CORRECTION D'ERREUR EN MQ.

Dans la théorie classique la façon usuelle de compenser les effets des perturbations (du bruit) dans le stockage ou la communication des données est d'introduire de la redondance de façon suffisamment structurée. C'est ce que l'on appelle la correction d'erreurs. La correction d'erreur est efficace dans la mesure où l'information est digitale. En effet pour des signaux analogues les perturbations dues au bruit sont continues et non pas discrètes ce qui rend la correction d'erreur quasiment impossible.

On pourrait naïvement croire que dans le cas quantique la correction d'erreur est aussi impossible (ou néanmoins difficile) car les états de l'espace d'Hilbert forment un continuum. Parfois des arguments ont été avancés dans ce sens, mais il s'est avéré (suite aux travaux de Shor) qu'il n'en est rien. La nature discrète de la MQ se manifeste dans la classification des perturbations possibles et on peut encore construire des codes correcteurs d'erreurs.

1. Bref rappel sur les codes linéaires classiques.

Le code correcteur le plus simple que l'on puisse imaginer est le code de répétition. Supposons que l'on veuille transmettre un bit 0 ou 1 à travers un canal BSC qui renverse le bit avec probabilité p. On pourrait utiliser le code de répétition

$$0 \to 000$$

$$1 \to 111$$

Par aucun ou un seul renversement on peut facilement détecter puis corriger l'erreur grâce à la règle de la majorité. Par exemple si 010 est reçu on en conclu que très probablement le deuxième bit a été renversé et on décode en faisant l'opération $010 \to 000$. En faisant peut-être que l'on fait une "erreur de décodage" car il pourrait qu'en fait 111 était effectivement transmis et que les premiers et troisièmes bit aient été renversés par le canal. Mais la probabilité de cet événement est très faible si p est assez petit.

Sans répétition la probabilité d'erreur de décodage sera toujours égale à p alors qu'avec répétition elle est égale à $p^3 + 3p^2(1-p) = 3p^2 - 2p^3$ et est inférieure à p pour tout $0 < p < \frac{1}{2}$.

binaire

De façon générale un code linéaire est obtenu par une application linéaire

$$\mathbb{F}_2^K \longrightarrow \mathbb{F}_2^n$$

$$\vec{u} \longmapsto G\vec{u} = \vec{x}$$

où $\vec{u} = \begin{pmatrix} u_1 \\ \vdots \\ u_K \end{pmatrix}$ un vecteur colonne contenant K bits d'information et G une matrice (d'éléments 0 ou 1) $n \times K$. Le vecteur $G\vec{u}$ est le mot du code, vecteur binaire à n composantes. Toutes les opérations sont faites mod 2. Pour que l'image de $\mathbb{F}_2^K$ dans $\mathbb{F}_2^n$ soit un sous-espace vectoriel de dimension K on prend une matrice G de rang K.

C'est à dire que les K colonnes de G sont linéairement indépendantes. Le code de répétition correspond à la matrice $G = \begin{bmatrix} 1 \\ \vdots \\ 1 \end{bmatrix}$ et $K = 1$. Un autre exemple correspond à

$$G = \begin{bmatrix} 1 & 0 \\ 1 & 0 \\ 1 & 0 \\ 0 & 1 \\ 0 & 1 \\ 0 & 1 \end{bmatrix}.$$ Les mots de code sont

$$G\begin{bmatrix} 0 \\ 0 \end{bmatrix} = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix} \quad G\begin{bmatrix} 0 \\ 1 \end{bmatrix} = \begin{pmatrix} 0 \\ 0 \\ 0 \\ \vdots \\ 1 \end{pmatrix} \quad G\begin{bmatrix} 1 \\ 0 \end{bmatrix} = \begin{pmatrix} 1 \\ \vdots \\ 1 \\ 0 \\ 0 \\ 0 \end{pmatrix}$$

et $G\begin{bmatrix} 1 \\ 1 \end{bmatrix} = \begin{pmatrix} \vdots \\ \vdots \\ \vdots \end{pmatrix}$.

G s'appelle la matrice génératrice, n la longueur du code et k sa dimension (la dimension du s-esp vect dans $\mathbb{F}_2^n$). On dit que l'on a un code (n, k). Le nombre de mots du code est 2^k et son "rendement" $\frac{k}{n}$.

Une autre représentation utile du code est en terme de sa matrice de parité (celle-ci n'est pas unique). Puisque un code linéaire est un sous espace vectoriel de $\mathbb{F}_2^n$, il peut être vu comme le noyau d'une matrice H, c'est à dire l'ensemble des vecteurs solutions de

$$H\vec{x} = 0 \quad , \; H \text{ de dim } (n-k) \times n$$

et H est dual de G c.-à-d $HG = 0$.

Pour construire H à partir de G on peut écrire $G = [\vec{g}_1, \dots \vec{g}_k]$ où $\vec{g}_1 \dots \vec{g}_k$ sont indépendants et prendre $n-k$ vecteurs orthogonaux $\vec{h}_1 \dots \vec{h}_{n-k}$ au s-esp engendré par $\vec{g}_1 \dots \vec{g}_k$. Alors

$$H = \begin{bmatrix} \vec{h}_1^{\,T} \\ \vdots \\ \vec{h}_{n-k}^{\,T} \end{bmatrix}.$$

Ici $\text{rang } H = n-k$ et donc $\dim(\ker H) = n - \text{rang } H = k$.

Réciproquement pour construire G à partir de H on prend les

$n-k$ vecteurs lignes indépendants de H et on construit k vecteurs orthogonaux (ou s. esp engendré par les lignes de H) qui forment les k colonnes de G.

