UE COM105

# Corrigé des exercices du polycopié

# Chapitre 1

**Exercice 1.2**

*Concaténation parallèle :*

1.1 L'ensemble des mots de code de $C_1$ et $C_2$ (avec leur poids) sont :

| $C_1$ | | | | | | $poids$ |
|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 1 | 1 | 0 | 1 | 3 |
| 0 | 1 | 0 | 0 | 1 | 1 | 3 |
| 0 | 1 | 1 | 1 | 1 | 0 | 4 |
| 1 | 0 | 0 | 1 | 1 | 1 | 4 |
| 1 | 0 | 1 | 0 | 1 | 0 | 3 |
| 1 | 1 | 0 | 1 | 0 | 0 | 3 |
| 1 | 1 | 1 | 0 | 0 | 1 | 4 |

| $C_2$ | | | | $poids$ |
|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 1 | 1 | 2 |
| 0 | 1 | 0 | 1 | 2 |
| 0 | 1 | 1 | 0 | 2 |
| 1 | 0 | 0 | 1 | 2 |
| 1 | 0 | 1 | 0 | 2 |
| 1 | 1 | 0 | 0 | 2 |
| 1 | 1 | 1 | 1 | 4 |

La distance minimale de $C_1$ est donc $d_{min,1} = 3$, et la distance minimale de $C_2$ est donc $d_{min,2} = 2$.

1.2 Le rendement du code $C_P$ est $r = \frac{3}{7}$.

1.3 Les mots de code de $C_P$ sont :

| $C_P$ | | | | | | | $poids$ |
|---|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 1 | 1 | 0 | 1 | 1 | 4 |
| 0 | 1 | 0 | 0 | 1 | 1 | 1 | 4 |
| 0 | 1 | 1 | 1 | 1 | 0 | 0 | 4 |
| 1 | 0 | 0 | 1 | 1 | 1 | 1 | 5 |
| 1 | 0 | 1 | 0 | 1 | 0 | 0 | 3 |
| 1 | 1 | 0 | 1 | 0 | 0 | 0 | 3 |
| 1 | 1 | 1 | 0 | 0 | 1 | 1 | 5 |

La distance minimale de $C_P$ est donc égale à 3.

2.1 Le rendement du code $C_P$ est $r = \frac{k}{n_1+n_2-k}$

2.2 Un mot de code de $C_P$ peut s'écrire

$$\begin{aligned} c_P &= [k \text{ bits d'information} \mid (n_1-k) \text{ bits de redondance de } C_1 \mid (n_2-k) \text{ bits de redondance de } C_2] \\ &= [C_1 \mid (n_2-k) \text{ bits de redondance de } C_2] \\ &\equiv [C_2 \mid (n_1-k) \text{ bits de redondance de } C_1], \end{aligned}$$

où nous utilisons le symbole $\equiv$ pour dénommer les mots de codes de deux codes équivalents.
Ainsi les mots de code sont tous de poids supérieur à $d_{min,1}$, puisque le poids minimal d'un mot de code de $C_1$ est $d_{min,1}$ auquel on rajoute les bits de redondance de $C_2$ avec au moins un non nul (sinon c'est le mot nul). De la même façon, les mots de code sont tous de poids supérieur à $d_{min,2}$. La distance minimale étant le poids minimal d'un mot de code, elle est strictement supérieure au $\max(d_{min,1}, d_{min,2})$.

On pourrait avoir des mots de code de $C_P$ composés soit par un mot de code de $C_1$ auquel se rajoute les $(n_2 - k)$ bits de redondance de $C_2$ tous à 1 ou à un mot de code de $C_2$ auquel se rajoute les $(n_1 - k)$ bits de redondance de $C_1$ tous à 1, d'où $d_{min} \leq \min((d_{min1} + n_2 - k), (d_{min2} + n_1 - k))$.

La concaténation parallèle des codes correcteurs d'erreurs permet d'obtenir un codage plus robuste des bits d'information, qui sont dans ce cas protégés par les bits de redondance des deux codes. Le décodage peut se faire en considérant le code équivalent, mais il peut se faire aussi de manière itérative, où à chaque itération on décodage le premier code ensuite le deuxième code. Ce type de concaténation parallèle a donné naissance aux Turbo Codes, code correcteur ayant un très fort pouvoir de correction. Les Turbo Codes sont adoptés dans plusieurs normes telles que le WiMax et le LTE Advanced.

*Concaténation série :*

1.1 Le rendement du code $C_S$ est $r = \frac{3}{7}$ .

1.2 Les mots code de $C_S$ sont :

| $C_S$ | | | | | | | $poids$ |
|---|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 1 | 1 | 0 | 1 | 1 | 4 |
| 0 | 1 | 0 | 0 | 1 | 1 | 1 | 4 |
| 0 | 1 | 1 | 1 | 1 | 0 | 0 | 4 |
| 1 | 0 | 0 | 1 | 1 | 1 | 0 | 4 |
| 1 | 0 | 1 | 0 | 1 | 0 | 1 | 4 |
| 1 | 1 | 0 | 1 | 0 | 0 | 1 | 4 |
| 1 | 1 | 1 | 0 | 0 | 1 | 0 | 4 |

.

Sa distance minimale est donc égale à 4.

2.1 Le rendement du code $C_S$ est $r = \frac{1}{3}$.

2.2 Les mots code de $C_S$ sont :

| $C_S$ | | | | | | $poids$ |
|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 1 | 1 | 1 | 1 | 0 | 4 |
| 1 | 0 | 1 | 0 | 1 | 0 | 3 |
| 1 | 1 | 0 | 1 | 0 | 0 | 3 |

.

Sa distance minimale est donc égale à 3.

3. Pour la question 3. on suppose que le code $C_2$ est un code systématique.

3.1 Le rendement de $C_S$ est : $r = \frac{k}{n_1} \cdot \frac{n_1}{n_2} = r_1 \cdot r_2 = \frac{k}{n_2}$.

3.2 Tous les mots de code de $C_1$ sont contenus dans les mots de code de $C_S$ ainsi on a $d_{min} \geq d_{min,1}$. De plus les mots de code de $C_S$ sont un sous-ensemble des mots de code de $C_2$, ils vérifient donc $d_{min} \geq d_{min,2}$. Nous pouvons donc déduire que $d_{min} \geq \max(d_{min,1}, d_{min,2})$.
La concaténation série est très utilisée en pratique, tel que dans le GSM où un code en bloc (Reed Solomon) est concaténé à un code convolutif (une autre famille de code correcteur d'erreur).

**Exercice 1.3**

1. La longueur du mot de code est $n = 8$, la longueur du mot d'information est $k = 4$ et le rendement est $r = \frac{1}{2}$.
2. Une matrice génératrice $G$ de $\mathcal{C}$ sous forme systématique :

$$G = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 1 & 1 & 1 \\ 0 & 1 & 0 & 0 & 1 & 1 & 1 & 0 \\ 0 & 0 & 1 & 0 & 1 & 1 & 0 & 1 \\ 0 & 0 & 0 & 1 & 1 & 0 & 1 & 1 \end{bmatrix} \tag{1}$$

La matrice de parité $H$ du code $\mathcal{C}$ est :

$$H = \begin{bmatrix} 0 & 1 & 1 & 1 & 1 & 0 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 & 1 & 0 & 0 \\ 1 & 1 & 0 & 1 & 0 & 0 & 1 & 0 \\ 1 & 0 & 1 & 1 & 0 & 0 & 0 & 1 \end{bmatrix}$$

3. Les deux codes sont équivalents. On peut voir cela à partir de la matrice génératrice $G_{\text{carré}}$ trouvé dans la question 9. du TD 1. En fait, en partant de cette matrice génératrice et

   (a) En permutant l'ordre des colonnes $(1;\ 2;\ 3;\ 4;\ 5;\ 6;\ 7;\ 8)$ en $(1;\ 2;\ 3;\ 5;\ 8;\ 4;\ 6;\ 7)$,

   (b) Puis en remplaçant la première ligne par la somme de la première et de la dernière ligne,

   on arrive à la matrice génératrice systématique trouvée dans la question 2, voir égalité (1) au dessus. Plus précisément, la matrice obtenue en appliquant (a) à $G_{\text{carré}}$ est

$$\tilde{G} = \begin{bmatrix} 1 & 0 & 0 & 1 & 1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 & 1 & 1 & 0 \\ 0 & 0 & 1 & 0 & 1 & 1 & 0 & 1 \\ 0 & 0 & 0 & 1 & 1 & 0 & 1 & 1 \end{bmatrix},$$

   et en appliquant (b) à $\tilde{G}$ on arrive à

$$\bar{G} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 1 & 1 & 1 \\ 0 & 1 & 0 & 0 & 1 & 1 & 1 & 0 \\ 0 & 0 & 1 & 0 & 1 & 1 & 0 & 1 \\ 0 & 0 & 0 & 1 & 1 & 0 & 1 & 1 \end{bmatrix}.$$

   Cette matrice est identique à la matrice génératrice trouvée dans l'égalité (1). Comme nous avons seulement permuté des colonnes et pris la somme de deux lignes de $G_{\text{carré}}$, cela veut dire que les deux codes sont équivalents.

