



Comité National Français de Radioélectricité Scientifique

Section française de l'

Union Radio Scientifique Internationale

Siège social : Académie des Sciences, Quai de Conti – Paris

JOURNÉES SCIENTIFIQUES DU CNFRS

" VERS DES RADIOCOMMUNICATIONS RECONFIGURABLES ET COGNITIVES "

PARIS, LES 28 ET 29 MARS 2006

Architecture reconfigurable pour les systèmes MIMO à base d'opérateurs CORDIC

Hongzhi Wang*, Pierre Leray*, and Jacques Palicot*

* Supélec – Campus de Rennes, Avenue de la Boulaie, 35511 Cesson-Sévigné Cedex, France

E-mail: {[Hongzhi.wang](mailto:Hongzhi.wang@supelec.fr), [pierre.leray](mailto:pierre.leray@supelec.fr), [jacques.palicot](mailto:jacques.palicot@supelec.fr)}@supelec.fr

Résumé

Le système MIMO est une technologie attractive pour les systèmes 3G/4G sans fil. Dans cet article nous proposons la réalisation sur FPGA d'un algorithme MIMO « V-BLAST Square Root » basée sur l'utilisation d'un nombre variable d'opérateurs CORDIC. L'opérateur CORDIC convient bien pour l'implémentation car il s'appuie seulement sur de simples techniques d'additions et de décalages entre vecteurs. Cette architecture de l'algorithme square root est reconfigurable pour s'adapter à différents nombres d'antennes et différents débits. L'architecture proposée peut atteindre un débit de 600Mbit/s dans un circuit FPGA VirtexII de Xilinx pour le système MIMO avec une modulation QPSK.

Mots clés : MIMO, V-BLAST Square Root, CORDIC, Architecture configurable

1. Introduction

Avec l'intégration de l'Internet et de nouvelles applications multimédia dans les systèmes de communications sans fil, la demande en terme de débit ne cesse d'augmenter. Plusieurs techniques ont été développées pour répondre à ce besoin. La technique MIMO découverte en 1997 par les chercheurs de Bell Labs reste la plus prometteuse, elle peut augmenter d'une manière substantielle l'efficacité spectrale. Par exemple, les standards de réseaux locaux sans fil à haut débit, tels que IEEE 802.11n et Hiperlan2, vont adopter les systèmes MIMO dans leurs futures normes. Le débit des transmissions de HSDPA (High Speed Downlink Packet Access) peut atteindre 21.6Mb/s en utilisant les systèmes MIMO. Cette technique a reçu beaucoup d'intérêt ces dernières années et a donné lieu à de nombreux travaux. Parmi les difficultés engendrées par cette technique, l'implémentation des algorithmes de démodulation des signaux MIMO est un sujet d'actualité. Afin de pouvoir gérer la multiplicité des standards de communication, le système MIMO doit supporter différents types de modulations et de propagation. Ce sont les raisons pour lesquelles une architecture reconfigurable trouve tout son intérêt dans les systèmes MIMO.

Les algorithmes MIMO sont généralement implémentés sur DSP (Digital Signal Processor), il est donc difficile d'atteindre des performances élevées en débit. La solution traditionnelle ASIC, quand à elle, ne supporte pas les diversités des futurs systèmes. Il est donc naturel d'utiliser des circuits programmables de type FPGA pour répondre à notre problème de reconfiguration. Ces circuits sont largement utilisés en Traitement du Signal grâce à la possibilité de reconfiguration et le haut niveau de parallélisme. Concernant l'implémentation d'algorithmes de réception MIMO, il existe peu de travaux publiés actuellement sur ce sujet.

Une implémentation de l'algorithme « Square Root » a été réalisée par Z.Guo dans [1], mais cette réalisation ne remplit pas les contraintes de notre problème. En effet, la réalisation décrite dans [1] n'est pas reconfigurable car elle est implantée sur un ASIC.

C'est pour y répondre que nous proposons dans cet article une implémentation à base d'opérateurs CORDIC sur un circuit reprogrammable FPGA.

La suite de l'article est structurée de la manière suivante : la section 2 présente brièvement le système MIMO. La section 3 décrit les différents algorithmes de réception et illustre notre choix de l'algorithme « V-BLAST Square Root » en terme de compromis Performances/Complexité. La section suivante détaille l'algorithme « V-BLAST Square Root », que nous avons retenu. La section 5 analyse l'architecture fonctionnelle de l'algorithme, c'est aussi dans cette section que nous décrivons l'opérateur CORDIC . La section 6 présente l'architecture reconfigurable, nous analysons l'implémentation par des exemples concrets, puis nous présentons les résultats de la synthèse des différentes architectures sur un FPGA. Enfin, nous insistons sur l'aspect dynamique (temps réel) de la reconfiguration.

2 Présentation du système MIMO

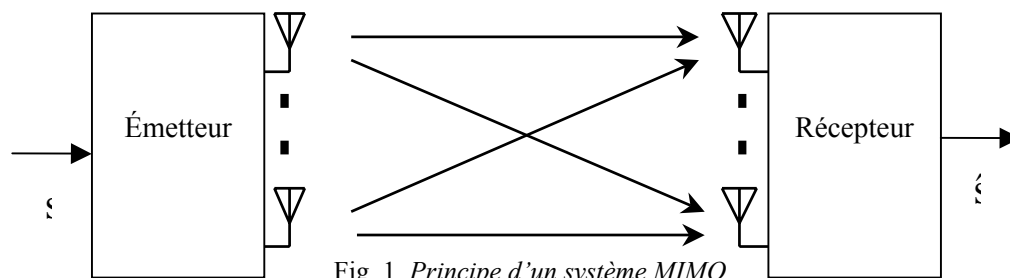


Fig. 1. Principe d'un système MIMO

L'apparition des systèmes MIMO a été motivée par le besoin accru en terme de débit par l'arrivée de nouveaux services tels que l'accès à Internet et la transmission d'images via les systèmes de communications sans fil, ainsi que la saturation des ressources en canaux de transmission, en particulier dans la bande de la téléphonie mobile. Les systèmes MIMO consistent à utiliser plusieurs antennes à l'émission et à la réception. En bénéficiant des traitements spatio-temporels associés, ces systèmes ont montré une augmentation considérable de l'efficacité spectrale (proportionnelle au nombre d'antennes utilisées) [2].

Partant du point de vue de la théorie de l'information, deux chercheurs des laboratoires Bell, Foschini [4] et Teletar [5] ont indépendamment montré que la capacité des systèmes multi-antennes augmentait linéairement avec le nombre d'antennes. Ces découvertes sont à l'origine des systèmes MIMO (voir figure 1) qui visent essentiellement à résoudre les problèmes d'encombrement et de limitation de capacité des réseaux sans fil large bande. L'idée de base dans les systèmes MIMO est le traitement spatio-temporel, où le temps (dimension naturelle) est complété par une dimension spatiale inhérente à l'utilisation de plusieurs antennes. Un tel système peut être vu comme l'extension des antennes intelligentes. La propriété clef d'un système MIMO est sa capacité à tourner la propagation multi-trajets (traditionnellement un inconvénient) en un avantage, en d'autres termes les systèmes MIMO exploitent les trajets multiples plutôt que de les supprimer.

Les techniques conventionnelles utilisées à la réception pour annuler la distorsion apportée par le canal MIMO nécessitent souvent, soit la connaissance du canal, soit l'utilisation d'une séquence de symboles connue au niveau du récepteur. Dans la pratique le canal est inconnu, donc une estimation de ce dernier est nécessaire. Souvent l'estimation du canal est basée sur l'utilisation des séquences d'apprentissage multiplexées avec les données utiles, ce qui diminue bien évidemment le débit utile. Pour des canaux invariants dans le temps, la perte n'est pas significative car un seul cycle d'apprentissage est nécessaire.

On distingue principalement deux types de techniques pour la transmission sur les systèmes MIMO la première est basée sur les codes spatio-temporels, l'autre sur le multiplexage spatial [2],[3].

