

ANALYSES ET INTERPRETATIONS DES RESULTATS.

4.1 Introduction.

Dans ce chapitre nous allons d'abord présenter en première lieu les codes polaires avec ces différentes décodeurs SC, SCL et CA-SCL en suite décrire le model proposés pour une transmission MIMO massif avec codage polaire et décodage CA-SCL et en fin l'analyses et l'interprétassions des résultats sont présenter .Une simulation d'antenne a antenne est proposer avec une analyse et des perspectives pour poursuivre les recherches dans ce domaine.

4.2 LES CODES POLAIRES.

Les codes polaires, comme premiers codes à capacité prouvée sur des entrées binaires sans mémoire discrète ce sont les seuls codes pour lesquels on peut démontrer mathématiquement qu'ils atteignent la limite de Shannon pour un code de taille infinie. [1], ont fait l'objet d'une attention particulière parmi divers codes de correction d'erreur directe (FEC). En raison de leur structure explicite et leur faible complexité $O(N\log_2 N)$ de codage/décodage, les codes polaires sont devenus un des codes les plus importants de la théorie du codage. A ce jour, de nombreux efforts ont porté sur plusieurs aspects théoriques des codes polaires [1]. Dans [11], [12], [19], Les décodeurs polaires à annulation successive (SC) ont été présentés. Ces architectures de faible complexité conviennent à des applications à contraintes de surface ; toutefois, en raison de la complexité inhérente des natures de l'algorithme SC, ces décodeurs SC ne sont pas à une longue latence et un faible débit.

4.2.1 Notation.

on notera a_i^N pour désigner un sous-vecteur (a_i, \dots, a_N) . \oplus Pour addition modulo 2 et \otimes pour désigner le produit de Kronecker.

4.2.2 La construction de Codes Polaires

Un Code Polaire, CP $(N; K)$, est un code en bloc linéaire [1] de taille $N = 2^n$, n étant un entier naturel, contenant K bits d'informations. La matrice génératrice du code est une sous-matrice de la n ème puissance de Kronecker de $K = \begin{bmatrix} 1 & 0 \\ 1 & 1 \end{bmatrix}$, notée $\mathcal{F} = K^{\otimes n} = \begin{bmatrix} K^{\otimes n-1} & 0_{n-1} \\ K^{\otimes n-1} & K^{\otimes n-1} \end{bmatrix}$, composée de K lignes. Ces K lignes sont choisies en supposant un décodage à annulation Successive (SC) qui permet de polariser la probabilité d'erreur des bits du message. Un Code Polaire peut être représenté sous forme matricielle à

partir de la matrice F et sous forme de factor graph. Ce dernier pouvant être utilisé pour le codage et pour le décodage.

4.2.3 Processus de codage

Comme tout code en bloc, les mots de codes associés à des Codes Polaires sont définis par une matrice génératrice, G, de dimension $(K \times N)$. Cette matrice génératrice, G, est obtenue en supprimant $N - K$ lignes de la matrice F. Le processus de codage équivalent consiste ensuite à multiplier un vecteur de taille K par cette matrice G. Un processus de codage alternatif consiste à construire un vecteur, noté U, contenant les K bits d'information et $N - K$ bits gelés (Ces bits gelés, forcés à 0, sont placés au niveau des indices les moins fiables) fixés à 0. Ce vecteur est construit de telle manière que les bits d'information soient localisés sur les indices les plus fiables correspondant aux K lignes de $K^{\otimes n}$ sélectionnées précédemment. Le mot de code correspondant, X, peut ensuite être calculé simplement tel que $X = U \times K^{\otimes n}$.

Exemple 1 Pour un code CP (8;4) et un message $U = [0, 0, 0, u_3, 0, u_5, u_6, u_7]$, avec la position des bits gelés arbitraire, le mot de code est obtenu par la multiplication matricielle suivante :

$$[0, 0, 0, u_3, 0, u_5, u_6, u_7] \times \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \end{bmatrix} = \begin{bmatrix} u_3 + u_5 + u_6 + u_7 \\ u_3 + u_5 + u_7 \\ u_3 + u_6 + u_7 \\ u_3 + u_7 \\ u_5 + u_6 + u_7 \\ u_5 + u_7 \\ u_6 + u_7 \\ u_7 \end{bmatrix}^T$$

Un code en bloc peut-être représenté sous la forme d'un factor graph. Dans le cas des Codes Polaires, nous avons vu que la construction de la matrice génératrice est récursive. Il est alors possible de montrer que la construction du graphe est également récursive.

Exemple 2. Dans cet exemple, deux matrices génératrices ainsi que leur représentation graphique sont présentées. Dans la figure 4.1 la matrice génératrice et le graphe de codage d'un Code Polaire de taille $N = 2$ et de message $U = [u_0, u_1]$ sont présentés.

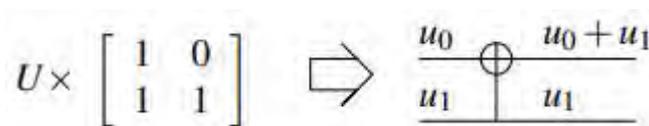


FIGURE 4.1 : Matrice génératrice et graphe de codeur Polaire CP(2,2).

De même dans la figure 4.2, la matrice génératrice et le graphe de codage d'un Code Polaire de taille $N = 4$ et de message $U = [u_0, u_1, u_3, u_4]$, sont donnés.

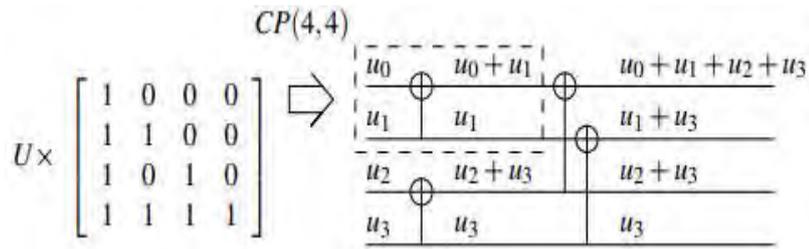


FIGURE 4.2 : Matrice génératrice et graphe de codeur Polaire CP(4,4).