4. On peut obtenir la distance minimale directement de $H$. En fait, $d_{\min}$ est égale au plus petit nombre de colonnes linéairement dépendantes de $H$. Comme $H$ ne contient pas la colonne nulle, et comme toutes les paires et tous les triplets de colonnes de $H$ sont linéairement indépendantes, on obtient que $d_{\min}$ n'est ni 1 ni 2 ni 3. On vérifie que les colonnes 1, 2, 3 et 6 de $H$ sont linéairement dépendantes. Donc, $d_{\min} = 4$.

5. Le nouveau ensemble de mots de code ne contient pas le mot 0. Le code n'est donc pas linéaire.

**Exercice 1.4**

1. Les mots de code sont $(0\ 0\ 0\ 0\ 0)$, $(0\ 1\ 1\ 0\ 1)$, $(1\ 0\ 1\ 1\ 0)$, et $(1\ 1\ 0\ 1\ 1)$.

2. La matrice de contrôle de parité est

$$H = \begin{bmatrix} 1 & 1 & 1 & 0 & 0 \\ 1 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 1 \end{bmatrix}.$$

   Et on trouve que $d_{\min} = 3$.

3. Le décodeur optimale qui minimise la probabilité d'erreur applique la règle du voisin le plus proche par rapport à la distance de Hamming (voir le résultat 1.3 dans le polycopié).

   Notons que tous les mots du code $\mathcal{C}$ sont différents. De plus, comme $d_{\min} = 3$, un mot reçu $\mathbf{y}$ ne peut pas être à distance de Hamming égale à 1 de plus d'un seul mot de code. Donc,
   — Si $\mathbf{y} \in \mathcal{C}$, le décodeur du voisin le plus proche décide $\hat{\mathbf{c}} = \mathbf{y}$.
   — Si pour un mot de code $\mathbf{c} \in \mathcal{C}$ il vaut que $d_{\text{H}}(\mathbf{y}, \mathbf{c}) = 1$, alors le décodeur du voisin le plus proche décide $\hat{\mathbf{c}} = \mathbf{c}$.

   Avec ces deux règles on peut déjà remplir 24 des 32 cases dans le tableau de décodage (tableau 1) sur la page suivante. En fait, ils existent 4 mots de code, et pour chaque mot de code ils existent 5 mots à distance de Hamming égale à 1.

   On peut vérifier que les 8 mots restants sont à distance de Hamming 2 de exactement deux mots de code. Pour garder la symétrie dans nos calculs de la probabilité d'erreur dans la question qui suit, nous choisissons un décodeur qui déclare ces deux mots de code avec la même probabilité $1/2$.

   Toutes ces reflexions mènent au décodeur optimal décrit dans le tableau 1.

4. La probabilité d'erreur s'exprime en générale comme suit :

$$P_e := \text{Pr}[\hat{\mathbf{C}} \neq \mathbf{C}] = 1 - \underbrace{\text{Pr}[\hat{\mathbf{C}} = \mathbf{C}]}_{=:P_{\text{c}}},$$

| mot reçu $\mathbf{y}$ | mot décodé $\hat{\mathbf{c}}$ |
|---|---|
| 0 0 0 0 0 | 0 0 0 0 0 |
| 0 0 0 0 1 | 0 0 0 0 0 |
| 0 0 0 1 0 | 0 0 0 0 0 |
| 0 0 0 1 1 | 0 0 0 0 0 et 1 1 0 1 1 avec probabilité $1/2$. |
| 0 0 1 0 0 | 0 0 0 0 0 |
| 0 0 1 0 1 | 0 1 1 0 1 |
| 0 0 1 1 0 | 1 0 1 1 0 |
| 0 0 1 1 1 | 0 1 1 0 1 et 1 0 1 1 0 avec probabilité $1/2$. |
| 0 1 0 0 0 | 0 0 0 0 0 |
| 0 1 0 0 1 | 0 1 1 0 1 |
| 0 1 0 1 0 | 0 0 0 0 0 et 1 1 0 1 1 avec probabilité $1/2$. |
| 0 1 0 1 1 | 1 1 0 1 1 |
| 0 1 1 0 0 | 0 1 1 0 1 |
| 0 1 1 0 1 | 0 1 1 0 1 |
| 0 1 1 1 0 | 0 1 1 0 1 et 1 0 1 1 0 avec probabilité $1/2$. |
| 0 1 1 1 1 | 0 1 1 0 1 |
| 1 0 0 0 0 | 0 0 0 0 0 |
| 1 0 0 0 1 | 0 0 0 0 0 et 1 1 0 1 1 avec probabilité $1/2$. |
| 1 0 0 1 0 | 1 0 1 1 0 |
| 1 0 0 1 1 | 1 1 0 1 1 |
| 1 0 1 0 0 | 1 0 1 1 0 |
| 1 0 1 0 1 | 0 1 1 0 1 et 1 0 1 1 0 avec probabilité $1/2$. |
| 1 0 1 1 0 | 1 0 1 1 0 |
| 1 0 1 1 1 | 1 0 1 1 0 |
| 1 1 0 0 0 | 0 0 0 0 0 et 1 1 0 1 1 avec probabilité $1/2$. |
| 1 1 0 0 1 | 1 1 0 1 1 |
| 1 1 0 1 0 | 1 1 0 1 1 |
| 1 1 0 1 1 | 1 1 0 1 1 |
| 1 1 1 0 0 | 0 1 1 0 1 et 1 0 1 1 0 avec probabilité $1/2$. |
| 1 1 1 0 1 | 0 1 1 0 1 |
| 1 1 1 1 0 | 1 0 1 1 0 |
| 1 1 1 1 1 | 1 1 0 1 1 |

TABLE 1 – Un décodeur optimal.

où $\mathbf{C}$ est le mot de code envoyé sur le BSC, $\hat{\mathbf{C}}$ est le mot de code décodé au récepteur et $P_{\mathrm{c}}$ indique la probabilité de succès.

En analysant le tableau 1 de notre décodeur optimal, nous observons que le décodeur trouve toujours le bon mot de code si :

— le canal ne change aucun bit du mot envoyé ;

— le canal change un des 5 bits du mot envoyé ;

et il trouve le bon mot de code avec une probabilité $1/2$ si

— le canal change les deux bits aux positions $i$ et $j$, pour $(i,j) \in \mathcal{J} := \{(1,2),\ (2,4),\ (1,5),\ (4,5)\}$.

Donc,

$$\begin{aligned} P_{\mathrm{c}} &= \Pr\big[\text{le canal ne change aucun bit d'entrée}\big] + \Pr\big[\text{le canal change un bit d'entrée}\big] \\ &\quad + \frac{1}{2} \sum_{(i,j)\in\mathcal{J}} \Pr\big[\text{le canal change les bits d'entrée } i \text{ et } j\big] \\ &= (1-\epsilon)^5 + 5\epsilon(1-\epsilon)^4 + \frac{1}{2}\cdot 4\epsilon^2(1-\epsilon)^3 \\ &= (1-\epsilon)^3\big((1-\epsilon)^2 + 5\epsilon(1-\epsilon) + 2\epsilon^2\big). \end{aligned}$$

