

TP 1 : Code détecteur et correcteurs d'erreurs

Question 0

Soit M le message et C le vecteur code :

$$C = \begin{pmatrix} h1 \\ h2 \\ h3 \\ b1 \\ b2 \\ b3 \\ b4 \end{pmatrix} \quad M = \begin{pmatrix} b1 \\ b2 \\ b3 \\ b4 \end{pmatrix}$$

Chaque bit de hamming (h_n) contrôle la parité d'un groupe de 3 bits. Ainsi on a :

$$h1 = 1 \times M + 0 \times M + 1 \times M + 1 \times M$$

$$h2 = 1 \times M + 1 \times M + 1 \times M + 0 \times M$$

$$h3 = 0 \times M + 1 \times M + 1 \times M + 1 \times M$$

Le code associé au message M peut s'exprimer par la relation matricielle suivante :

$$C = {}^t G \times M$$

$$h1 = e_1 \times b_1 + e_2 \times b_2 + e_3 \times b_3 + e_4 \times b_4$$

$$h2 = f_1 \times b_1 + f_2 \times b_2 + f_3 \times b_3 + f_4 \times b_4$$

$$h3 = g_1 \times b_1 + g_2 \times b_2 + g_3 \times b_3 + g_4 \times b_4$$

Eninf on trouve G, une matrice de taille (m,n) avec m=4 et n=7 :

$$G = \begin{pmatrix} e_1 & f_1 & g_1 & 1 & \cdots & \cdots & 0 \\ e_2 & f_2 & g_2 & \vdots & \ddots & & \vdots \\ e_3 & f_3 & g_3 & \vdots & & \ddots & \vdots \\ e_4 & f_4 & g_4 & 0 & \cdots & \cdots & 1 \end{pmatrix}$$

Question 3

L'opération réalisée à la ligne 20 est l'ajout d'une erreur sur chaque bit. On ajoute 1 afin d'avoir une valeur différente à chaque fois et tout cela en effectuant l'opération modulo 2.

Le vecteur Z représente à chaque itération le vecteur syndrome pour un bit donné de C1.

Si Z est égal au vecteur nul alors il n'y a pas d'erreurs. Sinon $Z=H.C1$

Les lignes 20 et 23 sont identiques mais elles n'ont pas le même but. La première ligne sert à ajouter une erreur sur chaque bit alors que la seconde sert à enlever l'erreur présente sur chaque bit de la même manière qu'elle a été créée.

H =

1	0	0	1	0	1	1
0	1	0	1	1	1	0
0	0	1	0	1	1	1

Zn =

1	0	0	1	0	1	1
0	1	0	1	1	1	0
0	0	1	0	1	1	1

Zn et H sont identiques : Zn = H

Chaque colonne de Zn représente le syndrome d'erreur pour chaque bit.

Question 4

La ligne 31 sert à comparer Z et H. Cette comparaison est effectuée à l'aide de la fonction symerr.

C =

1
1
1
1
1
0
1
1

Z =

0
1
1

i =

5

C_corrected =

1
1
1
1
1
1
1
1

R =

1
1
1
1

Question 5

- 1) On convertit le z10 scalaire à l'aide de la fonction `bi2de()` ce qui permet de convertir un vecteur binaire en un décimal.

```

Command Window

z10 =
    4

Z =
    1
    0
    0

z10 =
    2

Z =
    0
    1
    0

z10 =
    1

Z =
    0
    0
    1

P =

    0     0     0     0     0     0     0
    0     0     1     0     0     0     0
    0     1     0     0     0     0     0
    0     0     0     0     1     0     0
    1     0     0     0     0     0     0
    0     0     0     1     0     0     0
    0     0     0     0     0     1     0

T =

    0     0     0     0     0     0     0
    0     0     1     0     0     0     0
    0     1     0     0     0     0     0
    0     0     0     0     1     0     0
    1     0     0     0     0     0     0
    0     0     0     1     0     0     0
    0     0     0     0     0     1     0

>>

```

Les matrices T et P sont identiques. ON voit donc que la matrice de position d'erreur que nous avons créée est identique avec la table de syndrome T générée par fonction syndtable() de MATLAB.

Question 9

L'encodage à l'aide de la fonction convenc() permet de créer un code convolutif.

Pour le développement du treillis, cf annexe papier.

Question 10

Pour toutes les valeurs de i (la position de l'erreur), R reste constant.

```
Command Window

codefaux =

    1    1    1    0    1    0    1    1

R =

    1    0    0    1

i =

    7

codefaux =

    1    1    1    0    1    1    0    1

R =

    1    0    0    1

i =

    8

codefaux =

    1    1    1    0    1    1    1    0

R =

    1    0    0    1

>> |
```

Question 11

```
Command Window

M =
    1    1    1    0    0    0    0

code =
    1    1    0    1    1    0    0    1    1    1    0    0    0    0

codefaux =
    1    1    0    1    0    0    0    1    1    1    0    1    0    0

R =
    1    1    1    0    0    0    0

>> |
```

L'emplacement des erreurs est matérialisé en rouge ci-dessus.