

Architecture d'un détecteur itératif pour système à multi-antennes basé sur l'algorithme de propagation de croyance

Ali HAROUN¹, Charbel ABDEL NOUR², Matthieu ARZEL², Christophe JEGO¹

¹Bordeaux INP, Enseirb-Matmeca, CNRS IMS UMR 5218
351 Cours de la Libération, 33405 Talence Cedex, France

²Institut Mines-Télécom, IMT Atlantique, CNRS Lab-STICC UMR 6285
Technopôle Brest-Iroise, CS 83818, 29238 Brest Cedex 3, France

Ali.Haroun@ims-bordeaux.fr, Charbel.abdelnour@imt-atlantique.fr
Matthieu.Arzel@imt-atlantique.fr, christophe.jego@ims-bordeaux.fr

Résumé – Cet article présente une nouvelle architecture pour la détection souple itérative multi-antennes basée sur l'algorithme de propagation de croyance. Cette architecture correspond à une détection simplifiée basée sur les techniques détaillées dans [1]. A l'aide de cette architecture, nous pouvons calculer le rapport de vraisemblance du point initial d'une constellation en considérant la méthode de sous-régions dynamiques [1]. Ainsi, tous les points voisins du point détecté peuvent être obtenus à l'aide d'un calcul récursif. Grâce à un niveau approprié de parallélisme, l'architecture associée au détecteur souple permet des échanges d'information efficaces. Par conséquent, il favorise un bon compromis entre les performances au niveau taux d'erreurs binaires et la complexité calculatoire.

Abstract – This paper presents a novel architecture of an iterative multi-antenna soft detector based on the belief propagation (BP) algorithm. This architecture corresponds to a simplified detection based on the techniques detailed in [1]. By using this architecture, we can calculate the likelihood ratio of the initial point of a constellation by considering the dynamic region method [1]. Thus, all the points close to the detected point can be obtained by a recursive calculation. Thanks to an appropriate level of parallelism, the soft detector architecture enables efficient information exchange. Therefore, it promotes a compromise between the performance in terms of error rate and the computational complexity.

1 Introduction

Les systèmes multi-antennes, dit MIMO (*Multiple-Input Multiple-Output*) permettent d'accroître la capacité des liens de communications par rapport aux systèmes SISO (*Single-Input Single-Output*) [2]. Cette technique s'avère efficace pour augmenter les débits et la fiabilité tout en offrant une plus grande efficacité spectrale grâce à la diversité et au Multiplexage Spatial (MS). Malgré les atouts d'un processus itératif au niveau du récepteur MIMO, les coûts en termes de complexité de calcul et de latence demeurent un défi majeur. Dans la littérature, de nombreuses études se sont intéressées à la mise en œuvre de différents algorithmes de détection MIMO. Elles sont basées sur une détection de type Maximum de Vraisemblance (MV) [3], des critères de décodage sphérique [4] ou une détection de type MMSE [5]. Il est communément admis qu'une détection de type MV classique est optimale au niveau des performances. Cependant, sa complexité calculatoire est incompatible avec une implantation matérielle. Par ailleurs, il existe des détecteurs MIMO ayant une complexité polynomiale, à savoir le détecteur basé sur le décodage sphérique. Enfin, plusieurs études telles que [5] et [6] ont montré que les algorithmes de type MMSE permettent de réduire la complexité calculatoire et d'atteindre des performances s'approchant de celles d'un détecteur de type MV. Cependant, les détecteurs MIMO de type MMSE reposent sur une inversion matricielle. C'est pourquoi, la majorité des travaux consacrés à ce type de détection se sont intéressés aux techniques d'inversion matricielle pour faciliter l'implantation.

