

**TD 6****Codes en blocs systématiques**

Objectif du TD : On se propose dans ce TD d'étudier des codes en blocs systématiques. La forme systématique est très utilisée en pratique puisqu'elle permet de récupérer les bits d'information directement après la correction d'erreur sur le mot reçu sans inversion de l'opération de codage.

EXERCICE 1

Soit le code systématique C défini par les équations de parité suivantes :

$$c_1 = d_2 + d_3 + d_4$$

$$c_2 = d_1 + d_2 + d_3$$

$$c_3 = d_1 + d_2 + d_4$$

$$c_4 = d_1 + d_3 + d_4$$

où $[d_1, d_2, d_3, d_4]$ sont les bits d'informations et $[c_1, c_2, c_3, c_4]$ sont les bits de parité (bits de redondance) d'un mot de code. Un mot de code s'écrit donc $[d_1, d_2, d_3, d_4, c_1, c_2, c_3, c_4]$.

1. Déterminer les paramètres n , longueur du mot de code, k , longueur du mot d'information et le rendement de C .
2. Lister tous les mots de code de C et déterminer la distance minimale d_{min} du code.
3. Combien d'erreurs le code peut détecter et combien d'erreurs il peut corriger ?
4. Donner une matrice génératrice G de C sous forme systématique.
5. Déterminer la matrice de parité H du code C . Quel est le code dual du code C ?
6. On décide de construire un nouveau code C' en ne gardant que les mots de code de poids 4. Le code C' est-il un code linéaire ?

EXERCICE 2

Un code de parité consiste à rajouter un bit de redondance au mot d'information de telle façon de la somme XOR des bits du mot de code soit toujours nulle.

$$\text{Exemple : } d = [0\ 1\ 0\ 1\ 1] \mapsto c = [0\ 1\ 0\ 1\ 1\ 1] \text{ vérifiant } \sum c_i = 0.$$

Il est aussi envisageable de rajouter un bit de redondance de telle façon que $\sum c_i = 1$, on parle alors de parité impaire.

$$\text{Exemple : } d = [0\ 1\ 0\ 1\ 1] \mapsto c = [0\ 1\ 0\ 1\ 1\ 0\ 1] \text{ vérifiant } \sum c_i = 1.$$

1. Lister tous les mots du code de parité $C(3, 2)$ et donner sa d_{min} .
2. Lister tous les mots de code du code $C(3, 2)$ avec une parité impaire et donner sa d_{min} .
3. Lequel des deux codes vous choisiriez ?