7 - Compact Disc Audio : le code CIRC

Le code CIRC (Cross Interleaved Reed-Solomon Code) est un code correcteur d'erreur de type BCH particulièrement adapté aux problèmes du disque compact (CD). Ces problèmes se résument à retrouver l'information musicale, vidéo ou autre à partir d'un CD sale, rayé ou présentant un petit défaut de fabrication. Les normes de Philips exigent une correction d'une rayure de 0,2 mm, et le code CIRC permettra de corriger le paquet d'erreurs ou d'effacements de 4 096 bits consécutifs, ce qui correspond approximativement à une rayure d'un millimètre. Dans ce paragraphe, on se réfère à [7].

7.1 : Codes raccourcis

Théorème 7 - *Soit C un code linéaire [n, k, d] et $t \leqslant k$. Soit G une matrice génératrice de C de la forme $G = (A \mid B)$ où A est une matrice de taille $k \times t$ et de rang t. Le code C' dont les mots sont ceux de C dont on a effacé les t premières coordonnées, est un code [n–t, k–t, $d' \geqslant d$] appelé code raccourci de C.*

Preuve : Notons $G = [g_1, \ldots, g_n] = (A \mid B)$. Les mots du code C sont décrits par les produits xG quand x parcourt $\mathbb{F}_q$ soit $xG = ((x \mid g_1), \ldots, (x \mid g_n))$ où $(.|.)$ désigne la forme bilinéaire symétrique non dégénérée standard sur $\mathbb{F}_q^k$. Soit $C'' = \{c \in C / c_1 = \ldots = c_t = 0\}$, et notons $(e_1, \ldots, e_n)$ la base canonique de $\mathbb{F}_q^n$. Vérifions que la projection p sur $\bigoplus_{i=t+1}^{n} \mathbb{F}_q e_i$ parallèlement à $\bigoplus_{i=1}^{t} \mathbb{F}_q e_i$ induit un isomorphisme de C'' sur C'. $p : C'' \to C'$ est clairement injective et $p(C'') \subset C'$. Si $(c_{t+1}, \ldots, c_n) \in C'$, alors il existe $c = (c_1, \ldots, c_t, c_{t+1}, \ldots, c_n) \in C$. Comme A est de rang t, chaque vecteur $e_1, \ldots, e_t$ appartient à C, et $c - (c_1, \ldots, c_t, 0, \ldots, 0) = (0, \ldots, 0, c_{t+1}, \ldots, c_n)$ aussi et l'on a bien $C' \subset p(C'')$. Ainsi C' et C'' ont même dimension. Le sous-espace vectoriel

$$C'' = \{c \in \mathbb{F}_q / \exists x \ c = xG \text{ et } (x \mid g_1) = \ldots = (x \mid g_t) = 0\}$$

est l'image par l'application injective $x \mapsto xG$ de l'orthogonal $(\mathrm{Vect}(g_1, \ldots, g_t)^{\perp}$ dans $\mathbb{F}_q^k$, donc $\dim C'' = k - t$ et C' sera bien un code $[n - t, k - t]$.

Enfin, tout mot de code $c' \in C'$ est le projeté par p d'un mot $c' = (0, \ldots, 0, c)$ de C'' dont les t premières coordonnées sont nulles. Comme $C'' \subset C$, le poids $w(c')$ de c' vérifiera $w(c') = w(c) \geqslant d$ et la distance minimale de C' sera $\geqslant d$. ■

Le code raccourci d'un code de Reed Solomon conserve la même distance minimale. En effet, si d désigne la distance minimale du code de Reed Solomon C, et d' celle du code raccourci C', le théorème précédent, et la borne de Singleton permettent d'écrire

$$d \leqslant d' \leqslant (n-t) - (k-t) + 1 = d.$$

L'intérêt des codes raccourcis est de fournir une méthode de construction de k nouveaux codes de longueurs plus petites à partir d'un seul code de paramètres $[n, k, d]$ tout en conservant la capacité de correction du code, celle-ci étant directement liée à la distance minimale. C'est d'autant plus important que de nombreuses classes de codes, tels les codes BCH, ont des paramètres très spéciaux qu'il s'agit d'adapter aux diverses contraintes techniques, que les trois paramètres n, k, d d'un code ne sont pas indépendants entre eux mais liés par de nombreuses relations (telles les bornes de Hamming ou celle de Singleton). En règle générale, fixer 2 de ces paramètres revient à déterminer le troisième.

7.2 - Codes démultipliés

Dans la grande majorité des cas, des problèmes de lecture d'un disque compact surviennent lorsque le disque est sale, lorsque sa surface est rayée ou présente un défaut de fabrication. La perte d'information a lieu sur un nombre "important" de bits contigus, donnant naissance à ce qu'on appelle des "paquets d'erreurs". La correction de paquets d'erreurs est donc tout à fait typique des correcteurs présents sur un lecteur de CD, et fait en particulier appel aux codes démultipliés.

