Codes Détecteur et Correcteurs des Erreurs (CCE).

N. Lebedev

Table des matières

1 Introduction

1.1 Codage—solution algorithmique pour canaux bruités

La théorie de l'information et celle du codage sont les solutions « systèmes » , algorithmiques, pour la transmission à travers un canal bruité, contrairement aux autres solutions de type « matérielle », comme, par exemple, augmentation de la puissance de transmission, de la fiabilité des composants électroniques, et donc du coût.

Le canal est accepté tel qu'il est, avec un niveau de bruit qui n'est pas contrôlable par le concepteur du système de communication. Mais les dispositifs homologues du codage à l'émission et du décodage à la réception sont insérés dans le système avant et après le canal, afin de i) détecter les erreurs ii) en corriger certaines, Fig. 1. Ces techniques portent le nom des Codes Correcteurs des Erreurs (CCE).

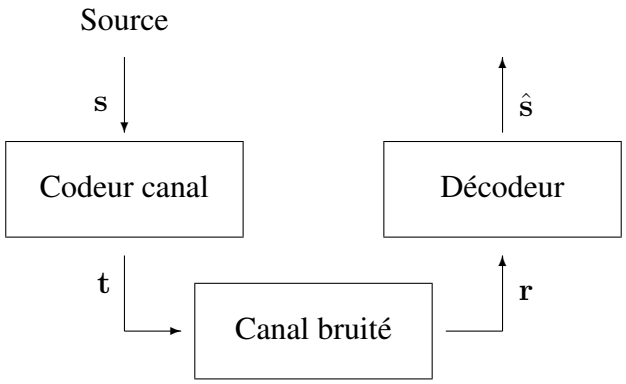

FIGURE 1 – Système de communication pour un canal bruité.

— Le codeur transforme le message issu de la source (ou codeur-compresseur de la source de type Huffman, LZW, codec voix) typiquement binaire $\mathbf{s}$, en un message codé $\mathbf{t}$ en rajoutant les bits de contrôle, donc les bits en plus, en créant de la redondance.
— Les bits $\mathbf{t}$ ainsi codés sont transmis dans le canal qui peut les altérer, en les transformant en message reçu $\mathbf{r} \neq \mathbf{t}$.
— Le décoder exploite ces bits de contrôle et essaie de de corriger les erreurs du canal pour produire une estimation $\hat{\mathbf{s}}$ du message source. Si toutes les erreurs n'ont pas pu être corrigées, $\hat{\mathbf{s}} \neq \mathbf{s}$, il s'agit des erreurs résiduelles, après le décodage. Cet événement sera caractérisé par la probabilité d'erreur résiduelle, que l'on souhaite la plus faible possible.

1.2 Codage canal : apports et compromis

Selon [Sklar 6.3.4, p.323, Fig 6.9] , pour différents scénarios de communication, différents apports et inconvénients du codage peuvent être analysés. Considérons la courbe sur Fig. 2 qui esquisse la probabilité d'erreur sur bit (PEB) pour les cas avec et sans codage, en fonction du ratio de l'énergie par bit et celle du bruit, sur la durée de transmission d'un bit dans un canal gaussien.

1.2.1 Compromis 1 : probabilité d'erreur vs bande passante

Soit, un système de communication conçu pour un fonctionnement type en point A. Mais la liaison est de mauvaise qualité, et la probabilité d'erreur qui peut être atteinte est seulement $P_b = 10^{-2}$, en moyenne, un bit sur cent se trouve erroné. Comment la réduire ?

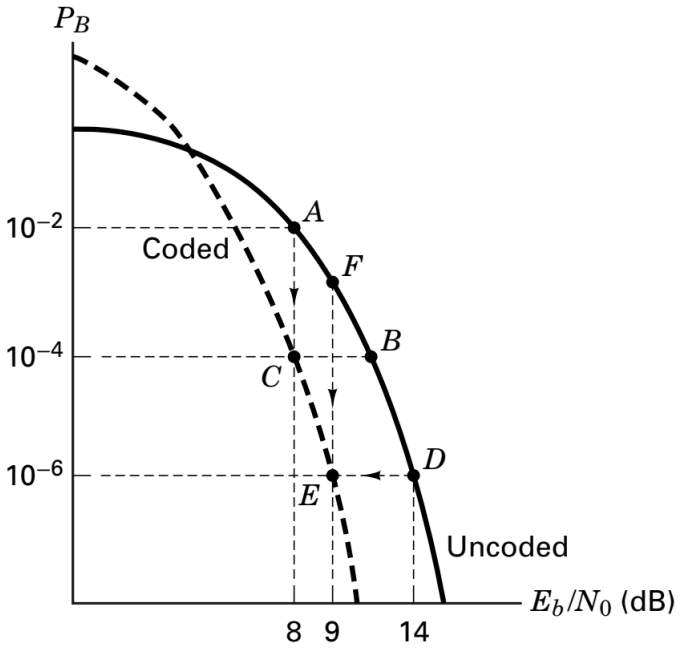

FIGURE 2 – $P_b = f(E_b/N_0)$ *codée et non-codée pour sur canal gaussien. [© Sklar].*

— 1ère idée : augmenter le RSB (Rapport Signal-sur-Bruit) pour déplacer le point d'opération en B, ($A \rightarrow B$). Mais cela peut être impossible si en A le système a déjà le ratio E_b/N_0 maximal. Par exemple, la puissance de transmission est maximale $P_t = max$ et le récepteur sur canal radio ou câble se trouve à la distance maximale d_{max}. Donc,

— 2nd idée : codage canal qui permet de réduire P_b (passage de $A \rightarrow C$), mais au prix de la redondance—la transmission de bits de contrôle en plus, avec le taux de codage $R_c = (k/n) \leq 1$.

- $R_b = 1/T_b$ fixé. Si l'objectif est de préserver le débit $R_b = 1/T_b$ constant des données utilisateur (application) non-codées, par exemple pour un système temps-réel, le débit des données codées transmises dans le canal $R_{b,coded} = 1/T_{b,coded}$ devra être plus élevé, car plus de bits, chacun sur un temps plus court, sont transmis dans le canal durant le même intervalle de temps, à savoir, $kT_b = nT_{b,coded}$. Cela mène à la dilatation de la bande (BW—bandwidth) utilisée par le signal par le facteur $n/k = 1/R_c$, donc $R_b = R_c R_{b,coded}$, qui peut être possible ou non, notamment, si plusieurs canaux en fréquence sont mutliplexés.
- $R_{chan} = R_{b,coded} = 1/T_{b,coded}$ fixé. Pour ce dernier cas, si le débit canal et donc l'occupation spectrale du signal est déjà imposée, maximale, R_{chan}. Avec le codage, ce débit sera celui de transmission des bits codés $R_{chan} = R_{b,coded}$, et, par conséquent, le débit utile des données non-codées, perçu par l'utilisateur, sera réduit par le facteur de taux de codage R_c, et comme avant $R_b = R_c R_{chan} = R_c R_{b,coded}$.

