N° d'ordre…

UNIVERSITE DE M'SILA

FACULTE DES SCIENCES ET DES SCIENCES DE L'INGENIORAT

DÉPARTEMENT DE MATHÉMATIQUES

MÉMOIRE

Présenté pour l'obtention du diplôme de Magister
Spécialité : MATHÉMATIQUES
Option : MATHÉMATIQUES DISCRÈTES.

Par :
HEBOUB LAKHDAR

SUJET

ETUDE DE TECHNIQUES DE DÉCODAGE DES CODES LINÉAIRES

Soutenu publiquement le 16/12/2009 devant le jury composé de :

BOUDAOUD Abdelmajid	**Prof, Université de M'sila**	**Président**
MIHOUBI Douadi	**M.C, Université de M'sila**	**Rapporteur**
BERBOUCHA Ahmed	**M.C, Université de Béjaia**	**Examinateur**
DAHMANE Achour	**M.C, Université de M'sila**	**Examinateur**

Promotion : 2009/2010

Remerciements

Je tiens à remercier le docteur D. Mihoubi d'avoir accepté de diriger ce travail et de créer autour de moi un environnement de recherche par ces conseils.

*Comme je tiens à remercier Monsieur le docteur***:** *BOUDAOUD Abdelmajid Pour avoir accepté la présidence du jury.*

*Je remercie également messieurs les docteurs***:** *BERBOUCHA Ahmed, DAHMANE Achour, Pour avoir accepté de juger ce travail.*

Je ne peux oublier de remercier tous les docteurs d'avoir contribué à notre formation durant l'année de la post- graduation, ainsi qu'à toute l'équipe du département de mathématiques

Dédicaces

A mes parents

A ma famille

A tous mes amis

A B.abd elhamid

Sommaire

خلاصة

يندرج هدا العمل في إطار نظرية الشفرات المصححة للأخطاء أكثر دقة دراسة مشاكل فك الشفرات الخطية و التي تنطلق من إيجاد الرسالة الأصلية المرسلة عبر قناة اتصال انطلاقا من الرسالة الواصلة. في البداية نقدم المفاهيم الأساسية لنظرية الشفرات المصححة الأخطاء ثم نتطرق إلى طرق فك الشفرات الخطية مستعملين أربع طرق مختلفة.

- باستعمال الجدول القياسي(المعياري).
- باستعمال اللانمطي (syndrôme).
- طريقة Meggitt.
- طريقة فك الشفرة محاصرة.

وأنهينا بطريقةMeggitt في الحالة غير الثنائية.

الكلمات المفتاحية : الحقول المنتهية، الشفرات الخطية، الشفرات الدورية، طرق فك شفرة.

Résumé

Ce travail se situe dans le cadre de la théorie des codes correcteurs d'erreurs. Plus précisément l'étude de problèmes de décodage des codes linéaires qui consiste à déterminer le message original envoyé via un canal de transmission, à partir du message reçu. Tout d'abord nous présentons les concepts fondamentaux de la théorie des codes correcteurs d'erreurs ensuite nous abordons les méthodes de décodage des codes linéaires, en utilisant quatre méthodes différentes:

-En utilisant le tableau standard.

-le syndrôme.

-la méthode de meggitt.

-la méthode de décodage par piégeage d'erreur.

Et on terminera par la méthode de décodage de meggitt au cas non binaire.

Mots clés: corps finis, codes linéaires, codes cycliques, méthode de décodage.

Abstract

This work is included in the frame of the theory of error correcting codes.

More precisely the study of the problem of the decoding linear codes that consist in determining the original message sent through a canal of transmission using the received message. First we present the basic concepts of the theory of error correcting codes then we discuss the methods of decoding linear codes, using four different methods:

- Using the standard table.

-Syndrome.

-The method of Meggitt.

-The method of decoding error trapping.

And we end with the method of Meggitt decoding non-binary case.

Key words: Finite Fields, Linear codes, cyclic codes, the methods of decoding.

Notations

$|G|$: L'ordre d'un groupe fini ou le cardinal d'un ensemble fini G.

$\mathbb{N}$: L'ensemble des entiers naturels.

$\mathbb{Z}$: L'ensemble des entiers relatifs.

$\mathbb{Z}/p\mathbb{Z}$: L'ensemble des entiers modulo p.

$C(n,k)$: Code correcteur de longueur n et dimension k.

$\overline{X}$: La classe de X modulo une relation d'équivalence.

$\mathbb{k}^*$: Le groupe multiplicatif d'un corps $\mathbb{k}$ avec $\mathbb{k}^* - \{0\}$.

$\mathbb{F}_q$: Un corps fini de cardinal q.

$A[x]$: L'anneau des polynômes à une déterminée x sur un anneau.

$(f(x))$: L'idéal engendré par $f(x)$ dans $A[x]$.

$\cong$: Isomorphisme de groupe, de corps, d'espaces vectoriels.

$[x]$: La partie entière d'un réel x.

$w(x)$: Le poids de Hamming d'un mots x.

rgH: Le rang d'une matrice H.

$\ker H$: L'espace nul d'une matrice H.

$d(x,y)$: Distance de Hamming entre x et y.

$C^{\perp}$: Le code dual du code considéré.

INTRODUCTION GENERALE

Le transfert d'informations prend de plus en plus d'importance dans notre société que ce soit pour la transmission de photographies de planètes éloignées, pour des communications entre ordinateurs ou encore pour la lecture de nos disques lasers. Les codes correcteurs d'erreurs sont utilisés pour corriger des erreurs quand les messages sont transmis par le biais d'un canal de communication comportant des parasites.

Par exemple nous pourrions transmettre une information binaire (un flot de os et de 1s) a travers un canal parasité aussi rapidement et aussi sûrement que possible. Le canal peut être une ligne téléphonique, une liaison de communication par satellite, une liaison radio haute fréquence. …

La perturbation (le parasite) pourrait être une erreur humaine, foudre, parasite thermal etc…..et pourrait conduire à des erreurs de telle sorte que l'information reçue est différente de celle transmise.

Le transfert de l'information n'est pas parfait, c'est pourquoi il est nécessaire de détecter, et dans certains cas de pouvoir même corriger les erreurs contenues dans le message reçu.

L'enjeu de la détection et de la correction d'erreurs est, donc essentiellement, dans la recherche d'algorithmes de décodage efficaces.

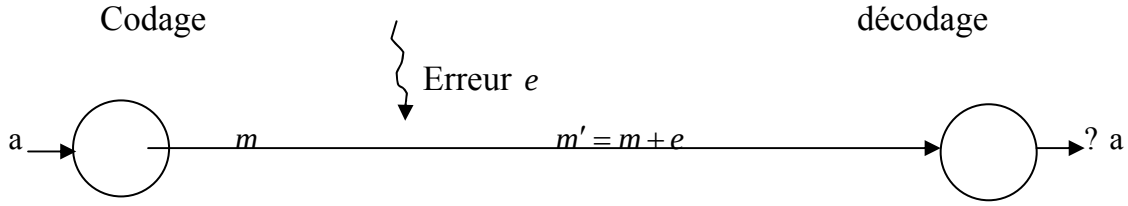
Les codes linéaires, parmi les codes correcteurs, correspondent à ce besoin et particulièrement les codes linéaires cycliques.

Position du problème

Soit q un nombre premier. On considère un code linéaire $C(n,k,d)$ sur un corps finis $\mathbb{F}_q$ le mot de code $m \in C$ déduit du message a est ensuite transmis.

Le canal de transmission peut introduire des erreurs, à la sortie de ce canal, on obtient au lieu de m un mot m'.

On a $m' = m + e$ où e est l'erreur de transmission, le problème du décodage est de retrouver m donc a à partir de m'. Schématiquement, ce processus se décrit ainsi.



e est la différence symbole à symbole entre m et m'. On appelle poids d'un mot m et on note $w(m)$ le nombre de composantes non nulles de ce mot. $w(e)$ est donc le nombre de composantes différentes de m et m'. A ce poids on associe la distance de Hamming sur $\mathbb{F}_q^n$:

$$d(m,m') = w(m - m') \text{ dans le cas linéaire.}$$

Nous pouvons immédiatement conclure que si $m' \in C$ et $m' \neq m$ alors on ne peut détecter et à fortiori corriger une quelconque erreur.

La stratégie naturelle est donc de choisir m de façon à minimiser $w(m - m')$.

C'est cela que doit accomplir l'algorithme de décodage.

L'objet de ce mémoire est l'étude du problème de décodage des codes linéaires qui consiste à déterminer le message original envoyé via un canal de transmission à partir du message reçu, dans le cas ou le nombre d'erreurs ne dépasse pas la capacité de la correction du code en utilisant les propriétés structurelles des codes linéaires. Dans ce mémoire on étudie quatre méthodes de décodage différentes. On donne tout d'abord le principe général pour chaque méthode de décodage puis ensuite on décrit en détails son algorithme.

Les méthodes utilisées sont:

1- le tableau standard défini à partir de la partition de $\mathbb{F}_q^n$ par une relation d'équivalence appropriée.

2-le syndrôme défini à partir de la matrice de contrôle du code.

3-la méthode de Meggitt définie à partir de la cyclicité du code.

4 -la méthode de décodage par piégeage d'erreurs qui est une méthode de Meggitt améliorée. On terminera cette étude par une généralisation de la méthode de Meggitt au cas non binaire.

Déroulement de la mémoire

Dans ce mémoire, on s'intéresse à l'étude de quelques techniques de décodage des codes linéaires.

Le premier chapitre est un chapitre d'introduction où nous présentons les notions et les propriétés fondamentales nécessaires pour la réalisation de ce travail tels que: ensemble quotient, anneaux de polynômes, corps finis et espaces vectoriels. Les notions citées dans ce chapitre représentent l'outil mathématique utilisé pour l'étude des codes correcteurs d'erreurs.

Le deuxième chapitre regroupe les définitions et les propriétés fondamentales des codes correcteurs d'erreurs.

Le troisième chapitre est consacré à l'étude de décodage des codes linéaires, nous étudions les définitions et les propriétés de décodages, puis on va présenter deux méthodes de décodage pour les codes linéaires.

Enfin, dans le quatrième chapitre, on va étudier deux méthodes de décodage des codes cycliques (la méthode de Meggitt; La méthode de décodage par piégeage d'erreurs), et on terminera par l'étude de la méthode de Meggitt dans le cas non binaire. Nous achevons notre travail par une conclusion.

Chapitre I

Définitions et propriétés élémentaires

Dans ce chapitre, on rappelle les notions de base dont on aura besoin par la suite, ensemble quotient, anneaux de polynômes, corps finis et espaces vectoriels.

1. Ensemble quotient

Définition 1

On appelle relation d équivalence R *sur un ensemble* E *toute relation binaire réflexive, symétrique et transitive. On note* xRy *le fait que les éléments* x, y *sont en relation par* R.

1.1. Classes d'équivalence modulo R. Ensemble quotient

Soit R une relation d'équivalence définie sur E. Deux éléments x et y de E sont équivalents modulo R lorsque xRy. La classe de l'élément $x \in E$ est notée par $\overline{x}$ et on a: $\overline{x} = \{y \in E, yRx\}$.

L'ensemble quotient de E modulo R est l'ensemble des classes d'équivalences, noté E/R.

Corollaire 1

Les classes d'équivalents modulo R *forment une partition de* E

i.e E/R *définie une partition de* E.

2. Anneaux de polynômes

Définition 2

Soit A un anneau commutatif unitaire, toute suite d'élément de A n'ayant qu'un nombre fini de termes non nuls est dite polynômes à coefficients dans A.

L'ensemble des polynômes sur A est noté $A[x]$.

Si $P=(a_0,a_1,.....a_n,0,0,.....)\in A[x]$ on notera $P=(a_0,a_1,.....,a_n)$.

Si $a_n\neq 0$ on appelle n le degré de P $(n=\deg P)$, Si $a_n=1,$ on dit que P est unitaire.

On pose $\deg(0,0,0,.....)=-\infty.$

Les polynômes de degré égal à 0 sont les constantes

Dans $A[x]$ on définit l'addition et la multiplication comme suit :

$$(a_0,a_1,.....)+(b_0,b_1,.....)=(a_0+b_0,a_1+b_1,.....)$$

$$(a_0,a_1,.....)(b_0,b_1,.....)=(c_0,c_1,.....) \text{ ou` } c_K=\sum_{i=0}^{k}a_ib_{k-i}.$$

Notons que si A est intègre on a:

$$\deg PQ=\deg P+\deg Q.$$

Muni des deux opération $+$ et $\bullet$ définies ci-dessus, $A[x]$ est un anneau commutatif avec unité $(1,0,0,.....)$.

$(A[x],+,\bullet)$ Est appelé l'anneau des polynômes sur A.

Dans $A[x]$, on définit $x=(0,1,0,.....), x^2=(0,0,1,0.....)$ ….avec

$x^0=(1,0,0,.....)$.

Ce qui permet d'écrire toute polynôme p de degré n comme suit:

$$p(x)=a_0+a_1x+.....+a_nx^n.$$

2.1. Division euclidienne des polynômes

Soient U et V deux polynômes de $A[x]$ ($A[x]$ l'anneau des polynômes à coefficients dans A). Supposons que le coefficient dominant de V soit inversible dans A. Il existe alors deux polynôme Q et R, uniquement déterminés, telque

$U = VQ + R$ avec $\deg(R) < \deg(V)$.

Définition 3

Soit $p(x), q(x) \in A[x]$, *on dit que* $p(x)$ *divise* $q(x)$ *et on note* $p(x) \mid q(x)$ *si* $q(x) = p(x)r(x)$ *avec* $r(x) \in A[x]$ *et* $\deg p(x) < \deg q(x)$, $p(x)$ *est alors appelé un diviseur propre de* $q(x)$.

Définition 4

Un polynôme $q(x)$ *de deg>1 qui n'a pas de diviseurs propres est appelé un polynômes irréductible.*

2.2. Idéaux de $\mathbb{F}[x]$

Soit $\mathbb{F}[x]$ l'anneau des polynômes sur un corps $\mathbb{F}$.

Définition 5

On appelle idéal de $\mathbb{F}[x]$ *toute partie non vide* I *de* $\mathbb{F}[x]$ *tel que:*

$\rightarrow I$ *est stable par* $+$

$\rightarrow \forall P \in I$ et $\forall Q \in \mathbb{F}[x]$, $PQ \in I$.

Exemple 1

$\{0\}$ *et* $\mathbb{F}[x]$ sont des idéaux triviaux dans l'anneau $\mathbb{F}[x]$.