Les nœuds de parités (\oplus) se comportent comme des XOR et les nœuds de variables (?) font simplement passer la valeur binaire à l'étage suivant. On peut remarquer alors qu'un code de taille $2N$ utilise deux fois la structure d'un code de taille N puis fait la somme des N premières lignes et transmet directement les N dernières lignes comme démontré ci-dessous :

$$U_{n+1} = [U_n, U'_n]$$

Avec :

U_n Les N premiers bits de U_{n+1}

U'_n Les N derniers bits de U_{n+1}

$$\begin{aligned} U_{n+1} \times \mathcal{F}_{n+1} &= [U_n, U'_n] \times \mathcal{F}_{n+1} \\ &= [U_n, U'_n] \times \begin{bmatrix} \mathcal{F}_n & 0_n \\ \mathcal{F}_n & \mathcal{F}_n \end{bmatrix} \\ &= [U_n \times \mathcal{F}_n + U'_n \times \mathcal{F}_n, U'_n \times \mathcal{F}_n] \\ &= [(U_n, U'_n) \times \begin{bmatrix} \mathcal{F}_n \\ \mathcal{F}_n \end{bmatrix}, U'_n \times \mathcal{F}_n] \end{aligned}$$

Plus généralement, le factor graph d'un Code Polaire, présenté dans [1], de taille $N = 2^n$ est composé de n étages de $\frac{N}{2}$ nœuds de parité de degré 3 et $\frac{N}{2}$ nœuds de variables de degré 3. Le degré d'un nœud représente son nombre de connections avec d'autres nœuds. Le factor graph peut être utilisé pour le codage et le décodage. Pour le codage, le vecteur d'entrée U , sur le côté gauche, est propagé dans le graphe dans le but de générer le mot de code X , sur la droite. Un Code Polaire de rendement $R = 0.5$ est donné en exemple dans la figure 1.13 ; cela signifie que la moitié des bits du vecteur U sont des bits gelés et l'autre moitié est composée de bits d'information d_i .

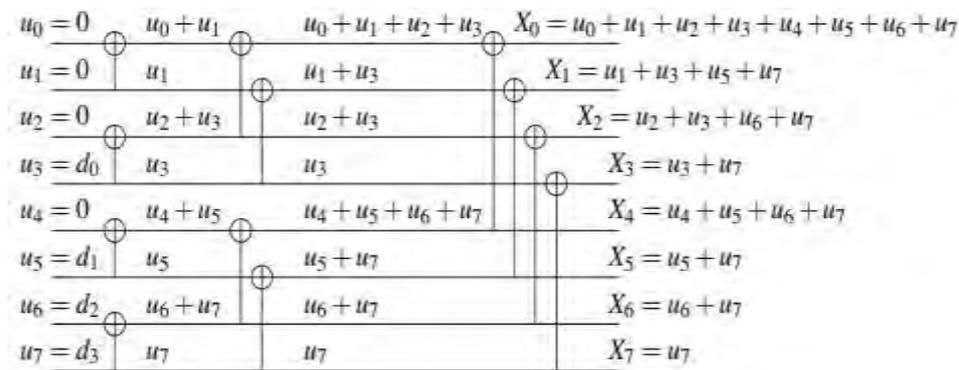


FIGURE 4.3 : Factor graph d'un codeur de Code Polaire CP (8;4) non systématique.

4.2.4 Le décodage par annulation successive (SC)

L'algorithme de décodage SC permet de décoder les Codes Polaires. Un **algorithme** est un ensemble de règles opératoires dont l'application permet de résoudre un problème énoncé au moyen d'un nombre fini d'opérations. (Dictionnaire Larousse) Un algorithme se caractérise en particulier par sa complexité. Cette dernière correspond au nombre d'opérations nécessaires pour compléter l'algorithme.

Un algorithme de décodage est une manière de traiter l'information reçue afin de récupérer, avec le moins d'erreurs possible, ce qui a été envoyé. L'information reçue correspond au mot de code qui a traversé le canal. Le décodage va alors permettre de récupérer le message transmis, avec le moins d'erreur possible, à partir de ces informations du canal. Les algorithmes sont plus ou moins complexes et performants. Les performances de décodage de l'algorithme représentent sa capacité à corriger les erreurs du mot reçu. En général, nous verrons que ces deux notions sont opposées. Pour les Codes Polaires, un premier algorithme a été proposé dans [1] et est détaillé dans le paragraphe qui suit.

-Décodage SC.

Une fois le message transmis au travers du canal de communication, la version bruitée,

$Y = Y_0^{N-1} = [Y_0, Y_1, \dots, Y_N]$, du mot de code $X = X_0^{N-1} = [X_0, X_1, \dots, X_N]$ est reçue. Le but du décodage est d'estimer le vecteur U à partir de la version bruitée du mot de code (Y). Arıkan a montré, que les Codes Polaires atteignent la capacité du canal sous l'hypothèse d'un décodage par annulation successive. Ce décodage consiste à estimer un bit u_i à partir de l'observation du canal et de la connaissance des bits précédemment estimés. La valeur du bit estimé est notée \hat{u}_i .