Dans [7], les auteurs ont proposé une autre approche de détection basée sur la propagation de croyance (BP), ayant une complexité de calcul linéaire, afin de réduire la complexité calculatoire et la latence des détecteurs MIMO. Cette dernière offre des avantages indéniables pour les implantations matérielles. Dans les travaux récents [8], [9]

et [10], une solution conjointe de détection et de décodage pour un système MIMO-BP (MIMO basé sur l'algorithme BP) est proposée dans un contexte itératif. Dans ces travaux, le principe de la propagation de croyance est appliqué aussi bien à la détection des symboles reçus qu'à leur décodage. Pour ce faire, un code correcteur d'erreurs de type LDPC non binaire (*Non-Binary Low-Density Parity-Check*) est exploité. Un nouveau schéma de réception itérative qui combine un détecteur MIMO-BP associé à un demapping avec une constellation (MAQ-64) suivi d'un décodeur LDPC non binaire a été proposé.

Dans cet article, nous allons décrire une solution architecturale pour le détecteur MIMO-BP étudié dans les travaux précédents [1]. A l'aide de cette architecture, le calcul du Rapport de Vraisemblance (RV) du point initial d'une constellation est effectué [1]. Dès lors, tous les points voisins du point détecté peuvent être déduits à l'aide d'un calcul récursif [10]. Grâce au niveau de parallélisme approprié, l'architecture associée au détecteur itératif permet un échange d'information efficace et par conséquent la montée en débit et une maîtrise de la latence du récepteur itératif.

2 Modèle et paramètres du système

Dans cet article, nous modélisons une chaîne de transmission MIMO intégrant un codeur LDPC non binaire basé sur le corps de Galois GF(64) ayant un rendement $R = 1/2$ et des degrés de connectivité des nœuds de variable et de parité $d_v = 2$ et $d_c = 4$. De plus, une modulation d'amplitude en quadrature MAQ-64 suivie d'un multiplexage spatial composé de $N_t = 2$ antennes à l'émission et de $N_r = 2$ antennes à la réception a été intégré dans notre système. Le canal de transmission retenu est un canal non sélectif en fréquence à évanouissements de type Rayleigh. À la réception, un détecteur BP-MIMO suivi d'un décodeur LDPC non binaire pour une longueur de trame $N = 384$

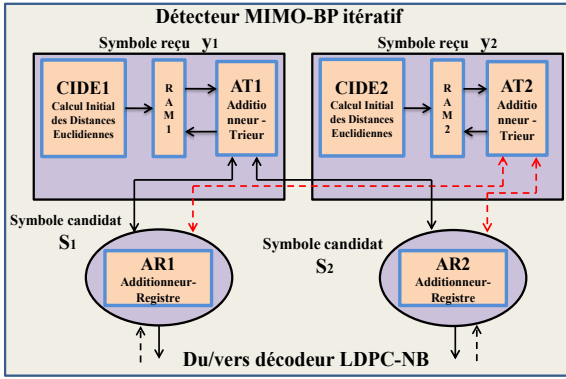


FIG. 1: Architecture du détecteur MIMO-BP itératif.

symboles sont utilisés. Afin d'améliorer les performances, les informations extrinsèques sont échangées entre le détecteur BP-MIMO et le décodeur LDPC non binaire au cours d'un processus itératif. Plus d'informations sont disponibles dans [8], [9] et [10].

3 Architecture du détecteur MIMO-BP

Dans cette section, les différents blocs du détecteur MIMO-BP sont successivement décrits. Pour tous les blocs, l'objectif est de parvenir à un compromis entre la complexité matérielle et les performances. Le détecteur MIMO-BP est basé sur le principe de détection simplifiée pour calculer le RV du point initial le plus proche du symbole reçu en considérant la méthode de sous-régions dynamiques décrite dans [1]. Puis, les distances Euclidiennes d'un nombre limité de points voisins sont déduites par un calcul récursif tel qu'expliqué dans [1], [10]. En fait, ce calcul représente la partie critique du détecteur.