Définition 6

Soit P *un polynôme de* $\mathbb{F}[x]$. *On défini l'idéal engendré par* P, *noté* (P) *par:*

$$(P)=\{PQ, Q \in \mathbb{F}[x]\}.$$

C'est donc l'ensemble des polynômes multiples de P.

2.3. Définitions et théorèmes

Définition 7

Un idéal engendré par un seul polynôme P, *dit de type* (P), *est appelé idéal principal.*

Théorème 1

Tout idéal de $\mathbb{F}[x]$*est principal. On dit donc que l'anneau* $\mathbb{F}[x]$ *est principal*.

Définition 8

Soit I *un idéal de* $\mathbb{F}[x]$, *on appelle générateurs de* I *les polynômes* w *tel que:*

$$I=(w).$$

Propriété 1

Les générateurs w *se déduisent les uns des autres par multiplication par une constante non nulle* $\lambda \in \mathbb{F}^*$.

De plus, si $I \neq \{0\}$, *alors les générateurs ont tous les mêmes:*

$$\deg(w)=\min\{\deg(P), P \in I-\{0\}\}.$$

Propriété 2

De la propriété précédente, on déduit que si:

$\rightarrow w \in I$

$\rightarrow \deg(w) = \min\{\deg(P), P \in I - \{0\}\}$ *Alors:*

$I = (w)$

Définition 9

L'unique générateur unitaire d'un idéal I non nul est appelé polynôme minimal de l'idéal I.

3. Corps finis

Nous rappelons dans cette section, des définitions et des propriétés liées au corps finis.

Définition 10

Un corps de q éléments est dit un corps fini de cardinal q et on le note $\mathbb{F}_q$.

Exemple 2

Pour p un nombre premier, $\mathbb{F}_p \simeq \mathbb{Z}/p\mathbb{Z}$ est un corps fini de cardinal p.

3.1. Caractéristique d'un corps fini

Soit $(\mathbb{F}_q, +, \cdot)$ un corps fini. La caractéristique de $\mathbb{F}_q$, notée car $(\mathbb{F}_q)$, est l'ordre additif de 1.

On a donc

$\text{car}(\mathbb{F}_q) = \inf\{n \in \mathbb{N}^*, n \cdot 1 = 0\}$.

Théorème 2

Soit $\mathbb{F}_q$ un corps fini. Alors:

1) La caractéristique de $\mathbb{F}_q$ est un nombre premier p.

2) $\mathbb{F}_q$ est un espace vectoriel de dimension finie n sur $\mathbb{F}_p$ et on a : $q = p^n$.

Théorème 3

Soit $\mathbb{F}_q$ un corps fini de cardinal q.

Le groupe multiplicatif $\left(\mathbb{F}_q^*, \cdot\right)$ est cyclique d'ordre $q-1$.

Théorème 4

Soit $\mathbb{F}_q$ un corps fini de cardinal q.

Pour tout $x \in \mathbb{F}_q^*$ on a: $x^{q-1} = 1,$ et pour tout $x \in \mathbb{F}_q$ on a : $x^q = x$.

Preuve :

D'après le théorème (3) l'ordre de $\mathbb{F}_q^*$ est $q-1$ donc pour tout $x \in \mathbb{F}_q^*$ on a: $x^{q-1} = 1$ avec 1 l'élément neutre pour la multiplication, et par conséquent pour tout $x \in \mathbb{F}_q$ on a : $x^q = x$.

Il en résulte du théorème (4), qu'un corps $\mathbb{F}_q$ a q éléments est l'ensemble des racines du polynôme $x^q - x$.

Définition 11

Un élément générateur du groupe cyclique $\mathbb{F}_q$ d'un corps fini $\mathbb{F}_q$ est appelé un élément primitif de $\mathbb{F}_q$.

Théorème 5

Soit α un élément primitif d'un corps fini $\mathbb{F}_q$ alors:

$$\mathbb{F}_q = \{0, 1, \alpha, \alpha^2, \ldots\ldots\ldots\ldots \alpha^{q-2}\}.$$

Avec : $\alpha^{q-1} = 1$ de plus α^k est primitif si et seulement si k et $q-1$ sont premiers entre eux.

3.2. Construction d'un corps fini

Pour déterminer les éléments d'un corps fini $\mathbb{F}_p$ on peut suivre une des méthodes suivantes:

1- Soit utiliser l'anneau quotients $\mathbb{F}_q[x] \big/ (f(x))$ où $f(x)$ est un polynôme irréductible sur $\mathbb{F}_q$.

2- Soit en utilisant le fait que $\mathbb{F}_q^*$ est un groupe cyclique où chaque élément est une puissance d'un même élément générateur α appartient à $\mathbb{F}_q^*$.

Théorème 6

Soit $\mathbb{F}_q$ un corps et $f(x) \in \mathbb{F}_q[x]$, alors $\mathbb{F}_q[x] \big/ (f(x))$, est un corps si et seulement si $f(x)$ est irréductible sur $\mathbb{F}_q$.

La preuve de ce théorème montre non seulement que $\mathbb{F}_q[x] \big/ (f(x))$ est un corps, mais nous donne aussi la façon d'obtenir ses éléments.

Preuve:

On note par I l'idéal principal $(f(x))$, supposons que $f(x)$ est réductible sur $\mathbb{F}_q$, c.à .d $f(x) = a(x)b(x)$ tel que $a(x), b(x)$, ont des degrés inférieur au degré de $f(x)$.

On montre dans ce cas que $\mathbb{F}_q[x]/I$ n'est pas un corps. Le degré de tout polynôme non nul de I doit être supérieur où égal au degré de $f(x)$, donc $\mathrm{a}(x) \notin I$, $b(x) \notin I$, par conséquent $I+\mathrm{a}(x)$, $I+b(x)$ sont des éléments non nuls de $\mathbb{F}_q[x]/I$.

Mais on a:

$$(I+\mathrm{a}(x))(I+b(x)) = I+\mathrm{a}(x)b(x) = I+f(x) = I$$

ce qui montre que $\mathbb{F}_q[x]/I$ ne peut être un corps donc $f(x)$ doit être irréductible sur $\mathbb{F}_q$.

Inversement, supposons maintenant que $f(x)$ est irréductible sur $\mathbb{F}_q$.

$\mathbb{F}_q[x]/I$ est un anneau commutatif d'élément unité $I+e$(où e est l'unité de $\mathbb{F}_q$), il suffit donc de démontrer que tout élément non nul de $\mathbb{F}_q[x]/I$ admet un inverse dans $\mathbb{F}_q[x]/I$.

Soit $I+p(x) \in \mathbb{F}_q[x]$ différent de zéro (c. à.d différent de I), donc $p(x) \notin I$, ce qui montre que $p(x)$ n'est pas multiple de $f(x)$, comme $f(x)$ est irréductible, alors $f(x)$ et $p(x)$ sont premiers entre eux et donc d'après le théorème de bézout

$\exists, u(x), v(x) \in \mathbb{F}_q[x]/I$ tel que:

$$f(x)\,u(x)+p(x)v(x)=e$$

alors on a:

$$e-p(x)v(x) = f(x)u(x) \in I$$

et par conséquent

$$I+e = I+p(x)v(x) = (I+p(x))(I+v(x)) = e$$

c.à..d $I+v(x)$ est l'élément inverse de $I+p(x)$

Théorème 7

Soit $\mathbb{F}_q$ *un corps fini et* $f(x) \in \mathbb{F}_q[x]$, *avec degré* $f(x) = n$. *Alors:*

$$\mathbb{F}_q[x] \Big/ (f(x)) = \left\{ a_0 + a_1 x + \ldots\ldots\ldots + a_{n-1} x^{n-1} : a_i \in \mathbb{F}_q \right\}$$

est un espace vectoriel sur F_q *de dimension* n *de base*

$\left\{1, \alpha, \ldots\ldots\ldots\ldots\ldots \alpha^{n-1}\right\}$, *avec* $\alpha = [x] + (f(x))$ *où* $\bar{\alpha} = 0$.

Le théorème précédent, nous montre la façon de déterminer les éléments d'un corps fini.

On sait que le corps fini $\mathbb{F}_{p^n}$ est un espace vectoriel de dimension n sur $\mathbb{F}_p$, de plus, $\mathbb{F}_{p^n}$ est une extension simple, c à d $\mathbb{F}_{p^n} = \mathbb{F}_p(\alpha)$, est tous les $(n+1)$ éléments de $\mathbb{F}_{p^n}$ seront linéairement dépendants.

Donc ils existent $a_0, a_1, \ldots\ldots\ldots, a_n \in \mathbb{F}_p$ tel que $a_0 + a_1\alpha + \ldots\ldots\ldots + a_n\alpha^n = 0$.

Ce qui montre que α est une racine du polynôme $a_0 + a_1 x + \ldots\ldots\ldots + a_n x^n \in \mathbb{F}_p[x]$.

Soit $f(x)$ le polynôme minimal de α (irréductible unitaire).

$$F_{p^n} = F_p(\alpha) \cong \mathbb{F}_p[x] \Big/ (f(x)).$$

On détermine un polynôme irréductible unitaire de degré n sur $\mathbb{F}_p$ et on construit $\mathbb{F}_p[x] \Big/ (f(x))$.

Exemple 3

Dans $\mathbb{F}_3$, le polynôme $g(x) = x^2 + x + 2$ est irréductible, on détermine les éléments de $\mathbb{F}_{3^2}$ en le regardent comme extension obtenue par adjonction à $\mathbb{F}_3$ d'une racine de $g(x)$, ainsi $\mathbb{F}_{3^2} = \mathbb{F}_3[x] \Big/ (g(x))$, soit α une racine de $g(x)$, alors $\{1, \alpha\}$ est une base de $\mathbb{F}_{3^2}$

$$\mathbb{F}_3[x]/(g(x)) = \{a_0 + a_1\alpha / a_0, a_1 \in F_3\} = \{0,1,2,2\alpha,1+\alpha,2+\alpha,1+2\alpha,2+2\alpha\}.$$

Cette dernière écriture s'appelle la représentation polynomial de ce corps, tout polynôme du corps $\mathbb{F}_3[x]/(g(x))$, peut être réduit modulo $g(\alpha)$ en utilisant le fait que: $g(\alpha)=0.$

Dans $\mathbb{F}_3$, c'est-à-dire que : $\alpha^2 = 2\alpha + 1,$ et on aura :

$$F_{3^2} = \mathbb{F}_3[x]/(g(x)) = \{0,1,\alpha,\alpha,^2\alpha^3,\alpha^4,\alpha^5,\alpha^6,\alpha^7\}.$$

(Représentation en puissance de α)

3.3. Calculs résiduels sur les polynômes

Soi$f(x) \in \mathbb{F}_q[x]$, $f(x)$ fixé ($\mathbb{F}_q$ corps) et soit $g(x)$, $h(x) \in \mathbb{F}_q[x]$, les deux polynôme $g(x)$ et $h(x)$ sont dits congrus modulo $f(x)$ en notation

$g(x) \equiv h(x)[f(x)]$ si $g(x) - h(x)$ est divisible par $f(x)$.

$\forall a(x) \in \mathbb{F}_q[x]$ on a : $a(x) \equiv r(x)[f(x)]$

Avec $a(x) = q(x)f(x) + r(x)$ et $r(x) = 0$ ou deg $r(x) <$ deg $f(x)$.

* Si $a(x)$, $h(x) \in \mathbb{F}_q[x]/(f(x)) = \mathbb{F}_q[x]/I$, $I = (f(x))$

1- $a(x) + h(x) \in \mathbb{F}_q[x]/I$

2- le produit $a(x)\,h(x)$ est calculé modulo $f(x)$.

Exemple 4

Dans $\mathbb{F}_2[x]/(x^2+x+1)$ On a:

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

Addition et multiplication dans $\mathbb{F}_2[x]/(x^2+x+1)$.

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

4. Espaces vectoriels

On dit que E est un espace vectoriel sur un corps $\mathbb{k}$ si et seulement si, pour des éléments u, v de E, et w de E on a:

1. $(u+v)+w = u+(v+w)$
2. $\exists 0 : u+0 = 0+u = u$
3. $\forall u \ \exists(-u) : u-u = 0$
4. $u+v = v+u$
5. $\forall c \in \mathbb{k}, c(u+v) = c\cdot u + c\cdot v$
6. $\forall a,b \in \mathbb{k}, (a+b)u = a\cdot u + b\cdot u$
7. $\forall a,b \in \mathbb{k}, (a\cdot b)u = a\cdot(b\cdot u)$
8. $1\cdot u = u$

Soit E est un espace vectoriel sur un corps $\mathbb{k}$. Une partie $\mathbb{F} \neq \phi$ est un sous-espace vectoriel de E si et seulement si:

$$\begin{cases} x+y \in \mathbb{F} & \forall x, y \in \mathbb{F} \\ \lambda x \in \mathbb{F} & \forall \lambda \in \mathbb{k}, \forall x \in \mathbb{F} \end{cases}$$

Si A est un sous –ensemble de E, alors le sous-espace engendré par A est l'ensemble de toute les combinaisons linéaires d'éléments de A.

Une *famille génératrice* de E est un sous-ensemble $G \subseteq E$ tel que le sous-espace engendré par G et E. Ou, de manière équivalente, que tout vecteur de E est une combinaison linéaire d'élément de G. E est dit de dimension finie s'il contient une famille génératrice finie.

Une *famille libre de* E est un sous-ensemble $L \subseteq E$ tel qu'aucun élément$v \in L$ n'est une combinaison linéaire d'autres éléments de L. Ou, de manière équivalente,la seule combinaison linéaire d'éléments de L qui est nulle est celle dont tous les coefficients sont nuls.

Une base de E est une famille d'éléments de E qui est libre et génératrice.
Si $(b_i)_{i\in I}$ est une base de E, $\exists!(\lambda_i)_{i\in I} \in \mathbb{k} : x = \sum_{i\in I} \lambda_i b_i$.
Si la dimension de E est finie alors chaque base de E est finie et toutes les bases ont les mêmes nombre d'élément. Ce nombre est la dimension de l'espace. Si la dimension de E est n, alors:

-Si $E = \mathbb{k}^n$, alors la famille $(e_i)_{i\in I}$ avec $I = \{1,2,3,.....,n\}$ et $e_i = (0,0,....1,0,.....,0)$ avec 1à la i^e position est la base canonique de $\mathbb{k}^n$.

Soient E et V deux espaces vectoriels sur un corps $\mathbb{k}$. Une *application linéaire* f de E dans V vérifie :

$\forall x \in E, \forall y \in E \quad f(x+y) = f(x) + f(y)$
$\forall \lambda \in \mathbb{k}, \forall x \in E \quad f(\lambda x) = \lambda \cdot f(x)$

Soit $f : E \to V$ une application linéaire et soit A la matrice associe à f.
$kerf = \ker A = \{e \in E / f(e) = 0\}$ est le noyau de f ou l'espace nul de la matrice A.