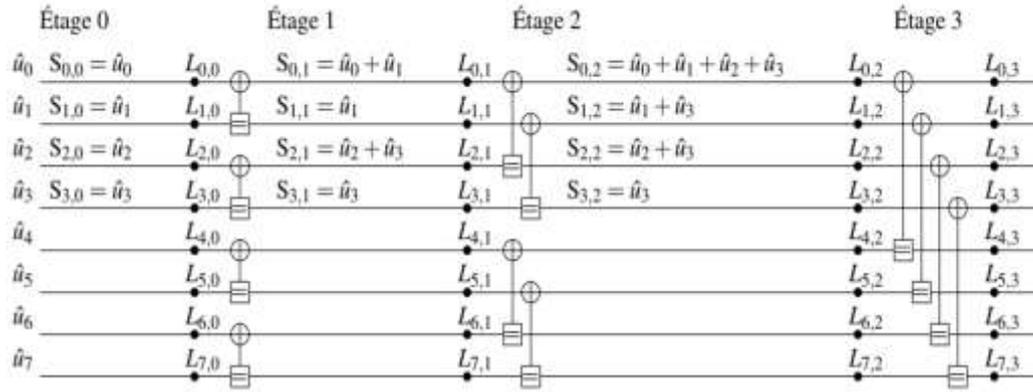


FIGURE 4.4 : Factor graph d'un Code Polaire de taille $N = 8$.

Chaque échantillon Y_i est converti dans un format appelé Likelihood Ratio - Rapport de vraisemblance (LR). Ces LR, notés $L_{i,n}$, sont positionnés sur le graphe comme dans la figure 1.14 pour $N = 8$, avec $0 \leq i \leq N - 1$. Les nœuds de parités (\oplus) et les nœuds de variables (\ominus), représentés différemment, ne symbolisent plus un simple XOR ni une simple copie. Au cours du décodage les différentes valeurs $L_{i,j}$ ($0 \leq j \leq n$) sont mise à jour ainsi que les valeurs $S_{i,j}$. Ces dernières, appelées sommes partielles, représentent le recodage des bits \hat{u}_i , au fur et à mesure de leur estimation. Le séquençement particulier des opérations est expliqué ci-après.

Tout d'abord, il est à noter que la mise à jour des LR, $L_{i,j}$, et des sommes partielles, $S_{i,j}$, peut être calculée efficacement en utilisant la représentation graphique des Codes Polaires. En effet, la matrice génératrice est inversible dans l'ensemble F_2 et son inverse est-elle même. On peut donc aisément montrer par récurrence que $\mathcal{F} \times \mathcal{F} = I_N$.

$$\begin{bmatrix} K^{\otimes n-1} & 0_{N/2} \\ K^{\otimes n-1} & K^{\otimes n-1} \end{bmatrix} \times \begin{bmatrix} K^{\otimes n-1} & 0_{N/2} \\ K^{\otimes n-1} & K^{\otimes n-1} \end{bmatrix} = \begin{bmatrix} I_{N/2} & 0_{N/2} \\ 0_{N/2} & I_{N/2} \end{bmatrix} \quad (4.1)$$

Nous pouvons alors exprimer l'égalité suivante :

$$[X] \times \mathcal{F} = [U \times \mathcal{F}] \times \mathcal{F} = U : \quad (4.2)$$

Pour estimer successivement chaque bit u_i , le décodeur se base sur l'observation du vecteur issu du canal $L_{(0,n)}^{(N-1)} = [L_{(0,n)} \dots L_{(N-1,n)}]$ et des bits précédemment estimé $\hat{u}_0^{i-1} = [\hat{u}_0 \dots \hat{u}_{i-1}]$. Dans ce but, le décodeur doit calculer les valeurs des LR suivants :

$$L_{i,0} = \frac{\Pr(Y_0^{N-1}, \hat{u}_0^{i-1} | u_i = 0)}{\Pr(Y_0^{N-1}, \hat{u}_0^{i-1} | u_i = 1)} \quad (4.3)$$

Au cours du décodage, lors de la mise à jour d'un étage $j > 0$, ce ne sont pas les bits \hat{u}_i qui sont utilisés directement, mais les sommes partielles, $S_{i,j}$, qui sont une combinaison de ces bits estimés. Lorsque le décodage met à jour un LR de l'étage 0, $L_{i,0}$, alors le décodeur prend une décision quant la valeur du bit \hat{u}_i est tel que :

$$\hat{u}_i = \begin{cases} 0 & \text{si } L_{i,0} > 1 \\ 1 & \text{sinon.} \end{cases} \quad (4.4)$$

Le décodeur a connaissance des bits gelés. Par conséquent, si u_i est un bit gelé, alors $\hat{u}_i = 0$ peu importe la valeur de $L_{i,0}$. L'algorithme de décodage est appelé Annulation Successive (SC), se déroulement est expliqué ci-après. Le décodeur estime successivement les bits \hat{u}_i à partir des LRs, $L_{i,j}$, et des sommes partielles, $S_{i,j}$, qui sont calculés tels que :

$$L_{i,j} = \begin{cases} F(L_{i,j+1}, L_{i+2^j,j+1}) & \text{si } B_{i,j} = 0 \\ G(L_{i-2^j,j+1}, L_{i,j+1}, S_{i-2^j,j}) & \text{si } B_{i,j} = 1 \end{cases}, \quad (4.5)$$

$$S_{i,j} = \begin{cases} H_1(S_{i,j-1}, S_{i+2^{j-1},j-1}) & \text{si } B_{i,j-1} = 0 \\ G(L_{i-2^j,j+1}, L_{i,j+1}, S_{i-2^j,j}) & \text{si } B_{i,j-1} = 1 \end{cases}, \quad (4.6)$$

Avec :

$$\begin{cases} F(a, b) = \frac{1+ab}{a+b} \\ G(a, b, S) = b \times a^{1-2S} \\ H_u(S) = S \\ H_1(S, S') = S \oplus S' \\ B_{i,j} = \frac{i}{2^j} \bmod 2, 0 \leq i < N \text{ and } 0 \leq j < n \end{cases} \quad (4.7)$$

Preuve :

Notations

Nous travaillons dans l'ensemble fini binaire dont les valeurs sont contenues dans l'ensemble $\{0,1\}$, d'où l'opération de somme et de ou exclusif sont équivalentes. Dans la suite les notations suivantes sont utilisées :

- La probabilité de transition du canal W est notée $W(y_j|x_i)$. Cela représente la probabilité d'obtenir y_j sachant que x_i a été envoyé à travers le canal.
- Les vecteurs sont notés $u_a^b = [u_a, u_{a+1}, \dots, u_b]$, $a < b$
- Le rapport de vraisemblance (Likelihood Ratio - LR) : sachant que N est la taille du mot de code et que \hat{u}_i est le i^{me} bit à être estimé au cours du décodage, on a :

$$L_N^i(y_1^N, \hat{u}_1^{i-1}) = \frac{\Pr(y_1^N, \hat{u}_1^{i-1} | u_i = 0)}{\Pr(y_1^N, \hat{u}_1^{i-1} | u_i = 1)} \quad (4.8)$$

- La combinaison de canaux :
 \mathcal{X} est l'ensemble des mots de codes possibles en entrée du canal.
 \mathcal{Y} est l'ensemble des mots de codes possibles en sortie du canal.