2.1 Système MIMO à base de codes spatio-temporels

Afin d'améliorer la qualité de la transmission, Alamouti [6] et Tarokh [7] ont conçu des systèmes basés essentiellement sur la diversité, proposant un codage et un étiquetage conjoints. Ce *codage spatio-temporel*

(CST) permet également des communications plus sûres, il consiste à ajouter de la redondance aux données binaires émises afin d'augmenter la diversité spatiale et éviter les évanouissements propres au canal MIMO. Pour plus de détails sur les codes spatio-temporels se référer à [2], [1], [6], [7].

2.2 Système MIMO à base de Multiplexage spatial

Le multiplexage spatial, ou V-BLAST (Vertical Bell Labs Layered Space-Time) [8], [9] peut être vu comme une classe spéciale des codes spatio-temporels, son principe consiste à diviser le flux de données à l'entrée en plusieurs sous-flux, ces derniers sont transmis sur des antennes différentes. Cette architecture vise principalement à augmenter la capacité du système. Une description simple est donnée sur la figure 1.

Les antennes d'émission transmettent, chacune un symbole différent, indépendant de celui des autres antennes, mais en utilisant la même modulation et la même fréquence porteuse. La bande passante utilisée reste identique à celle d'un système classique, mais comme plusieurs symboles sont émis, l'efficacité spectrale augmente.

Nous nous intéresserons dans la suite de cet article uniquement au multiplexage spatial et plus particulièrement à l'architecture V-BLAST.

Nous utilisons dans cette étude un modèle mono-utilisateur dans un canal non sélectif en fréquence. De plus, le modèle est sans mémoire. C'est le modèle de canal MIMO le plus utilisé, il consiste en une matrice H , dont chaque coefficient complexe h_{ij} représente la fonction de transfert entre la i^e antenne réceptrice et la j^e antenne émettrice. Pour un système avec M antennes à l'émission et N antennes en réception, le vecteur reçu r peut s'écrire :

$$r = Hs + v \quad (1)$$

Dans cette équation, $s = [s_1 \dots s_M]^T$ est le vecteur de symboles émis, H est la matrice de canal de dimension $M \times N$ et $v = [v_1 \dots v_N]^T$ est le vecteur de bruit additif gaussien en réception. Nous supposons que $E[ss^*] = I_N$, $E[vv^*] = R$ et $E[sv^*] = 0$.

Les éléments de H ont une phase uniformément distribuée et une amplitude qui suit une loi de Rayleigh. Ce modèle est typique d'un environnement avec de nombreux échos et un écart suffisant entre les antennes. Nous supposons aussi que le canal reste constant durant la transmission d'un bloc de données et que le récepteur connaît parfaitement la matrice de canal H . Cette connaissance peut s'obtenir soit par des symboles d'apprentissage soit par estimation aveugle du canal.

3. Algorithmes de réception

Il existe de nombreux algorithmes de réception envisageables pour récupérer les symboles dans les systèmes MIMO. Les moins complexes sont les récepteurs linéaires basés sur le critère de forçage à zéro (ZF) ou la minimisation de l'erreur quadratique moyenne (MMSE). L'algorithme de réception proposé pour le système V-BLAST [10] essaie d'éliminer successivement les interférences dues aux émetteurs autres que celui dont on estime les symboles. Ils sont sous-optimaux en termes de TEB. Le détecteur optimal est basé sur le maximum de vraisemblance (ML) qui demande une importante charge de calcul lorsque le nombre d'antennes et la taille de la constellation sont grands. Il existe de nombreux algorithmes sous-optimaux basés sur le ML (par exemple décodage sphère).

L'étude de nouveaux algorithmes performants ne relève pas de ce travail. Il existe de très nombreux algorithmes et notre objectif n'est pas d'en faire une recherche exhaustive. Nous souhaitons seulement dans cette section identifier, parmi les algorithmes les plus connus, les meilleurs algorithmes en terme de complexité/performance que nous pourrions implanter sur FPGA

➤ Maximum de Vraisemblance

Le récepteur du maximum de vraisemblance (ML) offre les meilleures performances en *Taux d'Erreur Binaire (TEB)*. En effet, il est optimal si les vecteurs émis s sont équiprobables, ce qui est le cas puisque les symboles s_i sont équiprobables et que les N voies émises en parallèle sont indépendantes. Il s'exprime classiquement de la façon suivante :

$$S = \arg \min ||r - Hs||^2$$

Toutefois sa charge de calcul devient rapidement très importante car elle augmente exponentiellement avec le nombre d'antennes à l'émission M et linéairement avec le nombre d'antennes en réception N . Le problème est NP complet et il n'existe pas d'algorithme permettant de trouver une solution optimale en temps

polynomial. Il est donc naturel de rechercher des algorithmes plus simples ayant des performances proches. Dans cette catégorie nous trouvons l'algorithme du décodage par sphères (SD) [12], [13], ainsi que l'algorithme « Branch & Bound ».

Nous ne listerons pas ici l'ensemble des récepteurs sous-optimaux, la littérature étant très importante sur ce sujet. Nous nous limitons aux plus connus.

➤ **« Branch & Bound »**

Le principe consiste à construire un arbre dont chaque branche devient un sous-problème plus simple à résoudre [15], [16]. La conservation d'une branche dans l'algorithme est régit par 2 bornes. Si la borne calculée dans la branche considérée est supérieure à celle de la branche conservée précédemment alors cette branche est abandonnée.

➤ **Décodage par sphères**

Il est cependant possible d'approcher les performances du maximum de vraisemblance en gardant une complexité raisonnable. [12], [13].

Le principe de cet algorithme est de se placer au niveau du signal reçu et de chercher le point du réseau le plus près à l'intérieur d'une sphère de rayon C . Cela diminue fortement le domaine de recherche par maximum de vraisemblance, puisque seuls les points du réseau situés à une distance inférieure à C du signal reçu sont considérés lors de la minimisation de la métrique. Le choix de C est donc crucial pour la vitesse de convergence de l'algorithme et pour la précision des résultats.

➤ **Récepteur linéaire du Forçage à Zéro (ZF)**

Ce récepteur est le plus simple et aussi le moins performant. Il cherche à annuler les contributions des autres émetteurs sur chaque symbole. Ceci revient à inverser la matrice de transfert du canal :

$$S = (H^*H)^{-1} H^* r$$

Lorsque H est mal conditionnée, son inversion multiplie le bruit et dégrade alors sérieusement les performances à faible SNR.

➤ **Récepteur linéaire MMSE**

Ce critère minimise l'erreur moyenne quadratique due à la fois au bruit et aux interférences entre symboles, contrairement au récepteur ZF qui ne s'occupe que des interférences entre symboles. Son expression est:

$$S = (H^*H + \sigma^2 I)^{-1} H^* r, \quad \text{avec } \sigma^2 \text{ la variance du bruit par antenne}$$

Ce récepteur résiste mieux au bruit que le récepteur ZF. À haut SNR, le récepteur MMSE tend vers le récepteur ZF car σ^2 tend vers 0.