La matrice de parité permet de détecter certaines erreurs (mais pas toutes). Supposons que le mot $\vec{x}$ soit transmis à travers un canal et $\vec{x}'$ reçu : $\vec{x}' = \vec{x} + \vec{e}$ où $\vec{e}$ est un vecteur contenant des 1 pour les bits erronés et 0 ailleurs. On a :

$$H\vec{x}' = H\vec{x} + H\vec{e} = H\vec{e}$$

On appelle $H\vec{e}$ le syndrome. Si celui-ci est nul il indique que le mot reçu est mauvais. Pour le canal BSC la probabilité d'avoir un vecteur d'erreur $\vec{e}$ est $p^{|\vec{e}|}(1-p)^{n-|\vec{e}|}$ et il est naturel (pour p petit) de décoder $\vec{x}'$ en déclarant que $\vec{x}$ est le vecteur qui minimise $d(\vec{x}, \vec{x}')$ (égal à $|\vec{e}|$) la distance de Hamming entre $\vec{x}$ et $\vec{x}'$. En effet $\vec{x}$ est le vecteur (transmis) le plus probable (qui maximise $p^{|\vec{e}|}(1-p)^{n-|\vec{e}|}$) étant donné que $\vec{x}'$ a été reçu (les vecteurs transmis sont choisis uniformément). C'est le décodeur dit de "distance minimale".

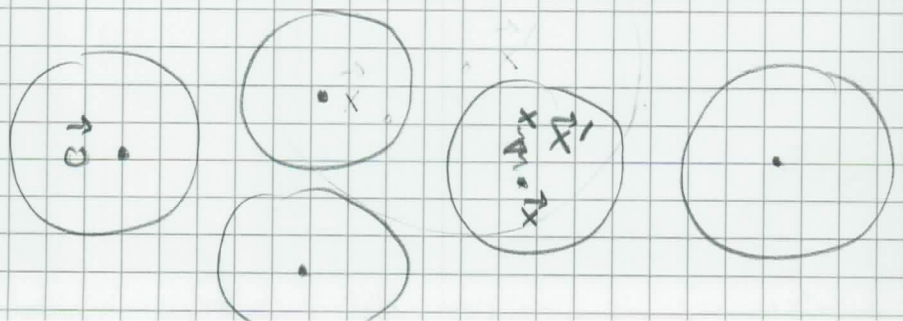
Un paramètre important qui caractérise la qualité d'un code est sa distance minimale $d = \min_{x, x' \in \mathcal{C}} d(\vec{x}, \vec{x}')$.

Pour un code linéaire on a aussi $d = \min_{\substack{\vec{x} \neq \vec{0} \\ \vec{x} \in \mathcal{C}}} d(\vec{x}, \vec{0})$.

On parle alors de code (n, k, d) : longueur n, dimension k et distance minimale d. Le théorème suivant est important (et facile à prouver)

Théorème : Soit un code (n, k, d). On peut toujours corriger jusqu'à t erreurs avec $2t+1 \leq d$ grâce au décodeur de la distance minimale.

Si on considère les boules de Hamming de rayon $\frac{d-1}{2}$ autour des mots de code, celles-ci ne s'intersectent pas. De plus si le nombre d'erreur est $\leq \frac{d-1}{2}$ alors le mot reçu $\vec{x}'$ est contenu dans une et une seule des boules. On déclare que le mot transmis est l'unique mot de code dans la même boule que $\vec{x}'$.



Une classe de codes importante sont les codes de Hamming. Ceux-ci sont communément décrits par leur matrice de parité. Soit $r \geq 2$ et H la matrice dont les colonnes sont tous les $2^r - 1$ vecteurs binaires non nuls de longueur r. On a $n = 2^r - 1$ et $n - k = r \Rightarrow k = 2^r - r - 1$. On obtient donc un code $(n, k) = (2^r - 1, 2^r - r - 1)$ de longueur $2^r - 1$ et dimension $2^r - r - 1$. Il faut vérifier que le rang de cette matrice est bien r: cela est vrai car parmi les $2^r - 1$ vecteurs binaires non nuls il y en a r indépendants qui engendrent tout $\mathbb{F}_2^r$.

On peut montrer que pour tout code linéaire, le nombre minimale de colonnes de H qui sont linéairement dépendantes donne exactement d (c.a.d : toute collection de $d-1$ colonnes sont lin indép et une collection de d colonnes est lin dép). Pour les codes de Hamming on voit facilement que $d = 3$. Ainsi ces codes corrigent une erreur. Par exemple le code de Hamming $(7, 4, 3)$ possède la matrice (ici $r = 3$)

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

Ici nous avons ordonné les colonnes dans l'ordre lexicographique car cela permet une correction d'erreur aisée. Supposons qu'une erreur soit produite par le 3ième bit. Alors $\vec{e} = \begin{pmatrix} 0 \\ 0 \\ 1 \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix}$ et le syndrome vaut $H\vec{e} = \begin{pmatrix} 0 \\ 1 \\ 1 \end{pmatrix}$ qui est égal à la 3ème colonne de H. Le syndrome nous montre automatiquement quel bit est erroné et il suffit de le renverser pour effectuer la correction d'erreur. Bien entendu que l'erreur est correctement détectée et corrigée seulement si elle est unique.

2) Codes de Répétition Quantique -

Par analogie avec le cas classique on peut considérer le canal "bit flip" et le code de répétition

$$|0\rangle \rightarrow |000\rangle$$

$$|1\rangle \rightarrow |111\rangle$$

Quel est le procédé de détection et correction dans le cas quantique? Il faut choisir correctement la base de mesure pour ne pas détruire immédiatement l'information contenue dans canal! Ici l'espace d'Hilbert à la sortie du canal est de dimension $2^3 = 8$ et il faut une base à 8 dimensions. Considérons les 4 projecteurs de dimension 2 chacun (8 dim au total)

$$P_0 = |000\rangle\langle 000| + |111\rangle\langle 111|$$

$$P_1 = |100\rangle\langle 100| + |011\rangle\langle 011|$$

$$P_2 = |010\rangle\langle 010| + |101\rangle\langle 101|$$

$$P_3 = |001\rangle\langle 001| + |110\rangle\langle 110|$$

P_0 projette dans le s-espace à 0 erreurs, P_1 dans le s-esp à une erreur, sur le premier bit, P_2 dans le s-esp à une erreur dans le

deuxième bit, P_3 de la même à une erreur sur le 3 rème bit.