K est le noyau de la matrice génératrice utilisée $K = \begin{bmatrix} 1 & 0 \\ 1 & 1 \end{bmatrix}$. Donc, pour coder un mot de code plus grand, la matrice est élevée à une puissance de Kronecker qui dépend de la taille du code.

Par exemple : $K^{\otimes 2} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 \\ 1 & 1 & 1 & 1 \end{bmatrix}$ $W_2 : x^2 \rightarrow y^2$

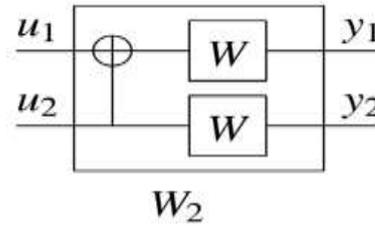


FIGURE 4.5 : codeur deux bits

$$w_2(y_1^2, u_1^2) = W(y_1|u_1 \oplus u_2) W(y_2|u_2) \quad w_2(y_1^2, u_1^2) = w_2(y_1^2, u_1^2) * G$$

– Probabilités associées à la règle de somme :

$$\Pr(A) = \sum_B \Pr(A, B) = \sum_B [\Pr(A|B)\Pr(B)]$$

Si $A = (y_1^2 | u_1 = 0)$

$$w_2(y_1^2 | u_1 = 0) = \sum_{u_2} [w_2(y_1^2 | u_1 = 0, u_2) \times \Pr(u_2)]$$

La plupart du temps $\Pr(u_2) = \frac{1}{2}$, donc

$$w_2(y_1^2 | u_1 = 0) = \frac{1}{2} \sum_{u_2} [w_2(y_1^2 | u_1 = 0, u_2)]$$

Décodage

Les seules données disponibles sont celles provenant du canal. Cette fois, le traitement ne correspond pas seulement à des additions car nous travaillons avec des Likelihood Ratio - Rapport de vraisemblance (LR). L'exemple suivant décrit le principe de décodage sur un code de taille 2.

- décodage \hat{u}_1 de fonction F

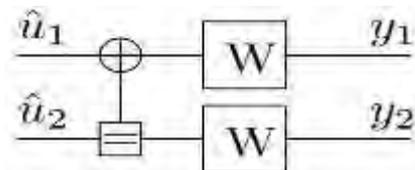


FIGURE 4.6 : Décodeur a deux bits

Nous cherchons à calculer le Likelihood Ratio - Rapport de vraisemblance (LR) de u_1 .

1 :

$$\Pr(y_1^2 | u_1 = 0) = \frac{1}{2} [w_2(y_1^2 | u_1 = 0, u_2 = 0) + w_2(y_1^2 | u_1 = 0, u_2 = 1)]$$

$$\begin{aligned}
&= \frac{1}{2} [W^2(y_1^2|u^*G) + W^2(y_1^2|u^*G)] \\
&= \frac{1}{2} [W(y_1|x_1=0) W(y_2|x_2=0) + W(y_1|x_1=1) W(y_2|x_2=1)]
\end{aligned}$$

2 : X

$$\begin{aligned}
\Pr(y_1^2|u_1=1) &= \frac{1}{2} [w_2(y_1^2|u_1=1, u_2=0) + w_2(y_1^2|u_1=1, u_2=1)] \\
&= \frac{1}{2} [W^2(y_1^2|u^*G) + W^2(y_1^2|u^*G)] \\
&= \frac{1}{2} [W(y_1|x_1=1) W(y_2|x_2=0) + W(y_1|x_1=0) W(y_2|x_2=1)]
\end{aligned}$$

3 : on applique le rapport entre 1 et 2

$$\begin{aligned}
L_2^{(1)}(y_1^2) &= \frac{\Pr(y_1^2|u_1=0)}{\Pr(y_1^2|u_1=1)} \\
&= \frac{\frac{1}{2} [W(y_1|x_1=0) W(y_2|x_2=0) + W(y_1|x_1=1) W(y_2|x_2=1)]}{\frac{1}{2} [W(y_1|x_1=1) W(y_2|x_2=0) + W(y_1|x_1=0) W(y_2|x_2=1)]} \\
&= \frac{\frac{W(y_1|x_1=0) W(y_2|x_2=0)}{W(y_1|x_1=1) W(y_2|x_2=1)} + 1}{\frac{W(y_2|x_2=0) + W(y_1|x_1=0)}{W(y_2|x_2=1) + W(y_1|x_1=1)}} \\
F(L_1^{(1)}(y_1), L_1^{(1)}(y_2)) &= \frac{1 + L_1^{(1)}(y_1) L_1^{(1)}(y_2)}{L_1^{(1)}(y_1) L_1^{(1)}(y_2)}
\end{aligned}$$

La fonction de décodage F est maintenant définie : $F(a, b) = \frac{1+a \times b}{a+b}$. Il reste encore l'autre LR à calculer.

