

Tp3 : Communication Numérique

10 janvier

ODIER Valentin – WAKIM Marie

Question 0

Question 1

Matrice G :

$$G = \begin{pmatrix} 1 & 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 1 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 & 0 & 0 & 1 \end{pmatrix}$$

Matrice H :

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

Matrice G', ans correspondant à G'.

$$\text{ans} = \begin{pmatrix} 1 & 0 & 1 & 1 \\ 1 & 1 & 1 & 0 \\ 0 & 1 & 1 & 1 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

$$C = \begin{pmatrix} 1 \\ 0 \\ 1 \\ 0 \\ 0 \\ 0 \\ 1 \end{pmatrix}$$

Pour le calcul des bits de parité, on regarde la valeur de M :

M =

0
0
0
1

$$h_1 = 1*0 + 0*0 + 1*0 + 1*1 = 1$$

$$h_2 = 0*0 + 1*0 + 1*0 + 0*1 = 0$$

$$h_3 = 1*0 + 1*0 + 0*1 + 1*1 = 1$$

C =

1
0
1
0
0
0
1

Dans notre cas on ne peut pas mettre en évidence l'utilité de C1 et/ou C2, on relance le programme pour obtenir un cas favorable :

C1 permet d'écrire C en termes de 0 et de 1.

C2 réalise le code de Hamming sur M.

Nous constatons que C2 égal C1. On en déduit qu'il effectue les deux opérations amenant à C1 :

```
C=G'*M                                %--> Codage de M
C1=mod(C,2)
C2=encode(M,7,4,'hamming')
R0=decode(C2,7,4,'hamming')
```

$$\begin{array}{l}
 C = \\
 \begin{array}{c} 2 \\ 1 \\ 2 \\ 0 \\ 0 \\ 1 \\ 1 \end{array} \\
 \\
 C1 = \\
 \begin{array}{c} 0 \\ 1 \\ 0 \\ 0 \\ 0 \\ 1 \\ 1 \end{array} \\
 \\
 C2 = \\
 \begin{array}{c} 0 \\ 1 \\ 0 \\ 0 \\ 0 \\ 1 \\ 1 \end{array}
 \end{array}$$

Les vecteurs M et R0 sont identiques, puisque R0 est le signal décodé.

Question 2

$$\begin{array}{l}
 Z = \\
 \begin{array}{c} 2 \\ 4 \\ 2 \end{array} \\
 \\
 Z = \\
 \begin{array}{c} 0 \\ 0 \\ 0 \end{array}
 \end{array}$$

Z est composé de nombre pairs. Puis on effectue un modulo 2 sur Z, donc finalement il vaut zéro.

Ici Z est seulement composé de zéro, il n'y a pas d'erreur. Un « 1 » correspond à une erreur.

Question 3

A la ligne 20 :

```
C1(i)=mod(C1(i)+1,2);
```

Cette ligne permet d'ajouter une erreur à C1.

A chaque itération Z correspond à la détection d'erreur.

La ligne 23 est la même que la ligne 20, elle permet de corriger l'erreur.

Valeur de Zn comparé à H :

Zn =

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

H =

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

Nous constatons que Zn est égale à H.

Chaque colonne de Zn correspond au vecteur syndrome pour une erreur sur un bit. La première colonne correspondant à une erreur sur le premier bit, la deuxième sur le second...etc

Question 4

```
C=[1 1 1 1 0 1 1]'; %--> Exemple de
C_corrected=C; % detection et
Z=mod(H*C,2) % de correction
for i=1:length(C)
    test=symerr(Z,H(1:k,i));
    if (test==0)
        i
        C_corrected(i)=mod(C_corrected(i)+1,2);
    end
end
end
```

```
Zn =
    1    0    0    1    0    1    1
    0    1    0    1    1    1    0
    0    0    1    0    1    1    1

C =
    1
    1
    1
    1
    0
    1
    1

Z =
    0
    1
    1

i =
    5

C_corrected =
    1
    1
    1
    1
    1
    1
    1
```

Le programme repère une erreur pour i=5 (sur le bit 5), c'est-à-dire b2. En modifiant la ligne 34, nous obtenons bien « C corrigé ».

```
C_corrected =
    1
    1
    1
    1
    1
    1
    1

R =
    1
    1
    1
    1
```

R est le décodage de C, notre signal de départ. Il est corrigé puisque la fonction encode() corrige les erreurs.

Question 5