$f(E) = AE = \{f(e)/e \in E\} = \{Ae/e \in E\}$ est l'espace image de f (ou de la matrice A).

L'espace nul d'une matrice A est l'ensemble, noté $NulA$, de toutes les solutions de l'équation homogène $AX = 0$.

Chapitre II

Codes correcteurs d'erreurs

1. Introduction

Dans ce chapitre on va présenter tout d'abord les définitions et le principe général des codes correcteurs d'erreurs, puis on passera à un cas particulier des codes qui est les codes linéaires, qui ont une structure algébrique riches, cette structure simplifie l'étude des codes linéaires, un code linéaire peut être décrit par sa matrice génératrice ou bien sa matrice de contrôle, la distance minimale est déterminée par le poids de Hamming, etc.. et plus précisément on va se concentrer sur les codes cycliques qui possèdent une riche structure mathématique, ainsi on peut effectuer, l'opération décalage qui consiste a obtenir un mot de code à partir d'un autre par décalage de symboles.
Tout les problème de la théorie des codes correcteurs d'erreurs consiste à construire des codes, qui détectent et corrigent le plus possible d'erreurs.

2. La nécessité du codage

Afin de préciser l'importance du codage de l'information on donne l'exemple suivant:
Supposons que nous ayons un code binaire de longueur 11 ayant $2^{11} = 2048$ mots.
Comme notre code contient toutes les chaînes de longueur 11, il ne détecte aucune erreur.
Supposons que la probabilité qu'un bit soit transmis sans erreur soit de $1-\frac{1}{10^8}$ et supposons que la transmission se fasse à un taux de 10^7 bits par seconde. Alors la probabilité qu'un mot soit transmis de façon incorrecte est

$$1-P\left(0 \text{ erreur}\right)=1-\left(1-\frac{1}{10^8}\right)^{11} \approx \frac{11}{10^8}.$$

Cela implique qu'il y aura approximativement

$$\frac{11}{10^8}\times\frac{10^7}{11}=0.1$$

mots par seconde qui seront transmis incorrectement sans être détecté. Cela représente donc 8640 mots incorrects par jour. Supposons maintenant que l'on ajoute à chaque mot un bit dit *bit de parité* (le bit de parité vaut 1 si le nombre de bits égaux à1 est impair, sinon il vaut 0). On doit avoir au moins 2 erreurs dans notre mot afin que celui-ci soit mal interprété. Cette probabilité est

$$1-P(0\text{ erreur})-P(1\text{ erreur})=1-\left(1-\frac{1}{10^8}\right)^{12}-\binom{12}{1}\left(1-\frac{1}{10^8}\right)^{11}\left(\frac{1}{10^8}\right)^{1}\approx\frac{66}{10^{16}}$$

Cela implique qu'il y aura approximativement

$$\frac{66}{10^{16}}\times\frac{10^7}{12}=5.5\times10^{-9}$$

mots par seconde qui seront transmis incorrectement sans être détecté. Cela représente donc une seule erreur à tous les 2000 jours !

3. Les codes

Soit Q un ensemble fini à q éléments, soient k et n deux entiers naturels non nuls avec $k \le n$.

Un $k-$uplets a de Q^k sera appelé un message ou un mot de longueur k ,et sera noté soit : $a = (a_1, a_2, ..., a_k)$ soit: $a = a_1 a_2 .....a_k$ sous forme concaténée.

L'ensemble des messages sera une partie E de Q^k, et l'on introduit une application injective.

$$f : E \to Q^n$$

$$a = (a_1, a_2 ....., a_k) \to c = (c_1, c_2, ..., c_n)$$

appelée application de codage ou encodeur, le message a de E sera encodé par le mot $f(a) = c = (c_1, c_2, ...c_n) \in Q^n$.

Notons $C = f(E)$ l'image de E par f.

C est appelé code de longueur n sur Q, et les éléments de C s'appellent des mots du code, le cardinal du code est par définition celui de C.

Définition 1

On définit le taux d'information du code C de longueur n le rapport $R = k/n$.

Exemple 1

$C = \{0000, 1011, 0101, 1110\}$ est un code de longueur 4 sur $Q = \{0,1\}$

3.1. Distance de Hamming

Pour compter le nombre d'erreurs, on introduit la distance de hamming sur Q^n. Elle permet de mesurer le degré de différence entre deux mots x et y de Q^n.

Dans la suite de ce document, on prendra Q un corps fini $\mathbb{F}_q$.

La distance de Hamming entre deux mots

$x = (x_1, x_2, ..., x_n)$ *et* $y = (y_1, y_2, ..., y_n)$ *de* $\mathbb{F}_q^n$ que l'on notera $d(x, y)$ est le nombre d'indices i tel que $x_i \neq y_i$

$$d(x,y) = card\{i : x_i \neq y_i\}.$$

Le poids d'un mots x de $\mathbb{F}_q^n$, noté $w(x)$, est égale au nombre de ses composante non nulles.

Exemple 2

$d(00011,00100) = 3$ $w(00011) = 2$

$d(11200,22001) = 4$ $w(11200) = 3$

Il faut remarquer que la distance de Hamming, est une vraie distance au sens métrique

Propriétés 1

Soit d la distance de Hamming .

Alors pour x, y et z , éléments de $\mathbb{F}_q^n$ on a :

1- $d(x,y) \in \mathbf{R}^+$

2- $d(x,y) = 0 \Leftrightarrow x = y$

3- $d(x,y) = d(y,x)$

4- $d(x,y) \leq d(x,z) + d(z,y)$

Définition 2

La distance minimale d d'un code C est définie par

$$d(C) = min\{d(x,y)/x, y \in C \text{ et } x \neq y\}.$$

un code C de longueur n, de cardinal M et de distance minimale d est appelé code (n, M, d), les nombre, n, M et d sont les paramètres du code.

La distance de Hamming nous permet de définir la notion de boule et de volume sur l'espace vectoriel $\mathbb{F}_q^n$.

L'ensemble

$$S_r(x) = \{y \in \mathbb{F}_q^n : d(x,y) \leq r\}$$

est appelé la sphère de rayon r et de centre x (r est un réel positif).

Et le volume $V_q(x,r)$ de la sphère $S_r(x)$ est le nombre d'éléments de $S_r(x)$

$$V_q(x,r) = \sum_{k=0}^{[r]} C_n^k (q-1)^k .$$

4. Codes linéaires

Dans notre étude des codes contenues dans $\mathbb{F}_q^n$, nous allons nous concentrer sur les codes linéaires, c'est-à-dire ceux qui ont une structure d'espace vectoriels. Ce qui permet d'utiliser les outils de l'algèbre linéaire.

Définition 3

Un code linéaire de dimension k et de longueur n est un sous – espace vectoriel de dimension k de $\mathbb{F}_q^n$.

Si la distance minimal de C est d, on dit que C est un code $[n,k,d]$ (ou simplement $[n,k]$), si $q=2$, on dit que C est un code binaire.

Pour un code linéaire C. On retrouve la distance de Hamming par la formule

$$d(x,y) = w(x-y).$$

Et la distance minimale du code C par :

$$d = min\{w(x)/x \in C \; et \; x \neq 0\}.$$

Exemple 3

Soit C le code linéaire de taille $(4,2)$

$C = \{0000,1011,0101,1110\}$

$d = min\{w(x)/x \in C \; et \; x \neq 0\}$

$= 2.$

Propriété 2

Si C est un code linéaire (n,k) sur $\mathbb{F}_q$, alors le nombre de mots de C est q^k.

Preuve:

Il suffit de remarquer que C est isomorphe à $\mathbb{F}_q^k$ en tant que $\mathbb{F}_q$ - espace vectoriel.

4.1. Encodage d'un Code linéaire

Etant donné qu'un code linéaire $C(n,k)$ est un sous espace vectoriel de dimension finie k, le code $C(n,k)$ peut être défini au moyen d'une matrice G à k lignes et n colonnes, appelée matrice génératrice, dont les lignes forment une base de C

4.1.1. Matrice génératrice

L'application linéaire $f : \mathbb{F}_q^k \to \mathbb{F}_q^n$ possède une matrice que l'on notera G^t dans les bases canoniques. Ici l'écriture G^t désigne la transposée de la matrice G qui possède k lignes et n colonnes, et tout mot de C s'écrira sous la forme :

$$c = f(x) = xG$$

où $c = (c_1, c_2, ..., c_n) \in \mathbb{F}_q^n$ et $x = (x_1, x_2, ..., x_k) \in \mathbb{F}_q^k$ sont des vecteurs lignes.

Soit C un code linéaire, une matrice génératrice de C, est une matrice génératrice G est donc de type $k \times n$ et de rang k telle que :

$$C = \{c \in \mathbb{F}_q^n / \exists x \in \mathbb{F}_q^k : c = xG\}.$$

Exemple 4

Soit $\{1100, 0111, 1010\}$ une base d'un code linéaire $C \subset \mathbb{F}_2^4$

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

et l'application associée est définie par :

$$\varphi_G : \mathbb{F}_2^3 \to \mathbb{F}_2^4$$

$$(x_1, x_2, x_3) \to (x_1, x_2, x_3) \begin{pmatrix} 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 1 \\ 1 & 0 & 1 & 0 \end{pmatrix}$$

$$\varphi_G(x_1, x_2, x_3) = (x_1 + x_3, x_1 + x_2, x_2 + x_3, x_2).$$

D'où : $C = \mathrm{Im}\varphi_G = \{0000, 1100, 0111, 1010, 1011, 0110, 1101, 0001\}$.

4.1.2. Matrice de contrôle

On peut aussi obtenir le code C, comme étant le noyau d'une application linéaire dont les lignes de la matrice H associée forment une base de l'espace nul de la matrice G. Si G est de type $k \times n$ alors H est de type $(n-k) \times k$ de rang $H = n-k$ on obtient donc l'application linéaire.

$$\varphi_H : \mathbb{F}_q^n \to \mathbb{F}_q^{n-k}$$

$$x \to Hx^t$$

et le code C est l'ensemble:

$$C = Ker\varphi_H = \{x \in \mathbb{F}_q^n : \varphi_H(x) = Hx^t = 0\}.$$

Exemple 5

Pour obtenir le code C (exemple précédente) a partir de la matrice de contrôle H on calcul tout d'abord l'espace nul de G.

$y \in \mathbb{F}_2^4$ Alors y$\in$ l'espace nul de G ssi $Gy^t = 0$

$$G\,y^t = 0 \Leftrightarrow \begin{pmatrix} 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 1 \\ 1 & 0 & 1 & 0 \end{pmatrix} \begin{pmatrix} y_1 \\ y_2 \\ y_3 \\ y_4 \end{pmatrix} = 0 \Leftrightarrow \begin{cases} y_1 + y_2 = 0 \\ y_2 + y_3 + y_4 = 0 \\ y_1 + y_3 = 0 \end{cases}$$

Les solutions du système sont $\{0000, 1110\}$.

Donc la base est $\{1110\}$ et la matrice $H = [1110]$

Soit $\varphi_H : \mathbb{F}_2^4 \to \mathbb{F}_2$

$$(x_1, x_2, x_3, x_4) \to (1110) \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{pmatrix}$$

$$\varphi_H(x_1, x_2, x_3, x_4) = x_1 + x_2 + x_3$$

et par conséquent, on a

$C = \ker \varphi_H = \{x \in \mathbb{F}_2^4 : x_1 + x_2 + x_3 = 0\}$ on a par exemple

$1111 \notin C$ Car $1+1+1 \neq 0$, $0111 \in C$ car $0+1+1=0$.

4.2. Code dual

La contrainte de la linéarité sur le code donne naturellement naissance à la notion de code dual d'un code linéaire. Puisque C est un espace vectoriel, on peut considérer l'ensemble des formes qui s'annulent sur C. Cet ensemble est un espace vectoriel que l'on appelle code dual de C.

Définition 4

Soit C un $[n,k,d]$-code linéaire .Soit $\langle .,. \rangle$ le produit scalaire euclidien usuel : $\langle c,v \rangle = \sum_{i=1}^{n} c_i v_i$. Le code dual, note $C^{\perp}$, est donc un code linéaire de la même longueur. Sa dimension est $n-k$.

$$C^{\perp} = \{ v \in \mathbb{F}_q^n : \forall c \in C : \langle c,v \rangle = 0 \}.$$

Il découle directement des définitions que si H est une matrice génératrice de $C^{\perp}$, elle est communément appelée matrice de contrôle du code C. De même, une matrice génératrice de C est une matrice de contrôle de $C^{\perp}$.

Exemple 6

Soit le code $C = \{000,011,101,110\}$ de longueur 3 sur $\mathbb{F}_2$ le dual $C^{\perp}$ de C est

$C^{\perp} = \{ y \in \mathbb{F}_2^3,\ y = abc,\ \forall c \in C,\ \langle c,y \rangle = 0 \}$

donc $y = 111$ ou $y = 000$ d'ou $C = \{111,000\}$.

4.3. Codes systématiques

A chaque mot $x=(x_1,x_2.....x_k)$ du message on adjoint $n-k$ symboles $c_{k+1}.....c_n$ dépendant linéairement des x_i pour obtenir le mot de code $c=f(x)$.

Les symboles c_i sont appelés bits de contrôle et

$$c=(x_1,.....x_k,c_{k+1},.....c_n)=(x_1,.....x_k)(I_k/A)$$

où (I_k/A) désigne la matrice $k\times n$ obtenue en écrivant cote à cote la matrice identité I_k de taille k et une matrice quelconque A.

Définition 5

Un code C sera dit systématique s'il possède une matrice génératrice de la forme

$$G=(I_k/A).$$

Dans ce cas

$c\in C$ si, et seulement si $c=xG=x(I_k/A)$ et $(-{}^tA/I_{n-k})\begin{pmatrix}I_k\\ {}^tA\end{pmatrix}^t x=-{}^tA^tx+{}^tA^tx=0$

C est inclus dans $\ker H$ où $H=(-{}^tA/I_{n-k})$.

Comme H est de rang $n-k$, les sous – espaces C et $\ker H$ ont même dimension k et donc $C=\ker H$.

La matrice $H=(-{}^tA/I_{n-k})$ est par conséquent une matrice de contrôle de C.

Exemple 7

L'application $f(x_1,x_2,x_3)=(x_1,x_2,x_3,x_1+x_2+x_3)$ définit un code systématique C de type $(4,3)$ sur $\mathbb{F}_2$ l'écriture.

$$(c_1,c_2,c_3,c_4)=(x_1,x_2,x_3)\left(\begin{array}{ccc|c}1&0&0&1\\0&1&0&1\\0&0&1&1\end{array}\right).$$

Met en évidence une matrice génératrice de C d'où l'on peut déduire de contrôle $H=(1\ 1\ 1)$.