**Exercice 1.5**

1. Il faut donc écrire $p(\mathbf{y}|\mathbf{c})$. Comme le canal est sans mémoire, on a
$$p(\mathbf{y}|\mathbf{c}) = \prod_{n=0}^{N-1} p(y_n|c_n).$$
Raisonnons maintenant sur une composante.
Quand $y_n = 0$, alors on vient de $y_n = c_n$ avec une probabilité de $1 - p_0$ et de $y_n \neq c_n$ avec une probabilité de $p_1$, d'où $p(y_n = 0|c_n) = (1-p_0)^{1-w_H(y_n,c_n)} p_1^{w_H(y_n,c_n)}$. De même, on a $p(y_n = 1|c_n) = (1-p_1)^{1-w_H(y_n,c_n)} p_0^{w_H(y_n,c_n)}$. Contrairement au cours, ici $p_0 \neq p_1$ et donc le cas $y_n = 0$ ne peut être réuni aussi facilement avec le cas $y_n = 1$. En fait
$$p(y_n|c_n) = p(y_n = 0|c_n)^{1-y_n} p(y_n = 1|c_n)^{y_n}.$$
On activera alors bien le terme adéquat pour cuaque valeur de $y_n$. Ainsi
$$p(y_n|c_n) = (1-p_0)^{(1-y_n)(1-w_H(y_n,c_n))} p_1^{(1-y_n)w_H(y_n,c_n)} (1-p_1)^{y_n(1-w_H(y_n,c_n))} p_0^{y_n w_H(y_n,c_n)}$$
d'où
$$\begin{aligned} p(\mathbf{y}|\mathbf{c}) &= (1-p_0)^{\sum_{n=0}^{N-1}(1-y_n)(1-w_H(y_n,c_n))} p_1^{\sum_{n=0}^{N-1}(1-y_n)w_H(y_n,c_n)} \\ &\times (1-p_1)^{\sum_{n=0}^{N-1} y_n(1-w_H(y_n,c_n))} p_0^{\sum_{n=0}^{N-1} y_n w_H(y_n,c_n)} \\ &= (1-p_0)^{N-w_H(\mathbf{y})-d_H(\mathbf{y},\mathbf{c})+\tilde{d}(\mathbf{y},\mathbf{c})} p_1^{d_H(\mathbf{y},\mathbf{c})-\tilde{d}(\mathbf{y},\mathbf{c})} \\ &\times (1-p_1)^{w_H(\mathbf{y})-\tilde{d}(\mathbf{y},\mathbf{c})} p_0^{\tilde{d}(\mathbf{y},\mathbf{c})} \end{aligned}$$
avec $\tilde{d}(\mathbf{y},\mathbf{c}) = \sum_{n=0}^{N-1} y_n w_H(y_n, c_n)$ (la somme à prendre dans $\mathbb{R}$ et non dans $\mathbb{F}_2$).
Quand $p_0 = p_1$, on retrouve bien la formule du cours ne faisant intervenir que la distance de Hamming $d_H(\mathbf{y},\mathbf{c})$.
2. Trouver le maximum de $p(\mathbf{y}|\mathbf{c})$ revient à trouver le maximum du logarithme. Donc on cherche à maximiser (en éliminant les termes constants et notamment ceux dépendent uniquement de $\mathbf{y}$)
$$(-d_H(\mathbf{y},\mathbf{c}) + \tilde{d}(\mathbf{y},\mathbf{c}))\ln(1-p_0) + (d_H(\mathbf{y},\mathbf{c}) - \tilde{d}(\mathbf{y},\mathbf{c}))\ln(p_1) - \tilde{d}(\mathbf{y},\mathbf{c})\ln(1-p_1) + \tilde{d}(\mathbf{y},\mathbf{c})\ln(p_0)$$
En réunissant les termes $w$, $d_H$ et $\tilde{d}$ ensemble, on a
$$d_H(\mathbf{y},\mathbf{c})\ln\left(\frac{p_1}{1-p_0}\right) + \tilde{d}(\mathbf{y},\mathbf{c})\ln\left(\frac{(1-p_0)p_0}{(1-p_1)p_1}\right)$$
En prenant l'opposé, on a
$$\hat{\mathbf{c}} = \arg\min_{\mathbf{c}} f(\mathbf{y},\mathbf{c})$$
avec
$$f(\mathbf{y},\mathbf{c})d_H(\mathbf{y},\mathbf{c})\ln\left(\frac{1-p_0}{p_1}\right) + \tilde{d}(\mathbf{y},\mathbf{c})\ln\left(\frac{(1-p_1)p_1}{(1-p_0)p_0}\right)$$
L'optimisation fait donc intervenir deux métriques : la distance de Hamming et un autre terme $\tilde{d}$.
3. Sur cet exemple, on va voir que le terme $\tilde{d}$ a bien une influence dans la minimisation.
Pour rappel, les deux mots de code sont $\mathbf{c}_1 = [0,0,0]$ et $\mathbf{c}_2 = [1,1,1]$. On considère recevoir le mot $\mathbf{y} = [1,0,0]$.
Dans un BSC, le ML conduit à la distance de Hamming minimale et donc au vote à la majorité. D'où $\hat{\mathbf{c}}_{\text{BSC}} = \mathbf{c}_1$.
Dans un BNSC avec $p_0 = 0.1$ et $p_1 = 0.4$, on va calculer $f(\mathbf{y},\mathbf{c}_1)$ et $f(\mathbf{y},\mathbf{c}_2)$. On a $d_H(\mathbf{y},\mathbf{c}_1) = 1$, $\tilde{d}(\mathbf{y},\mathbf{c}_1) = 1$, $d_H(\mathbf{y},\mathbf{c}_2) = 2$, $\tilde{d}(\mathbf{y},\mathbf{c}_2) = 0$, d'où
$$f(\mathbf{y},\mathbf{c}_1) = \ln\left(\frac{1-p_1}{p_0}\right) \text{ et } f(\mathbf{y},\mathbf{c}_2) = \ln\left(\frac{(1-p_0)^2}{p_1^2}\right)$$
et donc numériquement
$$f(\mathbf{y},\mathbf{c}_1) = \ln(6) \text{ et } f(\mathbf{y},\mathbf{c}_2) = \ln(5.0625).$$
Ainsi on a
$$\hat{\mathbf{c}}_{\text{BNSC}(0.1,0.4)} = \mathbf{c}_2.$$
Et ce n'est donc plus un vote à la majorité. La région de décision de $[1,1,1]$ s'est agrandie finalement car le signal reçu a plus de chances d'avoir des 0 car il y a plus de 1 émis qui se transforme en 0.

**Exercice 1.6**

1. On rappelle que si $y_n$ n'est pas un effacement, alors il a la valeur du $x_n$ émis. Donc $p(y_n \neq \Delta | c_n) = \delta_{0,w_H(y_n,c_n)}$ avec $\delta_{\ell,\ell'}$ l'indice de Kronecker qui vaut 1 si $\ell = \ell'$ et 0 sinon. En revanche $p(y_n = \Delta | c_n = 0) = p(y_n = \Delta | c_n = 1) = \varepsilon$. Finalement

$$p(\mathbf{y}|\mathbf{c}) = \left( \prod_{n \notin C_e} \delta_{0,w_H(y_n,c_n)} \right) \varepsilon^{|C_e|}$$

avec $C(e)$ l'ensemble des positions des effacements dans $\mathbf{y}$ et $|C_e|$ la cardinalité de cet ensemble.
Si on ne garde que les termes dépendant explicitement de $\mathbf{c}$, on obtient que

$$p(\mathbf{y}|\mathbf{c}) \propto \prod_{n \notin C_e} \delta_{0,w_H(y_n,c_n)}$$

ce qui implique qu'on va sélectionner le vecteur $\mathbf{c}$ qui a les positions non effacées de $\mathbf{y}$ identiques. S'il y a plusieurs vecteurs possibles, on le choisit aléatoirement.

2. On va d'abord calculer la probabilité d'erreur mot.

$$\text{Pr}(\hat{\mathbf{c}} \neq \mathbf{c}) = \sum_{\ell=0}^{2^k-1} \text{Pr}(\mathbf{c}^{(\ell)}) \text{Pr}(\hat{\mathbf{c}} \neq \mathbf{c}^{(\ell)})$$

avec $\mathbf{c}^{(\ell)}$ le $\ell^{\text{ème}}$ mot de code. D'où

$$\text{Pr}(\hat{\mathbf{c}} \neq \mathbf{c}) = \frac{1}{2^k} \sum_{\ell=0}^{2^k-1} \text{Pr}(\hat{\mathbf{c}} \neq \mathbf{c}^{(\ell)}).$$

On se trompera sur $\mathbf{c}^{(\ell)}$ s'il y a au moins $d_{\min}$ effacements. En effet en dessous de $d_{\min}$ effacements, le détecteur optimal sera retrouver le bon mot. Pour le prouver, il suffit de se rendre compte qu'on aura alors le mot de code émis $\mathbf{c}_1 = [\alpha_1, \beta]$ avec $\beta$ la partie non-effacée et $\alpha_1$ la partie effacée de longueur inférieur strictement à $d_{\min}$ et le mot de code détecté $\mathbf{c}_2 = [\alpha_2, \beta]$. Alors $\mathbf{c}_1$ et $\mathbf{c}_2$ sont à distance inférieure à $d_{\min}$ ce qui n'est pas possible. Ainsi l'événement $\{\hat{\mathbf{c}} \neq \mathbf{c}^{(\ell)}\}$ est inclus dans l'événement $\{\mathbf{y}$ admet au moins $d_{\min}$ effacements $\}$. D'où

$$\text{Pr}(\hat{\mathbf{c}} \neq \mathbf{c}^{(\ell)}) \leq \sum_{n=d_{\min}}^{N} C_N^n (1-\varepsilon)^{N-n} \varepsilon^n$$

ce qui implique

$$\text{Pr}(\hat{\mathbf{c}} \neq \mathbf{c}) \leq \sum_{n=d_{\min}}^{N} C_N^n (1-\varepsilon)^{N-n} \varepsilon^n$$

Si $\varepsilon$ est petit, on a la version approchéé suivante

$$\text{Pr}(\hat{\mathbf{c}} \neq \mathbf{c}) \approx \sum_{n=d_{\min}}^{N} C_N^n (1-(N-n)\varepsilon) \varepsilon^n.$$

En ne conservant que le terme dominant associé à la plus petite puissance, on a

$$\text{Pr}(\hat{\mathbf{c}} \neq \mathbf{c}) \approx C_N^{d_{\min}} \varepsilon^{d_{\min}}.$$

Comme les erreurs ont lieu avec les mots de code à distance minimale, il y a $d_{\min}$ erreurs sur $N$. Si elles sont équiréparties sur les positions, alors

$$P_b \approx \frac{d_{\min}}{N} C_N^{d_{\min}} \varepsilon^{d_{\min}}.$$

Comparer au cas BSC, on voit que $\varepsilon$ est élevée à la puissance $d_{\min}$ et non pas $\lfloor (d_{\min}-1)/2 \rfloor + 1 \approx d_{\min}/2$ et donc les performances du BEC($\varepsilon$) sont meilleures que celles du BSC($\varepsilon$). Dans le cours on avait un résultat similaire pour la capacité.