L'entrée du canal est le ket $\alpha|0\rangle + \beta|1\rangle$ codé :

$$\alpha|000\rangle + \beta|111\rangle$$

Si une seule erreur est produite, mettons pour le deuxième bit (avec prob p) le ket est

$$\alpha|010\rangle + \beta|101\rangle$$

Une mesure dans la base $\{P_0, P_1, P_2, P_3\}$ <u>préserve l'état avec probabilité 1</u> car

$$P_2(\alpha|010\rangle + \beta|101\rangle) = \alpha|010\rangle + \beta|101\rangle.$$

et ainsi on détecte que l'erreur est dans le deuxième bit sans détruire l'état. La correction d'erreur se fait en appliquant l'opérateur unitaire $\mathbb{1} \otimes X \otimes \mathbb{1}$ qui renverse le deuxième bit.

$$\mathbb{1} \otimes X \otimes \mathbb{1}(\alpha|010\rangle + \beta|101\rangle) = \alpha|000\rangle + \beta|111\rangle.$$

Bien sûr (comme dans le cas classique) on ne détecte pas

correctement des renversements de deux ou trois bits.

Un autre type d'erreur qui n'est pas détecté est le "phase flip"

$$\alpha|000\rangle + \beta|111\rangle \rightarrow \alpha|000\rangle - \beta|111\rangle$$

ou plus généralement des perturbations des coefficients α et β.

Nous verrons plus loin comment remédier à ce problème.

Il est utile de décrire le procédé de correction d'erreur d'un autre point de vue. Considérons un appareil de mesure qui mesure les observables

$$Z_1 \otimes Z_2 \otimes \mathbb{1} \quad \text{et} \quad \mathbb{1} \otimes Z_2 \otimes Z_3 .$$

Notons que la mesure simultanée de ces observables (avec un seul appareil de mesure) est possible car elles commutent. La mesure de chaque observable donne les résultats (± 1) et (± 1).

Ces mesures constituent l'analogue du "syndrome classique":

$+1, +1 \rightarrow$ pas d'erreur

$-1, +1 \rightarrow$ erreur dans le 1^{er} bit

$-1, -1 \rightarrow$ erreur dans le $2^{\text{ème}}$ bit

$+1, -1 \rightarrow$ erreur dans le $3^{\text{ème}}$ bit.

Une fois l'erreur détectée on la corrige en appliquant un opérateur unitaire :

par d'erreur $\rightarrow$ $\mathbb{1} \otimes \mathbb{1} \otimes \mathbb{1}$.

1er bit $\rightarrow$ $X \otimes \mathbb{1} \otimes \mathbb{1}$.

2e bit $\rightarrow$ $\mathbb{1} \otimes X \otimes \mathbb{1}$.

3e bit $\rightarrow$ $\mathbb{1} \otimes \mathbb{1} \otimes X$.

Toutes les opérations - encodage - transmission - mesure - correction - sont permises par le MQ (ops unitaires ou mesures projectives)

Exercice : proposez un circuit pour réaliser l'opération d'encodage. Remarquez notamment qu'il n'y a pas de contradiction avec le "No Cloning Thm".

Nous avons vu que ce code n'est pas efficace pour corriger les erreurs produites par un canal "phase-flip". Un code de répétition utile pour le canal "phase-flip" (mais inutile pour le bit-flip) est obtenu en travaillant dans la base $|+\rangle = \frac{|0\rangle + |1\rangle}{\sqrt{2}}$; $|-\rangle = \frac{|0\rangle - |1\rangle}{\sqrt{2}}$.

Encodage :

$|0\rangle \to |+\rangle \to |+++\rangle$

$|1\rangle \to |-\rangle \to |---\rangle$.

Détection d'erreur :

Mesurer les observables (qui commutent)

$X_1 \otimes X_2 \otimes \mathbb{1}$ et $\mathbb{1} \otimes X_2 \otimes X_3$.

Correction d'erreur :

Appliquez les opérateurs unitaire :

$\mathbb{1} \otimes \mathbb{1} \otimes \mathbb{1}$ (pas d'erreur)

$Z_1 \otimes \mathbb{1} \otimes \mathbb{1}$ (phase-flip du premier qubit)

$\mathbb{1} \otimes Z_2 \otimes \mathbb{1}$ (phase-flip 2ième qubit).

$\mathbb{1} \otimes \mathbb{1} \otimes Z_3$ (phase-flip 3ième qubit).

Par exemple si on transmet $|0\rangle$, on code par $|+++\rangle$ et si le canal produit $|++-\rangle$ on mesurera $(+1, -1)$ ce qui indique que le 3ème bit a subi un phase flip. On applique

$\mathbb{1} \otimes \mathbb{1} \otimes Z_3 \, |++-\rangle = |+++\rangle \longrightarrow |0\rangle$.

3). Le code de Shor.

La situation du paragraphe précédent n'est pas encore satisfaisante car en MQ les deux types d'erreurs peuvent essentiellement survenir en même temps et nous aimerions un procédé qui les corrige toutes. La solution de ce problème a été présentée pour la première fois par Shor et elle se révèle être à la base des autres constructions plus élaborées qui ont suivies.

Si nous pouvons corriger à la fois les bits-flips et les phases-flips, alors nous pouvons corriger du même coup une très large classe d'erreurs à 1 qubit. En effet nous pouvons corriger toutes les erreurs produites par des canaux du type

$$\mathcal{E}(\rho) = p_0 \rho + p_1 X \rho X + p_2 Y \rho Y + p_3 Z \rho Z$$

avec $p_0 + p_1 + p_2 + p_3 = 1$. Cette classe contient les canaux bit-flip, phase-flip, bit-phase-flip, dépolarisant cet...