- Décodage \hat{u}_2 de fonction F

Nous supposons que les bits précédents sont connus. Dans notre exemple \hat{u}_0 est connu et nous pouvons l'utiliser. Nous cherchons à calculer le Likelihood Ratio - Rapport de vraisemblance (LR) de u_2 .

$$L_2^{(2)}(y_1^2, \hat{u}_1) = \frac{\Pr(y_1^2, \hat{u}_1=0|u_2=0)}{\Pr(y_1^2, \hat{u}_1=0|u_2=1)}$$

1:

$$\Pr(y_1^2, \hat{u}_1=0|u_2=0) = \frac{1}{2} [w_2(y_1^2|\hat{u}_1, u_2=0)]$$

$$= \frac{1}{2} [W(y_1|x_1 = \hat{u}_1) W(y_2|x_2 = 0)]$$

2:

$$\begin{aligned} \Pr(y_1^2, \hat{u}_1 = 0 | u_2 = 1) &= \frac{1}{2} [w_2(y_1^2 | \hat{u}_1, u_2 = 1)] \\ &= \frac{1}{2} [W(y_1|x_1 = \hat{u}_1 \oplus 1) W(y_2|x_2 = 1)] \end{aligned}$$

3 : on applique le rapport entre 1 et 2

$$L_2^{(2)}(y_1^2, \hat{u}_1) = \frac{\frac{1}{2} [W(y_1|x_1 = \hat{u}_1) W(y_2|x_2 = 0)]}{\frac{1}{2} [W(y_1|x_1 = \hat{u}_1 \oplus 1) W(y_2|x_2 = 1)]}$$

Si $\hat{u}_1 = 0$:

$$\begin{aligned} L_2^{(2)}(y_1^2, \hat{u}_1) &= \frac{\frac{1}{2} [W(y_1|x_1 = 0) W(y_2|x_2 = 0)]}{\frac{1}{2} [W(y_1|x_1 = 1) W(y_2|x_2 = 1)]} \\ &= L_1^{(2)}(y_1, \hat{u}_1 = 0), L_1^{(2)}(y_2, \hat{u}_1 = 0) \end{aligned}$$

Si $\hat{u}_1 = 1$:

$$\begin{aligned} L_2^{(2)}(y_1^2, \hat{u}_1) &= \frac{\frac{1}{2} [W(y_1|x_1 = 1) W(y_2|x_2 = 0)]}{\frac{1}{2} [W(y_1|x_1 = 0) W(y_2|x_2 = 1)]} \\ &= \frac{L_1^{(2)}(y_2, \hat{u}_1 = 0)}{L_1^{(2)}(y_1, \hat{u}_1 = 0)} \end{aligned}$$

$G(L_1^{(2)}(y_1, \hat{u}_1 = 0), L_1^{(2)}(y_2, \hat{u}_1 = 0)) = L_1^{(2)}(y_2, \hat{u}_1 = 0) L_1^{(2)}(y_1, \hat{u}_1 = 0)^{1-2\hat{u}_1}$ La fonction G est maintenant définie : $G(a, b, SP) = b \times a^{SP_{i,j}}$. On remarque que pour calculer cette fonction, nous avons besoin d'un bit particulier noté $SP_{i,j}$. Ceci génère donc une dépendance, d'où un ordonnancement particulier pour le décodage.

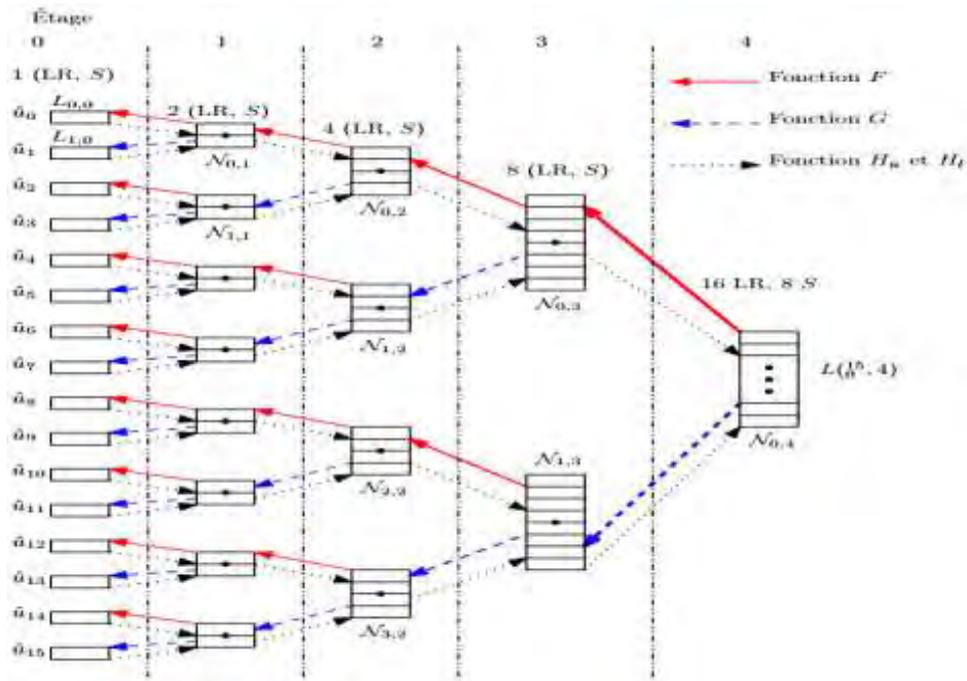


FIGURE 4.7 – Processus de décodage représenté sous la forme d’un arbre binaire.

La somme partielle $S_{i,j}$ correspond à la propagation des décisions dures dans le factor graph.

Par exemple, dans la figure 1.14, $S_{1,2} = \hat{u}_1 + \hat{u}_2$ (somme modulo 2).

