3.1 Comparative analysis between $\mathcal{L}_k^+$ and the code of Hamming

Table 4 below presents a comparative study between the Hamming code and $\mathcal{L}_k^+$ for some cases where the two codes are of the same length.

Table 4. Comparative study between Hamming code and $\mathcal{L}_k^+$

	HAMMING CODE	CODE BUILT FROM SUB-EXCEEDING FONCTION
Form	It is a linear code of the form $[2^r-1, 2^r-r-1, 3]$. For $r=4$, we have $[15, 11, 3]-$ linear code. For $r=6$, we have $[63, 57, 3]-$ linear code.	It is a linear code of the form $[3k, k, 6]$. For $k=5$, we have $[15, 5, 6]-$ linear code. For $k=21$, we have $[63, 21, 6]-$ linear code.
Parameters	Minimum distance: $d=3$ Correction capacity: $e_c=1$ For the $[15, 11, 3]-$ code Correction rate is $C_r=1/15$ For the $[63, 57, 3]-$ code Correction rate is $C_r=1/63$	Minimum distance: $d=6$ Correction capacity: $e_c=2$ For the $[15, 5, 6]-$ code Correction rate is $C_r=2/15$ For the $[63, 21, 6]-$ code Correction rate is $C_r=2/63$

These codes have the same length but different dimensions. However, the code $\mathcal{L}_k^+$ has 2 bits for the correction capability and the error detection capability was 5 comparing with the Hamming code which can correct one error and detect only 2 errors.

References

[1] Pierre Abbrugiati, *Introduction to error correcting codes*, 23 January 2006, 36 p. (in French)

[2] W. Adams, P. Loustaunau, *An Introduction to Groebner Bases* (Graduate Studies in Mathematics, vol. 3), Providence, RI, USA: American Mathematical Society, 1994, 289 p. ISBN:0-8218-3804-0.

[3] T. Becker, V. Weispfenning, *Groebner Bases. A Computational Approach to Commutative Algebra* (Graduate Texts in Mathematics, vol. 141), New York: Springer, 1998, 576 p. ISBN-10: 0387979719. ISBN-13: 978-0387979717.

[4] M. Borges-Quintana, M.A. Borges-Trenard, P. Fitzpatrick, E. Martinez-Moro, "Groebner bases and combinatorics for binary codes," *Applicable Algebra in Engineering, Communication and Computing*, vol. 19, no. 5, pp. 393–411, 2008.

[5] M. J. E. Golay, "Notes on digital coding," in *Proc. IRE*, 1949, vol. 37, pp. 657.

[6] *Combinatorial Analysis* (Proceedings of Symposia in Applied Mathematics, vol. 10), R. Bellman and M. Hall Jr., Eds. Providence, R.I., USA: American Mathematical Society, 1960, 311p. Electronic ISBN: 978-0-8218-9225-1.

[7] D. Knuth, *The Art of Computer Programming – Sorting and Searching*, 2nd ed., vol. 3, Redwood City, CA, USA: Addison Wesley Longman Publishing Co. Inc., 1998, xiv+780p. ISBN: 0-201-89685-0.

[8] D.H. Lehmer, "Teaching combinatorial tricks to a computer," in *Combinatorial Analisis* (Proc. Sympos. Appl. Math., vol. 10), Providence, R.I., USA: Amer. Math. Soc, 1960, pp. 179–193.

[9] F.J. MacWilliams and N.J.A. Sloane, *The theory of error correcting codes* (North-Holland mathematical library, vol. 16), Amsterdam-New York-Oxford, Netherlands: North-Holland Pub. Co., 1977, xx+762 p. ISBN: 0444850090 and 0444850104.

[10] Roberto Mantaci, "On the distribution of anti-excesses in the symmetric group and its subgroups," *Theoret. Comp. Science*, vol. 117, no. 1–2, pp. 243–253, 1993. (in French).

[11] Roberto Mantaci and Fanja Rakotondrajao, "A permutation that knows what Eulerian means," *Discrete Mathematics and Theortical Computer Science*, vol. 4, pp. 101–108, May 2001.

[12] Mehwish Saleemi and Karl-Heinz Zimmermann, "Groebner Bases For Linear Codes," *International Journal of Pure and Applied Mathematics*, vol. 62, no. 4, pp. 481–491, 2010.

[13] Melvyn el Kamel-Meurigne (Under the direction of Benoit Fabrèges), *Error correcting code*, 23 January 2006, 31 p. (in French).

[14] Senad Mohamed-Mahmoud, "Contribution to soft decision decoding of bulk correcting codes," Ph.D. dissertation, January 2016, HAL Id: tel-01356210. (in French).

[15] L. Rabefihavanana, "Encoding of Partition Set Using Sub-exceeding Function," *International Journal of Contemporary Mathematical Sciences*, vol. 13, no. 2, pp. 63–78, 2018.

[16] Luc Rabefihavanana, "Parts of a set and sub-exceeding function: Coding and Decoding," *Bulletin Of Mathematics And Statistics Research*, vol. 5., no. 3, pp, 38–53, 2017 (July-Sept) Ky Publications.

[17] J. H. van Lint, *Introduction to Coding Theory* (Graduate Texts in Mathematics, vol. 86), Berlin: Springer-Verlag Berlin Heidelberg, 1999, xii+186 p.

L. Rabefihavanana, H. Andriatahiny,
T. Rabeherimanana

Received November 16, 2018
Revised January 21, 2019

Luc Rabefihavanana
Department of Mathematics and Computer Science
Faculty of Sciences, University of Antananarivo, Madagascar
Phone: (+261) 34 74 877 40
E–mail: lucrabefihavanana@yahoo.fr

Harinaivo Andriatahiny
Department of Mathematics and Computer Science
Faculty of Sciences, University of Antananarivo, Madagascar
E–mail: aharinaivo@yahoo.fr

Toussaint Joseph Rabeherimanana
Department of Mathematics and Computer Science
Faculty of Sciences, University of Antananarivo, Madagascar
E–mail: rabeherimanana.toussaint@yahoo.fr