En fait on peut aussi montrer que le code de Shor corrige n'importe quelle type d'erreur à 1 qubit.

L'idée principale du code de Shor est de "concaténer" les deux codes de répétition. La concaténation est en fait une méthode de la théorie classique du codage qui permet de construire des nouveaux codes en assemblant des codes élémentaires.

Les mots de code sont construits comme suit :

$$(|0\rangle \to |+\rangle) \to |+++\rangle = \frac{|0\rangle + |1\rangle}{\sqrt{2}} \otimes \frac{|0\rangle + |1\rangle}{\sqrt{2}} \otimes \frac{|0\rangle + |1\rangle}{\sqrt{2}}$$

$$\to \frac{|000\rangle + |111\rangle}{\sqrt{2}} \otimes \frac{|000\rangle + |111\rangle}{\sqrt{2}} \otimes \frac{|000\rangle + |111\rangle}{\sqrt{2}}.$$

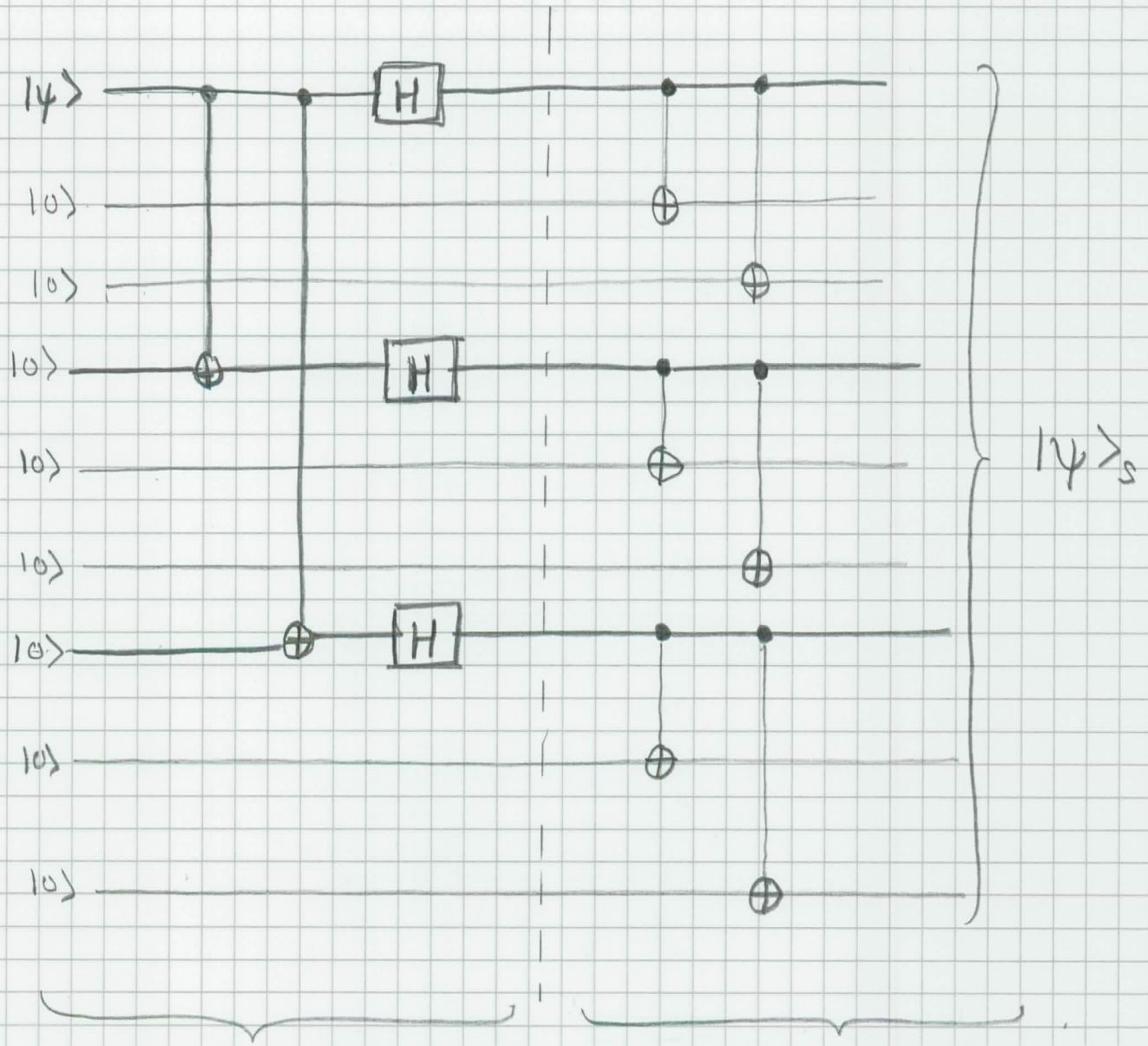
$$(|1\rangle \to |-\rangle) \to |---\rangle = \frac{|0\rangle - |1\rangle}{\sqrt{2}} \otimes \frac{|0\rangle - |1\rangle}{\sqrt{2}} \otimes \frac{|0\rangle - |1\rangle}{\sqrt{2}}$$

$$\to \frac{|000\rangle - |111\rangle}{\sqrt{2}} \otimes \frac{|000\rangle - |111\rangle}{\sqrt{2}} \otimes \frac{|000\rangle - |111\rangle}{\sqrt{2}}.$$

Ainsi chaque qubit est encodé par un état à 9 qubits (le "rendement" de ce code est $1/9$). Si nous appelons $|0\rangle_S$ et $|1\rangle_S$ les deux états à 9 qubits, un mot de code général est

$$|\psi\rangle = \alpha|0\rangle + \beta|1\rangle \longrightarrow |\psi_S\rangle = \alpha|0\rangle_S + \beta|1\rangle_S.$$

Le circuit suivant réalise l'opérateur de codage de façon unitaire



code de répétition protégeant les erreurs "phase-flip"

code de répétition protégeant les erreurs "bit-flip".