On a donc la relation suivante pour les deux cas :

$$R_c = k/n = T_{b,coded}/T_b = R_b/R_{b,coded} \leq 1.$$

En résumé :

— un des apports du codage est de permettre de réduire la probabilité d'erreur P_b pour le même RSB

— au prix de i) débit utilisateur réduit, et ii) plus grande complexité du système en matériel/logiciel pour codage-décodage.

La théorie de l'information s'intéresse aux modèles des canaux, des codes, et limites théoriques de leur aptitude de correction, tandis que la théorie du codage étudie les aspects pratique de conception et de la mise en œuvre des codeurs-décodeurs. Différentes techniques de CCE existent :

1. Code de parité
2. Codes en blocs linéaires
   — Codes en blocs
   — Codes cycliques
   — Codes BCH
      — Codes de Hamming
      — Codes Reed-Solomon
3. Codes linéaires en mode flux : codes convolutifs

1.3 Principes de codage—codes en blocs $\mathcal{C}(n, k)$

Un code par blocs fragmente les bits produits par une source en blocs de (k) bits et les convertit en blocs de (n) **bits appelés mots-codes (*CW—codewords)***, on écrit un code $\mathcal{C}(n, k)$, Fig 3.

FIGURE 3 – Codage par blocs.

La redondance est quantifiée par le taux ou le ratio de codage, qui correspond au facteur de réduction du débit utile vu à la fois par la source qui produit les bits non-codés, et par le destinataire de l'information utile décodée.

Déf 1.1. *Taux ou ratio du codage :* $R_c = \frac{k}{n} \leq 1$.

Codage $\Longleftrightarrow$ Redondance, rajout des bits de contrôle

Notons, qu'il y a 2^k messages source vecteurs $\mathbf{u} = [u_{k-1}, \ldots, u_0]$ chacun de longueur (k) bits qui sont chacun encodés sur 2^k mots-codes $\mathbf{c} = [c_{n-1}, \ldots, c_0]$ correspondants, chacun de longueur (n) bits. Les autres 2^{n-k} mots de longueur (n) bits ne sont pas les mots-codes, mais résultent de la transmission erronée dans le canal !

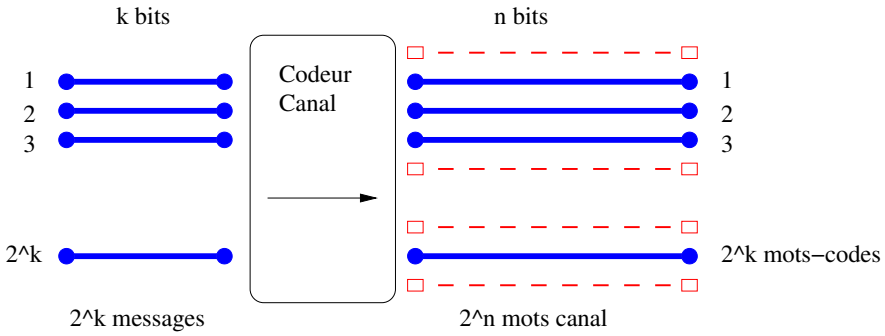

FIGURE 4 – *Redondance du codage en blocs : messages et mots-codes.*

1.3.1 Exemple 1 : Code de contrôle de parité « rectangulaire »$(n, k) = (24, 15)$

Le codage consiste à écrire les $k = 15$ bits en entrée dans une table de $m = 3$ lignes et $\ell = 5$ colonnes, $k = m\ell$. Puis, de calculer la parité par ligne et par colonne, puis la parité sur colonne et ligne du résultat. La table est ensuite relue ligne par ligne, pour donner $(m+1)(\ell+1) = n = 24$ bits codés. Le taux de codage ici est $R_c = 15/24 = 5/8$.

$$\begin{array}{rc|c} & \ell = 5 & \\ & (10101) & 1 \\ m = 3 & (01010) & 0 \\ & (11011) & 0 \\ \hline & (00100) & 0 \end{array} \tag{1}$$

Propriétés. Afin d'examiner les propriétés de correction et de détection des erreurs par ce code, il faut faire l'hypothèse sur la nature des erreurs qui peuvent impacter le mot-code de longueur (n), et donc sur le canal qui les induit. Ici, le canal CBS (Canal Binaire Symétrique) avec la probabilité d'erreur p est supposé. Pour rappel, le CBS est présenté sur la Fig. 5.

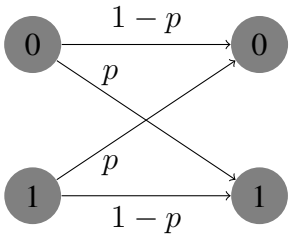

FIGURE 5 – CBS—Canal Binaire Symétrique.

Notons, qu'une seule erreur sur un bit transmis par le canal, va mener à l'erreur de contrôle de parité ligne et colonne. Ainsi, ce code a une aptitude de correction $t_{corr} = 1$ erreur (*). Si plus d'une erreur se produit, ce code est incapable de les corriger, il y aura donc les bits erronés dans le bloc reçu, sans que l'on puisse dire lesquels et combien. Cet événement est quantifié via la **probabilité d'erreur résiduelle après le décodage**, qui correspond à la probabilité pour le canal d'avoir introduit un nombre d'erreurs plus grand que le code n'est capable de corriger dans un mot-code de (n) bits ; pour ce code, c'est > 2 erreurs.