L'architecture globale du détecteur MIMO-BP est donnée dans la Figure 1. La première fonction effectuée par cette architecture est le calcul des distances Euclidiennes sur deux dimensions, ($N_t = N_r = 2$), de tous les symboles candidats pour chaque symbole reçu. Cette fonction est effectuée par le bloc CIDE (Calcul Initial des Distances Euclidiennes) qui représente l'élément critique au niveau de la complexité matérielle du détecteur comme contenant la plupart des ressources matérielles du dernier. Les distances calculées sont sauvegardées dans une mémoire RAM afin de les mettre à jour au cours de chaque itération entre le détecteur et le décodeur. Ensuite, un bloc AT (Additionneur-Trieur) est nécessaire pour additionner les informations extrinsèques provenant des nœuds candidats et du décodeur à celles qui sont associées dans la RAM. Puis, les sommes sont triées selon leur fiabilité. Les calculs au sein des nœuds associés aux symboles candidats sont effectués au sein d'un bloc AR (Additionneur-Registre). Ils permettent de tenir compte des informations *a priori* des autres symboles reçus. Ainsi, les informations associées aux symboles candidats sont mises à jour à partir des informations provenant des autres symboles reçus et des informations extrinsèques provenant du décodeur lors de l'itération précédente. L'information souple associée aux symboles est ensuite envoyée au décodeur.

3.1 Bloc CIDE : Le bloc CIDE est détaillé dans la Figure 2. Son rôle est de calculer les distances Euclidiennes initiales des vecteurs RV à deux dimensions entre le symbole reçu et tous les symboles candidats d'une constellation. Par conséquent, ce bloc est l'élément critique au niveau de la complexité matérielle autour duquel le détecteur est construit. Il est utilisé uniquement lors de la première itération globale entre le détecteur et le décodeur. Le bloc CIDE se décompose en quatre sous-blocs. Deux instances des sous-blocs B1, B2 et B3 sont nécessaires pour obtenir une bonne efficacité en termes d'allocation des ressources. En effet, il est nécessaire de calculer en parallèle les parties réelles et imaginaires de points détectés. La partie réelle I est calculée grâce aux sous-blocs B1 et B2. En parallèle, la partie imaginaire Q est obtenue grâce aux sous-blocs B1' et B2'. Notez qu'il n'est

pas obligatoire de dupliquer le sous-bloc B4 car les parties réelles et imaginaires sont combinées pour trier les distances Euclidiennes. En bref, le premier sous-bloc, désigné par B1 ou B1', permet de détecter le point le plus proche de l'observation reçue. Le second B2 (resp. B2') génère les parties réelles et imaginaires associées aux points détectés. Ensuite, les sous-blocs B3 (resp. B3') calculent les valeurs de distance Euclidiennes pour un nombre limité de points autour du point détecté par le bloc B1 (resp. B1') et B2 (resp. B2'). Enfin, le sous-bloc B4 reçoit séquentiellement les parties réelles et imaginaires de points détectés et les trie selon leur fiabilité.

3.1.1 Sous-bloc B1 ou B1' : Une nouvelle formulation pour estimer le point le plus proche du symbole reçu a été proposée dans [1]. Selon cette formulation, l'estimation des points les plus proches est obtenue en calculant un vecteur appelé P^I (resp. P^Q) correspondant aux symboles candidats S^I (resp. S^Q). La relation entre P^I et S^I est donnée par l'équation suivante :

$$S^I = \left\{ \frac{1}{\sigma_s} (-\sqrt{M} + 1 + 2P^I) \mid P^I = 0, 1, \dots, \sqrt{M} - 1 \right\} \quad (1)$$

Il est à noter que les points de constellation MAQ-64 sont normalisés avec un facteur de normalisation σ_s égal à $\sqrt{42}$ pour s'assurer que l'énergie moyenne sur tous les symboles est unitaire. M est égal à 64 lorsque la constellation est une MAQ-64. Ensuite, la distance euclidienne entre l'observation reçue et une valeur possible des deux symboles candidats, en remplaçant S^I par P^I , peut être exprimée comme suit :

$$DE(S1, S2)^2 = \frac{(D_E^I)^2 + (D_E^Q)^2}{\sigma_w^2} \text{ tel que} \quad (2)$$