➤ **Récepteur à retour de décision V-BLAST**

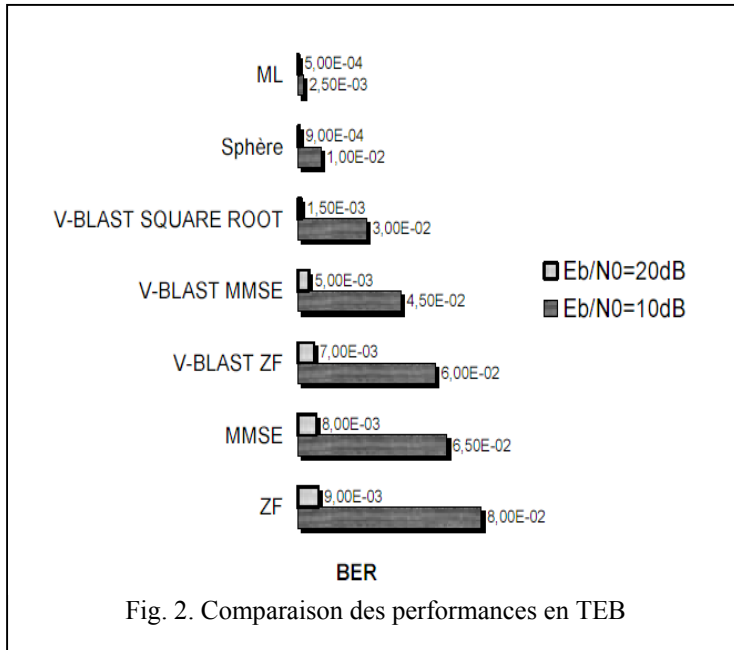
Le principe de l'algorithme du récepteur V-BLAST a été présenté dans [8], [10]. Il s'agit d'un égaliseur à retour de décision, adapté à la structure des systèmes MIMO. L'égaliseur peut utiliser soit le critère du forçage à zéro, soit le critère du MMSE. Son principe est le suivant : le symbole de l'émetteur le plus favorisé (possédant le meilleur TEB suivant le critère considéré) est démodulé en premier. Sa contribution au vecteur reçu r est ensuite annulée, ce qui augmente le SNR sur les autres émetteurs (à chaque bonne décision). Cette étape est répétée jusqu'au dernier émetteur, le moins favorisé. Ce récepteur est aussi noté dans la littérature OSIC (Ordered Successive Interference Cancellation). Comme pour tous les égaliseurs à retour de décision, son principal inconvénient est la propagation des erreurs. Une fois qu'une mauvaise décision a été prise sur la valeur d'un symbole, une mauvaise contribution est retirée au vecteur r , ce qui entraîne que les symboles suivants seront presque certainement mal estimés. C'est la raison pour laquelle l'ordonnancement est utilisé pour minimiser la propagation des erreurs.

De nombreux auteurs ont proposé des améliorations de l'algorithme, les uns sont basés sur une décomposition QR itérative comme M.DAMEN [12]. Les autres utilisent des méthodes de substitution comme W.Zha et S.D.Blostein [14]. Mais le TEB est dégradé.

➤ **« Square Root Algorithm »**

B.Hassibi présente en 2000 l'algorithme « Square Root » [19] qui permet de diminuer la complexité de $O(M^4)$ à $O(M^3)$ sans dégrader le TEB, dans le cas où $M=N$. Parmi les différents algorithmes de détection d'un signal MIMO l'algorithme « V-BLAST Square Root » réalise un bon compromis entre les performances attendues et la faible complexité de l'algorithme.

➤ **Comparaison succincte performance/complexité**



Dans cette section nous avons présenté les divers algorithmes de détection classique, tels que les structures linéaires et non linéaires ou les structures à maximum de vraisemblance.

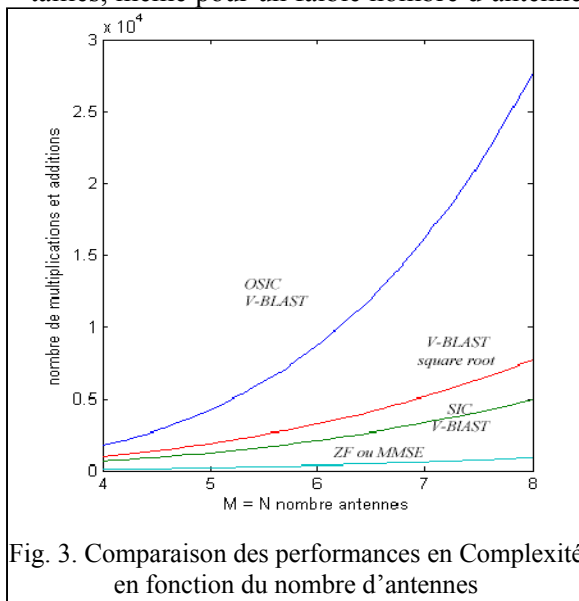
La figure 2 montre une évaluation du TEB pour les récepteurs que nous venons de présenter. Ces résultats sont obtenus dans un système MIMO muni de 2x2 antennes avec une modulation QPSK.

Le tableau 1 présente pour un système MIMO 4x4 la complexité en terme de nombre d'opérations. Sur la figure 3 nous avons représenté cette même complexité d'implémentation exprimée en nombre d'opérations (multiplications et additions) par symbole en fonction du nombre d'antennes.

Algorithmes	Nombre d'opération/symbole	Antennes 4x4
ZF	$5N^3/3-2N/3$	103
MMSE	$3MN^2+N^3/3+MN/2-N^2/2$	107
SIC V-BLAST	$2M^2N+2MN^2+11M^3$	960
OSIC V-BLAST	$M^2N^2+2NM^3+15/4M^4$	1728
V-BLAST Square Root	$5/3M^3+8M^2N+2MN^2+2MN-M^2$	763
Décodage Sphère		6822

Tableau.1 Comparaison de complexité des divers algorithmes

Il est important de noter que pour l'algorithme du maximum de vraisemblance, la taille de la constellation joue un rôle important, parce que le récepteur doit calculer les distances pour un ensemble de vecteurs possibles dans la constellation. Le temps de calcul devient vite excessif pour les constellations de grandes tailles, même pour un faible nombre d'antennes. Par exemple, dans le tableau 1, un système QAM 16 avec 4 antennes nous donne 16^4 possibilités soit 65536. Donc cet algorithme n'est pas retenu. Les performances des architectures de types linéaires semblent correctes mais pas suffisantes au regard des structures non linéaires.



A la lecture des figures précédentes, nous avons sélectionné l'algorithme « Square Root » comme un des meilleurs compromis Complexité/Performance. Afin de bien comprendre l'implantation sur FPGA que nous proposons, nous décrivons d'abord, en détail, cet algorithme dans la section suivante.

4. Description de l'algorithme « V-BLAST Square Root »

4.1 Principe du récepteur V-BLAST

Le principe de l'algorithme du récepteur V-BLAST a été présenté dans [10]. Il s'agit d'un égaliseur à retour de décision, adapté à la structure des systèmes MIMO. L'égaliseur peut utiliser soit le critère du forçage à zéro, soit le critère du MMSE. Son principe est le suivant : le symbole de l'émetteur le plus favorisé (possédant le meilleur TEB suivant le critère considéré) est démodulé en premier. Sa contribution au vecteur reçu r est ensuite annulée, ce qui augmente le SNR sur les autres émetteurs (à chaque bonne décision). Cette étape est répétée jusqu'au dernier émetteur, le moins favorisé.

4.2 Algorithme « V-BLAST Square Root »

L'algorithme « V-BLAST Square Root » est proposé par B.Hassibi [19] ; il évite le calcul répété de la pseudo-inverse de la matrice du canal et l'inversion matricielle en utilisant les transformations unitaires. Elle permet de diminuer la charge de calcul de $O(M^4)$ à $O(M^3)$ sans dégrader le TEB. Pour cela B.Hassibi utilise une relation de récurrence connue en filtrage adaptatif *RLS* (Recursive least square). Il démontre que si on applique une séquence de rotation de Givens à la relation de récurrence, on obtient $P^{1/2}$ après i itérations. Mais il reste alors à calculer Q_α . La meilleure solution sera d'appliquer une relation qui fournit $P^{1/2}$ et Q_α en même temps. C'est la raison pour laquelle B.Hassibi met en évidence une nouvelle relation de récurrence à partir du bloc matrice précédent auquel il ajoute un bloc vecteur.