# Chapitre 2

**Exercice 2.1**

1. On rappelle que
$$B \geq \frac{1}{T_s}$$
Or on a aussi
$$\frac{1}{T_s} = \frac{D_b}{\log_2(M)} \tag{2}$$
d'où
$$\log_2(M) \geq \frac{D_b}{B}.$$
L'application numérique de l'exercice conduit à $\log_2(M) \geq 3,6$ d'où l'utilisation minimale d'une 16-PAM.
2. Comme nous utilisons un filtre en racine de cosinus surélévé de facteur d'excès de bande $\rho$, on a
$$B = \frac{1+\rho}{T_s}.$$
Grâce à l'équation (2), on obtient que
$$1 + \rho = \frac{B \log_2(M)}{D_b}.$$
Comme $M = 16$, l'application numérique conduit à
$$\rho = 0.1111....$$

**Exercice 2.2**

1. On a
$$\begin{aligned}
P(\mathbf{x}_m \to \mathbf{x}_{m'}) &= \Pr\left(p(\mathbf{z}|\mathbf{x}_{m'}) > p(\mathbf{z}|\mathbf{x}_m)|\mathbf{x} = \mathbf{x}_m\right) \\
&= \Pr\left(\prod_{n=0}^{N-1} p(z_n|x_{m',n}) > \prod_{n=0}^{N-1} p(z_n|x_{m,n})\Big|\mathbf{x} = \mathbf{x}_m\right) \\
&= \Pr\left(\prod_{n=0}^{N-1} e^{-(z_n - x_{m',n})^2/N_0} > \prod_{n=0}^{N-1} e^{-(z_n - x_{m,n})^2/N_0}\Big|\mathbf{x} = \mathbf{x}_m\right) \\
&= \Pr\left(e^{-\sum_{n=0}^{N-1}(z_n - x_{m',n})^2/N_0} > e^{-\sum_{n=0}^{N-1}(z_n - x_{m,n})^2/N_0}\Big|\mathbf{x} = \mathbf{x}_m\right) \\
&= \Pr\left(\sum_{n=0}^{N-1}(z_n - x_{m',n})^2 < \sum_{n=0}^{N-1}(z_n - x_{m,n})^2\Big|\mathbf{x} = \mathbf{x}_m\right).
\end{aligned}$$
Or $z_n = x_{m,n} + w_n$, donc on a
$$P(\mathbf{x}_m \to \mathbf{x}_{m'}) = \Pr\left(\sum_{n=0}^{N-1}(x_{m,n} - x_{m',n} + w_n)^2 < \sum_{n=0}^{N-1} w_n^2\right).$$
En développant le carré, on obtient que
$$P(\mathbf{x}_m \to \mathbf{x}_{m'}) = \Pr\left(\omega \geq d_{m,m'}^2\right)$$
avec
$$\omega = -2\sum_{n=0}^{N-1} w_n(x_{m,n} - x_{m',n})$$
et
$$d_{m,m'}^2 = \sum_{n=0}^{N-1}(x_{m,n} - x_{m',n})^2.$$

Il nous faut donc juste connaître la densité de probabilité de $\omega$. C'est une gaussienne (car somme de gaussiennes) de moyenne nulle et de variance $\sigma_\omega^2$. Il est facile de vérifier que la variance vaut

$$\sigma_\omega^2 = 2N_0 d_{m,m'}^2.$$

Ainsi

$$P(\mathbf{x}_m \to \mathbf{x}_{m'}) = Q\left(\frac{d_{m,m'}}{\sqrt{2N_0}}\right).$$

2. On a

$$P_e = \frac{1}{M}\sum_{m=0}^{M-1}\sum_{m'\neq m} Q\left(\frac{d_{m,m'}}{\sqrt{2N_0}}\right).$$

Les termes dominants dans les sommes précédents sont asociées aux plus petits arguments intervenant dans les fonctions $Q$. Par conséquent, on ne garde que les couples d'indice $(m, m')$ tel que $d_{m,m'} = d_{\min}$ avec

$$d_{\min} = \min_{\ell,\ell'} d_{\ell,\ell'}$$

la distance minimale euclidienne entre les mots de code. (et non la distance minimale de Hamming).
Soit $N_{\min,m}$ le nombre de mots à distance minimale $d_{\min}$ de $\mathbf{x}_m$, alors

$$P_e = \frac{1}{M}\sum_{m=0}^{M-1} N_{\min,m} Q\left(\frac{d_{\min}}{\sqrt{2N_0}}\right)$$

d'où

$$P_e \approx N_{\min} Q\left(\frac{d_{\min}}{\sqrt{2N_0}}\right)$$

avec

$$P_e = \frac{1}{M}\sum_{m=0}^{M-1} N_{\min,m}$$

le nombre de mots de code moyen à distance minimale.

3. Si les codes sont modulés par une 2-PAM (d'amplitude $\pm A$) alors on a

$$d_{\min} = 2A\sqrt{d_H}.$$

4. La probabilité d'erreur mot vaut alors

$$P_e \approx N_{\min} Q\left(\frac{\sqrt{2} d_H A}{\sqrt{N_0}}\right).$$

Or $E_b = E_s/R$ avec $R$ le rendement du code et $E_s$ l'énergie symbole qui vaut $A^2$ pour une 2-PAM, d'où

$$P_e \approx N_{\min} Q\left(\sqrt{R d_H \frac{2E_b}{N_0}}\right).$$

Le gain de codage vaut $Rd_H$. Dans le cours de modulation, on a vu que le gain valait environ $Rd_H/2$ (en négligeant le $-1$ et la partie entière). Il y a donc un facteur 2 et donc 3dB d'écart en faveur de l'exemple de cet exercice par rapport à celui de cours. La différence réside dans le fait que dans l'exercice on décode directement à partir de l'observation et non à partir des bits détectés depuis l'observation. La détection préalable des bits pour ensuite chercher la structure du code (ce qu'on appelle la méthode *hard decision*) n'est pas optimale par rapport à conserver toute l'observation réelle (ce qu'on appelle la méthode *soft decision*).

**Exercice 2.3**

1. La probabilité de bonne détection sur le lien global est égale à $1 - P_e$. Pour que le lien global soit correct, il faut que tous les liens intermédiaires soient eux aussi corrects. La probabilité de bonne détection de ces liens intermédiaires est $1 - P_e^{(i)}$. Comme les liens intermédiaires sont indépendants, on obtient que
$$1 - P_e = (1 - P_e^{(i)})^{N+1}$$
ce qui conduit au résultat.
2. Pour un lien intermédiaire quelconque fixé, il faut déterminer l'énergie reçue par symbole, notée $E_s^{(r)}$. Il est clair que la relation entre les puissances émises et reçues s'appliquent aussi sur les énergies (puisque le temps-symbole est invariant). Par conséquent, on a
$$E_s^{(r)} = \frac{E_s}{D_R^2}.$$
De plus, une modulation 2-PAM est utilisée, donc selon le cours, on sait que
$$P_e^{(i)} = Q\left(\sqrt{\frac{2E_s^{(r)}}{N_0}}\right).$$
Ainsi, on obtient que
$$P_e^{(i)} = Q\left(\sqrt{\frac{2E_s}{D_R^2 N_0}}\right).$$
3. Comme $D_T = (N+1)D_R$ et comme $P_e = 1 - (1 - P_e^{(i)})^{N+1}$, on en déduit trivialement que
$$P_e = 1 - \left(1 - Q\left(\sqrt{\frac{2(N+1)^2 E_s}{D_T^2 N_0}}\right)\right)^{N+1}.$$
En notant
$$\beta = \sqrt{\frac{2E_s}{D_T^2 N_0}} > 0,$$
et
$$f_\beta(x) = 1 - \left(1 - Q\left(x+1\right)\right)^{x+1},$$
on a
$$P_e = f_\beta(N). \tag{3}$$
4. On veut optimiser $P_e$ par rapport à $N$. Par conséquent, on va étudier les variations de $P_e$ en fonction de $N$ (cf. Eq. (3)). En examinant rapidement la fonction $f_\beta$, on s'aperçoit vite qu'elle tend vers 0 quand $N$ tend vers l'infini. et donc que la probabilité d'erreur peut être rendu aussi petit que l'on souhaite en mettant un nombre infini de répéteurs. Ce résultat n'est en fait pas surprenant car, comme l'énergie par symbole est bornée par répéteur et non pour l'ensemble des répéteurs, plus on met de répéteur et plus l'énergie par symbole émis augmente ce qui implique une décroissance vers 0 de la probabilité d'erreur. En pratique, évidemment on ne met pas un nombre infini de répéteurs en raison du coût et en fait aussi parce que cela n'est pas nécessaire. En effet, aucun système ne souhaite travailler avec une probabilité d'erreur infiniment petite. Chaque système requiert une certaine qualité de service (QoS) avec une probabilité d'erreur cible qui sera amplement suffisante si elle est satisfaite. Pour la voix (2G), $P_e^{(0)} = 10^{-3}$ suffit. Pour les données (ADSL), $P_e^{(0)} = 10^{-7}$ suffit. Pour la vidéo (TNT), $P_e^{(0)} = 10^{-11}$ suffit.
5. On a
$$N_{\min} = \arg \min_{\substack{N \\ t.q.\ P_e^{(0)} \geq f_\beta(N)}} N.$$
6. Etant donné notre application numérique, on a
$$\beta = \sqrt{\frac{2.10^{10}}{(10^6)^2}} = \sqrt{2}.10^{-1}.$$
La fonction $f_{\sqrt{2}.10^{-1}}$ est représentée sur la Fig. 2.
On observe que
$$N_{\min} = 28.$$

