Département COMELEC UE COM105

Corrigé du TD 2

Exercice 1

1. Chaque mot de code est constitué de $n = 7$ bits, et plus précisément des bits $d_1, d_2, d_3, r_1, r_2, r_3, r_4$ décrits dans l'énoncé.
   Les trois premiers bits d_1, d_2, d_3 sont nommés bits d'information car ils peuvent être choisis librement. Les quatre derniers bits r_1, r_2, r_3, r_4 sont nommés bits de parité car ils sont entièrement déterminés par les bits d'information. En effet, l'énoncé décrit comment on trouve les bits r_1, r_2, r_3, r_4 à partir des bits d_1, d_2, d_3. Le code $\mathcal{C}$ contient donc 8 mots de code : un mot pour chaque triplet de bits d_1, d_2, d_3. Par conséquent, $k = \log_2 |\mathcal{C}| = \log_2(8) = 3$.
   Le rendement r du code vaut $r = \frac{k}{n} = \frac{3}{7}$.
2. La matrice génératrice est une matrice avec $k = 3$ lignes et $n = 7$ colonnes. On se rappelle qu'en général on trouve une matrice génératrice G d'un code $\mathcal{C}$ en choisissant k mots du code $\mathcal{C}$ qui sont linéairement indépendants, et en mettant ces k mots en tant que lignes de G. Comme $k = 3$, cela veut dire qu'on cherche trois mots de code $c^{(1)}, c^{(2)}, c^{(3)} \in \mathcal{C}$ qui sont différents et qui ne s'annulent pas :
   $$c^{(1)} + c^{(2)} + c^{(3)} \neq (0,0,0,0,0,0,0), \tag{1}$$
   où on se rappelle que les additions sont par composante et sur $\mathbb{F}_2$, c'est-à-dire $1 + 1 = 0$.
   Un choix possible pour les trois mots de code est $c^{(1)} = (1,0,0,1,1,1,0)$, $c^{(2)} = (0,1,0,0,1,1,1)$, et $c^{(3)} = (0,0,1,1,1,0,1)$, ce qui induit la matrice génératrice :
   $$G = \begin{bmatrix} 1 & 0 & 0 & 1 & 1 & 1 & 0 \\ 0 & 1 & 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 & 1 & 0 & 1 \end{bmatrix}.$$
   On note qu'on choisissant judicieusement les trois mots de code $c^{(1)}, c^{(2)}, c^{(3)}$, on obtient une matrice génératrice sous forme systématique, c'est-à-dire une matrice génératrice du type
   $$G = \left[\begin{array}{c|c} I_3 & P \end{array} \right]. \tag{2}$$
   où I_k désigne la matrice d'identité de dimension $k = 3$ et P la matrice
   $$G = \begin{bmatrix} 1 & 1 & 1 & 0 \\ 0 & 1 & 1 & 1 \\ 1 & 1 & 0 & 1 \end{bmatrix}.$$
   Comme vu dans le cours, à partir de cette matrice génératrice il est facile de trouver une matrice de contrôle de parité (voir les équations (1.15) et (1.16) dans le polycopié) :
   $$H = \left[\begin{array}{c|c} -P^{\mathsf{T}} & I_4 \end{array} \right] = \begin{bmatrix} 1 & 0 & 1 & 1 & 0 & 0 & 0 \\ 1 & 1 & 1 & 0 & 1 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 & 1 & 0 \\ 0 & 1 & 1 & 0 & 0 & 0 & 1 \end{bmatrix},$$
   où l'on note que $-1 = 1$ et $-0 = 0$ car on se trouve sur $\mathbb{F}_2$. Cette matrice de contrôle de parité est encore sous forme systématique.

3. On trouve $d_{\min} = 4$ ce qu'on peut voir de la façon suivante. D'après le résultat 1.7 du polycopié, $d_{\min}$ est égale au plus petit nombre de colonnes linéairement dépendantes de H. (On rappelle que sur $\mathbb{F}_2$ un ensemble de vecteurs est linéairement dépendant si la somme de ces vecteurs est égale au vecteur nul $\mathbf{0}$. L'addition est toujours à effectuer par composante et sous $\mathbb{F}_2$ où $1 + 1 = 0$.)

   Pour trouver $d_{\min} = 4$, on note d'abord qu'il n'y a pas de colonne nulle dans H et donc $d_{\min} \neq 1$. On constate aussi que $d_{\min} \neq 2$ parce que toutes les colonnes de H sont différentes et donc la somme de chaque paire de colonnes n'est pas nulle. De plus, $d_{\min} \neq 3$ parce que toutes les colonnes de H ont un nombre *impair* de 1, et toutes les sommes de deux colonnes de H ont un nombre *pair* de 1. Il n'y a donc pas de triplet de colonnes de H qui somment au vecteur nul. Finalement on note que $d_{\min} \leq 4$ parce que les colonnes $1, 2, 4, 7$ sont linéairement dépendantes. Toutes ces considérations réunies démontrent que $d_{\min} = 4$.