Sous l'hypothèse que les erreurs sur les bits sont équiprobables et indépendantes (caractéristique du canal CBS), la probabilité d'avoir (j) erreurs dans un mot-code de (n) bits suit la loi binomiale :

$$P(j, n) = \binom{n}{j} p^j (1-p)^{n-j} \, . \tag{2}$$

Pour un code capable de corriger au plus (t_{corr}) erreurs, mais pas $(t_{corr} + 1), (t_{corr} + 2), \ldots, n$ erreurs, la probabilité d'erreur résiduelle par bloc de (n) bits, P_B, est la somme des probabilités de tels événements :

$$P_B = \sum_{j=t_{corr}+1}^{n} \binom{n}{j} p^j (1-p)^{n-j} \simeq \binom{n}{t_{corr}+1} p^{t_{corr}+1} (1-p)^{n-(t_{corr}+1)} \, , \tag{3}$$

où l'approximation vient du fait que lorsque $p \ll 1$, le premier terme de la somme avec $p^{t_{corr}+1}$ en facteur, est dominant .

La probabilité d'erreur binaire, P_b dépend de nombreux éléments, notamment, du principe de codage et de décodage. Pour ce code rectangulaire, elle peut être approximée par :

$$P_b \approx \frac{1}{n} \sum_{j=t_{corr}+1}^{n} j \binom{n}{j} p^j (1-p)^{n-j}. \tag{4}$$

Cette démarche permet de mettre en évidence l'utilité et le gain du codage : pour $p \ll 1$, la probabilité d'erreur résiduelle avec le codage est inférieure à celle sans codage $P_B(n = 24, k = 15) < p$. Or, le codage permet d'avoir la transmission de meilleure qualité, en réduisant le nombre de bits erronés reçus.

Exemple numérique. Pour ce code rectangulaire $(24, 15)$ sur BSC avec $p = 0,01$:

$$P_B(n = 24, k = 15) = \sum_{j=2}^{n} \binom{24}{j} p^j (1-p)^{24-j} \simeq \binom{24}{2} p^2 (1-p)^{22} \ = 0,022. \tag{5}$$

Or, $P_b(n = 24, k = 15) = \frac{2}{n} P_B \approx 0,002$, ainsi $P_b < p$. ■

Codage $\Longleftrightarrow$ Réduction de la probabilité d'erreur après décodage

1.3.2 Exemple 2 : Code par répétition R_N (sur CBS)

Bits (s)ource **s**	Bits codés (t)ransmis **t**
0	000
1	111

TABLE 1 – Mots-codes pour un code à répétition R_3.

Intuitivement, répéter le message N fois augmenterait ses chances pour être correctement reçu. C'est une approche utilisé dans un codage à répétition R_N. Soit un code à répétition R_3 (Tab. 1) qui simplement retransmet à travers le canal chaque bit issu de la source $N = 3$ fois. Ce code, pour tout N n'a que deux mots-codes— [0 0 0] et [1 1 1]. Supposons qu'un message émis par la source **s** :

$$\mathbf{s} = 0 \ \ 0 \ \ 1 \ \ 0 \ \ 1 \ \ 1 \ \ 0$$

est transmis à travers le CBS décrit par $p = 0,1$ en utilisant ce code à répétition. L'effet du canal peut être décrit par l'inversion aléatoire indépendante de chaque bit, représenté par addition $\mathrm{mod}\ 2$ (1+1=0) du vecteur du bruit **n** au vecteur transmis **t**, résultant en vecteur reçu **r** , Fig. 6.

$$\mathbf{r} = \mathbf{t} + \mathbf{n}$$

s	0	0	1	0	1	1	0
t	000	000	111	000	111	111	000
n	000	001	000	000	101	000	000
r	000	001	111	000	010	111	000

FIGURE 6 – Exemple de transmission avec un code R_3.

Comment décoder chaque mot-code dans le vecteur reçu **r** ? L'approche intuitive sera correcte : pour chaque mot-code de $n = 3$ bits, faire un vote par majorité—décider en faveur de $\hat{s} = 0$ si

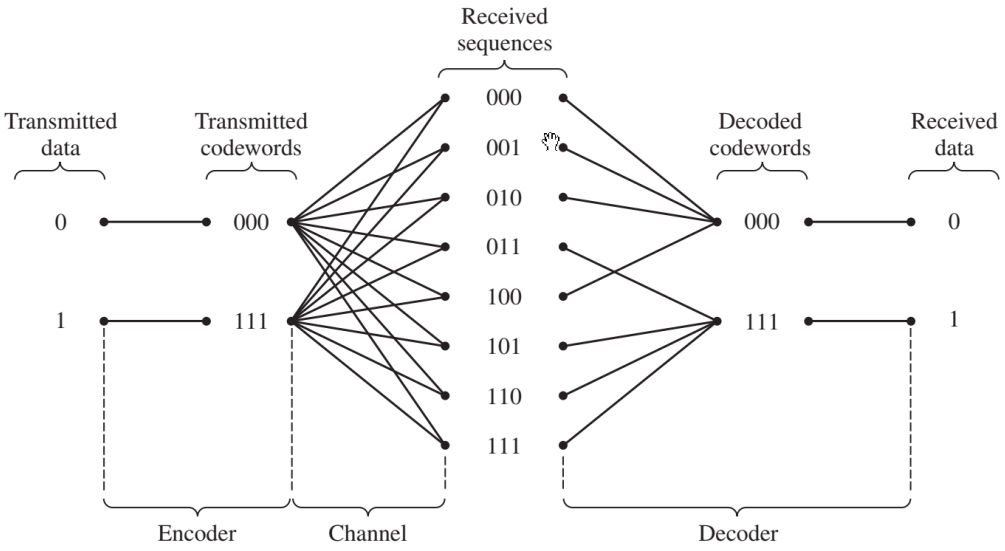

FIGURE 7 – *Code à répétition R_3 : décodage.*

le nombre de ”0” est plus grand que le nombre de ”1”, et vice versa. Par ailleurs, cette approche sera optimale de point de vue du critère de *maximum de probabilité a posteriori* (voir Annexe **??**). Graphiquement, c'est illustré sur la Fig. 7.

Ce procédé de décodage appliqué au vecteur reçu de la Fig. 6, les messages décodés $\hat{\mathbf{s}}$ sont obtenus : le premier triplet est décodé comme 0, le second également comme 0, et une erreur sera corrigée. Mais toutes les erreurs ne peuvent pas être corrigées, comme c'est le cas pour le 5ème triplet, où le canal altère 2 bits, il y a alors erreur du décodage, le bit détecté $\hat{s}_5 \neq s_5$ n'est pas celui transmis.

s	0	0	1	0	1	1	0
t	000	000	111	000	111	111	000
n	000	001	000	000	101	000	000
r	000	001	111	000	010	111	000
$\hat{\mathbf{s}}$	0	0	1	0	0	1	0
corrected errors		⋆					
undetected errors					⋆		

FIGURE 8 – Décodage par ”vote majoritaire”

Cette approche met en évidence la notion de la distance (de Hamming, section 2.2.2) entre les mots-codes. Lorsqu'un mot-code $cw_0 = [0\ 0\ 0]$ est transmis, et le bruit altère un seul bit, par exemple, le 1er, le vecteur bruité reçu $[1\ 0\ 0]$ du second triplet n'est pas un mot-code.