5. Codes linéaires cycliques

Dans la famille des codes linéaires, il existe une classe très importante, celle des codes en blocs linéaires et cyclique, dans un code cyclique toute opération de décalage cyclique appliquée à un mot de code fournit un autre mot de code.

La représentation vectorielle et matricielle des codes linéaires est remplacée par une représentation polynomial, l'effet de décaler un mot de code d'une case revient a multiplier le polynôme par l'indéterminée x et prendre le reste modulo $x^n - 1$.

Ainsi, il suffit d'additionner où de multiplier des polynômes en restant dans l'anneau quotient modulo $x^n - 1$, i, e l'anneau $\mathbb{F}_q[x] \big/ \left(x^n - 1\right)$.

Définition 6

Un code linéaire C de longueur n est cyclique s'il vérifie la propriété suivante:

$c_0 \ldots\ldots c_{n-1} \in C \Rightarrow c_{n-1} c_0 \ldots\ldots c_{n-2} \in C$.

Rappelons que $\mathbb{F}_q[x]$ et $\mathbb{F}_q[x] \big/ \left(f(x)\right)$, sont des anneaux principaux, de plus, nous savons qu'il y a un isomorphisme entre $V_n\left(\mathbb{F}_q\right)$ et l'anneau des polynôme de degré inférieur à n sur $\mathbb{F}_q$ noté $\mathbb{F}_q^n[x] \cong \mathbb{F}_q[x] \big/ \left(f(x)\right)$ où $f(x)$ est un polynôme de degré n. Cet isomorphisme nous permet de considérer indifféremment un mot de C comme un vecteur de $V_n\left(\mathbb{F}_q\right)$ ou comme un polynôme de $\mathbb{F}_q^n[x]$ de la façons suivante :

$V = a_0 a_1 \ldots\ldots a_{n-1} \in V_n\left(\mathbb{F}_q\right) \Rightarrow V(x) = a_0 + a_1 x + \ldots\ldots a_{n-1} x^{n-1} \in \mathbb{F}_q^n[x]$

Lorsque $f(x) = x^n - 1$, on notera $\mathbb{F}_q[x] \big/ \left(x^n - 1\right)$ par R_n.

La représentation polynomial de $c = c_0 c_1 \ldots\ldots c_{n-1}$ est le polynôme

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

Exemple 8

1) le code linéaire

$$C = \{000, 011, 101, 110\}$$

est un code cyclique puisque $011 \in C \Rightarrow 101 \in C$, $101 \in C \Rightarrow 110 \in C$ et $110 \in C \Rightarrow 011 \in C$.

La représentation polynomiale de 011 est $x + x^2$.

2) Le code linéaire $C = \{01010, 10101, 11111, 00000\}$ n'est pas cyclique. Par exemple, en décalant 01010 de 1 vers la droite, on obtient 00101 qui n'est pas un mot code.

La représentation polynomial d'un code C est l'ensemble des représentations polynomiales de ses mots.

Effectuer un décalage sur un mot revient à multiplier par x sa représentation polynomial, modulo $x^n - 1$. Donc C est cyclique si et seulement si pour tout mot c de C, $xc(x)$ calculé modulo $x^n - 1$ est la représentation polynomiale d'un mot de C. En pratique, on confond les mots du code et leur représentation polynomiale.

Exemple

Le code $C = \{000, 011, 101, 110\}$ correspond aux polynômes 0, $x + x^2$, $1 + x^2$, $1 + x$ pris modulo $x^3 - 1$. Sa représentation polynomiale est donc

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

Théorème 1

Une partie $C \subset R_n$ est un code cyclique si et seulement si elle satisfait :

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

2) $a(x) \in C$ *et* $r(x) \in R_n \Rightarrow r(x)a(x) \in C$.

Preuve :

On suppose que C est un code cyclique dans R_n donc C est linéaire par conséquent (1) est vérifié.

Supposons maintenant que $a(x) \in C$ et

$$r(x) = r_0 + r_1 x + ...r_{n-1} x^{n-1}. \in R_n$$

Puisque la multiplication par x correspond à un cyclique, on a : $x\,\mathrm{a}(x) \in C$ et alors

$x(x\mathrm{a}(x)) = x^2\mathrm{a}(x) \in C$ et $r(x)\mathrm{a}(x) = r_0 + r_1 x\mathrm{a}(x) + ...r_{n-1}x^{n-1}\mathrm{a}(x)$ est également dans C, ainsi chaque sommant est dans C, (2) et donc vérifié.

Supposons maintenant (1) et (2) sont vérifiés, soit $r(x)$ un scalaire cette condition implique que C est linéaire prenons $r(x) = x$ dans (2) ce qui montre que C est cyclique.

Rappelons aussi que

$$(f(x)) = \{r(x) f(x) / r(x) \in R_n\}.$$

L'idéal engendre part $f(x)$.

Théorème 2

Pour tout $f(x) \in R_n$ l'ensemble $(f(x))$ est un code cyclique dit engendré par $f(x)$.

Preuve :

1) si $\mathrm{a}(\mathrm{x}) f(x) \in (f(x))$ *et* $b(x) f(x) \in (f(x))$

alors $\mathrm{a}(\mathrm{x}) f(x) + b(x) f(x) = (\mathrm{a}(\mathrm{x}) + b(x)) f(x) \in (f(x))$.

2) Si $\mathrm{a}(\mathrm{x}) f(x) \in (f(x))$ *et* $r(x) \in R_n$

alors $r(x)(\mathrm{a}(\mathrm{x}) f(x)) = (r(x)\mathrm{a}(x)) f(x) \in (f(x))$.

Exemple 9

Soit le code $C = (1 + x^2)$ dans $\mathbb{F}_2[x] \big/ (x^3 - 1)$

L'éléments de C sont $0, 1+x, 1+x^2, x+x^2$

donc : $C = \{000, 110, 101, 011\}$.

Théorème 3

Soit C un code cyclique dans R_n

1- il existe un polynôme unique (unitaire) $g(x)$ de degré minimal dans C.

2- $C = (g(x))$.

3- $g(x)$ est facteur de $x^n - 1$.

Preuve:

1) supposons $g(x)$ et $h(x)$ deux polynômes unitaire de degré minimales alors $g(x), h(x) \in C$ et ayant un degré minimal ceci conduit à une contradiction.

Si $g(x) \neq h(x)$ ainsi $g(x)h(x)$ est dans C et de degré inférieur à $d^\circ g(x)$.

2) Supposons $a(x) \in C$, par l'algorithme de division par $g(x)$, $a(x) = q(x)g(x) + r(x)$ où $d^\circ r(x) < d^\circ g(x)$ mais $r(x) = a(x) - q(x)g(x) \in C$ en utilisant des propriétés du code cyclique données au théorème (1) grâce à la minimalité de $d^\circ g(x)$. On doit avoir $r(x) = 0$ et ainsi $a(x) \in (g(x))$.

3) En appliquant l'algorithme de division :

$x^n - 1 = q(x)g(x) + r(x)$ où $d^\circ r(x) < d^\circ g(x)$

mais $r(x) \equiv -q(x)g(x) \pmod{x^n - 1}$ et ainsi $r(x) \in (g(x))$

Pour la minimalité de degré de $g(x)$, on doit avoir $r(x) = 0$, ce qui implique $g(x)$ est un facteur de $x^n - 1$.

Théorème 4

Soit C *code un cyclique de polynôme générateur* $g(x) = g_0 + g_1x + g_2x^2 + ...g_rx^r$ *avec* $d^\circ g(x) = r$.

Alors, $\dim C = k = n - r$ *et sa matrice génératrice est :*

$$G = \begin{bmatrix} g_0\ g_1\ .....g_r\ 0\ 0\ 0\ 0\ .....0 \\ 0\ g_0\ g_1\ .....g_r\ 0\ 0\ 0\ 0\ .....0 \\ . \\ . \\ . \\ 00\ 0\ .....0\ g_0\ g_1\ .....g_r \end{bmatrix} = \begin{bmatrix} g(x) \\ x\,(g(x)) \\ \\ \\ \\ x^{k-1}(g(x)) \end{bmatrix}$$

Preuve :

Les $n-r$ lignes de la matrice G sont nécessairement linéairement indépendantes. Ces $n-r$ lignes représentent les mots du code.

$g(x), xg(x), x^2g(x), ...x^{n-r-1}g(x).$

Il reste à montrer que chaque mot de code dans C s'exprime à l'aide de ceux-ci.

Le théorème (3) montre que si $a(x)$ est un mot de code *on a* $a(x) = q(x)g(x)$, pour un polynôme $q(x)$ et que ceci est une égalité de polynôme dans $\mathbb{F}_q[x]$, qui ne requiert aucune réduction modulo $x^n - 1$ ainsi $d^\circ a(x) < n$ il s'en suit que $d^\circ q(x) < n - r$

d'où : $q(x)g(x) = \left(q_0 + q_1x + ...q_{n-r-1}x^{n-r-1}\right)g(x)$

$$= q_0g(x) + q_1xg(x) + ... + q_{n-r-1}x^{n-r-1}g(x).$$

Laquelle est la combinaison linéaire désirée.

Exemple 10

Le code de Hamming de paramètre $(7,4,3)$ et de polynôme générateur $g(x) = 1 + x + x^3$ admet pour matrice génératrice :

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

Exemple 11

Le code de paramètre $(4,1,4)$ sur $\mathbb{F}_3$ et de polynôme générateur $g(x)=x^3+2x^2+x+2$ admet pour matrice génératrice :

$G=[2,1,2,1]$.

Les mots du code sont les combinaisons linéaires des lignes de la matrice G.

Soit $g(x)$ le polynôme générateur de degré r d'un code linéaire cyclique de paramètre $[n,n-r]$. Aussi, nous savons que $x^n-1=g(x)h(x)$ où $h(x)$ est un polynôme de degré $n-r$ le polynôme $h(x)$ est appelé le polynôme de contrôle du code ayant comme générateur $g(x)$.

Le théorème suivant montre comment à laide de $h(x)$ nous pouvons obtenir la matrice de contrôle.

Théorème 5

Soit $h(x)$ le polynôme de contrôle d'un code linéaire cyclique C dans R_n.

1- le code C peut être représenté par :

$C=\{p(x)\in R_n / p(x)h(x)=0\}$.

2- soit $h(x)=h_0+h_1x+.....h_{n-r}x^{n-r}$ alors une matrice de contrôle du code C est donnée par :

$$H=\begin{bmatrix}h_{n-r}.....h_0\ 0\ 0\ 0\ 0\ 0\ 0\\0\,h_{n-r}\ .....h_0\,0\ 0\ 0\ 0\ 0\\0\ 0\ h_{n-r}.....h_0\,0\ 0\ 0\ 0\\.....\\0\ 0\ 0\ 0\qquad h_{n-r}\ .....\ h_0\end{bmatrix}$$

Exemple 12

Considérons le code binaire $C(7,4)$ généré par $g(x)=x^3+x+1$ sa matrice génératrice est composée à partir du polynôme générateur

$g(x)=g_0+g_1x+g_2x^2+.....+g_kx^k$ est

$$G=\begin{pmatrix} g_0 & g_1 & g_2 & g_3 & 0 & 0 & 0 \\ 0 & g_0 & g_1 & g_2 & g_3 & 0 & 0 \\ 0 & 0 & g_0 & g_1 & g_2 & g_3 & 0 \\ 0 & 0 & 0 & g_0 & g_1 & g_2 & g_3 \end{pmatrix}$$

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

Calculer la matrice de contrôle à partir de la matrice génératrice n'est pas en général aisé, par contre, on peut facilement trouver le polynôme de contrôle $h(x)$, qui est tel que:

$$h(x).g(x)=0$$

et donc

$$h(x)=\frac{x^7-1}{x^3+x+1}$$

$$=x^4+x^2+x+1$$

et la matrice de contrôle correspondant est

$$H=\begin{pmatrix} h_4 & h_3 & h_2 & h_1 & h_0 & 0 & 0 \\ 0 & h_4 & h_3 & h_2 & h_1 & h_0 & 0 \\ 0 & 0 & h_4 & h_3 & h_2 & h_1 & h_0 \end{pmatrix}$$

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

Pour encoder 0111 en utilisant le produit polynomial, on doit multiplier le polynôme $m(x)=x^3+x^2+x$ correspondant par $g(x)=x^3+x+1$.

On obtient

$$c(x) = m(x)\,g(x) = \left(x^3 + x^2 + x\right)\left(x^3 + x + 1\right)$$
$$= x^6 + x^5 + x$$
$$= c_6 x^6 + c_5 x^5 + c_4 x^4 + c_3 x^3 + c_2 x^2 + c_1 x + c_0.$$

Qui correspondant au mot code $c_0 c_1 c_2 c_3 c_4 c_5 c_6 = 0100011$.

Chapitre III

Décodage des codes linéaires

1. Introduction

Dans ce chapitre on va présenter les définitions et les principes de détection et correction d'erreurs, ensuite on va présenter deux méthodes de décodage, décodage par tableau standard et décodage par syndrôme, tout en donnant des exemples pour chaque méthode.

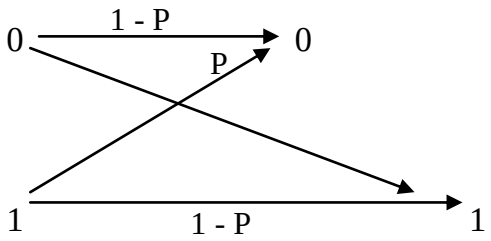
Décoder dans un code C désigne l'action d'associer un mot du code a un mot de l'espace vectoriel, on cherche le plus souvent à décoder en associant à un mot le mot de code du quel il est le plus proche.

Supposons que le code C est binaire. Soit $x = x_1.....x_n$ le mot du code transmis et $y = y_1.....y_n$ le mot reçu. on définit le vecteur erreur e par:

$$e = y - x = e_1.....e_n.$$

Avec $e_i = 1$ si le i^{ieme} symbole est faux et $e_i = 0$ si le i^{ieme} symbole est correct.

En général, on suppose que le canal est symétrique et sans mémoire (les bits sont indépendants) c.à.d une probabilité p de recevoir un symbole erroné et $1-p$ de la recevoir correctement (généralement il faut que $p < 1/2$)



Où

$$p(0/0) = p(1/1) = 1-p$$

et

$$p(0/1) = p(1/0) = p$$

avec $p(y_i/x_i)$ désigne la probabilité de recevoir y_i sachant que le symbole x_i qui a été transmit .