Montrons maintenant que ce code est capable de protéger l'état contre n'importe quelle erreur sur 1 qu-bit engendrée par les opérateurs $\mathbb{1}$ (pas d'erreur), X (bit-flip), Z (phase-flip) et Y (bit & phase flip).

En d'autre terme nous voulons montrer que si la sortie du canal est $|\psi_5\rangle$, $X_1|\psi_5\rangle$, $X_2|\psi_5\rangle$, $Z_1|\psi_5\rangle$ agit sur 1 des 9 qu-bits, il est possible de détecter l'erreur et de reconstruire $|\psi\rangle_5$ (et donc $|\psi\rangle$) grâce à des opérations permises de MQ.

Considérons d'abord l'code de X tout seul (bit-flip).

Comme avant, la mesure simultanée des observables Z_1Z_2 et Z_2Z_3 permet de décider si un des trois premiers qu-bit est renversé ou non. Ensuite il est facile de le corriger en appliquant X_i (si c'est $i=1,2,3$ qui est renversé). Pour pouvoir traiter tous les qu-bits de cette façon il faut faire une mesure des opérateurs suivants

$$Z_1Z_2 \text{ et } Z_2Z_3 \;;\; Z_4Z_5 \text{ et } Z_5Z_6 \;;\; Z_7Z_8 \text{ et } Z_8Z_9$$

Tous ces opérateurs commutent et la mesure simultanée est possible.

Considérons maintenant l'code de Z tout seul (phase flip).

Un phase flip sur le deuxième qu-bit change les états de base en (l'état à la sortie du canal est $Z_2|\psi\rangle_5$ avec prob p).

$$\frac{|000\rangle - |111\rangle}{\sqrt{2}} \otimes \frac{|000\rangle + |111\rangle}{\sqrt{2}} \otimes \frac{|000\rangle + |111\rangle}{\sqrt{2}}$$

et

$$\frac{|000\rangle + |111\rangle}{\sqrt{2}} \otimes \frac{|000\rangle - |111\rangle}{\sqrt{2}} \otimes \frac{|000\rangle - |111\rangle}{\sqrt{2}}$$

Cette erreur peut être détectée (sans détruire les états) grâce à une mesure de ($X_{1,2}$ qui commutent entre eux).

$$(X_1 X_2 X_3)(X_4 X_5 X_6) \quad \text{et} \quad (X_4 X_5 X_6)(X_7 X_8 X_9)$$

Par exemple, pour l'erreur ci-dessus

$$(X_1 X_2 X_3)(X_4 X_5 X_6)(Z_2|\psi_5\rangle) = -(Z_2|\psi_5\rangle).$$

et

$$(X_4 X_5 X_6)(X_7 X_8 X_9)(Z_2|\psi_5\rangle) = +(Z_2|\psi_5\rangle)$$

On conclut que l'erreur phase-flip est dans un des trois premiers qu-bits. Notez que l'on ne peut pas affirmer lequel des trois est erroné. Mais cela ne nous empêche pas d'effectuer la correction en appliquant l'opération $Z_1 Z_2 Z_3$ (vérifiez !)

$$Z_1 Z_2 Z_3 (Z_2|\psi_5\rangle) = |\psi_5\rangle.$$

Illustrons maintenant la correction d'une erreur bit & phase flip sur le $4^{\text{ème}}$ q-bit. L'état sortant du canal est

$$X_4 Z_4 |\psi_5\rangle.$$

D'abord on détecte les bit-flip comme suit :

$$Z_1 Z_2 \left(X_4 Z_4 |\psi_5\rangle\right) = \left(X_4 Z_4 |\psi_5\rangle\right)$$

$$Z_2 Z_3 \left(X_4 Z_4 |\psi_5\rangle\right) = \left(X_4 Z_4 |\psi_5\rangle\right)$$

$$Z_4 Z_5 \left(X_4 Z_4 |\psi_5\rangle\right) = -\left(X_4 Z_4 |\psi_5\rangle\right) \quad (\text{car } Z_4 X_4 = -X_4 Z_4)$$

$$Z_5 Z_6 \left(X_4 Z_4 |\psi_5\rangle\right) = \left(X_4 Z_4 |\psi_5\rangle\right)$$

$$Z_7 Z_8 \left(X_4 Z_4 |\psi_5\rangle\right) = \left(X_4 Z_4 |\psi_5\rangle\right)$$

$$Z_8 Z_9 \left(X_4 Z_4 |\psi_5\rangle\right) = \left(X_4 Z_4 |\psi_5\rangle\right)$$

On conclut qu'il y a un bit flip sur le $4^{\text{ème}}$ bit. Ensuite on détecte les phase flips comme suit

$$X_1 X_2 X_3 X_4 X_5 X_6 \left(X_4 Z_4 |\psi_5\rangle\right) = -\left(X_4 Z_4 |\psi_5\rangle\right)$$

(car X_4 et Z_4 anticommutent)

$$X_4 X_5 X_6 X_7 X_8 X_9 \left(X_4 Z_4 |\psi_5\rangle\right) = -\left(X_4 Z_4 |\psi_5\rangle\right).$$

et on conclut qu'il y a aussi un phase-flip sur le quatrième q-bit. L'état est corrigé en appliquant $(X_4 Z_4)$ sur la sortie du canal : $X_4 Z_4 \left(X_4 Z_4 |\psi_5\rangle\right) = |\psi_5\rangle$.

Correction d'erreurs à 1 qubit (générales.)