4. Sur un canal de transmission BEC il n'y a que des effacements. Il n'y a pas d'inversion de bits. Nous sommes donc sûrs qu'un bit reçu (non effacé) est égal au bit envoyé.

   Il faut donc trouver des valeurs $\{0, 1\}$ à mettre à la place des effacements de manière à obtenir un mot de code. Ceci est toujours possible, comme le canal n'a pas inversé de bits. En effet, quand $d_{\min}$ bits ou plus ont été effacés, il peut exister deux façons différentes de remplir les effacements. Par exemple, on considère deux mots du code $\mathcal{C}$ de distance $d_{\min}$. Si tous les $d_{\min}$ bits où les deux mots de code diffèrent sont effacés, alors il n'y a plus moyen de savoir lequel des deux mots de code a été envoyé. Par contre, quand moins de $d_{\min}$ bits ont été effacés, il existe une seule façon de remplir les effacements. S'il y avait deux façons, alors cela voudrait dire que le code $\mathcal{C}$ contient deux mots de code de distance inférieure à $d_{\min}$.

   Notre code peut donc remplir 1 ou 2 ou $d_{\min} - 1 = 3$ effacements.

5. Comme $d_{\min} = 4$, le code peut corriger toutes les configurations à une erreur, lorsqu'il est utilisé sur un BSC. Ceci est directement impliqué par le résultat 1.2 du polycopié.

6. On trouve la solution facilement en suivant les explications à la page 19 du polycopié.

   Soit $\mathbf{s}_i$ le syndrome dans le cas où le canal change seulement le bit à la position i. On considère alors le cas $\mathbf{y} = \mathbf{c} + \mathbf{e}_i$, où c est un mot du code $\mathcal{C}$ et $\mathbf{e}_i$ est le vecteur avec tous les éléments 0 sauf à la position i. Comme expliqué dans le polycopié, dans ce cas :

$$\mathbf{y}H^T = (\mathbf{c} + \mathbf{e}_i)H^T = \mathbf{e}_i H^T = \mathbf{h}_i^T,$$

   où $\mathbf{h}_i$ correspond à la $i$ème colonne de H.

   Donc, pour la matrice H trouvée dans la partie 2.) :

$$\mathbf{s}_1 = (1110); \quad \mathbf{s}_2 = (0111); \quad \mathbf{s}_3 = (1101); \quad \mathbf{s}_4 = (1000); \quad \mathbf{s}_5 = (0100); \quad \mathbf{s}_6 = (0010); \quad \mathbf{s}_7 = (0001). \tag{3}$$

   Soit $\mathbf{s}_{i,i+1}$ le syndrome dans le cas où le canal change les deux bits consécutifs aux positions i et $i + 1$ mais aucun autre bit. On considère alors le cas $\mathbf{y} = \mathbf{c} + \mathbf{e}_i + \mathbf{e}_{i+1}$, où c est un mot du code $\mathcal{C}$ et $\mathbf{e}_i$ et $\mathbf{e}_{i+1}$ sont les vecteurs avec tous les éléments 0 sauf à la position i ou à la position $i + 1$. Comme expliqué dans le polycopié, dans ce cas :

$$\mathbf{y}H^T = (\mathbf{c} + \mathbf{e}_i + \mathbf{e}_{i+1})H^T = (\mathbf{e}_i + \mathbf{e}_{i+1})H^T = \mathbf{h}_i^T + \mathbf{h}_{i+1}^T.$$

   Donc, pour la matrice H trouvée dans la partie 2.) :

$$\mathbf{s}_{1,2} = (1001); \quad \mathbf{s}_{2,3} = (1010); \quad \mathbf{s}_{3,4} = (0101); \quad \mathbf{s}_{4,5} = (1100); \quad \mathbf{s}_{5,6} = (0110); \quad \mathbf{s}_{6,7} = (0011). \tag{4}$$

   Noter que les 13 syndromes listés dans (3) et (4) sont tous différents.