La probabilité de chaque mot de longueur n transmis sans erreur est $(1-p)^n$ puisque chaque symbole a la probabilité $1-p$ et donc

$$p(y/y) = (1-p)(1-p).....(1-p) = (1-p)^n .$$

La probabilité que le vecteur reçu a une erreur est $p(1-p)^{n-1}$ et la probabilité que le vecteur reçu a i erreur est

$$p^i (1-p)^{n-i} .$$

Puisque $p < 1/2$ un vecteur est reçu avec aucune erreur est plus probable q'un vecteur contenant des erreurs.

Un vecteur reçu avec une erreur est plus probable qu'un autre avec deux …… etc.

Exemple 1

Considérons le code binaire de répétition de longueur 3, $C = \{000,111\}$.

Supposons le mot code 000 est transmit, les mots qui seront de $\mathbb{F}_2^3$ décodé par 000 sont $000,100,010,$ et 001.

Donc la probabilité que les vecteurs reçus sont décodés par le mot code 000 est

$$(1-p)^3 + 3p(1-p)^2 .$$

2. Détection et correction d'erreurs

On suppose avoir reçu un message qui n'est pas un mot du code. Il est clair qu'il y a eu une erreur au cours de la transmission et nous avons *détecté* la présence d'une (ou de plusieurs) erreurs. Si aucune erreur n'a été détecté, on a soit reçu un mot du code, soit reçu un mot qui comportait trop d'erreurs.

Le but de codage est évidemment la détection et surtout la correction d'erreurs. Il nous faut cependant définir ce que l'on entend par détection et correction d'erreurs dans le

cadre de la théorie de codage. Pour ce faire, nous utiliserons les définitions de distance de Hamming et de distance minimale.

Soit $x \in \mathbb{F}_q^n$ un mot de code expédie dans le canal et $y \in \mathbb{F}_q^n$ le message reçu.

Définition 1

On appelle vecteur erreur

$$e = x - y = (e_1, ..., e_n) \in F_q^n$$

et le nombre d'erreurs $N_e = d(x, y)$.

Exemple 2

Soit C le code de répétition: $C = \{000,111\}$ dans $\mathbb{F}_2^3$

Soit : $x = 000$ mot de code transmit et $y = 101$ le vecteur reçu.

Alors $e = 101$ et $N_e = d(000,101) = 2.$

Théorème 1

Le code (n, M, d) *détecte tout message faux, tant que le nombre d'erreur* N_e *vérifie*

$$0 < N_e < d.$$

Preuve :

Soit x le mot de code transmis et y le mot reçu, si $d(x, y) = N_e < d$, donc le message y ne peut être un mot de code et par suite on peut affirmer qu'il y'a au mois N_e erreurs commises. Si le message y est un mot de code, on ne peut pas affirmer que c'est le mot code qui a été envoyé, ou bien $y \neq x$ et dans ce cas $N_e > d$.

En résumé, le code peut détecter toutes les erreurs si $0 < N_e < d$.

Exemple 3

Dans l'espace vectoriel $\mathbb{F}_2^3$, on considère le code $\{000,110,101,011\}$ de distance minimale $d = 2.$ Si le mot de code envoyé est $x = 101$ et on reçoit le message $y = 111$ qui n'est pas un

mot de code, le récepteur peut constater selon le code C qu'au moins une erreur a été commise. par contre, si le récepteur reçoit $y = 000$ qui un mot de code (dans ce cas le mot code envoyé x est changé à travers le canal en un autre mot de code y), le récepteur considère que y est le bon mot de code qui à été envoyé, dans ce cas le nombre d'erreurs $N_e = d(101,000) = 2 > 2-1 = 1.$ En résumé ce code ne peut détecter qu'une seule erreur.

2.1. Erreurs de transmission

Quand l'expéditeur envoie un mot binaire x de longueur n, le destinataire reçoit un mot binaire y, et bien évidemment tout va pour le mieux si $y = x$.

Nous supposons toujours que y et x on la même longueur, autrement dit que les bits peuvent s'altérer mais pas se perdre.

Un dispositif permettant de transmettre des bits s'appelle un *canal binaire*.

Tout canal binaire comporte un risque d'erreur, et le rapport :

$$\frac{nombre\,de\ bits\,faux}{nombre\,de\ bits\ transmis}$$

donne une estimation de ce risque. Si l'on assimile l'expédition d'un bit à un phénomène aléatoire comportant deux issues :<<le bit est bien transmis>>, <<le bit est mal transmis>>, ce rapport est à peu près égal à p, on l'appelle la *probabilité d'erreur* du canal.

Evidemment $q = 1 - p$ est la probabilité que le bit soit bien transmis, sauf pour un canal très mauvais, p est très petit, et q est très voisin de 1.

Exemple 4

Si $p = 0,001$, et si l'on transmet un bit toutes les microsecondes, au bout d'une minute il y'aura environ $60\,000$ bits mal transmis.

Si la transmission de chaque bit est indépendante des autres, on résume cette propriété en disant que le canal est *sans mémoire*.

Théorème 2

Lorsqu'on transmet des mots binaires de longueur n *sur un canal sans mémoire dont la probabilité d'erreur est* p :

1) la probabilité que le vecteur d'erreur soit e *est :* $p^{w(e)}q^{n-w(e)}$.

2) la probabilité que le nombre d'erreur soit k *est :* $C_n^k p^k q^{n-k}$.

3) si l'on envoie le message c *la probabilité qu'on reçoive le message* y *est égal à :* $p^{d(c,y)}q^{n-d(c,y)}$.

Exemple 5

Si on a émie le signal 11010 et on a reçu 10011, les mots différent en deux positions .

En prenant $p = 0,1$ la probabilité de recevoir un mot pour lequel 2 bits ont été altérés est

$$\binom{5}{2}0,9^3.0,1^2 = 10.0,0073 = 0,073$$

et un mot avec une seule erreur est de $\binom{5}{1}0,9^4.0,1 = 5.0,06 = 0,3.$

On voit que, pour ce canal, il est probable de recevoir un mot avec une seule erreur de transmission qu'un mot avec deux erreurs de transmission.

Exemple 6

On expédie $c = 0110$, le tableau ci-dessous montre la probabilité de recevoir un autre mot de longueur 4 et il précise la valeur de cette probabilité quand $p = 0,001.$ par exemple l'événement :<<faire 2 erreurs >>comprend 6 issues : $0000, 0011, 0101, 1010, 1100, 1111$ sa probabilité est donc $6p^2q^2$.

Mot reçu	d	Probabilité et valeur si $p = 0,001$
0000	2	p^2q^2 $0,0000009....$
0001	3	p^3q^1 $0,0000000009.....$
0010	1	p^1q^3 $0,0009.....$
0011	2	p^2q^2 $0,0000009....$
0100	1	p^1q^3 $0,0009.....$
0101	2	p^2q^2 $0,0000009....$
0110	0	p^0q^4 $0,9.....$
0111	1	p^1q^3 $0,0009.....$
1000	3	p^3q^1 $0,0000000009.....$
1001	4	p^4q^0 $0,000000000001.....$
1010	2	p^2q^2 $0,0000009....$
1011	3	p^3q^1 $0,0000000009.....$
1100	2	p^2q^2 $0,0000009....$
1101	3	p^3q^1 $0,0000000009.....$
1110	1	p^1q^3 $0,0009.....$
1111	2	p^2q^2 $0,0000009....$

Puisque p est beaucoup plus petit que q les formules du théorème (2) montre que la probabilité qu'un mot se substitue à un autre pendant la transmission diminue très vite quand la distance qui les sépare est augmente ; dit autrement dit t, plus le poids du vecteur d'erreur augmente, moins il est probable.
Cette constatation pourrait nous amener à supposer que le vecteur d'erreur est l'élément le plus faible poids, ce qui nous conduirait à remplacer chaque mot reçu par le mot de code vraisemblable le plus proche.

Théorème 3

Un code linéaire C de distance minimum d peut détecter jusqu'à $d-1$ erreurs et en corriger jusqu'à $\left[\frac{d-1}{2}\right]$.

Preuve :

Supposons qu'on ait transmis le mot $u \in C$ et que l'on reçoive le vecteur $v = u + e$
e étant le vecteur d'erreur (dont les composantes non nulles représentent les erreurs de transmission).
Si $w(e) \geq d$ dans ce cas il se peut que $e \in C$. Si $e \in C$
Alors $u + e \in C$ car C est un espace vectoriel.
L'erreur ne sera donc pas détectée.
Considérons les sphères de rayon $t = \left[\frac{d-1}{2}\right]$ centrées sur les mots de code, par définition de d elles sont disjointes.
Si $w(e) \leq t$ alors v est dans la sphère centrée sur le mot du code u, on peut donc décoder v par u, les erreurs sont corrigées.
Si $w(e) > t$ alors v peut être dans une autre sphère centrée sur un mot de code diffèrent de celui émis. Le décodage peut donc fournir un mauvais résultat.

- La distance minimale d d'un code C permet d'obtenir le nombre maximum d'erreurs que le code peut corriger.

Si le message $c=(c_1.....c_n)$ a été envoyé avec moins de t erreurs de transmission, le message obtenu $x=(x_1,x_2,.....x_n)$ vérifie $d(x,c)\leq t$.

Ainsi l'on peut retrouver c à partir de x si, et seulement si, il existe un unique mot de code situé a une distance de x inférieur ou égale à t, cela revient a dire que les Boules fermées de rayon t centrées sur les éléments du code C soient disjointes.

Définition 2

Un code qui corrige jusqu'à t erreurs est appelé un code $t-$ correcteur.

Exemple 7

Si $d(C)=3$ le code corrige 1 erreur et détecte 2 erreurs. Généralement on a

$d(C)$	Le nombre d'erreurs détecté par C	Le nombre d'erreurs Corrige par C
1	**0**	**0**
2	**1**	**0**
3	**2**	**1**
4	**3**	**1**
5	**4**	**2**
6	**5**	**2**
7 ……	**6** ……..	**3** ……

Théorème 4

Si le code $C(n,k,d)$ *est* $t-$ *correcteur les paramètres* n,k,d *satisfont l'inégalité.*

$$q^k\left(1+(q-1)\binom{n}{1}+.....+(q-1)^t\binom{n}{t}\right)\leq q^n.$$

Définition 3

Un code linéaire $C(n,k,d)$ *,* $t-$ *correcteur, est dit parfait si l'égalité est satisfaite dans le théorème précédent.*

On s'intéresse maintenant naturellement au décodage. Il est à noté que l'opération de décodage n'est pas l'application inverse du codage.

3. Théorie algébrique du décodage

Dans toute cette section on considère un code linéaire $C(n,k,d)$ sur $\mathbb{F}_q$

3.1. position du problème

Etant donné

- que $x\in C$ est le`` message transmis''
- que x est perturbé dans un canal bruité par l'erreur $e\in\mathbb{F}_q^n$
- que $y=x+e$ ''message reçu'' est le seul mot auquel le décodeur a accès**,**

le problème du décodage est de retrouver x à partir de y.

3.2. Décodage par tableau standard

Soit C un code linéaire de dimension k et de longueur n sur F_q on définit sur l'espace vectoriel F_q^n la relation R par :

$$\forall x,y\in F_q^n: x\,R\,y\Leftrightarrow x-y\in C.$$

Proposition 1

La relation R est une relation d'équivalence, le code C est la classe d'équivalence de 0.

Démonstration :

- Réflexivité : pour tout $x \in \mathbb{F}_q^n$ on a $x-x=0\in C$ car le code est linéaire : donc xRx.
- Symétrie : si $xR\,y$, alors par définition $x-y\in C$ et on a $y-x\in C$ car C est linéaire donc. yRx.
- Transitivité : si xRy *et* yRz alors par définition $(x-y)\in C$ et $(y-z)\in C$

On a $x-z=(x-y)+(y-z)$: comme le code C est linéaire et que $(x-y)$ et $(y-z)$ sont des mots du code, $x-z$ est un mot du code xRz :

On note la classe d'équivalence d'un mot a par la relation R.

$\bar{\text{a}}=\{\text{a}+c \ / \ c\in C\}$.

En fin, il est clair que la classe de 0 est C : en effet

$cR\,0\Leftrightarrow c-0\in C$.

Proposition 2

Le cardinal de chaque classe d'équivalence d'un code linéaire $C(n,k)$ sur $\mathbb{F}_q$ est q^k.

Preuve:

On considère l'application : $\Phi:\begin{matrix} C\to \text{a}+C \\ c\mapsto \text{a}+c \end{matrix}$

Φ est une bijection de C dans $\text{a}+C$.

On a donc $|C|=|\text{a}+C|=q^k$.

Proposition 3

Pour un code linéaire $C(n,k)$ *sur* $\mathbb{F}_q$ *il existe* q^{n-k} *classes d'équivalences constituant une partition de l'espace vectoriel* $\mathbb{F}_q^n$.

L'ensemble quotient est $\mathbb{F}_q^n / R = \mathbb{F}_q^n / C = \{a + C / a \in \mathbb{F}_q^n\}$ et chaque classe contient q^k éléments d'où la partition de $\mathbb{F}_q^n$.

$$\mathbb{F}_q^n = C \cup (a^{(1)} + C) \cup (a^{(2)} + C) \cup ..... \cup (a^{(t)} + C).$$

Avec : $t = \frac{|\mathbb{F}_q^n|}{|C|} - 1 = \frac{q^n}{q^k} - 1 = q^{n-k} - 1.$

Chacune des classes de cette relation d'équivalence est appelée "classe latérale ou translaté ".

Le code C est la classe de l'élément de $\mathbb{F}_q^n$.

3.2.1. Recherche du vecteur d'erreur

Soit x un mot de code transmit et $y \in \mathbb{F}_q^n$ l'élément reçu, y doit être dans l'une des classe disant $a^{(i)} + C$. Le vecteur erreur $e = y - x \in a^{(i)} + C - x = a^{(i)} + C$,

(car C est un code linéaire $C - x = C$) et par conséquent l'erreur commise est dans la classe de $y = a^{(i)} + C$.

L'erreur minimale est obtenue en prenant l'élément α de poids minimale qu'on appelle chef de classe on décode y par le mot de code $x = y - \alpha$. On fait remarquer que si α est le chef de classe de $a^{(i)} + C$, on a $\alpha + C = a^{(i)} + C$. Cette règle de décodage conduit à construire le tableau standard (table de Slepain) .

3.2.2. Construction du tableau standard

Un tableau standard pour le code linéaire $C(n,k)$ sur $\mathbb{F}_q$ contient tous les mots de $\mathbb{F}_q^n$. Sur chaque ligne sont rangés tous les éléments d'une même classe d'équivalence.