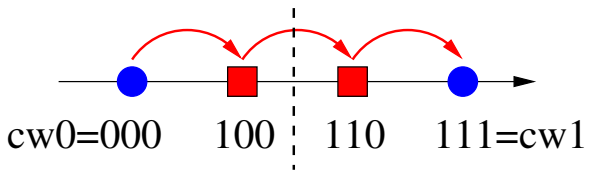

FIGURE 9 – *Code à répétition R_3 : mots-codes et erreurs.*

Il ne diffère du mot-code $[0\ 0\ 0]$ que d'un seul bit, les deux autres étant identiques, le décodeur prendra alors la décision en faveur du cw_0, **tout en corrigeant 1 erreur !**, et décidera en faveur du bit $\mathrm{s} = 0$ décodé. Mais toutes les erreurs ne peuvent pas être corrigées, comme c'est le cas pour le 5ème triplet, où le canal altère 2 bits, par exemple, le 1er et le 2nd bits.

Le vecteur bruité reçu $[1\ 1\ 0]$ ne diffère que d'un bit du mot-code $cw_1 = [1\ 1\ 1]$, c'est en faveur de ce dernier que le décodeur prendra la décision, pour décoder le bit $\mathrm{s} = 1$. Il y a alors erreur du décodage, le bit détecté $\hat{s}_5 \neq s_5$ n'est pas celui transmis. La Fig. 9 explicite cette notion.

Décodeur décide en faveur du mot-code « le plus proche » du mots reçu

Sachant que certaines erreurs ont pu être corrigées, on peut montrer (voir l'exercice) que pour ce code R_3 la probabilité d'erreur résiduelle après décodage $P_b \approx 0,03 < p = 0,1$, la probabilité d'erreur dans le CBS. Ce code a donc rempli son rôle.

Toutefois, cet avantage à un prix—réduction du facteur 3 du débit utile, car le taux de codage est $R_c = 1/3$, ou bien, si le débit source reste constant, l'augmentation du facteur 3 de la durée de la transmission, et donc $\times 3$ le prix d'une communication si facturée sur la durée !

On constate aussi que le gain n'est pas énorme, la probabilité d'erreur n'a même pas été divisée par 10. On peut inférer qu'en utilisant un code à répétition R_N avec N de plus en plus grand, on pourrait diminuer encore la probabilité d'erreur. Or, si celle-ci doit être rendue extrêmement faible, de l'ordre de 10^{-15}, par exemple pour le stockage des fichiers sur un support fiable pendant plusieurs années, combien de répétition seraient nécessaires pour encoder chaque bit (voir l'exercice) ?

1.3.3 Exemple 3. Probabilité d'erreur résiduelle par bit P_b pour un code à répétition R_N

Soit N impaire.

1. Déduire la probabilité d'erreur résiduelle pour ce code, supposant CBS avec $p = 0,1$.
2. Quel sera le terme dominant dans cette expression ?
3. En supposant $p = 0,1$, combien de répétitions sont nécessaires afin d'avoir $P_e \approx 10^{-15}$?

Solution.

1. Un code à répétition R_N a un taux de codage $R_c = 1/N$, et ne contient que deux mots-codes, le « tout à zéro » $[0\ 0, \ldots, 0]_{1\times N}$, et le « tout à un » $[1\ 1, \ldots, 1]_{1\times N}$. Or, une erreur de détection du mot-code de longueur N se produise, pour N impair, lorsque le canal alterne au moins (j) bits, où $j \geq (N+1)/2$. La probabilité d'erreur résiduelle par bloc est alors la somme des probabilités de ces cas d'erreurs :

$$P_B = \sum_{j=(N+1)/2}^{N} \binom{N}{j} p^j (1-p)^{N-j}, \tag{6}$$

2. dominée par le terme

$$\binom{N}{(N+1)/2} p^{(N+1)/2} (1-p)^{(N-1)/2} \tag{7}$$

3. Le calcul de ce dernier terme donnera : $N \approx 60$ répétitions seraient nécessaires. La conclusion serait très décevante : afin de pouvoir stocker un fichier de manière fiable sur un support (serveur) de mauvaise qualité avec $p = 0,1$, il aurait fallu... repliquer ce fichier sur ≈ 60-aine de serveurs de ce type ! Le taux de codage est $R_{N=60} = 1/60$, la redondance et donc le coût sont facteur 60, trop grand !

1.4 Limites de codage

La section précédente a mis en évidence un des compromis fondamentaux dans le codage entre **la probabilité d'erreur résiduelle** P_b qui est souhaitée la plus faible possible, et le taux de codage $R_c = k/n$ qui est une perte du débit utile. La Fig. 10 présente, parmi d'autres, les points (P_b, R) pour les codes à répétition, en linéaire à gauche, et en log à droite.