Soit C un code $[n, k]$ sur $\mathbb{F}_q$. Notons $q = p^m$ où p est premier, et appelons α un élément primitif de $\mathbb{F}_q$ (i.e. une racine primitive $(q-1)$-ième de l'unité dans $\mathbb{F}_q$. On sait que $(1, \alpha, \ldots, \alpha^{m-1})$ est une base du $\mathbb{F}_p$-espace vectoriel $\mathbb{F}_q$. Si $(x_{i1}, \ldots, x_{im})$ désignent les coordonnées de $x_i \in \mathbb{F}_q$ dans cette base, on peut définir l'application :

$$\begin{array}{ccccc} \psi : & \mathbb{F}_q & \to & \mathbb{F}_p^{nm} \\ & (x_1, \ldots, x_n) & \mapsto & (x_{11}, \ldots, x_{1m}, x_{21}, \ldots, x_{n1}, \ldots, x_{nm}) \end{array}$$

ψ est un isomorphisme de $\mathbb{F}_q$-espaces vectoriels appelés **démultiplication** et la bijection réciproque ψ^{-1} est appelée **contraction**. Le code C' démultiplié de C n'est autre que l'image $\psi(C)$ de C par ψ. C'est un code de paramètres $[nm, nk]$ sur $\mathbb{F}_p$. Si d désigne la distance minimale de C, rappelons que C corrige $e = [(d-1)/2]$ erreurs. Un mot $x = (x_1, \ldots, x_n)$ de longueur n de C sera démultiplié en un mot x' de longueur nm de C' suivant le schéma

x: x_1 x_2 ... x_e ... x_n

x': $x_{11}, \ldots, x_{1m}$ $x_{21}, \ldots, x_{2m}$ ... $x_{e1}, \ldots, x_{em}$ ... $x_{n1}, \ldots x_{nm}$

S'il y a moins de $m(e-1)+1$ erreurs consécutives dans le mot x', il y aura moins de e coordonnées erronées dans x, la correction de x est possible. Autrement dit :

Théorème 8 : *Si un code sur $\mathbb{F}_{pm}$* corrige *e erreurs, son démultiplié sur $\mathbb{F}_p$ corrigera jusqu'à $m(e-1)+1$ erreurs consécutives.*

7.3 - Effacements

Il arrive souvent que le lecteur ne puisse déterminer la valeur du bit qu'il est en train de lire. On dit qu'on a affaire à un effacement. Le problème des effacements est un peu différent de celui de l'erreur dans le sens où le décodeur sait qu'il y a eu disparition d'information à l'endroit occupé par le symbole effacé. Dans le cas d'une erreur, le décodeur est a priori incapable de dire si le symbole proposé est vrai ou faux. Cette différence est mise en valeur dans les deux résultats suivants :

Théorème 9 : *Un code (éventuellement non linéaire) de distance minimale d peut corriger $f = d - 1$* effacements.

Preuve : Si c et c' sont deux mots de code qui possèdent $n-(d-1)$ coordonnées identiques, alors c et c' diffèrent sur au plus $d-1$ coordonnées et $d(c, c') \leqslant d-1$. Cela entraîne $c = c'$.■

Théorème 10 : *Un code (éventuellement non linéaire) de long ueur n et de distance minimum $d \geqslant 2e+f+1$ corrige e erreurs et f effacements.*

Preuve : Soit $x = (x_1, \ldots, x_n) \in C$ et x'le mot entaché de $p \leqslant e$ erreurs et $q \leqslant f$ effacements. Soit H l'ensemble des mots de C identiques à x sur les $n-q$ coordonnées non effacées. Il faut montrer l'existence d'un seul mot du code C situé à une distance $\leqslant e$ de H. Raisonnons par l'absurde : si y et z sont deux mots de C et si $h,k \in H$ vérifient $d(y, h) \leqslant e$ et $d(z,k) \leqslant e$. Alors

$$d \leqslant d(y,z) \leqslant d(y,h) + d(h,k) + d(k,z) \leqslant 2e + q \leqslant 2e + f$$

ce qui est contraire à l'hypothèse.■

7.4 - Entrelacements

L'objectif est encore ici de corriger de gros paquets d'erreurs. Considérons t mots $x_1, x_2, \ldots, x_t$ d'un code C et écrivons ces mots sur t lignes. On obtient le tableau suivant :

Mot x_1 :	x_{11}	x_{12}	...	x_{1n}
Mot x_2 :	x_{21}	x_{22}	...	x_{2n}
⋮	⋮	⋮		⋮
Mot x_t :	x_{t1}	x_{t2}		x_{tn}

Au lieu d'envoyer ce tableau ligne après ligne, on décide de l'envoyer colonne après colonne. On considère par exemple le mot

$$x_{11}x_{21}\ldots x_{t1}x_{12}x_{22}\ldots x_{t2}\ldots x_{tn}$$

obtenu par l'entrelacement de t mots de C. L'ensemble de ces mots forme un code C' de longueur nt et de dimension kt, appelé **code entrelacé de profondeur t du code C**. Si C corrige l erreurs consécutives, l'examen du tableau ci-dessus permet de voir que C' corrigera des paquets d'erreurs de longueur lt, ce qui constitue une amélioration substancielle.