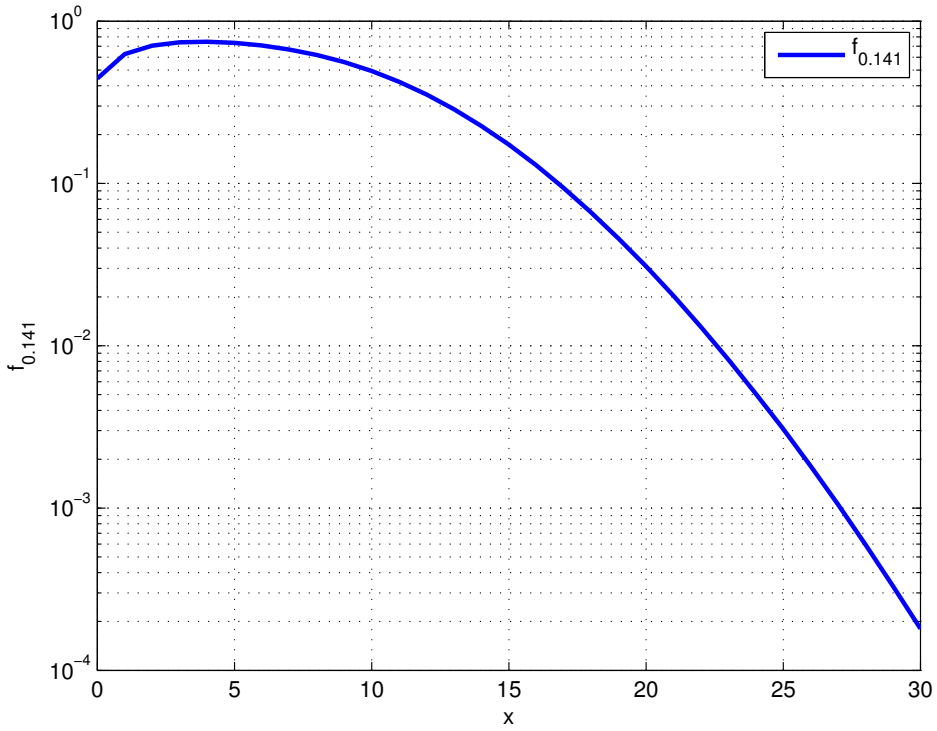



FIGURE 1 – $f_{\sqrt{2}.10^{-1}}(x)$ en fonction de $x$

**Exercice 2.4**

1.1 Comme les symboles d'une même voie sont indépendants et que le bruit est i.i.d, on peut travailler échantillon par échantillon et donc le maximum de vraisemblance s'écrit comme suit pour la voie $\#n$

$$\hat{s}_k^{(n)} = \arg\max_s \; p\left(z_k^{(n)}|s\right)$$

avec la maximisation qui est sur l'ensemble des $M^{(n)}$ symboles possibles.
Il nous suffit donc d'écrire la densité de probabilité conditionnelle $p(z_k^{(n)}|s)$. Comme le bruit est gaussien de moyenne nulle et de variance $N_0/2$, on a

$$p(z_k^{(n)}|s) \propto e^{-\frac{(z_k^{(n)} - h^{(n)}s)^2}{N_0}}$$

où $h^{(n)}$ est un paramètre connu et donc déterministe.
Par conséquent, on a

$$\hat{s}_k^{(n)} = \arg\min_s \; (\tilde{z}_k^{(n)} - h^{(n)}s)^2$$

qui est équivalent à

$$\hat{s}_k^{(n)} = \arg\min_s \; (\tilde{z}_k^{(n)} - s)^2$$

avec

$$\tilde{z}_k^{(n)} = \frac{z_k^{(n)}}{h^{(n)}}.$$

Donc le ML consiste à appliquer le détecteur à seuil de la modulation $M^{(n)}$-PAM sur le signal reçu divisé par la réponse du canal.

1.2 On note $m_1$ le nombre de bits émis par utilisation de canal (c'est-à-dire, à chaque fois que un symbole est émis sur chaque voie. En reprenant la formule donnée dans le polycopié à la section 3.5.3., on a

$$m_1 = \left\lfloor \frac{1}{2}\log_2\left(1+\frac{1}{\Gamma_1}\cdot\frac{|h^{(1)}|^2E_s}{N_0}\right)\right\rfloor + \left\lfloor \frac{1}{2}\log_2\left(1+\frac{1}{\Gamma_1}\cdot\frac{|h^{(2)}|^2E_s}{N_0}\right)\right\rfloor$$

avec, pour rappel, $\Gamma_1 = \left(Q^{(-1)}(P_e^{(0)}/2)\right)^2/6$.

2.1 Maintenant, le même symbole est émis sur les deux voies. Par conséquent, le ML ne peut plus travailler voie par voie. Par contre, il peut encore s'écrire instant par instant car les symboles émis à des instants différents sont indépendants et les bruits sont indépendants temporellement. Ainsi le ML s'écrit génériquement de la manière suivante

$$\hat{s}_k^{(n)} = \arg\max_s \; p\left(z_k^{(1)}, r_k^{(2)}|s\right)$$

Comme les bruits sont indépendants sur les deux voies, on a

$$\hat{s}_k^{(n)} = \arg\max_s \; p\left(z_k^{(1)}|s\right).p\left(z_k^{(2)}|s\right)$$

d'où

$$\hat{s}_k^{(n)} = \arg\min_s \; (z_k^{(1)} - h^{(1)}s)^2 + (z_k^{(2)} - h^{(2)}s)^2$$

En développant les carrés et en omettant les termes qui ne dépendent pas de $s$, on obtient que

$$\hat{s}_k^{(n)} = \arg\min_s \; f_1(s)$$

avec

$$f_1(s) = 2\frac{h^{(1)}z_k^{(1)} + h^{(2)}z_k^{(2)}}{|h^{(1)}|^2 + |h^{(2)}|^2}s + s^2 = 2z_k s + s^2$$

Il est clair que ce problème est équivalent au problème suivant

$$\hat{s}_k^{(n)} = \arg\min_s \; f_2(s)$$

avec

$$f_2(s) = (z_k - s)^2 = 2z_k s + s^2 + \text{constante}$$

Par conséquent, le ML conduit juste à appliquer le détecteur à seuil de la modulation $M^{(n)}$-PAM sur le signal $z_k$. On dit que le signal $z_k$ est la combinaison linéaire à maximum de SNR (en anglais, *Maximum Ratio Combiner - MRC*). On observe que le signal $z_k$ est un barycentre des signaux reçus et le poids est d'autant plus fort sur une voie que cette voie est bonne. Ce principe est notamment utilisé dans les récepteurs de téléphonies mobiles 3G lorsque les voies correspondent aux échos du canal de propagation. Le récepteur s'appele alors le *Rake receiver* et date de 1956.

2.2 On a facilement que

$$z_k = s_k + w_k$$

avec

$$w_k = \frac{h^{(1)}w_k^{(1)} + h^{(2)}w_k^{(2)}}{|h^{(1)}|^2 + |h^{(2)}|^2}.$$

Comme les bruits sont indépendants sur chaque voie, la variance de $w_k$ est égale à la somme des variances des deux bruits le constituant. Ainsi

$$\sigma_w^2 = \frac{N_0/2}{|h^{(1)}|^2 + |h^{(2)}|^2}.$$

Il suffit donc de reprendre la formule de probabilité d'erreur du cours pour la PAM en remplaçant $N_0/2$ par $\sigma_w^2$. On a alors

$$P_e = 2\left(1-\frac{1}{M}\right)Q\left(\sqrt{\frac{6(|h^{(1)}|^2 + |h^{(2)}|^2)\log_2(M)}{M^2-1}\frac{E_b}{N_0}}\right)$$

avec $E_b$ l'énergie consommée pour émettre un bit sur une des voies.

2.3 En négligeant le terme $1/M$ extérieur à la fonction $Q$ et en remplaçant $E_b$ par son expression en fonction de $E_s$, on a

$$P_e \approx 2Q\left(\sqrt{\frac{6(|h^{(1)}|^2+|h^{(2)}|^2)}{M^2-1}\frac{E_s}{N_0}}\right)$$

d'où

$$m_2 = \left\lfloor \frac{1}{2}\log_2\left(1+\frac{1}{\Gamma_2}\cdot\frac{E_s}{N_0}\right)\right\rfloor$$

avec

$$\Gamma_2 = \frac{\left(Q^{(-1)}(P_e^{(0)}/2)\right)^2}{6(|h^{(1)}|^2+|h^{(2)}|^2)}.$$

3. En supposant que $h^{(1)} = h^{(2)} = 1$, on a

$$m_1 = 2\left\lfloor \frac{1}{2}\log_2\left(1+\frac{1}{\Gamma}\cdot\frac{E_s}{N_0}\right)\right\rfloor$$

et

$$m_2 = \left\lfloor \frac{1}{2}\log_2\left(1+\frac{2}{\Gamma}\cdot\frac{E_s}{N_0}\right)\right\rfloor$$

avec $\Gamma = \left(Q^{(-1)}(P_e^{(0)}/2)\right)^2/6$.