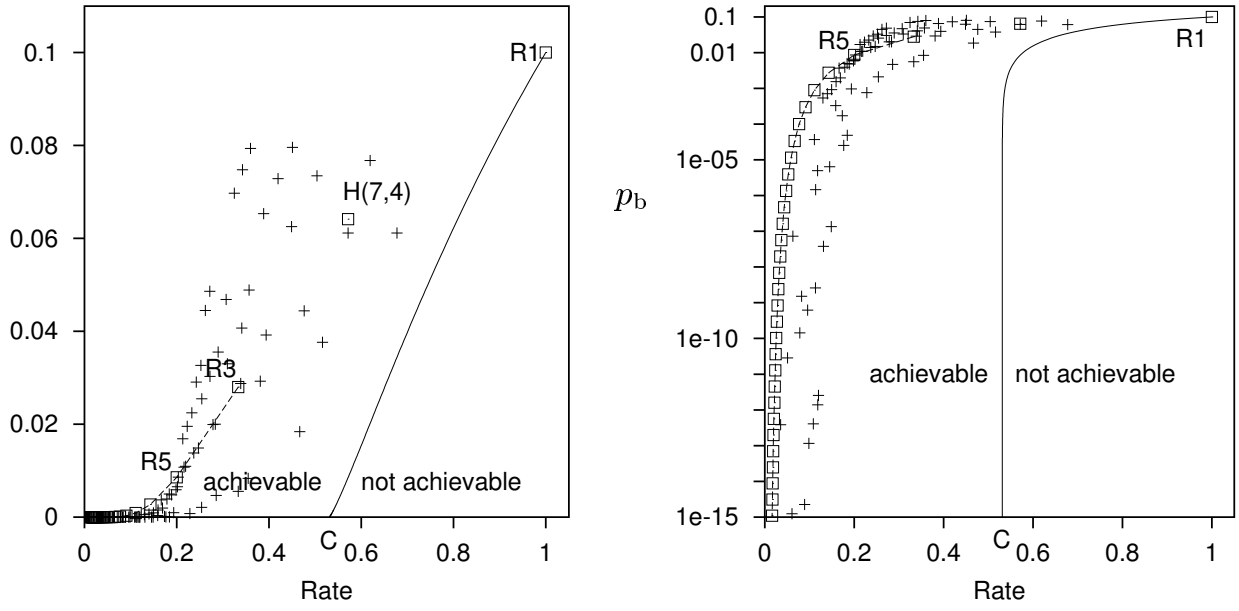

FIGURE 10 – *Shannon's noisy-channel coding theorem. The solid curve shows the Shannon limit on achievable values of* (P_b, R) *for the BSC with* $p = 0.1$. *[© MacKay]. "*$\square$*" — repetition codes ; "*$+$*" —other codes ; also the Hamming code* $(n, k) = (7, 4)$ *is shown.*

Plusieurs questions peuvent alors se poser :

1. Quelle est la région (P_b, R) qui peut être atteinte par les codes réalisable ? En d'autres mots, si une très faible P_b est souhaitée, le taux de codage, va-t-il lui aussi tendre vers 0, le temps de transmission allant jusqu'à l'infini ?
2. Si cette limite existe, est-il possible de construire les codes pratiques qui s'en rapprocheraient ?

La réponse à la première question a été donnée par C. Shannon en 1948 qui a établi la notion de capacité d'un canal bruité.

Théorème 1.1. *Codage pour un canal bruité. Pour tout canal, il existent les codes permettant d'atteindre la probabilité d'erreur aussi petite que souhaitée,* $P_b \to 0$ *au débit de codage* R_c *non-nul, proche d'une caractéristique propre d'un canal,* $R < C$*, sa capacité.*

Exemple 4 : La capacité d'un CBS avec $p = 0.1$**.**

$$C(f) = 1 - H_2(p) = 1 - \left[p \log_2 \frac{1}{p} + (1-p) \log_2 \frac{1}{1-p} \right], \tag{8}$$

où $H_2(p)$ est l'entropie binaire conditionnelle du canal CBS, avec la probabilité d'erreur p. Pour $p = 0,1$, la capacité du CBS est $C \simeq 0.53$. Cette limite est présentée sur la Fig. 10 par un trait continu. La conclusion immédiate remarquable, est que le taux de codage $R_c = 1/2$ permettrait

d'atteindre la capacité du CBS. Autrement dit, deux serveurs de mauvaise qualité seraient suffisants pour stocker le fichier, mais pas juste en le répliquant... ce qui amène la 2ème question :

Quel codage à utiliser ?

Cette question a donné naissance à la théorie du codage, et... 50 ans ont été nécessaires pour qu'en 1997 les turbo-codes et les codes LDPC soient inventés, permettant de s'approcher de la limite de Shannon pour les canaux point-à-point. Puis, les systèmes MIMO ont été introduits.

2 Codes en blocs linéaires

2.1 Linéarité

Rappelons que les codes en blocs effectuent une transformation d'un bloc ou vecteur de k bits d'information **u**tilisateur, un k-uplet noté $\mathbf{u}_{[1\times k]}$ en mot-code, (**c**odeword) un n-uplet $\mathbf{c}_{[1\times n]}$ de n bits, $n > k$. Seulement 2^k de ces n-uplets sont les mots-codes, notés $\mathbf{c}_i$.

Déf 2.1. *Linéarité. Le code en bloc est dit* **linéaire** *si*

- *chaque message* $\mathbf{u}_i$ *est transformé en mot-code* $\mathbf{c}_i$ *par une transformation linéaire* $\mathbf{u}_i \mapsto \mathbf{c}_i$.
- *si* $\mathbf{c}_i$ *et* $\mathbf{c}_j$ *sont les mots-codes, alors toute combinaison linéaire des mots-codes est un autre mot-code.*

$$a_i\mathbf{c}_i + a_j\mathbf{c}_j + \ldots + a_k\mathbf{c}_k = \mathbf{c}_\ell \in \mathcal{C} \ ,$$

où $a_i, a_j \in \{0, 1\}$ *et toutes les opérations se font* $\mod 2$

L'observation suivante est de l'ordre : un vecteur $[0, \ldots, 0]_{1\times n}$ est aussi un mot-code, car $\mathbf{c}_i + \mathbf{c}_i = \mathbf{0}$.

Question : Comment obtenir un code avec cette structure linéaire ?
Soit $\mathcal{V}_n$ ensemble de tous les 2^n n-tuples binaires, c'est à dire, vecteurs de $\dim\{\mathcal{C}\} = n$, dont les éléments sont $\in \{0, 1\}$. Tout ensemble $\mathcal{C}$ de 2^k n-tuples (mots-codes) formant un sous-espace de dimension k dans $\mathcal{V}_n$, est un code en blocs linéaire $\mathcal{C}(n, k) \in \mathcal{V}_n$. Les k vecteurs de base qui sont les mots-codes, permettent d'obtenir les autres par leurs combinaisons linéaires.