$$D_E^I = \underbrace{y_1^I - A}_{L} - \underbrace{\left(\widehat{h_{1,1}^I} \cdot P_1^I - \widehat{h_{1,1}^Q} \cdot P_1^Q + \widehat{h_{1,2}^I} \cdot P_2^I - \widehat{h_{1,2}^Q} \cdot P_2^Q \right)}_{L'} \quad (3)$$

$$D_E^Q = \underbrace{y_1^Q - B}_{L'} - \underbrace{\left(\widehat{h_{1,1}^Q} \cdot P_1^I + \widehat{h_{1,1}^I} \cdot P_1^Q + \widehat{h_{1,2}^Q} \cdot P_2^I + \widehat{h_{1,2}^I} \cdot P_2^Q \right)}_{L}$$

où σ_w^2 représente la variance du Bruit Blanc Additif Gaussien (BBAG) et y (resp. $\widehat{h_{j,i}}$) représente l'observation (resp. les coefficients) du canal. A, B et $\widehat{h_{j,i}}$ sont des termes qui doivent être estimés une fois par observation MIMO reçue.

$$A = \left[\frac{1}{\sigma_s} (-\sqrt{M} + 1) \right] \times \left(\widehat{h_{1,1}^I} - \widehat{h_{1,1}^Q} + \widehat{h_{1,2}^I} - \widehat{h_{1,2}^Q} \right) \quad (4)$$

$$B = \left[\frac{1}{\sigma_s} (-\sqrt{M} + 1) \right] \times \left(\widehat{h_{1,1}^Q} + \widehat{h_{1,1}^I} + \widehat{h_{1,2}^Q} + \widehat{h_{1,2}^I} \right)$$

et

$$\widehat{h_{j,i}} = \left(\frac{2 \cdot h_{j,i}}{\sigma_s} \right).$$

Basé sur l'équation (3), la différence entre les termes \widehat{y}_1^I et L (resp. \widehat{y}_1^Q and L') doit être minimisée pour détecter le point le plus proche de l'observation reçue. Ensuite, une approximation est appliquée pour obtenir l'égalité entre les deux termes en supposant que la différence entre le point reçu et le symbole candidat est négligeable. Par conséquent, pour chaque élément fixé de S_2 , le vecteur P_1^I peut être exprimé comme suit :

$$P_1^I = \underbrace{\widehat{h_{1,1}^I} \cdot \widehat{y}_1^I + \widehat{h_{1,1}^Q} \cdot \widehat{y}_1^Q}_{C} - \underbrace{\left(\widehat{h_{1,1}^I} \cdot \widehat{h_{1,2}^I} + \widehat{h_{1,1}^Q} \cdot \widehat{h_{1,2}^Q} \right)}_{D} \cdot P_2^I$$

$$+ \underbrace{\left(\widehat{h_{1,1}^I} \cdot \widehat{h_{1,2}^Q} - \widehat{h_{1,1}^Q} \cdot \widehat{h_{1,2}^I} \right)}_{E} \cdot P_2^Q \quad (5)$$

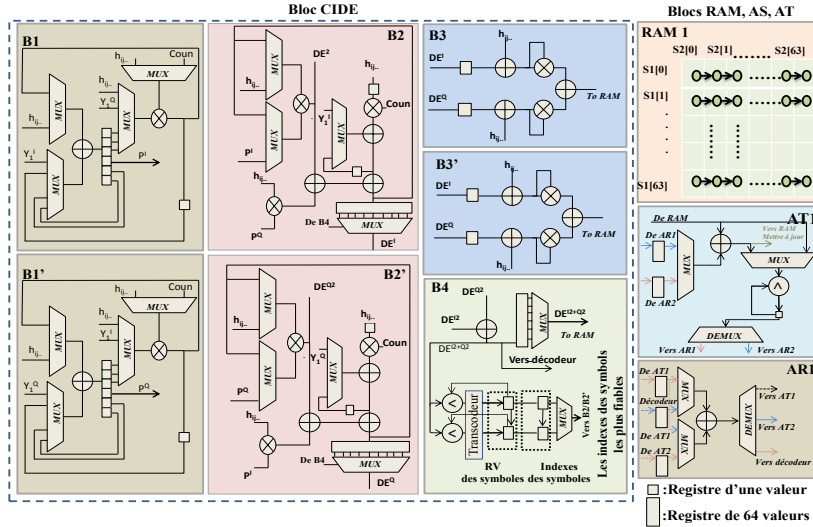


FIG. 2: Architecture détaillée du détecteur MIMO-BP itératif.