On procède de la façon suivante :

Soient : $\alpha_1, \alpha_2, \ldots\ldots\ldots\ldots\alpha_t$, les chefs de classe.

- **première ligne :** on liste les mots de C, en commençant par 0.
- **Deuxième ligne** : on choisit un mot, α_1 de poids minimum, qui n'est pas déjà dans le tableau (on obtient α_1 en parcourant tous les mots de poids $1,2,3\ldots\ldots$) on remplit alors le ligne en inscrivant $\alpha_1 + C$ dans la colonne ayant au sommet le mot du code C.
- **Troisième ligne** : on choisit un mot α_2, de poids minimum qui n'est pas déjà dans le tableau, on remplit alors la ligne en inscrivant $\alpha_2 + C$ dans la colonne ayant au sommet le mot du code C.
- On construire de la même façon jusqu'à ce que tous les mots du code soient inscrits et que des q^{n-k} lignes soient remplies

$c_1 = 0 \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots c_{q^k}$ classe des mots de code.

$\alpha_1 + c_1 \qquad \alpha_1 + c_2 \quad \ldots\ldots\ldots\ldots\ldots\ldots \alpha_1 + c_{q^k}$

$\alpha_2 + c_1 \qquad\qquad \alpha_2 + c_2 \ldots\ldots\ldots\ldots\ldots\ldots \alpha_2 + c_{q^k}$

.

.

.

.

$\alpha_t + c_1 + \qquad\qquad \alpha_t + c_2 \ldots\ldots\ldots\ldots \alpha_t + c_{q^k}$

} autres classes

Exemple 8

Soit C le code linéaire binaire de taille $(4,2)$ de matrice génératrice $G=\begin{pmatrix}1\,0\,1\,1\\0\,1\,0\,1\end{pmatrix}$,

c'est-à-dire le code $C=\{0000,1011,0101,1110\}$.

Construction du tableau standard :

0000	1011	0101	1110
1000	0011	1101	0110
0100	1111	0001	1010
0010	1001	0111	1100

Observons que la classe latérale $0001+C$ est identique à la classe latérale $0100+C$ puisque $0001\in 0100+C$.

Remarque 1

Le tableau standard n'est pas unique, en particulier, si dans une classe d'équivalence le mot de plus petit poids n'est pas unique le choix de celui qui est en tête de ligne dans le tableau aura une influence sur le décodage.

3.2.3. Utilisation du tableau standard

Méthode

Soit y le mot reçu, on cherche sa position dans le tableau puis on le corrige par le mot x situé en haut de la même colonne.

Cela revient à ajouter a y le vecteur d'erreur e situé en tête de sa ligne.

Exemple 9

Soit C le code linéaire binaire de taille $(4,2)$

$C = \{0000,1011,0101,1110\}$.

On obtient le tableau standard suivant :

0000	1011	0101	1110
1000	0011	1101	0110
0100	1111	0001	1010
0010	1001	0111	1100

Si on suppose avoir reçu le message 1111. on vérifie facilement que ce n'est pas un mot du code. Pour savoir de quel mot de code il provient, On cherche sa position dans le tableau standard et on lit le mot du code qui lui correspond sur la première ligne. Ainsi, le message transmis était 1011avec 0100 comme vecteur d'erreur.

Remarque 2

Le code de cet exemple peut corriger une erreur si celle –ci se rencontre sur une de trois première position du mot mais pas dans la quatrième. Par exemple, si le message 0101 *et altéré en* 0001. *On le décode convenablement. En revanche, le même message altéré sur sa dernière position donne* 0100 *qui est décode improprement* 0000. *On retrouve ainsi le fait que, comme* $d(C) = 2$, C *n'est pas un code correcteur, mais seulement détecteur d'une erreur.*

Exemple 10

Soit le code linéaire $C(5,2)$

$C = \{00000,10110,01011,11101\}$.

Sa matrice génératrice est $G = \begin{pmatrix} 1\,0\,1\,1\,0 \\ 0\,1\,0\,1\,1 \end{pmatrix}$

On a donc pour ce code $2^3 = 8$ classes d'équivalence différentes, C 'étant $1-$correcteur, on peut prendre comme représentant des 6 premiers classes le vecteur nul (pour le code lui-même) et les 5 vecteurs de poids 1 de $\mathbb{F}_2^5$ pour les deux derniers classes, on pourra prendre comme représentant de poids minimum un vecteur de poids 2 ne figurant pas dans les clases déjà remplies on a donc en effectuant cette construction :

00000	10110	01011	11101
10000	00110	11011	01101
01000	11110	00011	10101
00100	10010	01111	11001
00010	10100	01001	11111
00001	10111	01010	11100
11000	01110	10011	00101
10001	00111	11010	01100

Supposons maintenant que lors de la transmissions du mot du code $v = 10110$ il arrive une erreur par exemple, au niveau du cinquième bit, on recevra alors le mot $v' = 10111$ on cherche alors v' dans le tableau et on le décode par le mot situé sur la première ligne de sa colonne à savoir (10110). Le décodage est correct (on décode bien v' **par** v **).**

3.2.4. Mesure probabiliste de l'efficacité de la méthode si le code est binaire

Soit x un message envoyé, la probabilité que le message reçu soit un mot donné y dépend uniquement du vecteur d'erreur.

$e = x - y$, qui indique quelles erreurs de transmission doivent se produire

$$p(\text{Message reçu} = y) = p(\text{Vecteur d'erreur} = e)$$
$$= p^{w(e)}(1-p)^{n-w(e)}.$$

Exemple 11

Si le mot envoyé est 1101, la probabilité que le mot reçu soit 0111 est

$p(\textit{message reçu} = 0111) = p(\textit{vecteur} = 1010)$

$p\left(1^{er}\textit{ bit erroné}\right) \times p\left(2^{éme}\textit{ bit correct}\right) \times p\left(3^{éme}\textit{ bit erroné}\right) \times p\left(4^{éme}\textit{ bit correct}\right) = p^2(1-p)^2.$

Soit x le mot du code envoyé, le mot reçu y est corrigé en x si x et y sont dans la même colonne.

Par construction du tableau standard, x et y sont dans la même colonne si seulement si $x - y$ est dans la première colonne.

La correction fonctionne donc quand le vecteur erreur est dans la première colonne.

Proposition 4

On note α_i les mots de la 1^{er} colonne du tableau standard, x le mot du code envoyé, y le message reçu et $e = x - y$ le vecteur d'erreur, alors :

$$p(y \textit{ corrigé en } x) = \sum_{i=0}^{2^{n-k}} p(e = \alpha_i)$$

$$= \sum_{i=0}^{2^{n-k}} p^{w(\alpha_i)} (1-p)^{n-w(\alpha_i)}.$$

Exemple 12

On reprend le tableau standard construit précédemment :

0000	1011	0101	1110
1000	0011	1101	0110
0100	1111	0001	1010
0010	1001	0111	1100

La probabilité que le décodage soit correct est :

$$p(e=0000)+p(e=1000)+p(e=0100)+p(e=0010)=(1-p)^4+3p(1-p)^3.$$

Par exemple avec $p=10^{-4}$, la probabilité que le décodage soit correct est environ $0,9999$.

3.2.5. Inconvénients de la méthode

La méthode de correction par le tableau standard présente plusieurs inconvénients.

- Le tableau est long à construire.
- Dès que la taille des blocs est relativement importante ($n \geq 30$ environ), le tableau devient beaucoup trop gros pour être utilisable.
- La recherche du message reçu dans le tableau est lente.

On va voir une deuxième méthode de correction du message.

3.3. Le décodage par syndrôme

Soit C un code linéaire t- correcteur de matrice de contrôle H.

La linéarité des codes assure un décodage aisé, si un message x est reçu alors la détection d'erreur est réalisée par la matrice de control H. En effet, les altérations détectables ont eu lieu si et seulement si Hx^t est différent du vecteur nul. Si le nombre d'erreurs présentes dans le message est inférieur à t, le nombre d'altérations assurément détectable, alors Hx^t possède un unique antécédent e dans la boule fermé de centre le vecteur nul et de rayon $\boldsymbol{t}$. Le message corrigé est $x-e$ le vecteur Hx^t est appelé syndrome. Dans le cas où le nombre d'erreurs est supérieur à t il existe plusieurs antécédents de poids minimal et les altérations ne sont plus assurément corrigibles.

Définissons Tout d'abord l'application syndrôme.

Définition 4

On considère un code linéaire $C(n,k)$ *de matrice de contrôle* H, *l'application syndrôme est définie par*

$$S: \begin{array}{l} \mathbb{F}_q^n \to \mathbb{F}_q^{n-k} \\ x \mapsto Hx^t \end{array}$$

où $S(x)$ *est appelé syndrôme du vecteur* x.

Les propriétés du syndrôme sont les suivantes.

Lemme 1

C est $t-$*correcteur*

Soit $y \in \mathbb{F}_q^n$

1- $y \in C \Leftrightarrow S(y) = 0.$

2- *Si* $y = c + e$, *où* $c \in C$ *est le mot émis et* e *l'erreur, alors* $S(y) = S(e).$

3- $y_1, y_2 \in \mathbb{F}_q^n$ *si* $w(y_1) \le t$ *e tw* $(y_2) \le t$, *alors* $S(y_1) = S(y_2) \Rightarrow y_1 = y_2.$

Preuve :

1- par définition de S.

2- S étant linéaire, $S(y) = S(c+e) = S(c) + S(e).$

3-Si $w(y_1) \le t$ et $w(y_2) \le t$, alors $w(y_1 - y_2) \le 2t < d$.

De plus $S(y_1) = S(y_2) \Rightarrow S(y_1 - y_2) = 0 \Rightarrow y_1 - y_2 \in C.$

Alors $y_1 - y_2 \in C$, $w(y_1 - y_2) < t$ impliqué $y_1 - y_2 = 0$.

3.3.1. Méthode de décodage

Soit $C(n,k)$ un code linéaire $t-$correcteur de matrice de contrôle H. Voici une méthode pour décoder tout mot reçu $y \in \mathbb{F}_q^n$ pourvu que y soit affecté d'au plus t erreurs : $y = c + e_y$ avec $c \in C$ et $w(e_y) \leq t$.

On considère toutes les erreurs éventuelles, c'est-à-dire tous les $e \in \mathbb{F}_q^n$ tel que $w(e) \leq t$. Pour chaque $e \in \mathbb{F}_q^n$ tel que $w(e) \leq t$, on calcul $S(e)$. D'après l'assertion 3 du lemme (1), si $e \neq e'$ alors $S(e) \neq S(e')$. On fait une table contenant ces informations.

Soit $y = c + e_y \in \mathbb{F}_q^n$ un mot reçu. On calcule $s = S(y)$. On sait que $s = S(e_y)$.

[i] si s figure dans la table, associée à e_0, en décode y par $y - e_0$

[ii] si non, on peut dire que y est affecté de plus de $\boldsymbol{t}$ erreurs et on ne peut pas décoder.

Cette méthode de décodage est efficace mais coûteuse. Remarquons d'autre part que, même dans le cas [i] on fera une erreur de décodage si, en fait, y est affecté de plus de t erreurs.

Exemple 13

Soit $C(5,2)$ le code linéaire sur $\mathbb{F}_2$ de matrice génératrice $\begin{pmatrix} 1 & 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 1 & 1 \end{pmatrix}$

i.e $C = \{00000, 10110, 01011, 11101\}$ la matrice de contrôle de C est $H = \begin{pmatrix} 1 & 0 & 1 & 0 & 0 \\ 1 & 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 1 \end{pmatrix}$

syndrome z	Chef de classe $f(z)$
000	00000
110	10000
011	010000
100	00100
010	00010
001	00001

On calcule les syndrômes $S(e)$ des chefs de clase de e *par* $H^t e$:

$S(00000) = 000$ $S(10000) = 110$ $S(01000) = 011$
$S(00100) = 100$ $S(00010) = 010$ $S(00001) = 001$

On construit la table des syndromes :

Si on suppose avoir reçu le message $y = 11111$. On calcule le syndrôme

$$S(y) = \begin{pmatrix} 1 & 0 & 1 & 0 & 0 \\ 1 & 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 \\ 1 \\ 1 \\ 1 \\ 1 \end{pmatrix} = (010).$$

On en déduit que le mot de code est $y - f(010) = 11111 - 00010 = 11101$ et que le vecteur d'erreur était 00010.

3.3.2. Inconvénients de la méthode

La méthode de décodage par syndrôme présente plusieurs inconvénients:

1. Le tableau est long à construire.
2. Cette méthode est toute fois limitée car, pour de très grands codes, il devient impossible de trouver les leaders de classe.

Un code binaire de longueur 50 et de dimension 20 possède prés de 10^9 classes.

Chapitre IV

Décodage des codes linéaires cycliques

1. Introduction

Dans ce chapitre on s'intéresse au problème du décodage des codes cycliques. On va présenter deux méthodes de décodage des codes cycliques (la méthode de Meggitt, la méthode de décodage par piégeage d'erreur), et on terminera par l'étude de la méthode de Meggitt au cas non binaire.

Comme un code cyclique $C(n,k,d)$ sur un corps fini $\mathbb{F}_q$ n'est qu'un code linéaire, donc nous pouvons utiliser la même procédure de décodage que celle utilisée pour les codes linéaires à savoir la méthode du syndrôme.

Ainsi si $c(x) \in C$ est le mot envoyé et que $y(x)$ est reçu, $e(x) = y(x) - c(x)$ est le polynôme erreur. Nous définirons le poids d'un polynôme comme le nombre de coefficients différents de zéro.

Rappelons aussi que si *t* est la capacité de correction de C et si $z(x) \in \mathbb{F}_q[x] / (x^n - 1)$ on dira que « $z(x)$ est le mot reçu dont l'erreur est $e(x)$ ». Si $w(e(x)) \leq t$ et s'il existe $c(x) \in C$ tel que:

$$z(x) = c(x) + e(x)$$

(c'est à dire, $z(x)$ provient d'un mot de C entaché d'un nombre d'erreur au plus égal à t)

Il nous reste maintenant à voir ce que l'on entend par le syndrome d'un polynôme.

2. Syndrôme d'un polynôme

Définition 1

Soit C *un code cyclique de polynôme générateur* $g(x)$.

On appelle syndrôme polynomial (ou plus simplement syndrôme) d'un mot $z(x) \in \mathbb{F}_q[x] / (x^n - 1)$, *le reste de la division de* $z(x)$ *par* $g(x)$ *dans* $\mathbb{F}_q[x]$, *on le note* $S(z(x))$.

- Il est clair que $z(x) \in C \Rightarrow S(z(x)) = 0$

et que $S(z(x)) = S(z'(x)) \Leftrightarrow z(x) - z'(x) \in C$.