Présentons maintenant une autre technique d'entrelacement très efficace. Considérons toujours les mêmes mots $x_1, \ldots, x_t$ de C. La technique d'entrelacement avec un retard r consiste dans un premier temps à construire le tableau suivant dans lequel la première ligne est formée des premières coordonnées des mots $x_1, \ldots, x_t$, la seconde ligne est formée des secondes coordonnées de ces mots, mais décalées sur la droite de r symboles 0, et ainsi de suite sans oublier de décaler de r symboles vers la droite à chaque passage à la ligne.

$\underline{x_{11}}$	x_{21}	...	...	...	...	x_{t1}	0	0	...	0
0	0	...	0	$\underline{x_{12}}$	x_{22}	...	...	x_{t2}	0	⋮
⋮										0
0	0	...	...	...	...	...	0	$\underline{x_{1n}}$	x_{2n}	... x_{tn}

On transmet ensuite les colonnes de ce tableau pour obtenir un nouveau mot qui appartient à un code C''. C'' est le **code entrelacé de profondeur t avec un retard r du code C.**

Les symboles du premier mot x_1 ont été soulignés dans le tableau. r colonnes successives du tableau ne contiennent qu'un symbole figurant dans x_1. Par suite, si l'on perturbe les symboles de lr colonnes de ce tableau auxquels on peut rajouter l symboles "limitrophes", on touche à moins de l symboles consécutifs de chacun des motys $x_1, \ldots x_t$ de C. Cela signifie que si C corrige l erreurs consécutives, alors C'' corrigera $l(rn + 1)$ erreurs consécutives.

Exemple : Pour écrire l'entrelacement des trois mots

Mot a : a_1 a_2 a_3 a_4
Mot b : b_1 b_2 b_3 b_4
Mot c : c_1 c_2 c_3 c_4

avec un retard $r = 2$, on construit le tableau

$$\begin{array}{cccccccccc} a_1 & b_1 & c_1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & a_2 & b_2 & c_2 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & a_3 & b_3 & c_3 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & a_4 & b_4 & c_4 \end{array}$$

et le mot entrelacé avec retard sera

$$a_1000b_1000c_1a_2000b_2000c_2a_3000b_3000c_3a_4000b_4000c_4$$

18 symboles

Ici, $(n,t,r) = (4,3,2)$ et si C corrige des paquets de $l = 2$ erreurs, alors C'' corrigera des paquets de 18 erreurs.

7.5 - Code CIRC

Soit C le code de Reed-Solomon de polynôme générateur

$$g(x) = (x - \alpha)(x - \alpha^2)(x - \alpha^3)(x - \alpha^4)$$

où α est un élément primitif de $\mathbb{F}_{256}$. C est un code [255,251,5] sur $\mathbb{F}_{256}$ qui permet de construire deux codes raccourcis de distance minimale égale à 5 (cf. §7.1) :

- Un code C_1 de paramètres [28,24,5] - Un code C_2 de paramètres [32,28,5].

Après échantillonnage et quantification, le signal sonore se présente en trames de 24 octets. Ces 24 octets servent de symboles d'information du code C_1 et donnent naissance à des mots de 28 octets. Ces mots sont ensuite entrelacés avec une profondeur de 28 et un retard de 4 pour servir de symboles d'information du code C_2. Voyons maintenant comment s'opère le décodage des trames de longueur 32 lues par l'appareil...

• Action de C_2 : Le code C_2 est capable de corriger 2 erreurs, mais on l'utilise pour n'en corriger qu'une. Il peut détecter 3 erreurs. Si l'on note x le mot de $\mathbb{F}_{256}^{28}$ lu par l'appareil, trois cas sont possibles :

a1) Si $x \in C_2$, on estime que x représente le mot correct et qu'il n'y a pas eu d'erreur,

a2) Si x est à une distance de C_2 égale à 1, on estime qu'il y a eu une seule erreur, et le code C_2 la corrige,

a3) Dans tous les autrtes cas, on remplace x par 28 effacements qui seront désentrelacés puis envoyés sur le décodeur de C_1.