Les résultats obtenus jusqu'ici peuvent être obtenus aussi d'une façon plus unifiée qui montre leur généralité. Considérons la matrice densité décrivant la sortie d'un canal quelconque :

$$|\psi_s\rangle\langle\psi_s| \longrightarrow \mathcal{E}(|\psi_s\rangle\langle\psi_s|).$$

Une mesure des observables Z_1Z_2, Z_2Z_3, Z_4Z_5, Z_5Z_6, Z_7Z_8, Z_8Z_9 et $X_1X_2X_3X_4X_5X_6$, $X_4X_5X_6X_7X_8X_9$ va projeter l'état $\mathcal{E}(|\psi_s\rangle\langle\psi_s|)$ sur un des sous espaces propres de ces observables. Les valeurs propres sont toutes égales à ± 1. Si le bruit est faible et agit essentiellement sur un qubit au plus le syndrome sera avec grande probabilité du type discuté dans les paragraphe précédent, c.a.d correspondant à un bit-flip, phase-flip ou bit&phase flip. On peut l'identifier puis corriger l'erreur. Il existe une faible probabilité (si le bruit est faible) que l'erreur affecte plus d'un qubit (comme dans le cas classique) et alors la correction d'erreur est erronée.

En fin de compte le code de Shor est de longueur 9. Sa dimension est 2 et il corrige 1 erreur. Plus précisément les mots de code sont du type $\alpha|0\rangle_S + \beta|1\rangle_S$ donc vivent dans un sous espace de Hilbert de dimension 2^1, de l'espace à 9 qu-bits qui est lui de dimension 2^9. On dit que les paramètres de ce code quantique sont $[9, 1, 1]$ (longueur 9 et $\dim \mathcal{H} = 2^9$; $\dim \mathcal{C} = 2^1$; corrige 1 erreur).

4) Codes de Calderbank - Shor - Steane.

Le code de Shor corrige seulement 1 erreur (comme les codes de Hamming). Nous donnons maintenant la construction d'une large classe de codes capable de corriger t erreurs de type bit et phase flips. Nous verrons que pour une longueur n donnée, il est possible de trouver des codes de dimension k et corrigeant t erreurs t.q $\frac{k}{n}$ et $\frac{t}{n}$ soient $O(1)$.

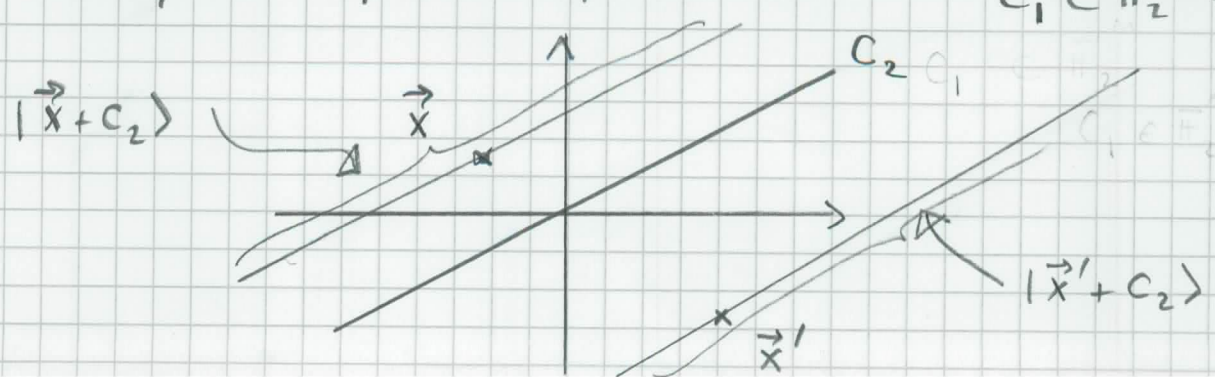
Soit C_1 et C_2 deux codes linéaires classiques t.q $C_2 \subset C_1$ de paramètres (n, k_1) et (n, k_2). On suppose aussi que C_1 et $C_2^{\perp}$ (le dual de C_2 qui est donné par tous

les mots de code $\underline{1}$ aux mots de C_2) corrigent t erreurs (p. ex leur distance minimale est $d = 2t+1$). Nous rappelons que C_1 est un s. esp vect de $\mathbb{F}_2^n$ de dimension K_1 et C_2 un s. esp vect de C_1, de dimension $K_2 \leq K_1$. Le code dual $C_2^\perp$ est le s. esp vect orthogonal à C_2 dans $\mathbb{F}_2^n$ et sa dimension est donc $n - K_2$.

En fait C_2 est aussi un s-groupe de C_1 et on peut considérer l'ensemble des classes d'équivalence C_1/C_2. Soit $\vec{x}$ un mot de C_1, la classe d'équivalence de $\vec{x}$ est l'ensemble des mots de la forme $\{\vec{x} + \vec{y} \,;\, y \in C_2\}$. C'est donc l'hyperplan // à C_2 passant par $\vec{x} \in C_1$. Posons

$$|\vec{x} + C_2\rangle = \frac{1}{\sqrt{|C_2|}} \sum_{\vec{y} \in C_2} |\vec{x} + \vec{y}\rangle$$

Ce Ket ne dépend que de la classe d'équivalence associée à $\vec{x}$ et non pas du représentant spécial choisi. $C_1 \subset \mathbb{F}_2^n$.



Le ket $|\vec{x} + C_2\rangle$ est la superposition cohérente de tous les états dans la classe d'équivalence $\vec{x} + C_2$. Pour deux classes d'équivalences différentes $\vec{x} + C_2$ et $\vec{x}' + C_2$ (c.a.d que $\vec{x} - \vec{x}' \notin C_2$) les kets associés sont perpendiculaires :

$$(\vec{x} + C_2 \mid \vec{x}' + C_2) = 0$$

Il y a $\frac{|C_1|}{|C_2|} = 2^{k_1 - k_2}$ classes d'équivalences et donc $2^{k_1 - k_2}$ vecteurs orthogonaux.