Ainsi cette définition de syndrôme est équivalente à celle, présentée pour les codes linéaires.

Propriété 1

Soit un mot $z(x) \in \mathbb{F}_q[x] / (x^n - 1)$, *alors* $z(x) \in C$ *si et seulement si* $g(x)$ *divise* $z(x)$ *dans* $\mathbb{F}_q[x]$.

Exemple 1

Soit C un code cyclique $(7,4)$ sur $\mathbb{F}_2$ de polynôme générateur $g(x) = x^3 + x + 1$,

Soit le mot reçu $y(x) = x^5 + x^4 + x^2$, le syndrôme du mot reçu est le reste de la division euclidienne de $y(x)$ par $g(x)$

$$x^5 + x^4 + x^2 = (x^3 + x + 1)(1 + x^3 + x + 1) + 1 + x^2.$$

Donc $S(y(x)) = 1 + x^2$.

Le décodage des codes cycliques s'effectue généralement en 3 étapes:

- Calcul du syndrôme.
- Association du syndrôme à l'erreur correspondante grâce à une table.
- Ajout de l'erreur au mot reçu.

Passons maintenant à un exemple permettant d'illustrer tout ce qui précède.

Exemple 2

Soit C un code cyclique $(7,4,3)$ sur $\mathbb{F}_2$ de polynôme générateur

$g(x)=x^3+x+1.$

Le code C corrige seulement une erreur, donc on à le tableau de syndrômes suivant :

Représentants de classes	syndrome
1	1
x	x
x^2	x^2
x^3	$x+1$
x^4	x^2+x
x^5	x^2+x+1
x^6	x^2+1

Supposons alors que ayant transmis $c(x)=x+x^2+x^3+x^6$, nous ayons reçu

$y(x)=x+x^3+x^6$, calculons son syndrôme

or $x^6+x^3+x=(x^3+x)(x^3+x+1)+x^2$.

Donc $S(y(x))=x^2$.

Le représentant de la classe de x^2 étant x^2, on décode $y(x)$ de la façon suivante

$$c(x)=y(x)-x^2=x+x^2+x^3+x^6.$$

Qui est bien le mot qui avait été envoyé.

Tel que nous l'avion mentionné avec les codes linéaires, un tel principe de décodage nécessite la tenue d'une table de représentant de classe qui peut devenir très longue. Ainsi nous sommes en droit de nous demander si une méthode de décodage tirant profit de la structure cyclique de code n'aurait pas pour effet d'améliorer le décodage. La réponse évidement est oui.

En effet il y a deux méthodes de décodage:

-la méthode de décodage de Meggitt.

-la méthode de décodage par piégeage d'erreur.

Pour une description détaillée voir [13]; [9]; [15].

3. Décodage de Meggitt

Supposons $C(n,k,d)$ un code cyclique sur $\mathbb{F}_2$ de polynôme générateur $g(x)$, C corrigera $t=\left[\frac{d-1}{2}\right]$ erreurs. Supposons que $c(x)\in C$ est transmis et $y(x)=c(x)+e(x)$ est reçu, où $e(x)$ est le vecteur erreur avec $w\left(e(x)\right)\le t$.

Dans cette section, on présente une technique pour le décodage des codes cycliques nommé le décodage de Meggitt.

La méthode de décodage de Meggitt s'applique aux codes cycliques binaires, mais elle peut se généraliser au cas non binaires, l'idée de base consiste en l'utilisation de la cyclicité du code pour restreindre la table des syndrômes et permettre des calculs récursifs. La méthode s'appuie sur le résultat suivant :

Proposition 1

Soit $e(x)$ le mot erreur du mot reçu $y(x)$.

Alors, pour tout entier, $0\le j\le n-1$.

a) le mot $x^j y(x)$ est un mot reçu l'erreur est $x^j e(x)$.

b) $S\left(x^j e(x)\right)=S\left(x^j y(x)\right)$.

(Tous les produits sont calculés dans $\mathbb{F}_2[x]/(x^n-1)$).

Preuve :

a) Supposons que $c(x) \in C$ est transmis et $y(x) = c(x) + e(x)$ reçu, où $e(x)$ est le vecteur erreur, on déduit $x^j\, y(x) = x^j c(x) + x^j e(x)$, le code C étant cyclique, on sait que $x^j c(x) \in C$.

D'autre par $w\left(x^j e(x)\right) = w\left(e(x)\right)$ car la multiplication par x^j ne modifié pas le poids d'un mot, l'égalité précédente montre donc que $x^j e(x)$ est le mot erreur du mot reçu $x^j y(x)$.

b) Puisque $y(x)$ est le mot reçu, il existe $c(x)$ multiple de $g(x)$ dan $\mathbb{F}_2[x] \big/ \left(x^n - 1\right)$ tel que :

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

On obtient donc, dans $\mathbb{F}_2[x] \big/ \left(x^n - 1\right)$ une relation de la forme

$$x^j y(x) = x^j c(x) + x^j e(x)$$

ceci implique, dans $\mathbb{F}_2[x]$, une égalité de la forme

$$x^j\, y(x) = x^j c(x) + x^j e(x) + b(x)\left(x^n - 1\right)$$

puisque $g(x)$ divise $x^n - 1$.

Dans $\mathbb{F}_2[x]$, on voit que $x^j y(x) \equiv x^j e(x) \bmod g(x)$), ce qui montre que $x^j y(x)$ et $x^j e(x)$ ont le même reste de la division par $g(x)$, c'est-à-dire le même syndrome.

On voit donc, d'après b), que si l'on trouve $S\left(x^j y(x)\right)$ dans une table de syndrome indiquant l'erreur correspondante, on peut retrouver $x^j e(x)$ et donc aussi $e(x)$.

La proposition suivante montre comment on peut calculer $S\left(x^j y(x)\right)$ à partir de $S\left(y(x)\right)$ de manière récursive.

Proposition 2

Avec les notations de la proposition précédente soit $S_i(x)$ *la suite de polynômes de* $\mathbb{F}_2[x]/(x^n-1)$ *définie par :*

$$S_0(x)=S(y(x)).....S_{i+1}(x)=S(xS_i(x)).$$

Alors pour tout entier $0\le j\le n-1$

$$S_j(x)=S(x^j y(x)).$$

Exemple 3

Soit C un code cyclique $(7,4)$ sur $\mathbb{F}_2$ de polynôme générateur

$$g(x)=x^3+x^2+1.$$

Soit le mot reçu $y(x)=x^4+x^3+x^2+1$ le syndrôme du mot reçu est le reste de la division euclidienne de $y(x)$ par $g(x)$.

Donc

$$S_0(x)=S(y(x))=1+x+x^2$$

$$S_1(x)=S(xy(x))=1+x$$

$$S_2(x)=S(x^2y(x))=x+x^2$$

3.1. Principe du décodage

Le décodeur de Meggitt effectue un décodage symbole par symbole. On corrige d'abord une composante erronée du mot reçu au moyen de la méthode décrite ci-dessous, puis on applique de nouveau la méthode au nouveau mot reçu ainsi obtenu.

3.2. Algorithme de décodage de Meggitt

Soit T *la table des syndrômes des erreurs dont la composante d'indice* $n-1$ *est erronée. Soit* $c(x)$ *le mot envoyé,* $y(x)$ *le mot reçu, et* $e(x)$ *le mot erreur avec* $w(e(x)) \leq t$.

La suite $S_i(x)$ *est définie comme dans la proposition (2).*

- *Calcule de* $S(y(x))$.
- *Si* $S(y(x)) = 0$ *alors* $c(x) = y(x)$ *et l'algorithme se termine.*
- *Si non.*
  - *Rechercher le plus petit entier j tel que* $S_j(x)$ *se trouve dans la table* T.
  - *Corriger la composante d'indice* $n-1-j$ *de* $y(x)$, *soit* $y'(x)$ *le nouveau mot obtenu.*
  - *Repartir au début de l'algorithme avec* $y'(x)$.

Exemple 4

[i] Soit C le code Hamming $(15,11,3)$ sur $\mathbb{F}_2$, soit le polynôme générateur, $g(x) = x^4 + x + 1$, le code Hamminig binaire est un code $1-$ correcteur, la table T des syndrômes des erreurs dont la composante d'indice 14 est égale a 1 se réduit au tableau suivant :

Erreur	x^{14}
Syndrome	$x^3 + 1$

Soit $y(x) = x^5 + x^4 + x^2 + x + 1$ le mot reçu, le syndrôme est le reste de la division de $y(x)$ par $g(x)$, c'est-à-dire

$$x^5 + x^4 + x^2 + x + 1 = (x^4 + x + 1)(x + 1) + x .$$

Donc $S(y(x)) = x$, or $S(y(x))$ ne figure pas dans la table T on recherche le plus petit entier j tel que $S_j(x) \in T$.

On a $j = 13$ est le plus petit entier tel que $S_j(x) \in T$, il y a donc une erreur en position 1 avec l'hypothèse que la capacité de correction égale à 1 n'est pas dépassé, l'erreur est $e(x) = x$ et le mot envoyé est

$$c(x) = y'(x) = c(x) = x^5 + x^4 + x^2 + 1.$$

[ii] Soit C le code Hamming $(7,4,3)$ sur $\mathbb{F}_2$, soit $g(x)$ le polynôme générateur, $g(x) = x^3 + x + 1$, le code Hamming binaire est un code $1-$ Correcteur, la table T des syndrômes des erreurs dont la composante d'indice 6 est égale a 1 se réduit au tableau suivant :

Erreur	x^6
Syndrome	$x^2 + 1$

-Soit $y(x)=x^6+x^4+x+1$ le mot reçu, le syndrôme est le reste de la division de $y(x)$ par $g(x)$, c'est-à-dire

$$x^6+x^4+x+1=\left(x^2+x+1\right)\left(x^3+1\right)+0.$$

Donc $S\left(y\left(x\right)\right)=0,$ alors le mot envoyé est $y\left(x\right).$

- Soit $y(x)=x^6+x^3+x$, le mot reçu, le syndrôme est le reste de la division de $y(x)$ par $g(x)$ c'est-à-dire

$$x^6+x^3+x=\left(x^3+x+1\right)\left(x^3+1\right)+x^2.$$

Donc $S\left(y(x)\right)=x^2,$ or $S\left(y(x)\right)$ ne figure pas dans la table T, et $j=4$ est le plus petit entier tel que $S_j\left(x\right)\in T$, il y a donc une erreur en position 2 avec l'hypothèse que la capacité de correction égale à 1 n'est pas dépassé, l'erreur est $e(x)=x^2$ et le mot envoyé est

$$c\left(x\right)=y'\left(x\right)=x^6+x^3+x^2+x.$$

4. Décodage par piégeage d'erreur

Supposons $C(n,k,d)$ un code cyclique sur $\mathbb{F}_q$ de polynôme générateur $g\left(x\right).$ C corrigera $t=\left[\frac{d-1}{2}\right]$ erreurs. Supposons que $c(x)\in C$ est transmis et $y(x)=c(x)+e(x)$ est reçu, où $e(x)$ est le vecteur erreur avec $w\left(e\left(x\right)\right)\le t.$

La méthode de décodage par piégeage d'erreur est une modification de la méthode de Meggitt, il s'agit de déplacer par décalage circulaire, c'est-à-dire "Piéger" en quelque sorte, les composantes non nulles de l'erreur sur certaines positions, on considère un code non nécessairement binaire, et on suppose que le nombre d'erreur ne dépasse pas la capacité de correction t.

Le principe du décodage par piégeage d'erreur s'appuie sur les résultats suivants.

Lemme 1

Soit $e(x)$ *le mot erreur du mot reçu* $y(x)$.

Si $\deg e(x) \le n-k-1$ *alors* $e(x) = S(y(x))$.

Preuve :

Dans $\mathbb{F}_q[x] / (x^n - 1)$ on a $y(x) = c(x) + e(x)$, avec $c(x) \in C$, soit encore dans $\mathbb{F}_q[x]$

$y(x) = a(x)g(x) + e(x) + b(x)(x^n - 1)$ puisque $g(x)$) devise $x^n - 1$, on trouve

$y(x) = d(x)g(x) + e(x)$, si $\deg e(x) \le n-k-1$ alors d'après l'unicité du reste dans la division par $g(x)$, on obtient $S(y(x)) = e(x)$.

Lemme 2

Soit $e(x)$ *le mot erreur du mot reçu* $y(x)$,

alors $w(S(y(x))) \le t$ *si et seulement si* $S(y(x)) = e(x)$.

Preuve :

La division dans $\mathbb{F}_q[x]$, de $y(x)$ par $g(x)$ s'exprime par

$$y(x) = g(x)a(x) + S(y(x))$$

Les conditions sur les degrés des polynômes intervenant dans cette égalité, font que celle ci est également vérifiée dans $\mathbb{F}_q[x] / (x^n - 1)$. la décomposition d'un mot reçu comme somme d'un mot du code et d'un mot de poids inférieur ou égal à t est unique.

Donc si $w(S(y(x))) \le t$, alors $S(y(x)) = e(x)$.

Réciproquement si $S(y(x)) = e(x)$, alors $w(S(y(x))) \le t$ puisque le poids de l'erreur est au plus t.

Théorème 1

Soit $e(x)$ *le mot erreur du mot reçu* $y(x)$.

Si s *est un entier* $0 \le s \le n-1$ *tel que* $w\left(S\left(y(x)\right)\right) \le t$, *alors*

$$e(x) = x^{-s} S\left(x^{s} y(x)\right).$$

Preuve :

Soit $y_1(x) = x^{s} y(x)$, c'est le mot reçu dont l'erreur est $e_1(x) = x^{s} e(x)$, d'après le lemme (2), si $w\left(S\left(y_1(x)\right)\right) \le t$, alors $S\left(y_1(x)\right) = e_1(x)$, soit

$S\left(x^{s} y(x)\right) = x^{s} e(x)$ *et* $e(x) = x^{-s} S\left(x^{s} y(x)\right)$.

Théorème 2

Soit $e(x)$ *le mot erreur du mot reçu* $y(x)$.

Si $e(x) = x^{m} e_1(x)$ *avec deg* $e_1(x) \le n-k-1$, *alors* $w\left(S\left(x^{m} y(x)\right)\right) \le t$.

Preuve:

D'après le lemme (1), si $deg\, e_1(x) \le n-k-1$, alors $e_1(x) = S\left(y_1(x)\right)$ et donc $w\left(S\left(y_1(x)\right)\right) \le t$, par ailleurs $y(x) = x^{m} y_1(x)$, donc $y_1(x) = x^{-m} y(x)$ et par suite $w\left(S\left(x^{-m} y(x)\right)\right) \le t$.

