TP N° 2: Codage de Hamming (comment ça marche ?)

Un canal bruité, à l'époque où Richard Hamming travaillait sur un calculateur de faible fiabilité produisent des erreurs qui se manifestent par le changement d'un état «0» en un «1» ou d'un «1» en un «0» lors d'une transmission de données numériques. Il existe des méthodes de codage qui permettent de détecter la présence d'une ou plusieurs erreurs dans un mot : il s'agit des codes détecteurs d'erreurs. La méthode de codage étudiée ici permet la détection mais aussi la correction d'erreurs se produisant lors de la transmission de données. Nous verrons plus exactement :

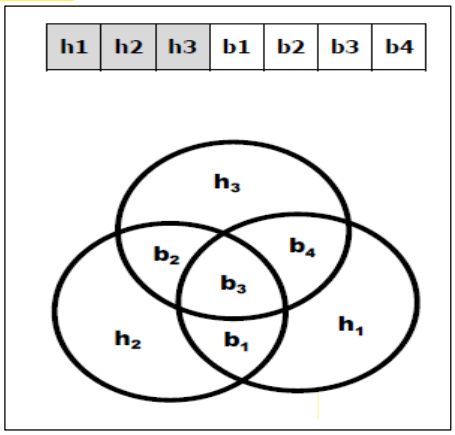
· Le code de Hamming (version (7,4)), fondé sur un contrôle de parité.

I. Codage de Hamming

D'une manière générale on associe à un message de m bits, k bits de contrôle de parité, ce qui totalise m+k bits à transmettre. Dans l'hypothèse d'une seule erreur par mot de (m+k) bits il faut distinguer (m+k) possibilités ainsi que l'absence d'erreur. Par conséquent, il faut choisir k de sorte que :

$2^k \geq m+k+1$.

Pour un message M de m=4 bits (b1,b2,b3,b4) il faut donc au moins k=3 bits de parité pour distinguer m+k=7 occurrences d'erreur et 1 absence d'erreur. Dans ce cas particulier du codage (7,4) le message codé peut être représenté dans le format ci-contre :



Chaque bit de hamming hi contrôle la parité d'un groupe de 3 bits. Le traitement de données sous MATLAB (schéma ci-contre) adopte la convention suivante :

· h1 examine la parité de (b1,b3,b4)

· h2 examine la parité de (b1,b2,b3)

· h3 examine la parité de (b2,b3,b4)

Le code associé au message M peut alors s'exprimer par une relation matricielle de la forme :

$C= G^t.M$ où :

L'opération d'addition est réalisée modulo 2, M est le vecteur message de m=4 lignes, « C » est le vecteur code de n=7 lignes, G une matrice (m*n).

II. Génération d'un code

1- Déterminer l'expression théorique de la matrice G qui génère le code C à partir du message M: $C= G^t.M$.

2- Exécuter le programme ci-contre et relever les matrices G et H dans la fenêtre de commande.

· Relever de la même manière la valeur de G' ($=G^t$).

```
clc; clf; clear;
m=4; k=3; n=m+k;

M=randint(m,1,2);          %--> Message
[H,G]=hammgen(k);

C=zeros(n,1); C1=zeros(n,1); C2=zeros(n,1);

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

Frag 1 : Codage et decodage.

· Relever la valeur de C dans la fenêtre de commande.

· Justifier ce résultat en calculant les bits de parité.

· Comparer les vecteurs C1 et C2.

· Comparer les vecteurs M et R0.

III. Syndrome et correction

Le vecteur de syndrome est facilement obtenu avec le produit matriciel Z=H.C.

3- Exécuter le programme ci-contre et relever la valeur de Z dans la fenêtre de commande. Commentaires.

```
14 -   Z=zeros(k,1); Zn=zeros(k,n);
15
16 -   Z=H*C;                  %--> Détection d'err
17 -   Z=mod(Z,2);             %--> Vecteur Syndrome
```

Frag 2 : Détection d'erreur, syndrome

```
19 - for i=1:length(C1)              %--> Simulation d'une
20 -     C1(i)=mod(C1(i)+1,2);       %    erreur sur chaque bit
21 -     Z=mod(H*C1,2);              %--> Comparer chaque Z
22 -     Zn(:,i)=Z;                  %    avec les colonnes de H
23 -     C1(i)=mod(C1(i)+1,2);
24 - end;
```

Frag 3 : Détection d'erreur, syndrome

4- Nous simulons l'occurrence d'une erreur sur chaque bit du message codé C1 ou C2 avec frag3.

· Quelle est l'opération réalisée à la ligne 20 ?

· Que représente le vecteur Z à chaque itération ?

· Exécuter frag3 et relever la valeur de la matrice Zn dans la fenêtre de commande.

· Comparer Zn et H. Que représente chaque colonne de cette matrice ?

5- Supposons l'arrivé du code C= [1 1 1 1 0 1 1] ^{t} avec une erreur évidente en b2 que nous souhaitons intercepter au moyen de frag4. Désignons le code corrigé par C_corrected.

```
27 -  C=[1 1 1 1 0 1 1]'              %--> Exemple de
28 -  C_corrected=C;                  %    detection et
29 -  Z=mod(H*C,2)                    %    de correction
30 -  for i=1:length(C)
31 -      test=symerr(Z,H(1:k,i));
32 -      if (test==0)
33 -          i
34 -          C_corrected(i)=mod(........,2);
35        end
36 -  end
37 -  C_corrected
38 -  R=decode(C,7,4,'hamming')       %--> Vérification
```

Frag 4 : Exemple d'application.

· Quelle est la valeur enregistrée par test (ligne 31) ?

· Compléter la ligne 34 et exécuter frag4. ?

· Relever les sorties C, Z, i, C_corrected et R dans la fenêtre de commande. Commentaires.