Hors, on peut montrer [1] facilement que $\log_2(1+x) > (1/2)\log_2(1+2x)$ pour $x$ strictement positif, ce qui implique que la première approche est plus efficace que la seconde approche en terme de débit (sauf à des SNR trop bas où la présence de la partie entière - non prise en compte dans la note de bas de page- inverse le résultat temporairement). Sur la Fig. 2, on présente en fonction du terme $E_s/(\Gamma N_0)$ exprimé en dB les débits atteints par les deux techniques. Clairement, l'approche 1 est meilleure que l'approche 2 sauf à bas SNR.

**Exercice 2.5**

1. La capacité de deux voies indépendantes est tout simplement la somme des capacités. Donc on a

$$C = \frac{1}{2}\log_2\left(1+|h^{(1)}|^2\frac{2E_s^{(1)}}{N_0}\right)+\frac{1}{2}\log_2\left(1+|h^{(2)}|^2\frac{2E_s^{(2)}}{N_0}\right).$$

2. Pour cela, on va utiliser la concavité de la fonction $f(\bullet) = \log_2(1+\bullet(2E_s/N_0))$. On pose $x_n = |h^{(n)}|^2$ et $\lambda_n = 1/2$. Alors

$$f(\lambda_1 x_1 + \lambda_2 x_2) \geq \lambda_1 f(x_1) + \lambda_2(x_2)$$

d'où

$$\log_2\left(1+\frac{2E_s}{N_0}\right) \geq \log_2\left(1+|h^{(1)}|^2\frac{2E_s}{N_0}\right)+\log_2\left(1+|h^{(2)}|^2\frac{2E_s}{N_0}\right)$$

Le terme $\log_2(1+2E_s/N_0)$ correspond à la capacité du canal quand $h^{(1)} = h^{(2)} = 1$ ce qui prouve le résultat.

3. Il faut donc optimiser la fonction

$$\frac{1}{2}\log_2\left(1+|h^{(1)}|^2\frac{2E_s^{(1)}}{N_0}\right)+\frac{1}{2}\log_2\left(1+|h^{(2)}|^2\frac{2E_s^{(2)}}{N_0}\right)$$

sous la contrainte $E_s^{(1)} + E_s^{(2)} = 2E_s$. Ceci pourrait se résoudre par des outils d'optimisation convexe car les fonctions sont concaves et linéaires. Mais vous n'avez pas encore ces outils. Donc on va contourner le problème (ce qui faisable ici, car on n'a que deux variables. On pose $x = E_s^{(1)}$ et donc l'autre variable $E_s^{(2)} = 2E_s - x$. Donc on va écrire le problème pour la variable $x$ comprise entre $[0, 2E_s]$. Ainsi le problème d'optimisation revient à maximiser

$$f(x) = \log_2(1+S_1 x)+\log_2(1+S_2(2E_s-x))$$

1. soit $f(x) = \log_2(1+x) - (1/2)\log_2(1+2x)$. On a $f'(x) = 1/(1+x) - 1/(1+2x) > 0$ pour $x$ strictement positif. Donc $f$ est strictement croissante et comme $f(0) = 0$, on a $f(x) > 0$.

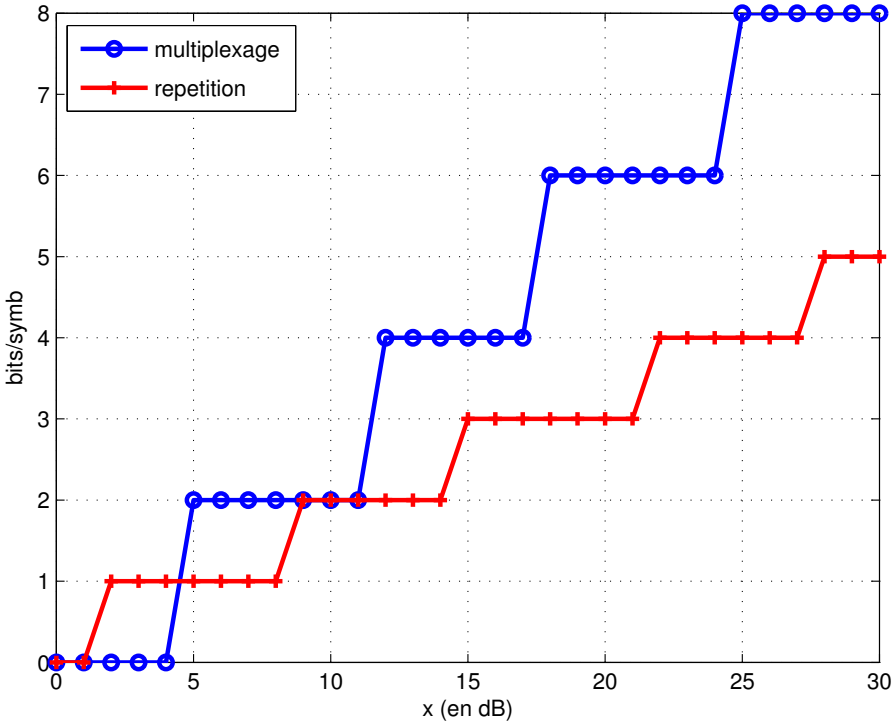



FIGURE 2 – $m_1$ et $m_2$ en fonction de $E_s/(\Gamma N_0)$

avec $S_n = 2|h^{(n)}|^2/N_0$ dans l'intervalle $[0, 2E_s]$.
Il suffit donc d'étudier la fonction $f$ sur $[0, 2E_s]$. Calculons sa dérivée :

$$f'(x) \propto \frac{S_1}{1+S_1 x} - \frac{S_2}{1+S_2(2E_s - x)}.$$

Cette dérivée s'annule en $x_0$ si $x_0$ vérifie

$$x_0 = E_s + \frac{1}{2}\frac{S_1 - S_2}{S_1 S_2}.$$

En dessous de $x_0$ la fonction est croissante, au dessus de $x_0$ elle est décroissante.
Le problème maintenant est de savoir si $x_0$ est dans l'intervalle $[0, 2E_s]$ C'est pourquoi, on obtient trois cas

— Si $(S_1 - S_2)/S_1S_2 \in [-2E_s, E_s]$, alors $x_0$ est dans l'intervalle voulu et donc $E^{(1)}_{s,\text{opt}} = E_s + \frac{1}{2}\frac{S_1 - S_2}{S_1 S_2}$ et $E^{(2)}_{s,\text{opt}} = E_s + \frac{1}{2}\frac{S_2 - S_1}{S_1 S_2}$.
— Si $(S_1 - S_2)/S_1S_2 \in (-\infty, -2E_s]$, alors $x_0$ est négatif et dans l'intervalle voulu la fonction est décroissante et donc il faut prendre $E^{(1)}_{s,\text{opt}} = 0$ et $E^{(2)}_{s,\text{opt}} = 2E_s$. Ceci arrive notamment quand $S_2 \geq S_1$ et donc que la voie 2 est bien meilleure que la voie 1. On n'allume alors que la voie 2.
— Si $(S_1 - S_2)/S_1S_2 \in [2E_s, \infty)$, alors $x_0$ est supérieur à $2E_s$ et dans l'intervalle voulu la fonction est croissante et donc il faut prendre $E^{(1)}_{s,\text{opt}} = 2E_s$ et $E^{(2)}_{s,\text{opt}} = 0$.

Cette attribution non-uniforme des énergies s'appelle le *waterfilling*.

# Chapitre 3

**Exercice 3.1**

1. Par la définition de l'information mutuelle

$$\begin{aligned} I(X;g(X)) &= H(X) - H(X|g(X)) \\ &= H(g(X)) - H(g(X)|X). \end{aligned} \tag{4}$$

On a $H(g(X)|X) = 0$ car $g(X)$ est une fonction de $X$. De plus $H(X|g(X)) \geq 0$ car toute entropie conditionnelle est positive. Elle vaut 0 si et seulement si $X$ est parfaitement déterminé par $g(X)$ ce qui ne peut arriver que lorsque $g$ est une fonction bijective. En combinant tous ces arguments, nous obtenons bien que

$$H(X) = H(g(X)) + \underbrace{H(X|g(X))}_{\geq 0} \geq H(g(X)) \tag{5}$$

avec égalité si et seulement si la fonction $g$ est bijective.

Nous avons démontré que la fonctionnelle d'une variable aléatoire peut seulement diminuer l'entropie et donc jamais l'augmente. Il n'y a pas de diminution si et seulement si le traitement est réversible.

2. Comme indiqué dans l'énoncé, pour toutes variables aléatoires $U$ et $V$ sur deux alphabets finis $\mathcal{U}$ et $\mathcal{V}$, la paire $(U, V)$ est une variable aléatoire sur l'alphabet fini $\mathcal{T} = \mathcal{U} \times \mathcal{V}$. Soit $T = (U, V)$ et soit $g$ la fonction

$$\begin{aligned} g\colon \qquad \mathcal{T} &\to \mathcal{W} \\ t = (u, v) &\mapsto w = u + v. \end{aligned} \tag{6}$$

En utilisant la résultat démontré dans 1, nous obtenons que

$$H(W) = H(g(T)) \leq H(T) = H(U, V). \tag{7}$$

Il y aura égalité si $g(\cdot)$ est bijective et donc si à partir de $W$ on peut récupérer la paire $(U, V)$.