$$\text{avec } \widehat{h_{1,1}^I}' = \frac{\widehat{h_{1,1}^I}}{\widehat{h_{1,1}^I}^2 + \widehat{h_{1,1}^Q}^2} \text{ and } \widehat{h_{1,1}^Q}' = \frac{\widehat{h_{1,1}^Q}}{\widehat{h_{1,1}^I}^2 + \widehat{h_{1,1}^Q}^2}.$$

$h_{j,i}^I$ and $h_{j,i}^Q$ sont des parties réelles et imaginaires des coefficients de canal de la $i^{\text{ème}}$ antenne de transmission vers la $j^{\text{ème}}$ antenne de réception. Il convient de noter que les coefficients de canal sont mémorisés une fois par trame pour chaque réalisation de canal. Ensuite, C, D et E sont des termes à estimer une fois par observation MIMO reçue. Une seule expression est obtenue pour les 64 valeurs possibles du symbole S_2 . Pour calculer P_1^I , il n'y a que huit valeurs pour P_2^I et pour P_2^Q sur la dimension de S_2 . Ainsi, le calcul des termes C, D et E nécessite 6 multiplications réelles et 5 additions réelles. En outre, les calculs des termes $D \times P_2^I$ et $E \times P_2^Q$ nécessitent 16 multiplications réelles parce que P_2^I et $P_2^Q \in \{0; 1; \dots; 7\}$. En effet, le processus de détection doit couvrir les 64 valeurs différentes du symbole S_2 . Par conséquent, cette étape nécessite de détecter le vecteur P_1^I des symboles S_1 lorsque S_2 s'étend à tous les valeurs possibles $\{S_2 = 0; 1; \dots; 63\}$. 22 multiplications réelles et 67 additions réelles sont alors nécessaires.

De même, un sous-bloc B1' permet de calculer les termes C', D' et E' pour estimer la composante en quadrature P_1^Q . Cette étape nécessite également 22 multiplications réelles et 67 additions réelles pour estimer les symboles P_1^Q lorsque P_2 s'étend sur toutes les valeurs possibles. Au total, 44 multiplications réelles et 134 additions réelles sont nécessaires. Globalement, l'architecture du B1 (resp. B1') conçue est composée de quatre multiplexeurs, d'un addresseur réel, d'un multiplieur réel et de 7 registres d'une valeur.

3.1.2 Sous-blocs B2 et B2' : Le sous-bloc B2 (resp. B2') calcule la partie réelle (D_E^I (Resp. D_E^Q imaginaire)) du point détecté pour chaque valeur du vecteur P_1^I (resp. P_1^Q). Il est composé de quatre multiplexeurs, trois additionneurs réels, trois multiplieurs réels, un registre de 64 valeurs et 2 registres d'une valeur. Basé sur cette architecture, le calcul du terme D_E^I (resp. D_E^Q) dans (3) ne nécessite qu'un seul cycle d'horloge. Un deuxième cycle est nécessaire pour calculer le terme au carré $D_E^{I^2}$ (resp. $D_E^{Q^2}$) qui sont ensuite triés au sein du bloc B4. Globalement, 128 cycles sont nécessaires pour calculer les 64 distances Euclidiennes D_E et leur carré D_E^2 du symbole S_1 . Noter que l'exécution des sous-blocs B2/B2' peut démarrer après un délai de 10 cycles d'horloge au niveau B1/B1'.