L'algorithme est résumé ci-dessous :

Etape 1 : Calcul de $P^{1/2}$ et de Q_α ,

- Initialisation

$$\left[\begin{array}{cc} 1 & H_i^{1 \times M} P_{i-1}^{1/2 M \times M} \\ 0^{M \times 1} & P_0^{1/2 M \times M} \\ -e_i^{N \times 1} & Q_0^{N \times M} \end{array} \right] \text{ avec } P_0^{1/2 M \times M} = \beta I, Q_0 = 0^{N \times M}$$

- pour $i = 1$ à N

$$\left[\begin{array}{cc} 1 & H_i^{1 \times M} P_{i-1}^{1/2 M \times M} \\ 0^{M \times 1} & P_{i-1}^{1/2 M \times M} \\ -e_i^{N \times 1} & Q_{i-1}^{N \times M} \end{array} \right] \Theta_i = \left[\begin{array}{c} \times \quad 0^{1 \times M} \\ \times \quad P_i^{1/2 M \times M} \\ \times \quad Q_i^{N \times M} \end{array} \right] \quad (2)$$

Fin

Après N itérations on obtient $P_N^{1/2} = P^{1/2}$, $Q_N = Q_\alpha$,

Q_i représentant la $i^{\text{ème}}$ itération avec Q_0 initialisé à 0,

e_i indique la $i^{\text{ème}}$ colonne de la matrice identité,

Θ_i correspond à une transformation unitaire, qui transforme la matrice de l'équation (2) en une matrice triangulaire inférieure. Les méthodes pour trouver ce type de transformation unitaires sont connues [19]

Itération : Pour $i=0$ à $M-1$

Etape 2 : Déterminer la norme minimale des lignes de $P^{1/2}$ et permuter cette ligne pour qu'elle soit la dernière. Permuter également l'indice du symbole reçu en rapport. Effectuer une transformation unitaire qui satisfasse la relation (3).

$$P^{1/2}\Sigma = \begin{bmatrix} P^{1/2}_{i-1} & P_i^{i-1} \\ 0 & p_i \end{bmatrix}, \quad (3)$$

Comme pour l'équation (2), Σ est une transformation unitaire qui transforme $P^{1/2}$ en matrice triangulaire supérieure.

Etape 3 : Actualiser Q_α à partir de $Q_\alpha\Sigma$

Etape 4 : Calculer le *nulling vector* $w_i = p_i q_{\alpha,i}^*$ (4)

Etape 5 : Calculer \hat{y}_i $\hat{y}_i = w_i r_i$ (5)

Etape 6 : Calculer le symbole estimé $\hat{s}_i = \text{décision}(\hat{y}_i)$ (6)

Etape 7 : Annuler les interférences $r_{i-1} = r_i - h_i \hat{s}_i$ (7)

Fin

5. Description Fonctionnelle de l'algorithme « V-BLAST Square Root »

L'architecture du décodeur MIMO « V-BLAST square-root » est illustrée par la figure 4. Elle est composée de 6 modules de traitement.

Les entrées sont constituées des messages reçus r et des valeurs de la matrice de canal H . Les trois premiers modules (M_1, M_2, M_3) effectuent la décomposition de la matrice H en utilisant des transformations unitaires. Ces modules calculent les grandeurs $P^{1/2}$ (Etape 1), Q_α (Etape 1), p_i (Etape 2) et $q_{\alpha,i}^*$ (Etape 3). Le module

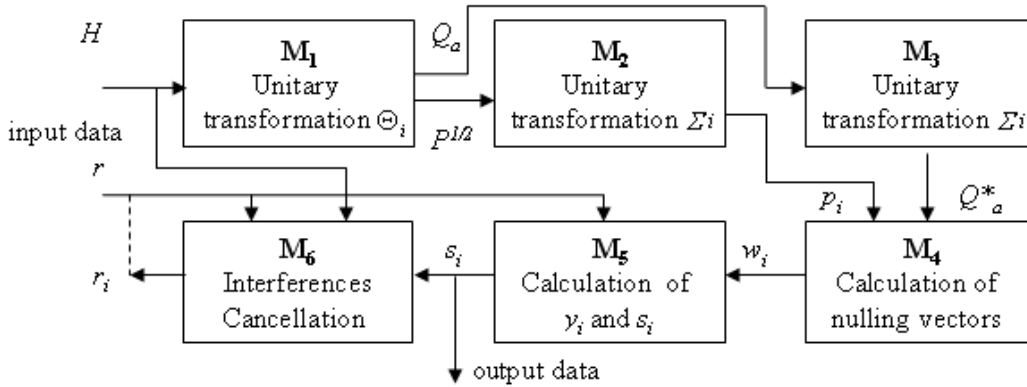


Fig. 4 Architecture fonctionnelle du décodeur MIMO « V-Blast square-root »

suivant M_4 détermine l'ordre optimal de décodage et calcule le vecteur w_i (*nulling vector*) (Etape 4). Le module M_5 décide du vecteur symbole transmis (Etape 6) et le dernier module M_6 réalise l'annulation d'interférences entre les symboles (Etape 7).

Les trois modules (M_1, M_2, M_3) présentent une architecture similaire. Ces modules sont conçus en utilisant des opérateurs CORDIC (voir l'exemple ci-dessous pour les calculs de $P^{1/2}$ et de Q_α dans le module M_1). Au lieu d'exécuter la décomposition QR par un réseau triangulaire, nous utilisons une séquence de rotation de Givens basée sur CORDIC.

Les trois derniers modules (M_4, M_5, M_6) sont basés sur les processeurs élémentaires (PE) constitués d'un multiplieur-accumulateur, un soustracteur et un buffer.

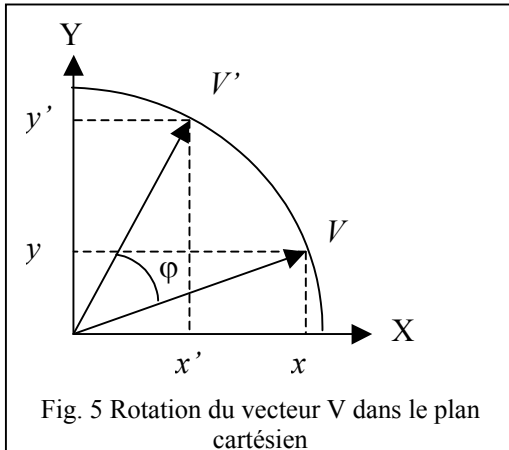
5.1 Opérateur CORDIC

Les séquences de rotation de Givens peuvent être implémentées sur une architecture composée d'un simple processeur. Cette architecture basée sur l'opérateur CORDIC (COordinate Rotation Digital Computing) est proposée par Rader [20]. C'est une méthode simple et efficace pour le calcul d'une gamme de fonctions complexes qui s'appuie sur une technique d'additions et de décalage entre vecteurs. L'algorithme calcule par approximation la plupart des fonctions basées sur la trigonométrie. Il exécute des rotations sans utiliser d'opérations de multiplications.

5.1.1 Principe de l'algorithme CORDIC

La résolution de fonctions telles que sinus ou cosinus par l'algorithme de CORDIC s'appuie sur une méthode de rotation de vecteur dans le plan cartésien. Supposons la rotation du vecteur $V(x,y)$ d'un angle φ tel qu'illustré à la figure 5.

Les coordonnées du vecteur V' sont exprimées selon les équations :



$$\begin{aligned} x' &= x \cos(\varphi) - y \sin(\varphi) \\ y' &= y \cos(\varphi) + x \sin(\varphi) \end{aligned}$$

Si on restreint l'angle de rotation à $\tan^{-1}(\pm 2^{-i})$ où $i = 0, 1, 2, 3, \dots$, on obtient alors φ par une série de rotations élémentaires successives de l'ordre de:

$$a_{i+1} = a_i - d_i \cdot \tan^{-1}(2^{-i}) \quad \text{où } d_i = \pm 1$$

L'indice d_i indique le sens de rotation de l'angle pour chaque itération. Cet indice est déterminé à chaque itération selon le résultat d'une comparaison. Le mode d'utilisation de l'indice sera expliqué un peu plus loin. Chaque vecteur itératif $V_{i+1}(x_{i+1}, y_{i+1})$ est représenté par :

$$\begin{aligned} x_{i+1} &= K_i [x_i - y_i \cdot d_i \cdot 2^{-i}] \\ y_{i+1} &= K_i [y_i + x_i \times d_i \times 2^{-i}] \quad \text{où } K_i = \cos(\tan^{-1} 2^{-i}) = (1 + 2^{-2i})^{-1/2} \end{aligned}$$