Nous notons aussi que pour tout $U$ et $V$ l'entropie cojointe $H(U, V)$ vérifie l'inégalité :

$$H(U, V) \leq H(U) + H(V) \tag{8}$$

avec égalité si et seulement si $U$ et $V$ sont indépendants.

En combinant les équations (7) et (8), nous obtenons

$$H(W) \leq H(U) + H(V) \tag{9}$$

avec égalité si et seulement si $U$ et $V$ sont indépendants et $(U, V) \mapsto U + V$ est inversible.

Un exemple pour lequel $H(W) = H(U) + H(V)$ :
— $U$ uniforme sur $\{1, 7\}$ ;
— $V$ indépendant de $U$ et uniforme sur $\mathcal{V} = \{10, 20, 30, 40\}$.
Dans ce cas $W$ est de loi uniforme sur $\{11, 17, 21, 27, 31, 37, 41, 47\}$ et donc son entropie est égale à $H(W) = \log_2(8) = 3$. Comme $H(U) = 1$ et $H(V) = 2$, on voit que $H(W) = H(U) + H(V)$.

**Exercice 3.2**

1. Le choix de $\hat{x}$ qui minimize $p_{\mathrm{e}}(\hat{x})$ est l'élément de $\mathcal{X}$ de plus grande probabilité. Donc, dans l'exemple, le meilleur choix est $\hat{x} = 1$ et $p_{\mathrm{e}}^{\star} = \sum_{i=2}^{m} p_i = 1 - p_1$.
2. Soit $\tilde{X}$ une variable aléatoire sur l'alphabet $\{2, \ldots, m\}$ suivant la loi $P_{\tilde{X}}(i) = \frac{p_i}{p_{\mathrm{e}}^{\star}}$, pour $i = 2, \ldots, m$. On

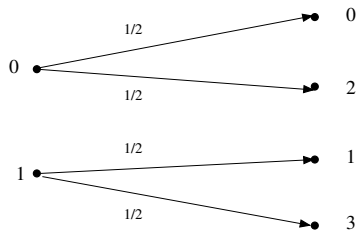
trouve

$$\begin{aligned} H(X) &= -\sum_{i=1}^{m} p_i \log_2(p_i) \\ &= -p_1 \log_2(p_1) - \sum_{i=2}^{m} p_i \log_2(p_i) \\ &= -(1-p_{\mathrm{e}}^\star)\log_2(1-p_{\mathrm{e}}^\star) - p_{\mathrm{e}}^\star \sum_{i=2}^{m} \frac{p_i}{p_{\mathrm{e}}^\star} \log_2\left(\frac{p_i}{p_{\mathrm{e}}^\star}\cdot p_{\mathrm{e}}^\star\right) \\ &= -(1-p_{\mathrm{e}}^\star)\log_2(1-p_{\mathrm{e}}^\star) - p_{\mathrm{e}}^\star \sum_{i=2}^{m} \frac{p_i}{p_{\mathrm{e}}^\star} \log_2\left(\frac{p_i}{p_{\mathrm{e}}^\star}\right) - p_{\mathrm{e}}^\star \sum_{i=2}^{m} \frac{p_i}{p_{\mathrm{e}}^\star} \log_2(p_{\mathrm{e}}^\star) \\ &= -(1-p_{\mathrm{e}}^\star)\log_2(1-p_{\mathrm{e}}^\star) - p_{\mathrm{e}}^\star \log_2(p_{\mathrm{e}}^\star) + p_{\mathrm{e}}^\star H(\tilde{X}) \\ &= H_{\mathrm{b}}(p_{\mathrm{e}}^\star) + p_{\mathrm{e}}^\star H(\tilde{X}). \end{aligned} \tag{10}$$

3. Comme l'alphabet de $\tilde{X}$ prend $m-1$ valeurs, $H(\tilde{X}) \leq \log_2(m-1)$. En utilisant l'inégalité prouvée dans la question 2., cela justifie l'inégalité de l'énoncé.

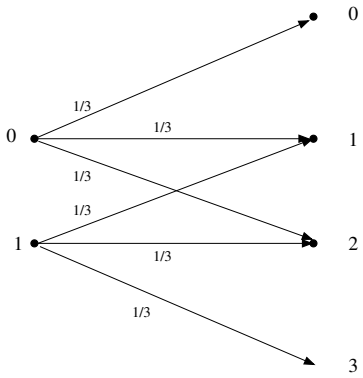
**Exercice 3.3**

1.1 Pour $\mathcal{Z} = \{0,2\}$, l'alphabet de sortie est $\mathcal{Y} = \{0,1,2,3\}$ et le diagramme du canal est :




Nous avons donc affaire à un canal sans bruit où à partir des symboles de sortie on peut toujours récupérer les symboles d'entrée. Donc, $H(X|Y) = 0$ et $I(X;Y) = H(X)$. Dans ce cas, la capacité du canal est égale au logarithme (en base 2) de la cardinalité de l'alphabet d'entrée $\mathcal{X}$ qui est ici $\{0,1\}$. D'où $C = \log_2(2) = 1$.

2.1 Pour $\mathcal{Z} = \{0,1,2\}$, l'alphabet de sortie est $\mathcal{Y} = \{0,1,2,3\}$ et le diagramme du canal est :




2.2 On calcule $I(X;Y) = H(X) - H(X|Y)$. Si $X$ suit une loi de Bernoulli-$q$, on a

$$H(X) = H_{\mathrm{b}}(q). \tag{11}$$

Les valeurs de sortie $y \in \{0,3\}$, correspondent à une entrée unique. Donc $H(X|Y=0) = H(X|Y=3) = 0$ et

$$H(X|Y) = \sum_{y=0}^{3} P_Y(y) H(X|Y=y) = \sum_{y=1}^{2} P_Y(y) H(X|Y=y). \tag{12}$$

Notons que par la formule des probabilités totales et la formule de Bayes,

$$P_Y(1) = P_{XY}(0,1) + P_{XY}(1,1) = P_X(0)P_{Y|X}(1|0) + P_X(1)P_{Y|X}(1|1) = (1-q)\frac{1}{3} + q\frac{1}{3} = \frac{1}{3}. \tag{13}$$

Par des arguments similaires, nous trouvons aussi que

$$P_Y(2) = 1/3. \tag{14}$$

Pour trouver $H(X|Y=1)$, calculons $P_{X|Y}(1|1)$. En applicant la formule de Bayes deux fois, on a

$$P_{X|Y}(1|1) = \frac{P_{X,Y}(1,1)}{P_Y(1)} = \frac{P_{X,Y}(1,1)}{P_Y(1)} = \frac{P_X(1)P_{Y|X}(1|1)}{P_Y(1)} = \frac{q\frac{1}{3}}{\frac{1}{3}} = q. \tag{15}$$

Donc, si la sortie $Y$ est égale à 1, alors l'entrée $X$ suit une loi Bernoulli-$q$ ce qui implique que

$$H(X|Y=1) = H_{\mathrm{b}}(q). \tag{16}$$

De la même façon,

$$H(X|Y=2) = H_{\mathrm{b}}(q). \tag{17}$$

Grâce aux équations (12), (13), (14), (16), (17), on obtient que :

$$H(X|Y) = 2\frac{1}{3}H_{\mathrm{b}}(q) = \frac{2}{3}H_{\mathrm{b}}(q). \tag{18}$$

En utilisant alors l'équation (11), on a finalement :

$$I(X;Y) = H_{\mathrm{b}}(q) - \frac{2}{3}H_{\mathrm{b}}(q) = \frac{1}{3}H_{\mathrm{b}}(q). \tag{19}$$

2.3 La capacité du canal est donc :

$$C = \max_{q\in[0,1]} I(X;Y) = \max_{q\in[0,1]} \frac{1}{3}H_{\mathrm{b}}(q) = \frac{1}{3}. \tag{20}$$

En fait, le canal dans cette partie 2., peut être considéré comme un BEC avec 2 symboles d'effacements différents : 1,2. En effet, si on observe $Y = y \in \{1,2\}$, alors cela ne nous donne aucune information sur l'entrée : dans ces cas $H(X|Y=1) = H(X|Y=2) = H(X)$ et donc ces observations ne diminuent pas l'entropie sur l'entrée !