Le code $CSS(C_1, C_2)$ est le sous-espace d'Hilbert engendré par ses $2^{k_1 - k_2}$ vecteurs orthogonaux. C'est un sous-espace de l'espace d'Hilbert à n-qubits $\underbrace{\mathbb{C}^2 \otimes \mathbb{C}^2 \otimes \cdots \otimes \mathbb{C}^2}_{n \text{ fois}}$.

lequel est de dimension 2^n. Les paramètres du code sont $[n, k_1 - k_2]$; longueur n, $\dim \mathcal{H} = n$ et $\dim \mathcal{C} = 2^{k_1 - k_2}$.

Montrons maintenant que si C_1 et $C_2^\perp$ corrigent t erreurs alors $CSS(C_1, C_2)$ corrige t bit et phase flips.

Soit $\vec{e}_1$ et $\vec{e}_2$ les vecteurs possédant des 1 et 0 avec la coordonnée 1 à l'endroit du bit et phase flip.

Le vecteur corrompu par le bruit est :

$$|\psi\rangle = \frac{1}{\sqrt{|C_2|}} \sum_{\vec{y} \in C_2} (-1)^{(\vec{x}+\vec{y}).\vec{e}_2} |\vec{x} + \vec{y} + \vec{e}_1\rangle$$

Pour effectuer la correction d'erreur il faut faire des opérations permises par la HQ qui reconstitue $|\vec{x} + C_2\rangle$. Comme il faudra calculer et stocker les "syndromes" du bit et phase flip on ajoute des bits auxiliaires de stockage et on travaille avec l'état

$$|\psi\rangle \otimes |0_m\rangle \otimes |0_m\rangle = \frac{1}{\sqrt{|C_2|}} \sum_{\vec{y} \in C_2} (-1)^{(\vec{x}+\vec{y}).\vec{e}_2} |\vec{x} + \vec{y} + \vec{e}_1\rangle \otimes |0_m\rangle \otimes |0_m\rangle$$

Correction du bit-flip

On calcule les syndromes $H_1(\vec{x} + \vec{y} + \vec{e}_1) = H_1 \vec{e}_1$ (grâce à la matrice de parité du code C_1). Ce calcul peut être implémenté de façon unitaire :

$$U_1 |\vec{x} + \vec{y} + \vec{e}_1\rangle \otimes |0_m\rangle \otimes |0_m\rangle = |\vec{x} + \vec{y} + \vec{e}_1\rangle \otimes |H_1 \vec{e}_1\rangle \otimes |0_m\rangle$$

si bien que

$$U_1 |\psi\rangle \otimes |0_m\rangle \otimes |0_m\rangle = \frac{1}{\sqrt{|C_2|}} \sum_{\vec{y} \in C_2} (-1)^{(\vec{x}+\vec{y}).\vec{e}_2} |\vec{x} + \vec{y} + \vec{e}_1\rangle \otimes |H_1 \vec{e}_1\rangle \otimes |0_m\rangle$$

Une mesure du second registre dans la base computationnelle donne $|H_1 \vec{e}_1\rangle$ avec probabilité 1 (car le ket $|H_1 \vec{e}_1\rangle$ peut être factorisé dans la dernière expression). Donc on connait le syndrome $H_1 \vec{e}_1$ et si $|\vec{e}_1| \leq t$ on peut évaluer $\vec{e}_1$ à partir de ce syndrome classique. Cela permet d'appliquer l'opération

$$\overline{\prod_{\substack{\text{composantes de } \vec{e}_1 \\ \text{égales à } 1}}} X_i \equiv \tilde{U}_1$$

pour obtenir

$$\tilde{U}_1 U_1 |\psi\rangle \otimes |0_m\rangle \otimes |0_m\rangle =$$

$$= \frac{1}{\sqrt{|C_2|}} \sum_{\vec{y} \in C_2} (-1)^{(\vec{x}+\vec{y}).\vec{c}_2} |\vec{x}+\vec{y}\rangle \otimes |H_1 \vec{e}_1\rangle \otimes |0_m\rangle .$$

Correction du phase-flip.

D'abord on applique les portes de Hadamard $H^{\otimes m}$ ce qui donne

$$H^{\otimes m} \tilde{U}_1 U_1 |\psi\rangle \otimes |0_m\rangle \otimes |0_m\rangle$$

$$= \frac{1}{2^{m/2}} \frac{1}{\sqrt{|C_2|}} \sum_{\vec{y} \in C_2} (-1)^{(\vec{x}+\vec{y}).\vec{c}_2} \sum_{\vec{z}} (-1)^{\vec{z}.(\vec{x}+\vec{y})} |\vec{z}\rangle \otimes |H_1 \vec{e}_1\rangle \otimes |0_m\rangle .$$

$$= \frac{1}{2^{n/2}} \frac{1}{\sqrt{|C_2|}} \sum_{\vec{z}} \sum_{\vec{y} \in C_2} (-1)^{(\vec{z}+\vec{e}_2).(\vec{x}+\vec{y})} |\vec{z}\rangle \otimes |H_1 \vec{e}_1\rangle \otimes |0_m\rangle$$

$$= \frac{1}{2^{n/2}} \frac{1}{\sqrt{|C_2|}} \sum_{\vec{z}} \sum_{\vec{y} \in C_2} (-1)^{\vec{z}.(\vec{x}+\vec{y})} |\vec{z}+\vec{e}_2\rangle \otimes |H_1 \vec{e}_1\rangle \otimes |0_m\rangle$$