3.1.3 Sous-blocs B3 et B3' : L'objectif du sous-bloc B3 (respectivement B3') est de calculer récursivement un nombre fixe de points voisins pour les distances Euclidiennes obtenues par le bloc B2/ B2'. Il fonctionne en deux phases. Dans la première phase, il calcule quatre points voisins pour toutes les distances Euclidiennes générées par B2/

B2'. Ils sont traités en parallèle au cours de deux cycles d'horloge. Par conséquent, le vecteur total est traité en 128 cycles d'horloge. Durant la deuxième phase, 20 points voisins sont calculés pour chacune des deux distances Euclidiennes les plus fiables parmi toutes les distances générées par B2/ B2' comme expliqué dans [10]. Par conséquent, cela coûte 20 cycles d'horloge. Pour ce faire, on ajoute à la partie réelle (ou imaginaire) du point généré par le bloc B2 (resp. B2') le terme $\{2 \times h_{j,i}^I\}$, (resp. $\{2 \times h_{j,i}^Q\}$). Ensuite, les deux termes obtenus sont élevés au carré et additionnés entre eux. Notez que le calcul des points voisins est traité en parallèle par deux sous-blocs pour diminuer la latence du bloc CIDE. L'architecture proposée est composée de trois additionneurs réels, de deux multiplieurs réels et de deux registres d'une valeur comme le montre la Figure 2. Puisque le calcul d'une distance Euclidienne nécessite un cycle d'horloge, l'exécution de sous-blocs B3 (resp. B3') peut commencer après un seul cycle d'exécution du sous-bloc B2 (resp. B2').

3.1.4 Sous-blocs B4 : Ce bloc génère tout d'abord les valeurs des distances Euclidiennes $D_E^{I^2}$ et $D_E^{Q^2}$ grâce à un addresseur réel. Ensuite, ces distances Euclidiennes sont triées en fonction de leur fiabilité. En fait, nous proposons de trier les deux distances Euclidiennes les plus fiables parmi les 64 provenant du bloc B2/B2'. Par conséquent, les indices de ces deux distances les plus fiables sont envoyés au bloc B2/B2' et puis au bloc B3/B3' pour calculer récursivement les 20 points voisins. L'architecture résultante contient un addresseur, deux comparateurs, un transcodeur, quatre registres d'une valeur, un registre de 64 valeurs et deux multiplexeurs comme montré dans la Figure 2. Les registres enregistrent les valeurs de distance Euclidiennes ainsi que leurs indices. Tous les deux cycles d'horloge, le sous-bloc B4 reçoit une nouvelle valeur de distance euclidienne. Grâce à deux comparateurs et un transcodeur, la comparaison s'opère en un cycle d'horloge. Le cas échéant, les valeurs des registres sont mises à jour. L'exécution du sous-bloc B4 peut démarrer après une latence de 11 cycles après que le sous-bloc B1/B1' démarre l'étape de calcul.

3.2 Bloc RAM : Un bloc RAM est réservé pour enregistrer les vecteurs RV bi-dimensionnels du symbole S_1 (resp. S_2) qui ont déjà été calculés par le bloc CIDE. Chaque vecteur de S_1 (resp. S_2) est composé de 64 valeurs associées au symbole S_2 (resp. S_1) comme représenté dans la Figure 2. Lors de la première inter-itération, ce bloc transmet directement les valeurs calculées par le bloc CIDE et triées par le bloc AT aux nœuds des symboles candidats S_1 , associé au bloc AR1, et S_2 , associé au bloc AR2. Dès la deuxième inter-itération, le bloc mémoire se met à jour via le bloc AT à partir des informations extrinsèques provenant du décodeur lors de l'itération précédente et mises à jour par le bloc AR.