On montre que, pour un nombre croissant d'itérations, le produit des facteurs K_i tend vers une constante égale à 0.6073. Ainsi le gain de l'algorithme de rotation est approximativement 1.647. Le gain précis est fonction du nombre n d'itérations que l'on peut exprimer par :

$$A_n = \prod_n (1 + 2^{-2i})^{1/2}$$

Puisque pour un nombre relativement élevé d'itérations, le produit tend vers un résultat constant, il nous est possible d'appliquer ce dernier plus loin dans l'algorithme. Ainsi on obtient un ensemble d'équations simplifiées et spécifiques au calcul des opérations mathématiques recherchées :

$$\begin{aligned} x_{i+1} &= x_i - d_i \cdot dy_i \\ y_{i+1} &= y_i + d_i \cdot dx_i \\ a_{i+1} &= a_i - d_i \cdot da_i \\ \text{où} \\ dy_i &= y_i \times 2^{-i} \\ dx_i &= x_i \times 2^{-i} \\ da_i &= \tan^{-1}(2^{-i}) \\ d_i &= \pm 1 \text{ selon le signe de } a_i \text{ ou } y_i \end{aligned}$$

Ces équations servent donc au calcul d'opérations. Le principe général de l'algorithme de CORDIC consiste à faire pivoter dans le sens approprié le vecteur de rotation par un angle de plus en plus petit jusqu'à ce que l'angle a ou les valeurs x et y soient approximativement égales à 0.

5.1.2 Architecture de l'opérateur CORDIC

La structure itérative de l'algorithme de CORDIC permet une implantation utilisant une structure pipeline (Fig.6), limitant ainsi la longueur des chemins critiques de façon à accroître la vitesse de fonctionnement.

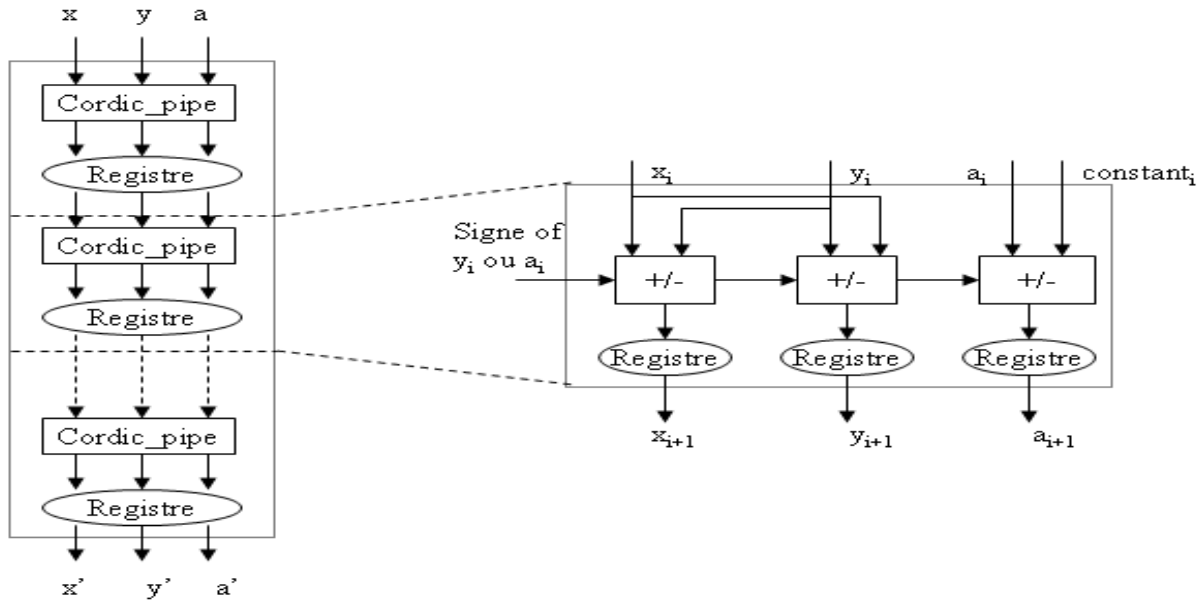


Fig. 6 Architecture pipeline de l'opérateur CORDIC

5.2 Implémentation parallèle de CORDIC (Un exemple : calcul de $P^{1/2}$ et de Q_α)

Les calculs de $P^{1/2}$ et de Q_α sont effectués dans le premier module. C'est la première étape pour décomposer la matrice du canal H . Pour des raisons de stabilité numérique et de facilité d'implémentation nous utilisons une séquence de rotations de Givens basées sur des opérateurs CORDIC.

$$\begin{bmatrix} 1 & H_i^{1 \times M} P_{i-1}^{1/2 M \times M} \\ 0^{M \times 1} & P_{i-1}^{1/2 M \times M} \\ -e_i^{N \times 1} & Q_{i-1}^{N \times M} \end{bmatrix} \Theta_i = \begin{bmatrix} \times & 0^{k \times M} \\ \times & P_i^{1/2 M \times M} \\ \times & Q_i^{N \times M} \end{bmatrix} \quad \text{avec } P_0^{1/2 M \times M} = \beta I, Q_0 = 0^{N \times M} \quad (2)$$

Considérons maintenant, dans le reste de cet article, un modèle MIMO 2×2 antennes. Alors le premier terme de la seconde colonne de l'équation (2) $H_i^{1 \times M} P_{i-1}^{1/2 M \times M}$ est représentée ci-dessous :

$$H_i^{1 \times M} P_{i-1}^{1/2 M \times M} = \begin{bmatrix} HP_{11} & HP_{12} \\ HP_{21} & HP_{22} \end{bmatrix}$$

Les angles de rotation de Givens sont définis à partir de cette matrice comme ci-dessous :

$$\begin{aligned} \theta_1 &= \tan^{-1} \frac{\text{Im}(HP_{11})}{\text{Re}(HP_{11})}, & \theta_2 &= \tan^{-1} \frac{\text{Im}(HP_{12})}{\text{Re}(HP_{12})}, \\ \varnothing_1 &= \tan^{-1} \frac{\text{Module}(HP_{11})}{\text{Module}(HP_{12})}, & \varnothing_2 &= \tan^{-1} \frac{1}{\text{Module}(HP_1)}, \\ \text{Avec } \text{Module}(HP_1) &= \sqrt{\text{Module}(HP_{11})^2 + \text{Module}(HP_{12})^2} \\ \theta_3 &= \tan^{-1} \frac{\text{Im}(HP_{21})}{\text{Re}(HP_{21})}, & \theta_4 &= \tan^{-1} \frac{\text{Im}(HP_{22})}{\text{Re}(HP_{22})}, \end{aligned}$$

$$\theta_3 = \tan^{-1} \frac{\text{Module}(HP_{21})}{\text{Module}(HP_{22})}, \theta_4 = \tan^{-1} \frac{1}{\text{Module}(HP_2)},$$

Avec $\text{Module}(HP_2) = \sqrt{\text{Module}(HP_{21})^2 + \text{Module}(HP_{22})^2}$

Une fois les angles pré-calculés, nous pouvons exécuter les transformations unitaires à l'aide des opérateurs CORDIC. Le calcul de l'équation (2) peut alors être décomposé en 29 rotations.

