TP: Codes correcteurs d'erreurs.

Objectifs : Ce TP s'intéresse à la mise en œuvre de codes correcteurs d'erreurs.

Comme nous l'avons vu au cours des TP précédents, s'il est possible de réduire notablement le taux d'erreurs lors de la transmission numérique d'un message, ce dernier n'est en général, et malgré tous les efforts fournis, jamais nul.

Des erreurs persistent toujours dans la transmission, causées par exemple par des interférences, une altération du support de l'information, des rayonnements parasites . . ..

L'objectif de la théorie des codes correcteurs d'erreurs est de protéger l'information transmise de ces éventuelles altérations.

L'objectif de ce TP est, en complément des précédents permettant de générer des messages binaires, de les convertir en un signal numérique adapté au canal de transmission porteur de l'information, puis de les décoder de manière adaptée pour *tenter* de retrouver le message de départ, de mettre en œuvre deux types de codes correcteurs d'erreurs : un code de parité et un code de Hamming.

Ces deux codes correcteurs sont des codes linéaires en bloc.

La mise en place de ces codes devrait ainsi aboutir à une communication fiable du message sur un canal de transmission réel.

1 Découpage en blocs du message

Ecrire une fonction Matlab *bloc* permettant de découper un message de n bits en blocs de k bits et retournant le $p^{\text{ème}}$ bloc. (on pourra éventuellement utiliser la fonction Matlab *reshape*).

Remarque. Pour simplifier, on pourra n'utiliser par la suite que des messages de longueur n bits, où n est un multiple de k.

2 Code de parité

Nous allons mettre en œuvre dans cette partie un code à parité $(8, 7, 1)$. La distance minimale de ce code est 1 : il ne permet pas la correction d'une éventuelle erreur, mais simplement sa détection et la localisation du bloc contenant l'erreur.

Générer un message binaire be n bits, n étant un multiple de 7.

Découper ensuite ce message par blocs de 7 bits, puis, à chacun des blocs, ajouter le bit de parité adéquat.

On rappelle que la matrice génératrice d'un tel code est :

$$G = \left[\begin{array}{cc} I_7 & \begin{array}{c}1\\ \vdots \\ 1\end{array}\end{array}\right] = \begin{bmatrix} 1&0&0&0&0&0&0&1\\ 0&1&0&0&0&0&0&1\\ 0&0&1&0&0&0&0&1\\ 0&0&0&1&0&0&0&1\\ 0&0&0&0&1&0&0&1\\ 0&0&0&0&0&1&0&1\\ 0&0&0&0&0&0&1&1 \end{bmatrix}$$

tandis que sa matrice de contrôle est :

$$H = [1\ \ 1\ \ 1\ \ 1\ \ 1\ \ 1\ \ 1\ \ 1\ ]$$

Utiliser ensuite l'ensemble de la chaîne de transmission : codage en ligne (type polaire NRZ par exemple), transmission dans un canal réel, réception et décodage (utiliser le filtrage adapté ...).

Après avoir découpé le message reçu par bloc de 8 bits, déterminer le nombre d'erreurs et les localiser.

Eventuellement, les blocs erronés peuvent-être retransmis, jusqu'à ce qu'il ne subsiste plus aucune erreur ... du moins détectable par un code de parité.

3 Code de Hamming $(7, 4, 3)$

Au contraire du code précédent, un code de Hamming permet la détection <u>et</u> la localisation, donc la correction, d'une erreur par bloc.

Il n'est pas donc pas toujours nécessaire d'emettre une requête de réenvoie lors de la réception d'un bloc erroné.

Matrice de contrôle H. On peut former la matrice de contrôle H en écrivant les entiers de 1 à 7 sous forme binaire dans chaque colonne de H :

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

Matrice génératrice G. La matrice génératrice associée à la matrice de contrôle H est alors la matrice G donnée ci-dessous (le vérifier !) :

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

Remarque. La matrice génératrice G n'est pas sous sa forme systématique. La méthode du pivot de Gauss permet, si on le souhaite, d'y remédier (attention, la matrice de contrôle H doit aussi être modifiée dans ce cas).

La commande matlab *rref* pemret d'effectuer cette conversion.

1) **Distance et pouvoir correcteur du code.** Expliquer pourquoi la distance de ce code est de 3. Combien d'erreurs permet-il alors de corriger ?

2) **Codage d'un message.** Générer un message binaire de N bits et l'encoder suivant le code de Hamming.

   Chaque bloc d de 4 bits (utiliser la fonction de la $1^{\text{ère}}$ partie) est codé en un bloc de 7 bits c à l'aide de la matrice génératrice G du code de Hamming suivant le produit matriciel (attention toutes les opérations se font sur des nombres binaires, voir la fonction Matlab *mod*) :

   $$c = d\,G$$

   On peut remarquer que les bits de redondances sont en position 1,2 et 4 dans ce bloc. En d'autres termes, le bloc original est donné par les $3^{\text{ème}}$, $5^{\text{ème}}$,$6^{\text{ème}}$ et $7^{\text{ème}}$ bits du bloc codé.

3) **Bruitage du message.** On pourra dans un premier temps, simplement créer un vecteur d'erreurs apparaissant avec une probabilité donnée : il s'agit d'un vecteur binaire dont la proportion de 1 est la probabilité d'erreur choisie.

   Générer un tel vecteur.

   Le message transmis et bruité s'obtient alors en ajoutant (toujours l'addition binaire ...) le message codé au bruit.

4) **Détection et correction des erreurs.** La détection d'une erreur se fait par bloc (de 7 bits maintenant), et est basée sur le calcul du syndrôme.
   Pour chaque bloc b de 7 bits, on calcule le syndrôme s à l'aide la matrice de contrôle H :

   $$s = H\,b^T$$

   a) Si le syndrôme est nul, il n'y a pas d'erreur dans le bloc (ou éventuellement plus de deux erreurs)

   b) Si le syndrôme s est non nul, alors s est une colonne de la matrice H. Le numéro de la colonne donne la position de l'erreur.

      L'intérêt tout particulier de la forme de la matrice de contrôle H, justement sous forme non systématique, est que la $j^{\text{ème}}$ colonne de H est l'expression du nombre j en base 2.
      Ainsi, le syndrôme s est exactement la position de l'erreur exprimé en base 2.

Pour chaque bloc, calculer le syndrôme associé, et le corriger si besoin est.

5) **Décodage du message.** Reconstruire alors le message complet, en ne sélectionnant que les bits d'information (oter les bits de redondance), et comparer le taux d'erreurs introduit de la question 3), au taux d'erreurs effectif (celui introduit en 3)).

6) **Mise en œuvre du code de Hamming pour la transmission numérique dans un canal réel.**

   Reprendre l'étude précédente complète du code de Hamming, en utlisant cette fois un canal réel de transmission :

   a) génération d'un message binaire,

   b) encodage suivant le code de Hamming 2),

   c) codage en ligne du message (en polaire NRZ par exemple),

   d) transmission dans un canal à caratéristiques réelles (gain, puissance du bruit, et bande passante)

   e) réception et filtrage (adapté !) du signal

   f) échantillonnage, et décision sur le signal reçu

   g) correction d'éventuelles erreurs par bloc 4),

   h) décodage du code de Hamming 5).