3.3 Bloc AT (Additionneur-Trieur) : Le traitement appliqué dans ce bloc est basé sur le principe de la propagation de croyance détaillée dans [8]. Pendant la première inter-itération, les valeurs stockées dans la RAM sont triées au sein du bloc AT1 (resp. AT2) et directement envoyées aux blocs AR1 et AR2. En effet, aucune information extrinsèque n'est disponible à la sortie du décodeur. A partir de la seconde inter-itération, le bloc AT1 (resp. AT2) ajoute les informations extrinsèques provenant des blocs AR1 et AR2 aux valeurs mémorisées. Puis, les résultats sont triés pour trouver les valeurs les plus fiables et envoyés de nouveau aux blocs AR1 et AR2. Pour ce faire, un additionneur, un comparateur, un registre à une valeur, deux registres à 64 valeurs, deux multiplexeurs et un démultiplexeur sont ajoutés à l'architecture proposée.

3.4 Bloc AR (Additionneur-Registre) : Les calculs au sein d'un nœud de candidat sont effectués avec un additionneur, trois registres à 64 valeurs, deux multiplexeurs et un démultiplexeur comme montré dans la Figure 2. Tout d'abord, le bloc AR1 (resp. AR2) additionne les informations provenant du bloc AT1 et du bloc AT2. Puis, cette information est transmise au décodeur LPDC non binaire. Dans un second temps, l'information associée au bloc AT1 (resp. AT2) est mise à jour à partir des informations fournies par le décodeur LPDC non binaire et les informations provenant du bloc AT2 (resp. AT1). Enfin, ces informations sont enregistrées dans des registres à 64 valeurs au sein du bloc AT1 (resp. AT2).

4 Résultats de synthèse logique

Le Tableau 1 récapitule la latence et la complexité matérielle pour chaque bloc du détecteur MIMO-BP décrit dans la section précédente. L'architecture globale est composée de douze multiplieurs réels, dix-sept additionneurs réels, trois comparateurs, un transcodeur, vingt-sept registres à une valeur, huit registres à 64 valeurs, deux démultiplexeurs et vingt-deux multiplexeurs. Pendant la première itération entre le détecteur et le décodeur, les sous-blocs B1/B1', B2/B2' du bloc CIDE nécessitent une latence de 11 cycles d'horloge pour détecter les symboles qui seront envoyés au décodeur LDPC non binaire. En parallèle et durant 128 cycles d'horloge, les sous-blocs B3/B3' peuvent être exécutés pour trouver les points adjacents des distances Euclidiennes initiales. Pendant ce temps, B4 trie les deux distances Euclidiennes les plus fiables parmi les 64 distances Euclidiennes provenant du bloc B2/B2'. Au total 159 cycles d'horloge sont nécessaires pour que le tableau bi-dimensionnel du premier symbole candidat soit prêt comme indiqué dans le Tableau 1. Au cours des autres inter-itérations du détecteur souple, 92 cycles d'horloge sont nécessaires pour la mise à jour des informations mémorisées dans le bloc RAM.

Pour estimer la complexité matérielle de l'architecture proposée, la synthèse logique pour un circuit FPGA a été réalisée en utilisant l'environnement Xilinx ISE ciblant un Virtex-4 XC4VLX200. Ce dispositif est choisi pour une comparaison équitable avec une implantation de décodeur LDPC non binaire détaillée dans [11]. Le détecteur MIMO-BP nécessite des multiplieurs pouvant être implémentés dans des blocs DSP48. Nous constatons que l'architecture du détecteur MIMO-BP consomme 5013 *slices* (5802 *Flip-flops* et 4203 *LUT*), 2 blocs RAM et 5 blocs DSP48 comme indiqué dans le Tableau 2. La fréquence d'horloge maximale est de 58 MHz. Le chemin critique a été défini

TAB. 1: Complexité matérielle et nombre de cycles d'horloge

	Mult.	Add.	Comp.	Trans.	R1	R64	Cycl.
B1/B1'	2	2			14		79
B2/B2'	6	6			4	2	11+128
B3/B3'	4	6			4		11+128+20
B4		1	2	1	4	1	128
AT		1	1		1	2	92
AR		1				3	64
Total	12	17	3	1	27	8	159