On note que

$$\sum_{\vec{y} \in C_2} (-1)^{\vec{z}.\vec{y}} = \begin{cases} |C_2| & \text{si } \vec{z} \in C_2^{\perp} \\ 0 & \text{sinon. } (\text{c.a.d } \vec{z} \notin C_2^{\perp}). \end{cases}$$

ce qui donne l'état :

$$= \frac{1}{\sqrt{|C_2^{\perp}|}} \sum_{\vec{z} \in C_2^{\perp}} (-1)^{\vec{z}.\vec{x}} |\vec{z}+\vec{e}_2\rangle \otimes |H_1 \vec{e}_1\rangle \otimes |0_m\rangle,$$

On implémente maintenant le calcul du syndrome de $C_2^{\perp}$ de façon unitaire en stockant le résultat dans le dernier registre

$$U_2 \, |\vec{z}+\vec{e}_2\rangle \otimes |H_1 \vec{e}_1\rangle \otimes |0_m\rangle$$

$$= |\vec{z}+\vec{e}_2\rangle \otimes |H_1 \vec{e}_1\rangle \otimes |H_2^{\perp}(\vec{z}+\vec{e}_2)\rangle$$

$$= |\vec{z}+\vec{e}_2\rangle \otimes |H_1 \vec{e}_1\rangle \otimes |H_2^{\perp} \vec{e}_2\rangle.$$

où ici $H_2^{\perp}$ est la matrice de parité de $C_2^{\perp}$.

L'état obtenu est :

$$U_2 H^{\otimes m} \tilde{U}_1 U_1 |\psi\rangle \otimes |0_m\rangle \otimes |0_m\rangle$$

$$= \frac{1}{\sqrt{|C_2^{\perp}|}} \sum_{\vec{z} \in C_2^{\perp}} (-1)^{\vec{z}.\vec{x}} |\vec{z} + \vec{e}_2\rangle \otimes |H_1 \vec{e}_1\rangle \otimes |H_2^{\perp} \vec{e}_2\rangle .$$

La mesure du dernier registre (dans la base computationnelle) projette sur cet état lui-même avec probabilité 1. Donc $H_2^{\perp} \vec{c}_2$ est connu et $\vec{e}_2$ peut être calculé si $C_2^{\perp}$ corrige t erreurs (on suppose $|\vec{e}_2| \leq t$). Cela permet alors d'appliquer l'opérateur unitaire

$$\tilde{U}_2 = \overline{\prod_{\substack{\text{comp non} \\ \text{nulles de } \vec{e}_2}}} X_i$$

pour trouver l'état :

$$\tilde{U}_2 U_2 H^{\otimes m} \tilde{U}_1 U_1 |\psi\rangle \otimes |0_m\rangle \otimes |0_m\rangle$$

$$= \frac{1}{\sqrt{|C_2^{\perp}|}} \sum_{\vec{z} \in C_2^{\perp}} (-1)^{\vec{z}.\vec{x}} |\vec{z}\rangle \otimes |H_1 \vec{c}_1\rangle \otimes |H_2^{\perp} \vec{c}_2\rangle .$$

La dernière étape consiste maintenant à appliquer une fois de plus $H^{\otimes n}$. Cela donne

$$H^{\otimes n}\,\tilde{U}_2\,U_2\,H^{\otimes n}\,\tilde{U}_1\,U_1\;|\psi\rangle\otimes|0_n\rangle\otimes|0_n\rangle$$

$$= \frac{1}{\sqrt{|C_2^\perp|}}\,\frac{1}{2^{n/2}}\sum_{\vec{z}\in C_2^\perp}\sum_{\vec{y}}(-1)^{\vec{z}\cdot(\vec{x}+\vec{y})}\,|\vec{y}\rangle\otimes|H_1\vec{e}_1\rangle\otimes|H_2^\perp\vec{e}_2\rangle$$

$$= \frac{1}{\sqrt{|C_2^\perp|}}\,\frac{1}{2^{n/2}}\sum_{\vec{z}\in C_2^\perp}\sum_{\vec{y}}(-1)^{\vec{z}\cdot\vec{y}}\,|\vec{y}+\vec{x}\rangle\otimes|H_1\vec{e}_1\rangle\otimes|H_2^\perp\vec{e}_2\rangle$$

Maintenant on note (comme avant) :

$$\sum_{\vec{z}\in C_2^\perp}(-1)^{\vec{z}\cdot\vec{y}} = \begin{cases} |C_2^\perp| & \text{si } \vec{y}\in(C_2^\perp)^\perp = C_2 \\ 0 & \text{sinon.} \end{cases}$$

L'état final obtenu est donc :

$$\left(\frac{1}{\sqrt{|C_2|}}\sum_{\vec{y}\in C_2}|\vec{x}+\vec{y}\rangle\right)\otimes|H_1\vec{e}_1\rangle\otimes|H_2^\perp\vec{e}_2\rangle.$$

Nous voyons que le mot de code initial a été reconstruit dans le premier registre !

Exemple : Le code de Steane.

Le code de Steane est un code $CSS(C_1, C_2)$ avec $C_1 = \text{Hamming}[7, 4]$ et $C_2 = C_1^{\perp}$. On montre que le code possède 7 qubits, est de dimension 2 et corrige une erreur. Les mots de code sont (voir exercice)

$$|0\rangle_{\text{Steane}} = \frac{1}{\sqrt{8}} \{ |0000000\rangle + |1001101\rangle + |0101011\rangle + |0010111\rangle + |0111100\rangle + |1011010\rangle + |1100110\rangle + |1110001\rangle \}$$

et

$$|1\rangle_{\text{Steane}} = \frac{1}{\sqrt{8}} \{ |1111111\rangle + |0110010\rangle + |1010100\rangle + |1101000\rangle + |1000011\rangle + |0100101\rangle + |0011001\rangle + |0001110\rangle \}.$$

Le mot $|0\rangle_{\text{Steane}}$ correspond à la classe d'équivalence de $(0000000) \in C_1$ et le deuxième mot $|1\rangle_{\text{Steane}}$ correspond à la classe d'équivalence de $(1111111) \in C_1$.