2.2 Formulation

2.2.1 Matrice-génératrice du code

La notation utilise les vecteurs pour les messages d'utilisateur $\mathbf{u} = [u_0, u_1, \ldots, u_{k-1}]$ et pour les mots-codes, $\mathbf{c} = [c_0, u_1, \ldots, c_{n-1}]$, et la matrice pour l'opérateur linéaire de codage, appelé $\mathbf{G}_{[k\times n]}$, matrice-génératrice du code, dont k lignes $\mathbf{g}_i, i = 1, ..., k$ sont les mots-codes formant la base de sous-espace du code.

$$\mathbf{c} = \mathbf{uG} = \sum_{i=0}^{k-1} u_i \cdot \mathbf{g}_i \tag{9}$$

$$\mathbf{G} = \begin{pmatrix} \mathbf{g}_0 \\ \mathbf{g}_1 \\ \vdots \\ \mathbf{g}_{k-1} \end{pmatrix} = \begin{pmatrix} g_{0,0} & g_{0,1} & \cdots & g_{0,N-1} \\ g_{1,0} & g_{1,1} & \cdots & g_{1,N-1} \\ \vdots & \vdots & \ddots & \vdots \\ g_{K-1,0} & g_{K-1,1} & \cdots & g_{K-1,N-1} \end{pmatrix} \tag{10}$$

En utilisation les combinaisons linéaires, c'est à dire les additions $\mod 2$ des lignes de $\mathbf{G}$, celle-ci peut être mise sous forme dite *systematique* , qui fait ressortir dans les k premières positions

du mot-code les k bits du message source u_k , suivi de $(n-k)$ bits dit de contrôle de parité :

$$\mathbf{G} = [\mathbf{I_K} \mid \mathbf{P}] = \begin{pmatrix} 1 & 0 & 0 & \dots & 0 & p_{0,0} & p_{0,1} & \cdots & p_{0,n-k-1} \\ 0 & 1 & 0 & \dots & 0 & p_{1,0} & p_{1,1} & \cdots & p_{1,n-k-1} \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots & \\ 0 & 0 & 0 & \dots & 1 & p_{k-1},0 & p_{k-1,1} & \cdots & p_{K-1,n-k-1} \end{pmatrix} . \tag{11}$$

$$\mathbf{c} = \mathbf{u} \times [\mathbf{I_K} \mid \mathbf{P}] = [u_0, u_1, \dots, u_{k-1}; p_0, \dots, p_{n-k-1}] , \tag{12}$$

où I_k est la matrice identité et $\mathbf{P}$ est le bloc de parité.

2.2.2 Distance de Hamming. d_{min} du code CBL

Il a été observé précédemment de manière empirique, que plus les mots-codes sont « éloignés » les uns des autres en termes du nombre de bits qui les différencie, moins ils sont vulnérables aux erreurs du canal. Formalisons ces notions.

Déf 2.2. *Poids de Hamming d'un mot-code* $\mathbf{c}$ *:*

$$w_h(c_i) = \textit{nombre de "1"} = \sum_{\ell=1}^{n} c_i(\ell)$$

Déf 2.3. *Distance de Hamming entre n'importe quels deux mots-codes* $\mathbf{c}_i$ *et* $\mathbf{c}_j$ *est :*

$$d_h(c_i, c_j) = w(c_i \oplus c_j) = w(c_k)$$

Comme c'est un code linéaire, la somme mod2 des mots de code c_i et c_j est un autre mot-code c_k, et le poids de ce dernier, "weight" $w(c_k)$ est donc la distance entre ces mots de code, c_i et c_j.

Le code est caractérisé par sa d_{min}, qui est le nombre minimal de positions dans lesquels n'importe quels deux mots de code diffèrent, donc le nombre de "1" dans le résultat de l'addition mod2, ou encore leur distance de Hamming d_h.

La différence minimale, c'est à dire, les paires de mots de code les plus proches correspondent aux mots de code les plus vulnérables au bruit, donc au pire cas qui limite les performances du code en termes de la détection et de la correction des erreurs.

Mais puisque $d_h(c_i, c_j) = w(c_k)$, pour caractériser la distance minimale d'un code, cela revient à trouver le(s) mot(s) de code de poids minimal (sauf le mot $[0...0]$). Pour résumer :

$$d_{min}(\mathcal{C}ode) = \min_{\{c\}} w(c)$$

Conclusion : parmi tous les mots de code, il y en a un certain nombre, notés c_m qui ont le poids min $= d_{min}$ du code. Cette notion de la distance minimale est fondamentale pour quantifier l'aptitude d'un code à détecter et à corriger les erreurs. Empiriquement, sur l'exemple de code à répétition à partir de la Fig. 9, on peut déduire les métriques suivantes du code en bloc :

Déf 2.4. *Aptitude de détection et de correction des erreurs*

$$t_{det} = d_{min}{-}1$$
$$t_{corr} = \lfloor \frac{d_{min}{-}1}{2} \rfloor .$$

Exemple 5 : quelle est la distance minimale d'un code à répétition R_N ?

2.3 Etude de cas : code de Hamming (7,4)

2.3.1 Définition du code de Hamming. Matrice génératrice

2.3.2 Matrice de contrôle de parité. Code dual.

2.3.3 Décodage par syndrome du code en blocs.

2.3.4 Exemple 6 : décodage par syndrome de code en blocs (5,2)

2.3.5 Exemple : Probabilité d'erreur du code de Hamming

(2*) Calculer les probabilités d'erreur résiduelles par bloc et binaire pour le code de Hamming $(7, 4)$ pour la transmission dans un canal CBS de $p = 0, 1$. Le taux de codage est :

$$R_c = \frac{k}{n} = 4/7$$

1. P_B, par (B)loc ; Dans le cas du code de Hamming, l'erreur du décodage se produit lorsque le bruit alterne $t_{corr}+1$ ou plus de bits dans un mot-code transmis. L'expression générale de P_B est donnée. Le terme dominant est celui de poids $t_{corr}+1 = 2$ qui est l'ordre $\mathcal{O}(f^{t_{corr}+1})$ pour ce code, ce qui permet les approximations qui suivent.

$$\begin{aligned} P_B &= \sum_{j=t_{corr}+1}^{n} \binom{n}{j} f^p (1-p)^{n-j} \simeq \binom{n}{t_{corr}+1} p^{t_{corr}+1} (1-p)^{n-(t_{corr}+1)} \\ &\simeq \binom{7}{2} p^2 (1-p)^5 = 21 p^2 (1-p)^5 \,. \end{aligned} \tag{13}$$

2. P_b, binaire ou par (b)it pour les bits d'un mot-code.
   Pour le calcul de p_b, la logique est la suivante : lorsqu'une erreur du décodage se produit, un autre mot-code que celui transmis est détecté, qui diffère en d_{min} bits (qui sont donc erronés) sur n transmis. Comme l'erreur affecte les bits de manière aléatoire, on peut écrire :

$$P_b \approx \frac{d_{min}}{n} \cdot P_B = \frac{3}{7} P_B. \blacksquare \tag{14}$$

3 Codes cycliques

3.1 Définition d'un code cyclique

Les codes cycliques sont une sous-catégorie des CBL, codes en blocs linéaires.