Les cas a1 et a2 peuvent donner lieu à une erreur de décodage, mais la probabilité d'erreur est infime. Pour le voir, supposons que tous les mots de $\mathbb{F}_{256}^{28}$ soient équiprobables, ce qui constitue une hypothèse pénalisante. Le mot x sera mal décodé par le processus a1 ou a2 dès que x appartient à l'une des boules fermées de rayon 1 centrée sur un mot du code C_2 sans être le centre de cette boule. Une boule fermée de rayon 1 de $\mathbb{F}_{256}^{28}$ contient $1 + 255.C_{32}^{1}$ mots. C_2 contient 256^{28} mots et $\mathbb{F}_{256}^{32}$ en contient 256^{32}. La probabilité d'une erreur au décodage sera donc majorée par :

$$\frac{256^{28}\left(1 + 255.C_{32}^{1}\right)}{256^{32}} \approx 1{,}9.10^{-6}$$

ce qui est négligeable. Le même raisonnement pratiqué dans le cas où l'on utiliserait la double capacité de correction d'erreurs de C_2 fournirait le majorant suivant, bien trop grand pour être retenu :

$$\frac{256^{28}\left(1 + 255.C_{32}^{1} + 255^{2}.C_{32}^{2}\right)}{256^{32}} \approx 7{,}5.10^{-3}$$

• Action de C_1: Dans les cas a1 et a2, le mot, éventuellement corrigé par C_2 est désentrelacé puis décodé par C_1 pour fournir l'information originelle. Plaçons-nous maintenant dans le cas a3. Les 28 symboles du mot ont été effacés et constituent une colonne de la matrice d'entrelacement. Pour le voir, il faut se référer au §7.4 et bien comprendre qu'ici, nous avons entrelacé 28 mots de $\mathbb{F}_{256}^{28}$ donnés en lignes dans le tableau

Mot x_1:	$x_{1,1}$	$x_{1,2}$	$\cdots$	$x_{1,28}$
Mot x_2 :	$x_{2,1}$	$x_{6,2}$	$\cdots$	$x_{2,28}$
$\vdots$	$\vdots$	$\vdots$		$\vdots$
Mot x_{28} :	$x_{28,1}$	$x_{28,2}$	$\cdots$	$x_{28,28}$

et que la matrice d'enlacement est

$\underline{x_{1,1}}$	$x_{2,1}$	$x_{3,1}$	$x_{4,1}$	$\cdots$	$\cdots$					$x_{28,1}$
0	0	0	0	$\underline{x_{1,2}}$	$x_{2,2}$	$x_{3,2}$	$x_{4,2}$	$\cdots$		$x_{28,2}$
$\vdots$								$\ddots$		$\vdots$
0	0	0	0	0	$\cdots$	$\cdots$	0	$\underline{x_{1,28}}$	$x_{2,28}$	$\cdots$ $x_{28,28}$

C_1 corrige 4 effacements donc sera capable de corriger $4 \times 4 = 16$ colonnes entières de cette matrice d'entrelacements. Chacune de ces 16 colonnes de 28 symboles provient d'un mot de 32 symboles (soit 32×8 bits 0 ou 1) avant traitement par le code C_2. Par suite, le code CIRC sera capable de corriger des effacements contigus de longueur $16 \times 32 \times 8 = 4096$ bits sur la surface du disque compact, ce qui correspond à des rayures d'environ 1,23 mm sur le disque et se trouve bien au-dessus des normes de Philips (0,2 mm).

Références

[1] Arnoux P., *Minitel, codage de l'information et corps finis*, Pour la Science n° 125, mars 1988.

[2] Epreuve de l'Informatique de l'Agrégation externe de mathématiques, session 1996, *rapport du jury*, CNDP, 1996.

[3] Lidl R. & Niederreiter H., "*Finite Fields*", Encyclopedia of Mathematics and its Applications, vol. 20, Addison-Wesley Pubblishing Company, 1983.

[4] Mercier D-J, "*Cryptographie Classique et Cryptographie publique à clé révélée*", A.P.M.E.P., *Bulletin* n°406, septembre-octobre 1996.

[5] Mac William F.J. & Sloane N.J.A., "*The theory of Error-Correcting Codes*", North-Holland Mathematical Library, vol. 16, 2nd reprint 1983.

[6] Papini O. & Wolfmann J., "*Algèbre discrète et codes correcteurs*", Springer-Verlag, Mathématiques & Appllications n° 20, 1995.

[7] Zanotti J.P., "*Codage d'un signal Audionumérique sur un Support à Lecture Optique, Erreurs au décodage et Codes MDS*", Mémoire de DEA, Université d'Aix-Marseille II, Faculté des Sciences de Luminy, 1992.