La figure 7 illustre une implémentation totalement parallèle des calculs utilisant 29 opérateurs CORDIC. La structure parallèle est conçue pour obtenir un très haut débit, mais elle est coûteuse en ressources matérielles. Pour les applications à plus faible débit, toute la puissance de calcul disponible n'est pas utilisée. Pour tendre

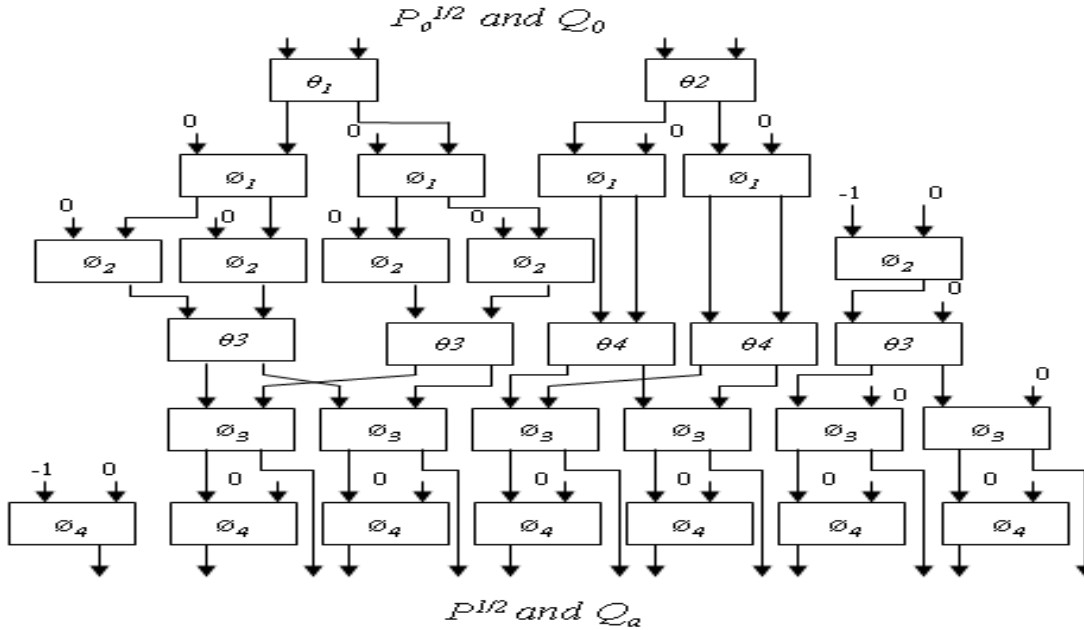


Fig. 7 29 opérateurs CORDIC pour calculer $P^{1/2}$ et Q_a

vers une implémentation optimale du décodeur MIMO, nous proposons une architecture reconfigurable dont la structure est fonction des caractéristiques de la transmission.

6. Architecture reconfigurable

Nous proposons une structure de traitement où le nombre d'opérateurs CORDIC s'adapte aux caractéristiques de débit et de nombre d'antennes pour une implémentation optimale. Une architecture entièrement parallèle offre bien sûr une capacité de calcul importante mais au prix d'une très grande complexité. Au contraire une structure itérative construite autour d'un seul opérateur réduit fortement la complexité mais limite la cadence de traitement. C'est en combinant ces 2 approches que l'on obtiendra une architecture optimale, en étant guidé par les relations qui doivent lier les caractéristiques de la chaîne de communication (débit, nombre d'antennes) et l'architecture du décodeur (nombre d'opérateurs CORDIC et leur organisation).

6.1 Equation

Nous avons établi des relations entre le débit et le nombre d'opérateurs CORDIC en parallèle. Elles sont établies pour un système MIMO à M antennes émettrices et N antennes réceptrices.

Au niveau du récepteur, le débit de traitement est exprimé par l'équation (8).

$$\text{Débit} = (\text{Freq} \times N \times b) \times (\text{NC}_{\text{utilisé}} / \text{NC}_{\text{total}}) \quad (8)$$

Dans cette équation, NC_{total} représente le nombre total d'opérateurs CORDIC à utiliser pour un système MIMO donné. Il est obtenu en fonction du nombre d'antennes à l'émission et à la réception. Freq est la cadence de calcul d'un opérateur CORDIC.

L'équation (8) peut être réécrite sous la forme suivante:

$$NC_{\text{utilisé}} = (\text{Débit} \times NC_{\text{total}}) / (\text{Freq} \times N \times b) \quad (9)$$

Cette équation permet d'obtenir le nombre optimal d'opérateur CORDIC à implanter pour s'adapter au débit de la chaîne de communication. Sous l'hypothèse que le canal reste stationnaire pendant une période trame, ce débit correspond au débit symbole à l'émission ($\text{Débit}_{\text{symbole}}$). Nous exprimons cette période de stationnarité par le nombre de symboles par trame ($N_{\text{symboles/H}}$). Dans ces conditions nous pouvons exprimer le nombre de CORDIC optimal comme étant le nombre de CORDIC minimal à utiliser pour que le débit binaire à l'émission soit égal au débit de traitement à la réception (cf. équation (10)):

$$NC_{\text{optimal}} = (M/N) \times (NC_{\text{total}} \times F_s) / (\text{Freq} \times N_{\text{symboles/H}}) \quad (10)$$

Avec $F_s = \text{Débit}_{\text{symbole}} / (M \times b)$

Tableau 1: Notations utilisées dans les équations (8), (9) et (10)

Freq	Fréquence horloge déterminée par la cadence de calcul d'un opérateur CORDIC
b	Bits par symbole
$NC_{\text{utilisé}}$	Nombre d'opérateurs CORDIC utilisé
NC_{minimal}	Nombre minimal d'opérateurs CORDIC
NC_{total}	Nombre total d'opérateurs CORDIC (en fonction de M et N)
F_s	Fréquence symbole
$N_{\text{symboles/H}}$	Nombre de symboles par trame
$\text{Débit}_{\text{symbole}}$	Débit symbole à l'émission

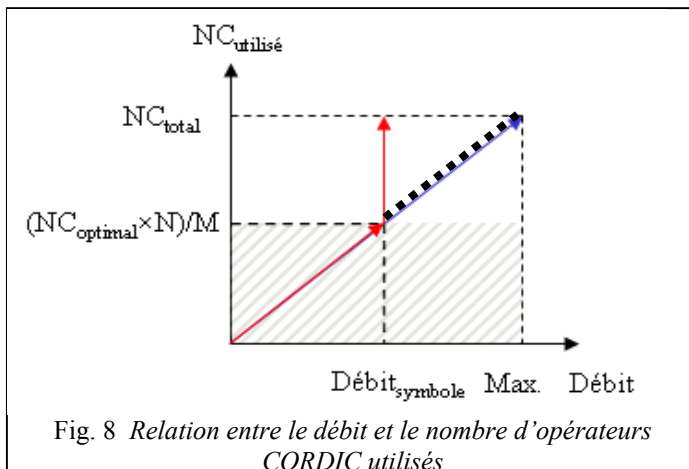


Fig. 8 Relation entre le débit et le nombre d'opérateurs CORDIC utilisés

La relation entre le nombre d'opérateurs CORDIC et le débit est illustré dans la figure 8. Les autres facteurs dans l'équation (10) sont considérés comme des constantes dans un même système MIMO. Le nombre d'opérateurs CORDIC minimal pour atteindre le débit souhaité est NC_{optimal} . Si on diminue le nombre d'opérateurs CORDIC, le débit va chuter linéairement. En revanche, le débit n'augmente pas linéairement avec le nombre d'opérateurs dans la figure 8. Il reste constant comme indiqué parce que le système calcule la même matrice du canal.

6.2 Exemple de résolution des équations et d'implémentation: calcul de $P^{1/2}$ et de Q_α

Avec les équations (9) et (10), nous pouvons calculer exactement le nombre d'opérateurs CORDIC nécessaire pour s'adapter aux débits donnés en diminuant la complexité.

L'architecture du module est représentée dans la figure 9. Lors que le débit demandé change, nous pouvons modifier le nombre d'opérateurs CORDIC ainsi que leurs interconnexions

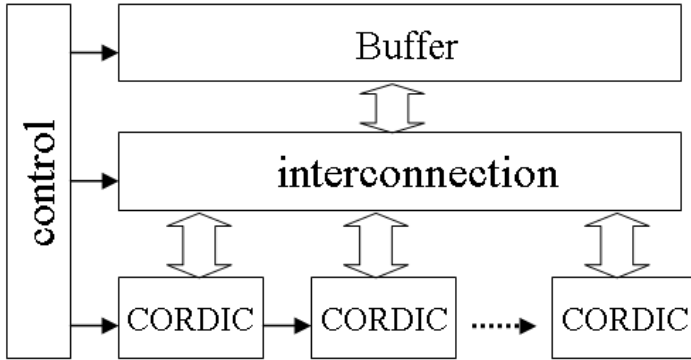


Fig. 9 Diagramme des blocs de transformations unitaires($M1, M2, M3$)

Par exemple, si nous définissons une trame comme un paquet de 7 symboles ($N_{\text{symboles/H}} = 7$), alors, pour atteindre 600 Mbits/s, le nombre optimal d'opérateurs CORDIC est de 5, en supposant que la fréquence de fonctionnement d'un opérateur CORDIC est de 150 MHz. Les calculs de $P^{1/2}$ et de Q_α peuvent être alors exécutés par 5 opérateurs CORDIC en parallèle en 7 cycles (Fig.10-a et 10-b).