TAB. 2: Résultats de synthèse pour une cible FPGA

FPGA XC4VLX200	MIMO-BP	LDPC non binaire in [11]
Nb. de Slices	5013	13385
Nb. de Flip Flops	5802	7521
Nb. de LUTs	4203	22981
Nb. de DSP48s	5	6
Nb. de RAMs	2	6

comme le temps d'exécution d'un multiplieur réel et de deux additionneurs réels. Dans ce contexte, la latence pour détecter une trame est de 49 μ s. On peut noter que le détecteur MIMO-BP est moins complexe en termes de ressources matérielles que le décodeur LDPC non binaire. De plus, nous pouvons constater que l'architecture proposée est 10 fois moins complexe au niveau des ressources matérielles (additionneur, multiplieur, ..) par comparaison au détecteur donné dans [6].

5 Conclusion

Dans cet article, nous avons détaillé une conception architecturale pour un détecteur itératif basé sur l'algorithme de propagation de croyance pour des systèmes MIMO. L'objectif de cette architecture est de limiter la complexité matérielle d'un tel détecteur. Pour ce faire, une nouvelle formulation de l'algorithme MIMO-BP a été considérée pour diminuer la complexité sans dégrader les performances. De plus, une organisation temporelle du détecteur itératif a été définie. Les résultats de synthèse montrent que les ressources logiques sont divisées par 10 par rapport à un détecteur MMSE [6]. A notre connaissance, il s'agit d'une première architecture dédiée à un détecteur MIMO-BP. Ces résultats sont une première étape vers une implantation matérielle efficace.

Références

- [1] A. Haroun, C. Abdel Nour, M. Arzel, and C. Jégo, "Low Complexity Soft Detection of QAM demapper for a MIMO System," *IEEE Communications letters*, vol. 99, pp. 1–4, 2016.
- [2] E. Telatar, "Capacity of Multi-Antenna Gaussian Channels," *Europ. Trans. on Telecom.*, vol. 10, pp. 585–595, 1999.
- [3] B. Steingrimsson, Z. Luo, and K. Wong, "Soft Quasi-Maximum-Likelihood Detection for Multiple-Antenna Wireless Channels," *IEEE Trans. Signal Process.*, vol. 51, pp. 2710–2719, 2003.
- [4] A. Burg and al., "VLSI Implementation of MIMO Detection Using the Sphere Decoding Algorithm," *IEEE J. Solid-State Circuits*, vol. 40, pp. 1566–1577, 2005.
- [5] M. Myllyla and al., "Complexity Analysis of MMSE Detector Architectures for MIMO OFDM Systems," *IEEE Conf. on Sign. Sys. and Computers*, pp. 75–81, 2005.
- [6] D. Karakolah, C. Jégo, C. Langlais, and M. Jezequel, "Design of an Iterative Receiver for Linearly Precoded MIMO Systems," *IEEE Inter. Sympo. on Cir. and Systems (ISCAS)*, pp. 597–600, 2009.
- [7] M. Kaynak, T. Duman, and E. Kurtas, "Belief Propagation over MIMO Frequency Selective Fading Channels," *IEEE Inter. Conf. on Net. and Services*, vol. 34, pp. 564–570, 2005.
- [8] A. Haroun, C. Abdel Nour, M. Arzel, and C. Jégo, "Symbol-based BP Detection for MIMO Systems associated with Non-Binary LDPC Codes," *IEEE Wir. Commun. Netw. Conf. (WCNC)*, pp. 212–217, 2014.
- [9] —, "Low-Complexity layered BP-based Detection and Decoding for a NB-LDPC Coded MIMO System," *IEEE Int. Conf. on Commun. (ICC)*, pp. 5107–5112, 2014.
- [10] —, "Low-Complexity LDPC-coded Iterative MIMO Receiver Based on Belief Propagation algorithm for Detection," *IEEE Int. Sympo. on Turbo Codes and Iterative Inf. Process. (ISTC)*, pp. 213–217, 2014.
- [11] E. Boutillon, Y. Eustache, P. Bomel, A. Haroune, and L. Conde-Canencia., "Performance measurement of davinci code by emulation," *DAVINCI Deliverable D6.2.3*, 2010.