7. L'algorithme de décodage suivant est capable de corriger toutes les configurations d'une seule erreur et toutes les configurations de deux erreurs *consécutives*.
   — Calculer le syndrome $\mathbf{s} = \mathbf{y}H^T$.
   — Si $\mathbf{s} = \mathbf{0}$, alors déclarer que le mot envoyé est égal au mot reçu : $\mathbf{x} = \mathbf{y}$.
   — Autrement, comparer $\mathbf{s}$ aux 13 syndromes listés au-dessus, c'est-à-dire à $\mathbf{s}_1, \ldots, \mathbf{s}_7$ et $\mathbf{s}_{1,2}, \ldots, \mathbf{s}_{6,7}$.
   Si $\mathbf{s} = \mathbf{s}_i$ pour un $i \in \{1, \ldots, 7\}$, inverser le bit i du mot reçu $\mathbf{y}$ et déclarer que ce mot a été envoyé.
   Si $\mathbf{s} = \mathbf{s}_{i,i+1}$ pour un $i \in \{1, \ldots, 6\}$, inverser les deux bits i et $i + 1$ de $\mathbf{y}$, et déclarer que ce mot a été envoyé.
   — Si $\mathbf{s}$ n'est ni $\mathbf{0}$ ni parmi les 13 syndromes listés au-dessus, déclarer une erreur.

   Noter que la valeur de $d_{\min} = 4$ indique seulement que toutes les configurations d'une seule erreur peuvent être corrigées.

8. Non, un tel algorithme ne peut pas exister comme $d_{\min} < 5$. Plus précisément, le problème est qu'il existe deux mots de codes $\mathbf{x}$ et $\mathbf{x}'$ qui avec des configurations de deux erreurs donnent lieu au même mot reçu $\mathbf{y}$. Par exemple, les deux mots de code $\mathbf{x} = (1010011)$ et $\mathbf{x}' = (0000000)$ et les deux vecteurs d'erreurs de poids 2, $\mathbf{e} = (1, 0, 1, 0, 0, 0, 0)$ et $\mathbf{e}' = (0, 0, 0, 0, 0, 1, 1)$, donnent lieu au même mot reçu

$$\mathbf{y} = \mathbf{x} + \mathbf{e} = \mathbf{x}' + \mathbf{e}' = (0000011).$$

A partir de ce mot reçu $\mathbf{y}$, c'est donc impossible de trouver avec certitude le mot de code envoyé, même si on sait que le canal introduit deux erreurs.

Exercice 2

1. *Code étendu* :

1.1 G étant sous forme systématique on déduit la matrice H de $\mathcal{C}$ facilement (voir les équations (1.15) et (1.16) dans le polycopié) :

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

Sa distance minimale est égale à 3 correspondant au nombre minimal de colonnes de H linéairement dépendantes. (Pour plus d'explications voir le résultat 1.7 dans le polycopié et la correction de la partie 3. de l'exercice 1 au dessus.)

1.2 On se rappelle que le code original est un code $\mathcal{C}(6, 3)$, ce qui veut dire qu'il est de longueur 6 et de dimension 3.

Pour le code étendu on ajoute un bit à chaque mot de code ce qui augmente la longueur du code de 1. La longueur du code étendu est donc $n = 6 + 1 = 7$.

Le bit ajouté à chaque mot de code est entièrement déterminé par les autres bits du mots de code. Le nombre de mots de code donc n'augmente pas, et la dimension du code reste $k = 3$.

1.3 Puisque $n = 7$ et $k = 3$, la matrice de parité H_E a $n = 7$ colonnes et $n - k = 4$ lignes. On recherche donc une matrice de contrôle de parité de taille 4×7 et rang 4 telle que $\mathbf{c}H_E^T = \mathbf{0}$ pour tous les mots du code étendu.

On déduira les trois premières lignes de H_E à partir de H. Pour cela on note que les 6 premiers bits des mots de code étendus sont égaux aux bits des mots du code original. Si on ajoute alors une colonne nulle à la matrice H on garde l'orthogonalité avec les mots du code étendu :

$$\mathbf{c} \left[\begin{array}{c|c} H & \mathbf{0} \end{array} \right]^T = \mathbf{0}, \quad \text{pour tous les mots du code étendu.} \tag{5}$$

Il reste donc à trouver une dernière ligne à ajouter qui est orthogonale à tous les mots du code étendu et est linéairement indépendante des trois lignes de la matrice $[H \mid \mathbf{0}]$. Pour cette ligne on peut choisir par exemple le vecteur composé uniquement de symboles 1 :

$$\mathbf{v}_4 = (1, 1, 1, 1, 1, 1, 1). \tag{6}$$

En effet, on peut facilement vérifier que

$$\mathbf{c}\mathbf{v}_4^T = \mathbf{0}, \quad \text{pour tous les mots de codes du code étendu,} \tag{7}$$

car

$$\sum_{i=1}^{7} c_i = \sum_{i=1}^{6} c_i + c_7 = \sum_{i=1}^{6} c_i + \sum_{i=1}^{6} c_i = \sum_{i=1}^{6} (c_i + c_i) = \sum_{i=1}^{6} 0 = 0. \tag{8}$$

(On se rappelle que les additions sont toujours dans $\mathbb{F}_2$ et donc $1 + 1 = 0$.) Sinon, on peut vérifier l'orthogonalité simplement en notant que chaque mot du code étendu a un nombre pair de symboles 1.