L’algorithme de décodage peut être représenté par un arbre binaire complet, comme illustré dans la figure 1.17. L’arbre de profondeur 4 représente le décodage d’un mot de code de taille $N = 16$. Les branches symbolisent les fonctions F , G , H_u et H_l . Les nœuds représentent les LR et les sommes partielles intermédiaires, calculés durant le processus de décodage. Les fonctions F et G traitent les LR de la droite vers la gauche et stockent les résultats dans les nœuds de gauche. Les fonctions H récupèrent les sommes partielles du nœud de gauche, les traitent puis les stockent dans le nœud de droite. Dans la description actuelle de l’algorithme de décodage, un nœud N est mis à jour lorsque tous les nœuds qui sont à sa gauche sont mis à jour et que toutes ses sommes partielles sont mises à jour. Cet algorithme récursif s’exécute à partir de la racine de l’arbre. Le nœud en cours de mise à jour est appelé N . Si N a des fils, alors la branche supérieure est calculée. Les fonctions F traitent les LR de N et stockent les LR résultant dans le fils supérieur de N noté N_u . Quand N_u est mis à jour, la fonction H_u récupère les sommes partielles de N_u et les stocke dans N . Ensuite, la branche inférieure est calculée : les fonctions G traitent les LR de N , en utilisant les sommes partielles précédemment récupérées. Les LR résultant sont ensuite stockés dans le fils inférieur à N noté N_l . Quand N_l est mis à jour, la fonction H_l combine les sommes partielles de N_l et N et stocke le résultat dans N . Le nœud N est maintenant mis à jour. Si le nœud N n’a pas de fils

(niveau des feuilles) alors la somme partielle est obtenue en fonction de la valeur du LR de N. Le processus de décodage est récapitulé dans l’algorithme.

Algorithme 1 : Algorithme de décodage par annulation successive

Résultat	:	Sommes	partielles	de	N	mises	à	jour.
Entrées	:				Nœud			N
MAJ		(Node	N)	-	Mise	À		Jour
si		N	a	des	fil			alors
		1.	Calculer F à partir des LR de N et stocker les LR résultant dans N_u					
		2.			MAJ			(N_u)
		3.	Calculer la fonction H_u : elle récupère les sommes partielles de et N_u les stocke dans					N
		4.	Calculer G à partir des LR et sommes partielles de N et stocker les LR résultants					N_1
		5.			MAJ			(N_1)
		6.	Calculer la fonction H_1 : elle combine les sommes partielles de N_1 et N, les stocke		ensuite			N
sinon								
								Prend une décision pour la somme partielle de N suivant la valeur de son LR

Le décodeur d’annulation successive (SC) est simple et de faible complexité mais présente des limites. Par conséquent, des progrès importants ont été réalisés en ce qui concerne l’amélioration des performances avec une longueur de bloc courte ou modérée [17][18][19] d’où la proposition du décodage SCL.

4.2.5 Décodage SCL

On n’a montré au paragraphe précédent que le décodeur SC a des limites. On peut améliorer les performances de décodage médiocres SC pour des faibles tailles de codes en utilisant le décodeur SCL. Le décodeur SCL présenté dans cette section permet d’améliorer les performances pour les Codes Polaires de petite taille. L’algorithme de décodage SC-LIST a été introduit dans Vardy [23]. C’est une généralisation du décodage SC. L’algorithme de décodage peut considérer jusqu’à L chemins différents en parallèle au cours du processus de décodage. Par exemple, lorsqu’un bit u d’information est estimé, alors le processus de décodage est dupliqué. Un des processus considère que la valeur estimée est un 0, l’autre processus considère que la valeur estimée est un 1, comme dans l’exemple de décodage avec une liste de taille $L = 2$ d’un Code Polaire $CP(4;4)$ dans la figure 1.13. Il est à noter que lorsqu’un bit gelé est rencontré, le processus n’est pas dupliqué, car la valeur du bit gelé n’est pas modifiable. Quand plus de L processus sont considérés en parallèle, alors seulement les L

processus de décodage les plus probables, qui ont la plus grande métrique dans la liste, sont conservés, comme illustré dans la figure 1.13.

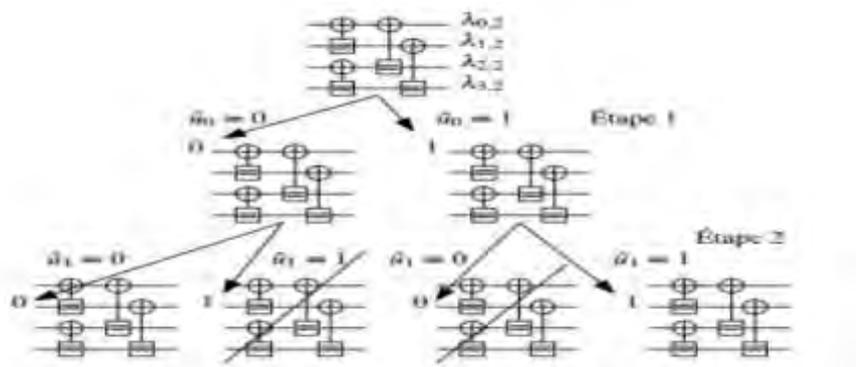


FIGURE 4.8 : Décodage SCL d'un Code Polaire $CP(4;4)$ avec une liste de taille $L = 2$.

4.2.6 Décodage CA-SCL.

Les codes polaires avec décodage SCL assisté par contrôle de redondance cyclique (CA-SCL) peuvent être supérieurs aux codes de contrôle de parité à faible densité (LDPC) en termes de performance et de complexité sous certaines conditions [20][21]. Ainsi, le décodage CA-SCL est le plus pratiqué. Malgré cela, ses performances et la complexité en longueur finie sont encore limitées dans les communications. Actuellement, beaucoup de recherches ont été faites sur la base de Codes polaires CA-SCL incluant l'amélioration de la performance et la réduction de la complexité.