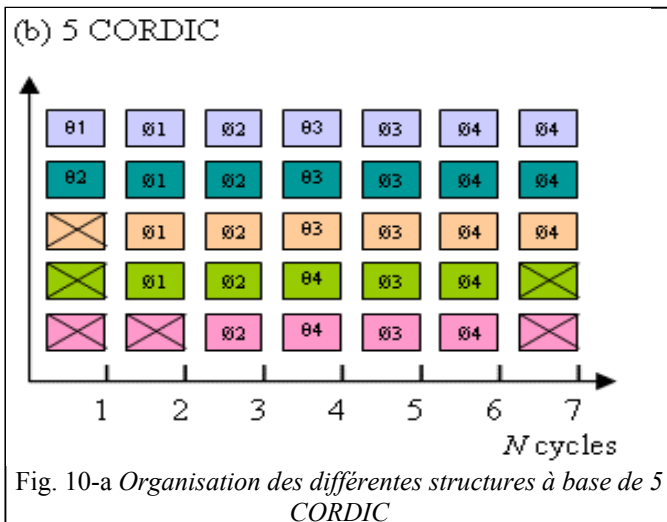


Fig. 10-a Organisation des différentes structures à base de 5 CORDIC

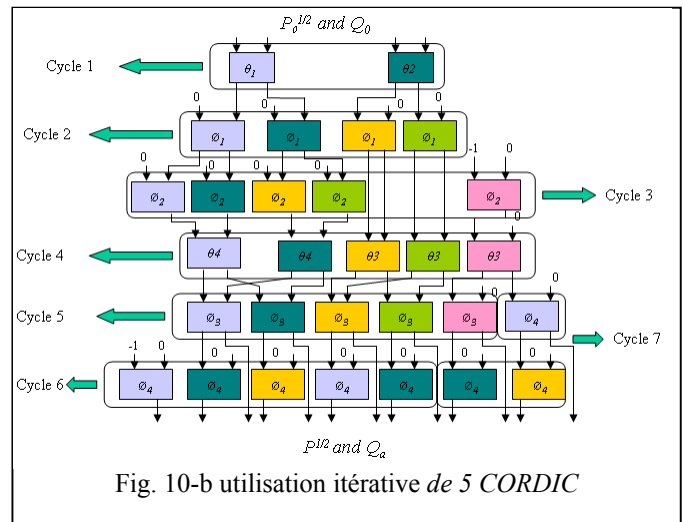


Fig. 10-b utilisation itérative de 5 CORDIC

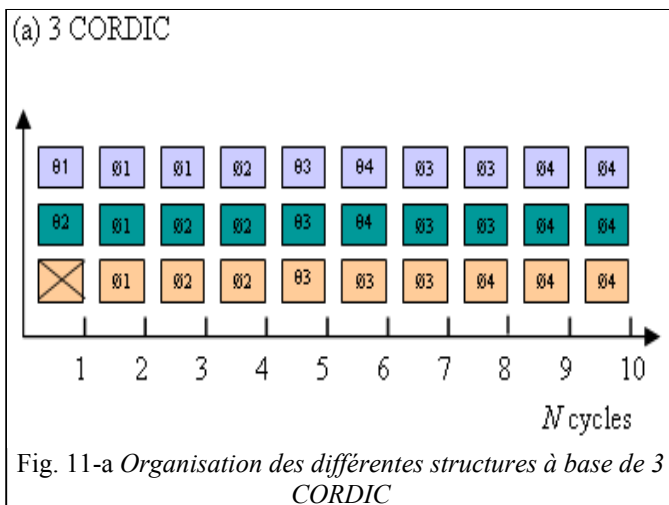


Fig. 11-a Organisation des différentes structures à base de 3 CORDIC

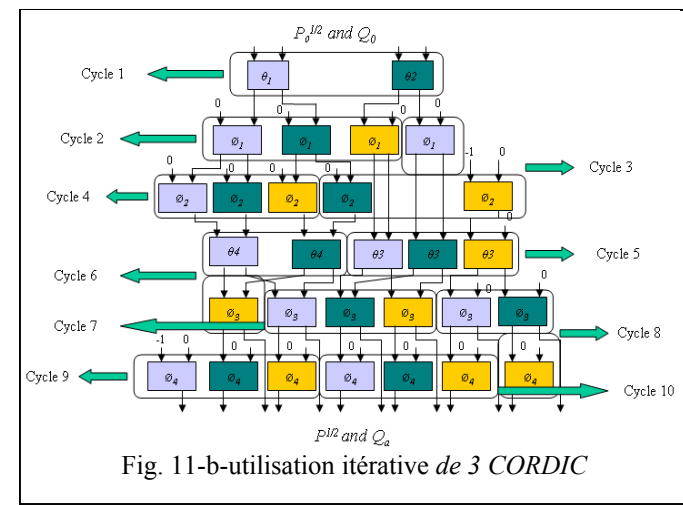


Fig. 11-b-utilisation itérative de 3 CORDIC

Maintenant pour obtenir un débit plus faible de 400 Mbits/s nous changeons le nombre d'opérateurs CORDIC. Le nombre d'opérateurs devant être implantés est de 3 en appliquant l'équation (10). Les calculs de $P^{1/2}$ et de Q_α peuvent alors être exécutés en 10 cycles. L'organisation de l'architecture des opérateurs CORDIC est présentée dans les figures 11-a et 11-b.

Dans la structure parallèle de la figure 7, 29 opérateurs CORDIC étaient utilisés. Dans cette nouvelle architecture, nous n'utilisons plus que 5 opérateurs CORDIC en parallèle (voir figures 10-a et 10-b). Nous utilisons ces 5 opérateurs CORDIC itérativement pendant 7 cycles. Dans le premier cycle, nous utilisons 2 opérateurs CORDIC pour effectuer 2 rotations de Givens avec les angles θ_1 et θ_2 . Ensuite dans le cycle suivant, 4 rotations de Givens avec les angles différents (indiquer dans la figure 10-b) sont effectués par 4 opérateurs CORDIC dont les deux premiers sont les mêmes que dans le premier cycle et réutilisés dans le deuxième cycle, et ainsi de suite pour les cycles suivants. L'ensemble du traitement est donc effectué par 5 opérateurs CORDIC en 7 cycles. L'opération est plus longue qu'avec la structure toute parallèle, mais nous optimisons les ressources matérielles

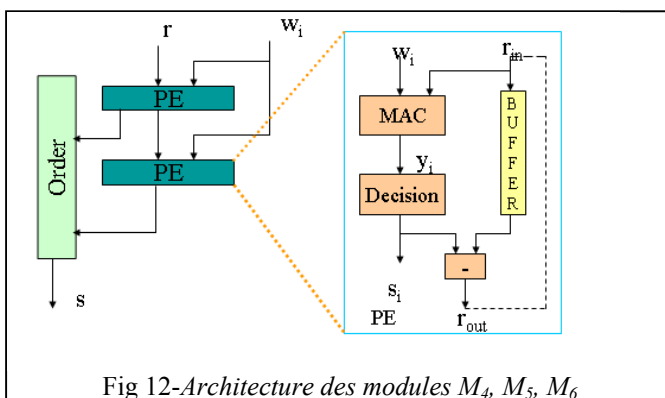
6.3 Résultat de l'implémentation du calcul de $P^{1/2}$ et de Q_α

Nous comparons les implémentations des trois architectures dans un circuit FPGA Virtex de Xilinx afin de déterminer l'architecture la plus adaptée à la réalisation matérielle. La première est une architecture parallèle et pipeline qui donne la meilleure performance en débit mais avec un coût en surface important. Puis nous l'optimisons en surface en diminuant le nombre d'opérateurs CORDIC. Enfin une autre optimisation utilise 3 CORDIC. Elle peut être utilisée quand le débit n'est pas essentiel ; elle permet de diminuer la complexité de l'architecture. La durée du chemin critique est de 6.729 ns, donc la fréquence maximum est de 148.6 MHz. Les messages reçus sont traités sur 16 bits. Les symboles estimés sont calculés en parallèle. Chaque cycle nous pouvons obtenir 2 symboles en parallèle. La modulation utilisée est une QPSK. Le tableau 2 résume les résultats obtenus en fonction du nombre d'opérateurs CORDIC utilisés.