Le résultat obtenu dans l'équation (20) indique que la capacité de ce BEC avec 2 symboles d'effacement est égale à la capacité d'un BEC avec un seul symbole d'effacement qui se réalise avec une probabilité égale à la somme des probabilité des 2 symboles d'effacement. De manière générale, on pourrait montrer que la capacité est toujours égale à

$$C = 1 - \text{somme des probabilités de chaque effacement.} \tag{21}$$

**Exercice 3.4**

1. On choisit une loi d'entrée équiprobable :

$$P_X(x) = 1/4, \qquad \forall x \in \{0,1,2,3\}. \tag{22}$$

Pour ce choix, $Y$ est aussi équiprobable, $P_Y(i) = 1/4$, pour $i = 0,1,2,3$. En appliquant la règle de Bayes, nous avons

$$P_{X|Y}(x|y) = \frac{P_{XY}(x,y)}{P_Y(y)} = P_{Y|X}(y|x).$$

Donc, pour chaque $y \in \{0,1,2,3\}$, les quatre valeurs de $P_{X|Y}(x|y)$ pour les quatre valeurs de $x \in \{0,1,2,3\}$ sont 0, 0, $p$, et $1-p$. Donc

$$H(X|Y=y) = H_{\mathrm{b}}(p), \qquad \forall y \in \{0,1,2,3\}.$$

Avec notre choix de $P_X$ dans (22) nous obtenons donc pour l'information mutuelle

$$I(X;Y) = H(X) - H(X|Y) = 2 - \sum_{y\in\{0,1,2,3\}} H_{\mathrm{b}}(p) P_Y(y) = 2 - H_{\mathrm{b}}(p).$$

2. Pour toute loi d'entrée $P_X$ :

$$I(X;Y) = H(Y) - H(Y|X) \leq 2 - \sum_{x\in\{0,1,2,3\}} H_{\mathrm{b}}(p) P_X(x) = 2 - H_{\mathrm{b}}(p).$$

Avec le résultat dans 1., nous obtenons que la capacité du canal est égale à $C = 2 - H_{\mathrm{b}}(p)$.

**Exercice 3.5**

1. Soient $Z_1 := X_1 \oplus Y_1$ et $Z_2 := X_2 \oplus Y_2$. Les deux variables aléatoires $Z_1$ et $Z_2$ indiquent si les deux BSCs inversent le bit d'entrée ou pas. Elles suivent donc une loi Bernoulli $p$, et elles sont mutuellement indépendantes et indépendantes de la paire $(X_1, X_2)$.
Notons que

$$\begin{aligned}
I(X_1,X_2;Y_1,Y_2) &= H(Y_1,Y_2) - H(Y_1,Y_2|X_1,X_2) \\
&= H(Y_1,Y_2) - H(Z_1 \oplus X_1, Z_2 \oplus X_2|X_1,X_2) \\
&= H(Y_1,Y_2) - \sum_{x_1\in\{0,1\}} \sum_{x_2\in\{0,1\}} H(Z_1 \oplus x_1, Z_2 \oplus x_2|X_1=x_1,X_2=x_2) P_{X_1X_2}(x_1,x_2) \\
&= H(Y_1,Y_2) - \sum_{x_1\in\{0,1\}} \sum_{x_2\in\{0,1\}} H(Z_1,Z_2|X_1=x_1,X_2=x_2) P_{X_1X_2}(x_1,x_2) \\
&= H(Y_1,Y_2) - H(Z_1,Z_2|X_1,X_2) \\
&= H(Y_1,Y_2) - H(Z_1,Z_2) \\
&= H(Y_1,Y_2) - H(Z_1) - H(Z_2) \\
&= H(Y_1,Y_2) - 2H_{\mathrm{b}}(p),
\end{aligned}$$

où la troisième et la cinquième égalités proviennent de la définition de l'entropie conditionelle ; la quatrième égalité tient parce que pour chaque pair $(x_1, x_2)$ il existe une fonction bijective entre $(Z_1, Z_2)$ et $(Z_1 \oplus x_1, Z_2 \oplus x_2)$ ; et la sixième et la septième égalités sont obtenues parce que $Z_1$ et $Z_2$ sont mutuellement indépendantes et aussi indépendantes de $(X_1, X_2)$.
2. Comme $Y_1$ et $Y_2$ sont binaires, on a $H(Y_1,Y_2) \leq H(Y_1) + H(Y_2) \leq 2$.
3. Notons que $C = \max_{P_{X_1,X_2}} I(X_1,X_2;Y_1,Y_2)$. Selon la question 2., nous avons donc $C \leq 2(1 - H_{\mathrm{b}}(p))$. De plus, en choisissant $X_1$ et $X_2$ i.i.d. de loi Bernoulli-$1/2$, les variables de sorties du canal $Y_1$ et $Y_2$ sont aussi i.i.d. et de loi Bernoulli-$1/2$. Ceci implique que les inégalités dans la réponse à la question 2. sont effectivement des égalités. Nous concluons donc que $C = 2(1 - H_{\mathrm{b}}(p))$, c'est-à-dire la capacité est égale à la somme des capacités des deux BSC.

**Exercice 3.6**

1. Le canal composé est un BSC($\delta/2$). Son diagramme est dessiné sur la figure 3.
2. Le premier canal $P_{Z|X}$ est un BEC($\delta$). Sa capacité est donc $1-\delta$.
La capacité du deuxième canal $P_{Y|Z}$ est égale à 1. D'un côté la capacité ne peut pas dépasser 1 comme la sortie du deuxième canal est binaire. D'autre côté, elle est au moins égale a 1, car $I(Z;Y) = 1$ quand $Z$ prend les deux valeurs 0 et 1 avec probabilité $1/2$ chacune.
La capacité du canal composé est $1 - H_{\mathrm{b}}(\delta/2)$ comme c'est un BSC($\delta/2$).
La capacité la plus grande est donc celle du « BEC inversé ».

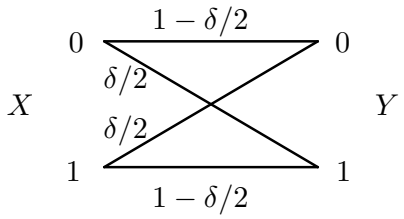



FIGURE 3 – Diagramme de canal

3. Selon la définition 2.6 et le théorème 2.5, la capacité $C$ d'un DMC est le supremum de tous les taux qui sont atteignables sur ce DMC. Comme indiqué dans le polycopié, ceci vaut aussi quand l'émetteur ou le récepteur font le codage ou le décodage de façon aléatoire.
   Nous prouvons d'abord que la capacité du DMC composé $P_{Y|X}$ ne peut pas dépasser la capacité du premier DMC $P_{Z|X}$. Soit $R > 0$ un taux qui est atteignable sur le DMC composé $P_{Y|X}$ et soit $\{(n, k = \lfloor nR \rfloor, f^{(n)}, g^{(n)})\}_{n=1}^{\infty}$ une suite de codes pour ce DMC tel que la probabilité d'erreur $P_{\text{e}}^{(n)}$ tend vers 0 quand $n \to \infty$.
   Nous allons construire une suite de codes $\{(n, k = \lfloor nR \rfloor, \tilde{f}^{(n)}, \tilde{g}^{(n)})\}_{n=1}^{\infty}$ pour le DMC $P_{Z|X}$ telle que sa probabilité d'erreur $\tilde{P}_{\text{e}}^{(n)}$ tend vers 0 quand $n \to \infty$. Ceci impliquera que le taux $R$ est aussi atteignable sur le DMC $P_{Z|X}$. Comme $R$ a été choisi arbitrairement au dessous de la capacité du DMC $P_{Y|X}$ cela prouvera que la capacité du DMC $P_{Y|X}$ ne peut pas dépasser la capacité du DMC $P_{Z|X}$.
   Nous choisissons pour tout $n$ :
   — $\tilde{f}^{(n)} = f^{(n)}$ ; et
   — $\tilde{g}^{(n)}$ se construit en deux étapes : le récepteur passe d'abord les symboles reçus $Z_1, \ldots, Z_n$ sur un DMC $P_{Y|Z}$, et il applique la fonction de décodage $g^{(n)}$ aux symboles $Y_1, \ldots, Y_n$ ainsi générés.
   On observe que pour tout $n$ :
   $$\tilde{P}_{\text{e}}^{(n)} = P_{\text{e}}^{(n)},$$
   ce qui prouve que $\tilde{P}_{\text{e}}^{(n)}$ tend vers 0 quand $n \to \infty$ comme désiré, et conclut la preuve que la capacité du DMC $P_{Y|X}$ ne peut pas dépasser la capacité du DMC $P_{Z|X}$.
   La preuve que la capacité du DMC composé $P_{Y|X}$ ne peut pas dépasser la capacité du deuxième DMC $P_{Y|Z}$ se fait de manière analogue. Il suffit de changer la preuve comme suit :
   — remplacer $P_{Z|X}$ par $P_{Y|Z}$ partout ;
   — choisir de construire $\tilde{f}^{(n)}$ en deux étapes : l'émetteur applique la fonction de codage $f^{(n)}$ aux bits d'information $D_1, \ldots, D_k$ et passe les symboles codés $X_1, \ldots, X_n$ par un DMC $P_{Z|X}$. Finalement, il envoie les symboles $Z_1, \ldots, Z_n$ ainsi générés sur le DMC $P_{Y|Z}$ ; et
   — choisir $\tilde{g}^{(n)} = g^{(n)}$.
4. La capacité d'un BSC($p$), c'est-à-dire $1 - H_{\text{b}}(p)$, est strictement décroissante en $p \in (0, 1/2)$. Donc, la capacité d'un BSC($\epsilon$) doit être inférieure à celle d'un BSC($\epsilon/2$). Or, dans la question 3. nous avons prouvé que la capacité d'un BEC($\epsilon$) ne peut pas être inférieure à la capacité d'un BSC($\epsilon/2$), la capacité d'un BEC($\epsilon$) dépasse celle d'un BSC($\epsilon$) pour tout $0 < \epsilon < 1/2$.