Les codes polaires associés CRC (Cyclic Redundancy Check) sont proposés dans [22] en mettant l'accent sur la réduction de la complexité de décodage et de diminuer le temps de décision par rapport à une méthode de décodage SCL assisté par un seul CRC. Les résultats montrent que l'espace mémoire de décodage et le délai peuvent être réduits considérablement grâce à une conception raisonnable de multi-CRC. Cependant, la déficience des multi-CRC polaires est que la perte de performance ne peut être ignorée à des niveaux de rapport signal/bruit (SNR) importants, il s'agit donc d'un algorithme sous-optimal qui ne peut pas atteindre le décodage du maximum de vraisemblance (ML) le niveau le plus performant. Dans [23], le décodage polaire segmenté CA-SCL est proposé, qui fonctionne de telle sorte que de nombreuses CRC devraient être nécessaires. Le délai de décodage est réduit en mettant fin au calcul préalable des trajectoires inutiles et ce schéma peut atteindre les mêmes performances que le décodage CA-SCL. Mais, le décodage CA-SCL segmenté augmente la complexité du codage par rapport à la polarité d'origine de l'encodage avec un seul CRC.

Compte tenu de ce point, le code polaire CA-SCL est proposé [24.9][25.10]. Cette méthode est également utile pour améliorer les performances du BLER (**B**lock **E**rror **R**ate). Cependant, il n'y a pas de règle claire pour placer les bits de contrôle de parité mais on peut les placer en introduisant en fonction du poids de polarisation (PW) des canaux [26.11].

4.2.7 Performance des codes polaires.

Le code polaire [1], proposé en premier lieu par Arikan en 2009, est l'un des plus prometteurs en matière de correction d'erreurs ces dernières années. C'est le seul code correcteur d'erreur qui atteint mathématiquement la capacité des canaux symétriques discrets sans mémoire (BI-DMS) à entrer binaire. Par conséquent, il bénéficie d'une attention importante du à l'appui théorique de beaucoup de recherche.

4.2.8 Modèle de transmission mimo massif avec codage/décodage polaire.

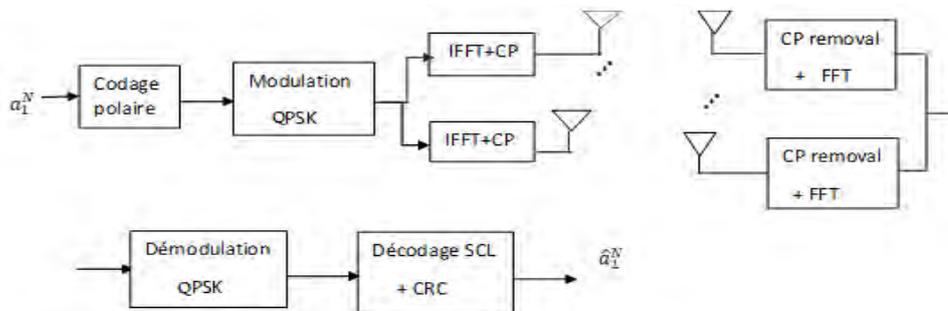


FIGURE 4.9 : Modèle de transmission mimo massif avec codage/décodage polaire.

Dans cette modèle de transmission on émet un message a_1^N qui est codé avec les codes polaire et moduler, pour chaque antenne d'émission on ajoute le transformée de Fourier rapide inverse IFFT et le CP est ajouter aussi mais au niveau de la réception on applique une transformé de Fourier rapide FFT, un removal CP, une démodulation et en fin le décodage CA-SCL est appliquer et le message reçu est sous la forme \hat{a}_1^N .

4.2.9 Analyses et interprétations des résultats.

La figure 4.2 étudie l'évolution des codes LDPC par rapport aux codes polaires en augmentant le nombre d'antenne progressivement pour les codes polaires, la figure 4.2 représente le FEC (10^0 à 10^{-3}) en fonction du SNR(-10 à 30dB).

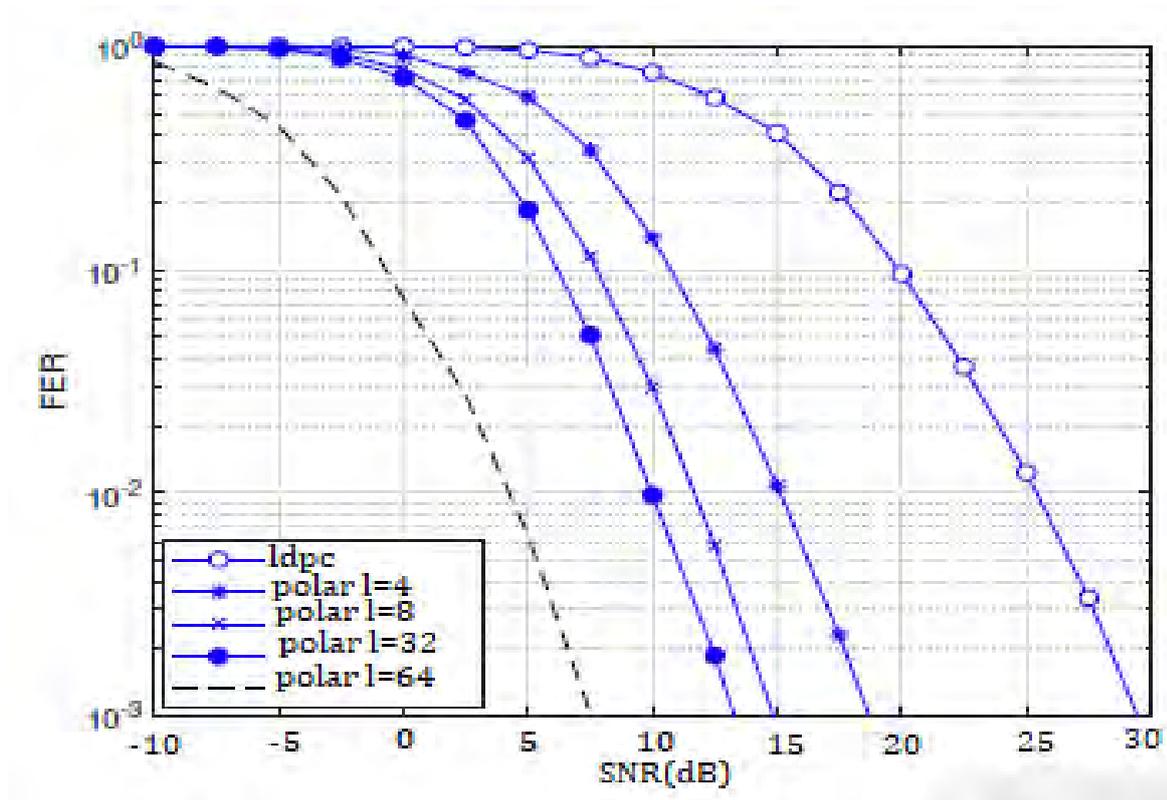


FIGURE 4.10 : Performances des codes polaires par rapport au code LDPC appliqué à un système MIMO MASSIF avec $N = 256$, $R = 0,5$ et une modulation QPSK.