Un choix possible pour la matrice H_E est donc le suivant :

$$H_E = \left[\begin{array}{c|c} H & \mathbf{0} \\ \hline \multicolumn{2}{c}{\mathbf{v}_4} \end{array} \right] = \begin{bmatrix} 1 & 0 & 1 & 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 1 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 & 1 & 0 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 \end{bmatrix}.$$

On note que cette matrice n'est pas sous forme systématique.

1.4 On trouve d'abord une matrice de contrôle de parité sous forme systématique. On se rappelle que toute matrice H_E de taille 4×7 et de rang 4 dont toutes les lignes sont orthogonales à tous les mots du code étendu, est une matrice de contrôle de parité pour ce code étendu. Par la linéarité du produit matriciel (ou du produit entre des vecteurs), on obtient que si pour des vecteurs $\mathbf{v}_1, \mathbf{v}_2, \mathbf{v}_3, \mathbf{v}_4$:

$$\mathbf{c}\mathbf{v}_i^T = \mathbf{0}, \quad \text{pour } i = 1, 2, 3, 4 \text{ et pour tous les mots de codes du code étendu} \tag{9}$$

alors il est aussi vrai que

$$\mathbf{c}\left(\sum_{i=1}^{4} \mathbf{v}_i\right)^T = \mathbf{0}, \quad \text{pour tous les mots de codes du code étendu.} \tag{10}$$

On a donc le droit de remplacer la dernière ligne de H_E par la somme des quatre lignes. Ceci nous amène à une matrice de contrôle de parité $H_{E,s}$ sous forme systématique :

$$H_{E,s} = \begin{bmatrix} 1 & 0 & 1 & 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 1 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 & 1 & 0 \\ 0 & 1 & 1 & 0 & 0 & 0 & 1 \end{bmatrix}. \tag{11}$$

En utilisant à nouveau les équations (1.15) et (1.16) dans le polycopié où on identifie la matrice $P = \begin{bmatrix} 1 & 0 & 1 \\ 1 & 1 & 0 \\ 1 & 1 & 1 \\ 0 & 1 & 1 \end{bmatrix}$, on obtient aussi une matrice génératrice sous forme systématique :

$$G_{E,s} = \begin{bmatrix} 1 & 0 & 0 & 1 & 1 & 1 & 0 \\ 0 & 1 & 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 & 0 & 1 & 1 \end{bmatrix}. \tag{12}$$

1.5 En examinant la matrice de contrôle de parité on trouve que la distance minimale du code étendu est égale à $d_{\min} = 4$. Ainsi, l'extension du code a permis d'augmenter la distance minimale du code. Pour trouver $d_{\min} = 4$ on peut suivre les étapes détaillées dans la partie 3. de l'exercice 1 au dessus. En particulier, on note que la première, quatrième, cinquième, et sixième colonne de $H_{E,s}$ sont linéairement dépendantes.

2. *Code rallongé* :

2.1 On rajoute un bit d'information et donc la dimension du code augmente et la dimension du code rallongé est $k = 3 + 1 = 4$. La longueur du code augmente aussi de 1 car on ajoute un bit à chaque mot de code. Donc $n = 6 + 1 = 7$.

2.2 La matrice de contrôle de parité H_R du code rallongé est de dimension 3×7. Ainsi le rallongement du code revient à rajouter une colonne à H.

2.3 Le seul choix possible de H_R qui permet de conserver la même distance minimale que le code originale, qui est $d_{\min} = 3$, est le suivant :

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

Pour tout autre choix de colonne à ajouter à H, la distance minimale $d_{\min}$ va diminuer.
Le code H_R n'est autre que le code de Hamming $(7, 4, 3)$.

3. *Code expurgé* :

3.1 Le sous ensemble $\mathcal{S}$ des mots de code de $\mathcal{C}$ est $\{(000000), (010011), (001101), (011110)\}$.

3.2 Connaissant le premier bit de tous les mots de code de $\mathcal{S}$, nous pouvons le supprimer. Pour le code expurgé on a : $n = 5$, $k = 2$, et $d_{min} = 3$.

3.3 A la réception, il suffit de rajouter au message reçu un premier bit à 0 et de procéder au décodage de $\mathcal{C}$.

A titre d'exemple, le code correcteur d'erreur utilisé dans la TNT, est un code expurgé $(204, 188)$ obtenu à partir du code Reed Solomon $(255, 239)$ en considérant les 51 premier bits à 0. Cette modification est nécessaire dans ce cas pour s'adapter à la taille des paquets de 188 bits délivrés par le codeur de source.