Déf 3.1. *Si* $\mathbf{c} = [c_{n-1}, c_{n-2}, \ldots, c_1, c_0]$ *est un vecteur représentant un mot-code d'un code cyclique, alors* $[c_{n-2}, c_{n-3}, \ldots, c_0, c_{n-1}]$ *obtenu par décalage cyclique de* $\mathbf{c}$ *est aussi un mot-code.*

Plus généralement, $(n-1)$ des 2^k mots-code (de longueur (n) bits) sont obtenus par décalage cyclique d'un seul mot-code.

Question. Comment sont obtenus les autres mots-codes ?

Cette propriété cyclique induit une structure qui peut être exploitée pour concevoir les algorithmes et les circuits de codage-décodage performants.

La définition et l'analyse des codes cycliques est typiquement faite via les polynômes, définis dans le corps fini GF(2). Or, dans la suite de cette section, toutes les additions sont $\mod 2$ (ou $\oplus$), l'addition et la soustraction sont donc les opérations identiques. A chaque mot-code $\mathbf{c}$ on associe un polynôme de degré au plus $(n-1)$:

$$c(x) = c_{n-1}x^{n-1} + c_{n-2}x^{n-2} + \cdots + c_1 x + c_0 \tag{15}$$

Considérons un polynôme $x \cdot c(x)$ correspondant à un décalage de $c(x)$ à gauche, ajoutons et soustrayons c_{n-1} :

$$x \cdot c(x) = c_{n-1}x^n + \underbrace{c_{n-2}x^{n-1} + \cdots + c_1 x^2 + c_0 x + c_{n-1}}_{c^{(1)}(x)} + c_{n-1} \tag{16}$$

Le $\deg\{x \cdot c(x)\}$ peut être égal à n, il ne peut donc représenter un mot-code. Dans l'équation (16), regroupons les termes et écrivons la division polynômiale de $x \cdot c(x)$ par $(x^n + 1)$:

$$\underbrace{x \cdot c(x)}_{A(x)} = \underbrace{(x^n+1)}_{B(x)}\underbrace{c_{n-1}}_{Q(x)} + \underbrace{c^{(1)}(x)}_{R(x)}, \tag{17}$$

où le reste de la division

$$c^{(1)}(x) = c_{n-2}x^{n-1} + c_{n-1}x^{n-2} + \cdots + c_0 x + c_{n-1} \tag{18}$$

correspond au mot-code $\mathbf{c}^{(1)} = [c_{n-2}, c_{n-3}, \ldots, c_0, c_{n-1}]$ qui est un décalage cyclique d'une position à gauche de $\mathbf{c}$. On dit que $c^{(1)}(x) = x \cdot c(x) \mod (x^n + 1)$.

On peut donc généraliser pour statuer que pour $i = 0, \ldots, n-1$, les mot-codes sont obtenus par décalage cyclique (ici à gauche) et l'expression suivante est correcte :

$$x^i \cdot c(x) = (x^n + 1)Q_i(x) + c^{(i)}(x) \tag{19}$$

3.2 Génération d'un code cyclique

On utilise un polynôme-générateur $g(x)$ de degré $(n-k)$ qui divise $x^n + 1$, tel que $x^n + 1 = g(x)h(x)$, il est le polynôme de degré minimal de tous les 2^k mots-codes du code cyclique. Sa forme générale est, avec $g_{n-k} = 1$:

$$g(x) = x^{n-k} + g_{n-k-1}x^{n-k-1} + \cdots + g_1 x + 1 \tag{20}$$

Supposons que chacun des 2^k messages à encoder est décrit par un polynôme de degré $\leq (k-1)$:

$$u(x) = u_{k-1}x^{k-1} + \ldots + u_1 x + u_0 \tag{21}$$

Sachant que le degré du polynôme $u(x)g(x)$ est $\leq (n-1)$, il peut représenter un mot-code.

Déf 3.2. *Tout mot-code* $\mathbf{c}$ *correspondant au message* $\mathbf{u}$ *est représenté par son polynôme* $c(x)$*, multiple de* $g(x)$ *:*

$$c(x) = u(x)g(x) \tag{22}$$

Il y a 2^k polynômes $\{u_m(x)\}$, qui pourraient alors produire 2^k mots-codes possibles. On peut montrer que les polynômes $u_m(x)g(x),\ m = 1, \ldots, 2^k$ satisfont la propriété cyclique et sont tous égaux $\mod (X^n + 1)$, ils correspondent alors bien aux mots de code. En effet, prenons n'importe quel de ces polynômes, $c(x) = u(x)g(x)$, appliquons le décalage et la division (en binaire avec les opérations $\mod 2$, on peut réécrire l'équation (17) comm $R(x) = Q(x)B(x) + A(x)$), ce qui donne un autre mot-code :

$$\begin{aligned} c^{(1)}(x) &= q(x)(x^n + 1) + xc(x) + = q(x)h(x)g(x) + x\, u(x)g(x) \\ &= [q(x)h(x) + xu(x)] \cdot g(x) = u_1(x)g(x). \end{aligned} \tag{23}$$

Ainsi, l'existence d'un code cyclique (n, k) est déterminé par l'existence d'un polynôme $g(x)$ de degré $(n - k)$ qui divise $x^n + 1$.

Déf 3.3. *Tout polynôme facteur de $x^n + 1$ engendre un code cyclique.*

Déf 3.4. *Parmi les polynômes facteurs de $x^n + 1$, on utilise les polynômes irréductibles.*

Théorème 3.1. *Tout polynôme irréductible dans GF(2) de degré m divise $x^{2^m-1} + 1$.*

L'exemple de la décomposition en produit de polynômes irréductibles de $p^{2^m-1} - 1$ pour $m \leq 5$ est donnée dans le Tab. 2.

m	facteurs de $p^{2^m-1} - 1$
2	$x^3 + 1 = (x + 1)(x^2 + x + 1)$
3	$x^7 + 1 = (x + 1)(x^3 + x^2 + 1)(x^3 + x + 1)$
4	$x^{15} = (x + 1)(x^2 + x + 1)(x^4 + x^3 + 1)(x^4 + x + 1)(x^4 + x^3 + x^2 + x + 1)$
5	$x^{31} = (x + 1)(x^5 + x^3 + 1)(x^5 + x^2 + 1)(x^5 + x^4 + x^3 + x^2 + 1)(x^5 + x^4 + x^3 + x + 1)(x^5 + x^4 + x^2 + x + 1)(x^5 + x^3 + x^2 + x + 1)$

TABLE 2 – Décomposition de $x^{2^m-1} + 1$ en polynômes irréductibles.