Les résultats on montrer une diminution du SNR en fonction du FER des codes polaire par rapport au code LDPC cela confirme que les codes polaires sont plus performants que les codes LDPC. On note une nette diminution du SNR pour les codes polaire en augmentant le nombre d'antenne de 4 à 64 et le gain passe de 13dB et sa confirme les résultats obtenus par [Aliou Thiame] plus le nombre d'antennes est élevé plus les performances des systèmes mimo massif augmente cela est due à une augmentation du nombre de degré de liberté, qui donne plus de possibilité aux antennes de la station de basse BS.

La figure 4.11 représente la variation du FEC (10^0 à 10^{-3}) en fonction du SNR (-5 à 30dB) par rapport au code polaire et LDPC en variant le nombre d'antenne et le nombre d'itération

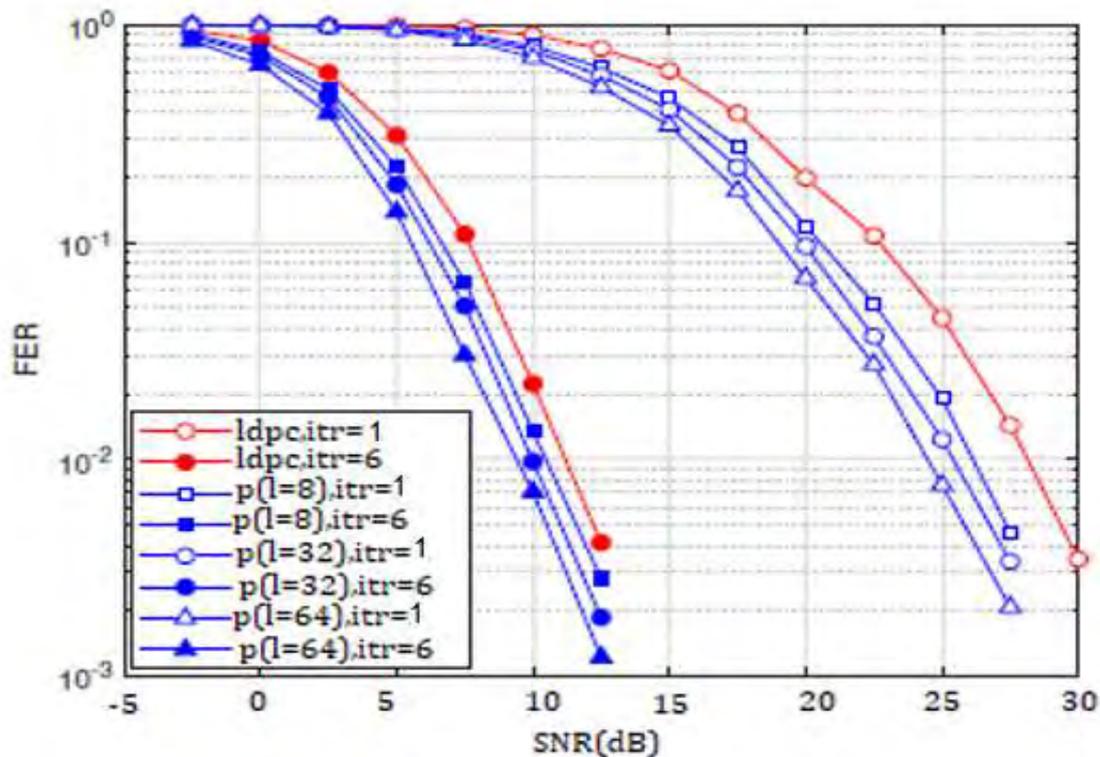


FIGURE 4.11 : Performances FER pour un système MIMO massif à codage polaire et LDPC par rapport au SNR avec $N = 256$, $R = 0.5$ et modulation QPSK.

Les résultats montrent une nette différence si on augmente le nombre d'itération, par exemple pour le code LDPC en augmentant le nombre d'itération on obtient un gain de 17.5dB même chose pour les codes polaires si on prend par exemple le code polaires à $L=8$ et $L=64$ on obtient respectivement un gain de 15 et 15.3dB, ces résultats montrent qu'une augmentation du nombre d'itération démontre le SNR ce qui rend plus performant le code cela confirme les résultats obtenus par [61] qui a montré que plus le nombre d'itération augmente plus le système est performant.

4.1.10 Conclusion.

Dans ce chapitre, nous avons montré en premier lieu la construction des Codes Polaires, l'algorithme de décodage SC, a été détaillé car beaucoup de travaux en découlent, suite nous avons montré, la performance des codes polaires par rapport au code LDPC et la dernière partie expliquée l'évolution des décodeurs SC vers le SCL et le CA-SCL en montrant les limites en terme de performance et de complexité. Les codes polaires on

montrait des avantages par rapport à leur simple décodeur et de faible complexité. Nous avons montré que les performances des systèmes MIMO massifs son proportionnel au nombre d'antenne de la station de base SB et qu'on peut améliorer les performances des codes polaires et LDPC en augmentant le nombre d'itération. D'après tous ces résultats on peut dire que le système MIMO massif associé au codes polaires son performant et il constitue un candidat potentiel pour la cinquième génération 5G.