4.1. Principe du décodage

De nouveau on peut calculer $S\left(x^{s} y(x)\right)$ récursivement en utilisant la suite $S_i(x)$ définie au paragraphe précédent, on cherche s tel que $w\left(S\left(x^{s} y(x)\right)\right) \le t$, puis $e(x)$ par le théorème (1), dans le cas du théorème (2), ensuite il existe un tel s, et $e(x)$ est alors donné par le théorème (2).

4.2. Algorithme de décodage par piégeage d'erreur

Soit $y(x)$ *le mot reçu,* $e(x)$ *le mot erreur avec* $w(e(x)) \le t$.

- *Calcule de* $S(y(x))$.
- *Si* $S(y(x)) = 0$ *alors* $e(x) = 0$.
- *Si non*
  - *si* $w(S(y(x))) \le t$ *alors* $e(x) = S(y(x))$.
  - *si non on cherche le plus petit entier* s *tel que*

$w(S(x^s y(x))) \le t$, *alors* $e(x) = x^{-s} S(x^s y(x))$.

- *le mot envoyé est* $c(x) = y(x) - e(x)$.

Exemple 5

[i] Soit C le code Hamming $(15,11,3)$ sur $\mathbb{F}_2$, soit $g(x)$ le polynôme générateur

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

Soit $y(x) = x^5 + x^4 + x^2 + x + 1$, le mot reçu. Le syndrôme est le reste de la division de $y(x)$ par $g(x)$ soit $x^5 + x^4 + x^2 + x + 1 = (x^4 + x + 1)(x + 1) + x$.

Donc $S(y(x)) = x$ et $w(S(y(x))) = 1$, en supposant que la capacité d'erreurs n'est pas dépassée l'erreur est $e(x) = S(y(x))$, et le mot transmis est donc

$$c(x) = y(x) - e(x) = x^5 + x^4 + x^2 + 1.$$

Soit maintenant $y(x) = x^{14} + x^5 + x^2 + x + 1$, le mot reçu. Le syndrôme est le reste de la division de $y(x)$ par $g(x)$ soit,

$$x^{14} + x^5 + x^2 + x + 1 = (x^4 + x + 1)(x^{10} + x^7 + x^6 + x^4 + x^2) + (x^3 + 1).$$

Donc $S(x) = x^3 + 1$ et$w(S(y(x))) = 2$, puisqu'on suppose que le poids de l'erreur est au plus1, $S(y(x)) \neq e(x)$, l'entier $s = 4$ est le plus petit entier tel que,

$w(S(x^4 y(x))) \le t$.

$$e(x) = x^{-4} S\left(x^4 y(x)\right)$$

$$S\left(x^4 y(x)\right) \equiv x^3 \left(\bmod g(x)\right)$$

$$e(x) = x^{-4} x^3 \left(\bmod g(x)\right)$$

$$e(x) = x^{14}.$$

Le mot transmis est donc

$$c(x) = y(x) - e(x) = x^5 + x^4 + x^2 + 1.$$

[ii] Soit C le code Hamming$(7,4,3)$ sur $\mathbb{F}_2$, soit $g(x)$ le polynôme générateur

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

Soit $y(x) = x^6 + x^3 + x$ le mot reçu.

le syndrôme est le reste de la division de $y(x)$ par $g(x)$

Soit $x^6 + x^3 + x = (x^3 + x)(x^3 + x + 1) + x^2$, donc $S(y(x)) = x^2$ et $w\left(S\left(y(x)\right)\right) = 1$, en supposant que la capacité d'erreurs n'est pas dépassée l'erreur est $e(x) = S(y(x))$ et le mot transmis est donc $c(x) = x^6 + x^3 + x^2 + x$.

Soit maintenant $y(x) = x^6 + x^2 + x$ le mot reçu, le syndrôme est le reste de la division de $y(x)$ par $g(x)$ soit :

$$x^6 + x^2 + x = (x^3 + x)(x^3 + x + 1) + (x + 1).$$

Donc $\left(S(y(x))\right) = x + 1$ et $w\left(S\left(y(x)\right)\right) = 2$, puisqu'on suppose que le poids de l'erreur est au plus 1, $S(y(x)) \neq e(x)$, l'entier $s = 4$ est le plus petit entier s tel que

$$w\left(S\left(x^s y(x)\right)\right) \leq 1.$$

$$e(x) = x^{-4} S\left(x^4 y(x)\right)$$

$$S\left(x^4 y(x)\right) \equiv 1\left(\bmod g(x)\right)$$

$$e(x) \equiv x^4 1\left(\bmod g(x)\right)$$

$$e(x) = x^3.$$

Le mot transmis est donc

$$c(x) = x^6 + x^3 + x^2 + x\,.$$

Exemple 6

Soit C le code cyclique $(4,1,4)$ sur $\mathbb{F}_3$, soit $g(x)$ le polynôme générateur, $g(x) = x^3 + 2x^2 + x + 2,$ le code est un code $1-$correcteur,

- soit $y(x) = 2x^3 + x^2 + 2x$, le mot reçu, le syndrome est le reste de la division de $y(x)$ par $g(x)$ c'est-à-dire

$$2x^3 + x^2 + 2x = 2\left(x^3 + 2x^2 + x + 2\right) + 2$$

donc $S(y(x)) = 2$ et $w\left(S\left(y\left(x\right)\right)\right) = 1,$ en supposant que la capacité d'erreurs n'est pas dépassée l'erreur est $e(x) = S\left(y\left(x\right)\right),$ et le mot transmis est donc

$$c(x) = y(x) - e(x) = 2x^3 + x^2 + 2x + 1.$$

- Soit maintenant $y(x) = x^2 + 2x + 1$ le mot reçu, le syndrome est le reste de la division de $y(x)$ par $g(x)$ soit :

$$x^2 + 2x + 1 = 0\left(x^3 + 2x^2 + x + 2\right) + x^2 + 2x + 1.$$

Donc $S(y(x)) = x^2 + 2x + 1$ et $w\left(S\left(y\left(x\right)\right)\right) = 3,$ puisqu'on suppose que le poids de l'erreur est au plus 1, $S\left(y\left(x\right)\right) \neq e\left(x\right),$ l'entier $s = 1$ est le plus petit entier s tel que

$$w\left(S\left(x^s y(x)\right)\right) \leq 1$$

$$e(x) = x^{-1} S\left(xy(x)\right)$$

$$S\left(xy(x)\right) \equiv 1 \left(\operatorname{mod} g\left(x\right)\right)$$

$$e(x) \equiv x^{-1} 1 \left(\operatorname{mod} g\left(x\right)\right)$$

$$e\left(x\right) = x^3.$$

Le mot transmis est donc

$$c\left(x\right) = y\left(x\right) - e\left(x\right) = 2x^3 + x^2 + 2x + 1.$$

5. La méthode de décodage de Meggitt au cas non binaire

Supposons $C(n,k,d)$ un code cyclique sur $\mathbb{F}_q$ de polynôme générateur $g(x)$, C corrigera $t=\left[\frac{d-1}{2}\right]$ erreurs. Supposons que $c(x)\in C$ est transmis et $y(x)=c(x)+e(x)$ est reçu, où $e(x)$ est le vecteur erreur avec $w\left(e(x)\right)\le t$.

La méthode s'appuie sur le résultat suivant.

Proposition 3

Soit $e(x)$ *le mot erreur du mot reçu* $y(x)$.

Alors, pour tout entier j, $0\le j\le n-1$ *:*

a) *le mot* $x^j y(x)$ *est un mot reçu dont l'erreur est* $x^j e(x)$.

b) $S\left(x^j e(x)\right)=S\left(x^j y(x)\right)$.

(Tous les produits sont calculés dans $\mathbb{F}_q[x]/\left(x^n-1\right)$ *).*

La proposition suivante montre comment on peut calculer $S\left(x^j y(x)\right)$ à partir de $S\left(y(x)\right)$ de manière récursive.

Proposition 4

Avec les notations de la proposition précédente soit $S_j(x)$ *la suite de polynômes de* $\mathbb{F}_q[x]/\left(x^n-1\right)$ *définie par :*

$$S_0(x)=S\left(y(x)\right).....S_{i+1}(x)=S\left(xS_i(x)\right).$$

Alors pour tout entier $0\le j\le n-1$

$$S_j(x)=S\left(x^j y(x)\right).$$

Exemple 7

Soit C le code cyclique $(4,1,4)$ sur $\mathbb{F}_3$, soit $g(x)$ le polynôme générateur,

$g(x) = x^3 + 2x^2 + x + 2$

Soit le mot reçu $y(x) = 2x^3 + x^2 + 2x$ le syndrome du mot reçu est le reste de la division euclidienne de $y(x)$ par $g(x)$.

Donc

$$S_0(x) = S(y(x)) = 2$$

$$S_1(x) = S(xy(x)) = 2x$$

$$S_2(x) = S(x^2 y(x)) = 2x^2$$

$$S_3(x) = S(x^3 y(x)) = 2x^2 + x + 2$$

5.1. Algorithme de décodage de Meggitt au cas non binaire

Soit T la table des syndrômes des erreurs dont la composante d'indice $n-1$ est erronée. Soit $c(x)$ le mot envoyé, $y(x)$ le mot reçu, *et $e(x)$ le mot erreur avec $w(e(x)) \le t$.*

La suite $S_j(x)$ est définie comme dans la proposition (4).

- *calcule de $S(y(x))$.*
- *Si $S(y(x)) = 0$ alors $c(x) = y(x)$ et l'algorithme se termine.*
- *Si non.*
  - *Rechercher le plus petit entier j tel que $S_j(x)$ se trouve dans la table T.*
  - *Corriger la composante d'indice $n-1-j$ de $y(x)$, soit $y'(x)$ le nouveau mot obtenu. $y'(x) = y(x) - ax^{n-1-j}$ $(a \in \mathbb{F}_q)$.*
  - *Repartir au début de l'algorithme avec $y'(x)$.*

Exemple 8

Soit C le code cyclique $(4,1,4)$ sur $\mathbb{F}_3$, soit $g(x)$ le polynôme générateur, $g(x) = x^3 + 2x^2 + x + 2$, le code C est un code $1-$ correcteur, la table T des syndrômes des erreurs dont la composante d'indice 3 est erronée se réduit au tableau suivant :

Erreur	x^3	$2x^3$
Syndrome	$x^2 + 2x + 1$	$2x^2 + x + 2$

-Soit $y(x) = 1 + 2x + x^2 + 2x^3$ le mot reçu, le syndrôme de $y(x)$ est le reste de la division de $y(x)$ par $g(x)$, c'est-à-dire

$$1 + 2x + x^2 + 2x^3 = 2(x^3 + 2x^2 + x + 2) + 0.$$

Donc $S(y(x)) = 0$, alors le mot envoyé est $y(x)$.

- Soit $y(x) = 2x + x^2 + 2x^3$, le mot reçu, le syndrôme est le reste de la division de $y(x)$ par $g(x)$ c'est-à-dire

$$2x + x^2 + 2x^3 = 2(x^3 + 2x^2 + x + 2) + 2.$$

On recherche le petit entier j tel que $S_j(x) \in T$.

j	$S_j(x)$
0	2
1	$2x$
2	$2x^2$
3	$2x^2 + x + 2$

On a $j=3$ est le plus petit entier tel que $S_j(x) \in T$, il y a donc une erreur en position $4-3-1=0$ avec l'hypothèse que la capacité de correction égale à 1 n'est pas dépassée, l'erreur est $e(x)=2$ et le mot envoyé est

$$c(x)=y(x)-e(x)=y'(x)=2x^3+x^2+2x+1.$$

Conclusion

Nous avons présenté dans ce travail une étude sur les techniques de décodage des codes linéaires, qui consiste à déterminer le message original envoyé via un canal de transmission à partir du message reçu, elle est basée sur les propriétés structurelles des codes linéaires.

On a présenté quatre méthodes:

- *Le décodage par tableau standard.*
- *Le décodage par syndrôme.*
- *Le décodage de Meggitt.*
- *Le décodage par piégeage d'erreur.*

Et on a termine par l'étude de la méthode de Meggitt au cas non binaire.

Ces méthodes se basent sur le fait que le nombre d'erreur ne dépasse pas la capacité de correction.

Bibliographie

[1] Bassem Sakkour. *Etude et amélioration du décodage des codes de reed-muller d'ordre deux*. Thèse de doctorat en science. École polytechnique.2007.

[2] Bruno Martin. *Codage, cryptologie et applications***,** presses polytechniques et universitaires romaindes 2004.

[3] Coste M, Paugam A, Quarez R. *Codes correcteurs préparation à l'agrégation mathématiques Université de Rennes*1. juin2002.

[4] Christine Bachoc,*cours des codes(Uecode,signal)*
université Bordeaux, master CSI2-2004.

[5] Dany–Jack Merier. *Utilisation d'algèbre dans les systèmes* d'*informations*. IUFM des Antilles et de GUYANE mai 2001.

[6] Jacque Vélu. *Méthodes mathématiques pour L'informatique,* Dunod, paris,1994.

[7] Hans Bherer, théorie *algébrique du codage.* Mémoire présenté à la Faculté des études supérieures de l'université Lavel pour l'obtention du grade de maître ès sciences (M.Sc.) septembre 2000.

[8 Ladjlat. Lahcene *Etudes de l'équivalence de deux codes sur un corps finis*. Mémoire présenté pour l'obtention du diplôme de magistèr-2004-université de m'sila.

[9] Mac Williams, *F. J et Sloane, N-J-A, The theory of error – correcting codes*, Amsterdam- new York – oxford. North Holland 1977.

[10] Michel Demasure. *Cours d'algèbre. Primalité, divisibilité, codes*. Nouvelle bibliothèque mathématique, Cassini.1997.

[11] Mohamed zitouni, *algèbre*, Office des Publications Universitaires 1993

[12] Nicola.Sendrier. *Codes correcteurs d'erreurs à haut pouvoir de correction*. Thèses de doctorat de l'université Paris VI, Décembre 1991.

[13] O.papini et J.wolfman, *Algèbre discrète et codes correcteurs*, Springer Vergal 1995.

[14] P.V, Koseleff, codes *correcteurs d'erreurs.*
MIAC24-M3-Résumé du cours-V-2004.
[15]R. HILL A *first course in coding theory* clarendon press.oxsford
[16] R. Lidl et G. Pilz. *Applied Abstract Algebra*, springer verlag, New York – Berlin – Heidelberg, October 1997.
[17] Saadi Ameur. *Etude sur les bornes des codes correcteurs d'erreurs*.
Mémoire présenté pour l'obtention du diplôme de magistèr-2000-université de m'sila
[18] Steven Roman. *Coding and information theory*, springer verlag,
New York-Berlin-Heidelberg,1992.