3.3 Matrice génératrice d'un code cyclique

peut être obtenue à partir de $g(x)$ du code cyclique, en utilisant les k mots de code linéairement indépendants (polynômes orthogonaux) :

$$\{g(x), xg(x), x^2g(x), \ldots, x^{k-1}g(x)\},$$

comme base de dimension (k) qui sont utilisés comme lignes de matrice génératrice :

$$\mathbf{G} = \begin{pmatrix} x^{k-1}g(x) \\ \vdots \\ x^2g(x) \\ xg(x) \\ g(x) \end{pmatrix} \tag{24}$$

Exemple 6. Pour un code de Hamming $(n, k) = (7, 4)$, le polynôme $x^n + 1$ se factorise comme :

$$x^n + 1 = (x + 1)(x^3 + x^2 + 1)(x^3 + x + 1) \tag{25}$$

Ainsi, les deux polynômes-générateurs sont disponibles, chacun pouvant donner un code $(7, 4)$ différent : $g_1(x) = (x^3 + x^2 + 1)$ et $g_2(x) = (x^3 + x + 1)$. Prenons l'exemple de $g_1(x)$, il peut être écrit sous une forme d'un mot binaire dont les éléments sont les coefficients auprès de puissances de (x) : $[1101]$. Or, la matrice génératrice (non-systématique) de ce code est :

$$\mathbf{G} \begin{pmatrix} 1 & 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 0 & 1 \end{pmatrix} .\blacksquare \tag{26}$$

De manière générale, on peut considérer la factorisation suivante $x^n + 1 = g(x)h(x)$, où $h(x)$, avec $deg(h) = k$ est un polynôme de contrôle de parité, orthogonal à $g(x)$, qui peut être utiliser, après transformation, pour générer un code dual. Notons, que les vecteurs et donc les mots-codes engendrés par $g(x)$ et $h(x)$ sont orthogonaux uniquement si l'ordre des éléments d'un des vecteurs est inversé. Ainsi, définissons le *polynôme réciproque* comme :

$$x^k h(x^{-1}) \quad = \quad 1 + h_{k-1}x + h_{k-2}x^2 + \ldots + h_1 x^{k-1} + x^k \tag{27}$$

Ce polynôme est également multiple de $x^n + 1$, par conséquent, $x^k h(x^{-1})$ est un polynôme générateur pour un code $(n, n - k)$, qui est un *code dual* du code (n, k) et constitue l'espace nul de ce dernier.

Exemple(suite) Considérons le code $(7, 4)$ avec $g_1(x) = x^3 + x^2 + 1$. Son code dual est un code $(7, 3)$ associé au polynôme de contrôle de parité $h_1(x) = (x+1)(x^3+x+1) = x^4+x^3+x^2+1$, à noter que le coefficient h_4 est auprès de terme x^4. Le polynôme réciproque est alors $1+x^2+x^3+x^4$, avec h_4 auprès de $x^0 = 1$; il permet de générer le code dual donné sur la figure 12.

A noter, qu'aucun de ces deux exemples (ex 7.9-1 et 7.9-3) n'est systématique.

3.4 Forme systématique

3.5 Codage

TABLE 7.9–3

The (7, 3) Dual Code with Generator Polynomial
$X^4 h_1(X^{-1}) = X^4 + X^2 + X + 1$

Information Bits			Codewords						
X^2	X^1	X^0	X^6	X^5	X^4	X^3	X^2	X^1	X^0
0	0	0	0	0	0	0	0	0	0
0	0	1	0	0	1	0	1	1	1
0	1	0	0	1	0	1	1	1	0
0	1	1	0	1	1	1	0	0	1
1	0	0	1	0	0	1	1	0	0
1	0	1	1	0	1	1	0	1	1
1	1	0	1	1	0	0	0	1	0
1	1	1	1	1	1	0	1	0	1

FIGURE 11 – *Code dual* $(7, 3)$ *ex 7.9-3. © [Proakis].*

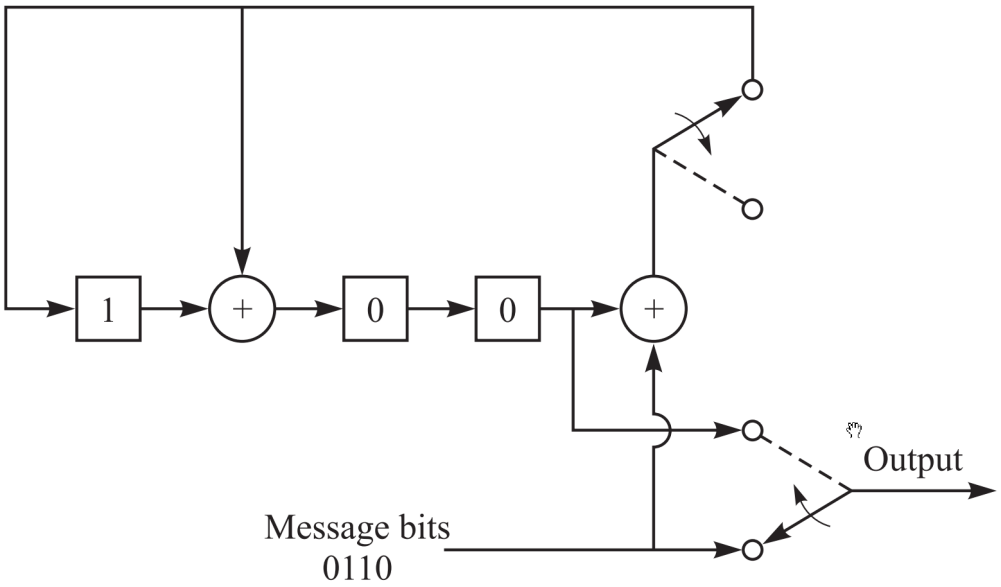

FIGURE 12 – *Codeur systématique pour un code* $(7, 4)$, $g_1(x) = x^3 + x + 1$.