FPGA Xilinx Virtex	Nombre de slices	Max.Freq (MHz)	Débit (Mbits/s)
29 CORDIC	16840	148.6	600
5 CORDIC	10967	148.6	600
3 CORDIC	8635	148.6	300

Tableau 2 : Résultat de la synthèse pour le module M1

6.4 Implémentation de l'ensemble de l'algorithme



Les modules M_1, M_2, M_3 ont été implantés comme vu précédemment et les modules M_4, M_5, M_6 sont implantés de la manière suivante : Ces modules sont constitués de plusieurs étages de PE pour atteindre un haut débit du traitement des messages reçus (Fig.12). Chaque étage PE se compose d'un multiplieur-accumulateur, un comparateur, un soustracteur et une mémoire. Le débit peut être encore amélioré en parallélisant plusieurs étages PE. Si le débit n'est pas un critère exigeant, on peut utiliser un seul PE de façon itérative afin de diminuer la complexité du module.

Après avoir implantés tous les modules de base, nous pouvons les intégrer dans une architecture globale. Cette architecture globale du détecteur est illustrée dans la figure 13. Nous utilisons un contrôleur pour organiser les interconnexions entre tous les modules. Les données sont stockées dans une mémoire FIFO prédéfinie par Xilinx. La taille de la mémoire dépend du nombre d'opérateurs utilisés (CORDIC et PE).

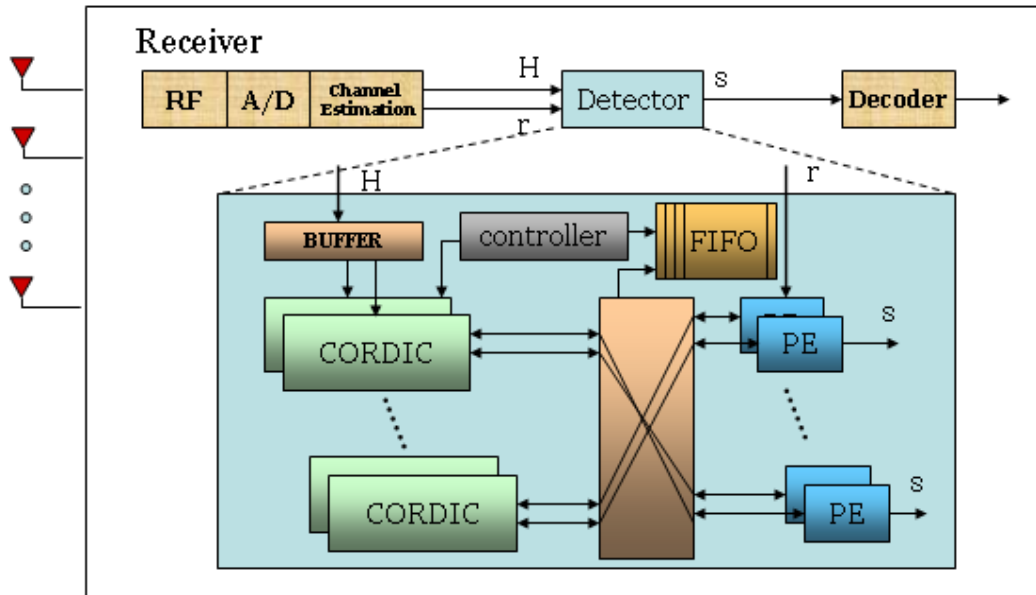


Fig. 13 Architecture du détecteur V-BLAST Square Root

6.5 Résultats de l'implémentation de l'ensemble de l'algorithme

FPGA Xilinx Virtex	Nombre de slices	Max.Freq (MHz)	Débit (Mbits/s)
50 CORDIC	29036	148.6	600
16 CORDIC	14380	148.6	600
8 CORDIC	9936	148.6	300

Tableau 3 : Résultat de la synthèse de l'ensemble de l'algorithme

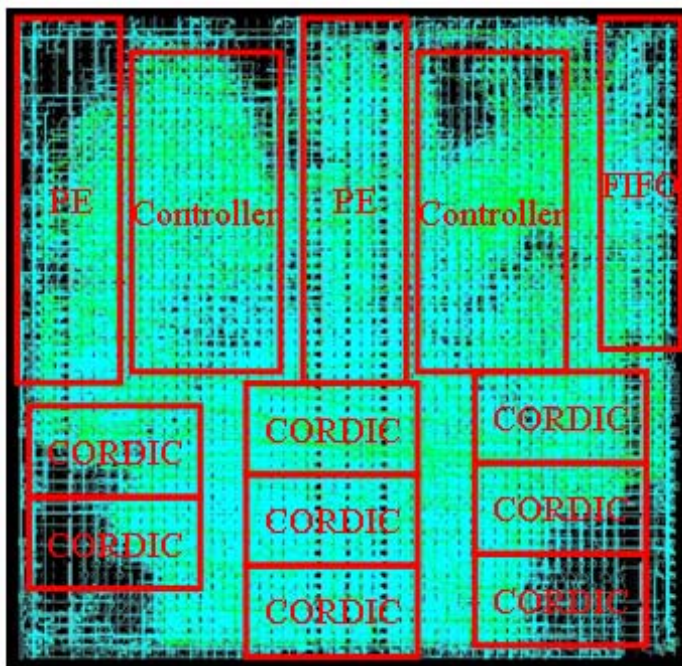


Fig.14-Implémentation de l'architecture dans un FPGA (FPGA editor) pour une architecture à 8 CORDIC

Les messages reçus sont traités sur 16 bits. Les symboles estimés sont calculés en parallèle. Chaque cycle nous obtenons 2 symboles en parallèle. La modulation utilisée étant une QPSK., nous obtenons alors les résultats du tableau 3. Ces résultats sont donnés en fonction du nombre d'opérateurs CORDIC utilisés. Les modules de l'architecture sont synthétisés individuellement par l'outil de synthèse de Xilinx ISE 6.3. Avec l'aide de PLANHEAD de Xilinx, nous pouvons définir les zones de chaque module de l'architecture dans un FPGA Virtex. Ensuite nous utilisons ISE 6.3 pour placer et router ces modules. La figure 14 montre une vue de l'architecture du détecteur V-BLAST Square Root après placement-routage dans un FPGA virtex II.

6.6 Reconfiguration dynamique de l'architecture

Nous avons montré que pour un ensemble de paramètres fixés caractérisant la transmission, nous pouvons en déduire l'architecture optimale de réalisation garantissant les performances de l'algorithme pour une complexité matérielle minimale. En utilisant la technologie FPGA, nous pouvons reconfigurer le système de traitement de façon à implémenter cette architecture optimale mais pour une combinaison des paramètres donnée et en ayant préalablement réalisé les différentes implémentations pour différentes combinaisons.

Dans un système MIMO où l'on envisage une adaptation dynamique des paramètres de la transmission (débit variable, structure d'antennes évolutive), il faut concevoir une plateforme matérielle de réalisation du décodeur capable de modifier son architecture en fonction de ces changements de contexte. La technologie

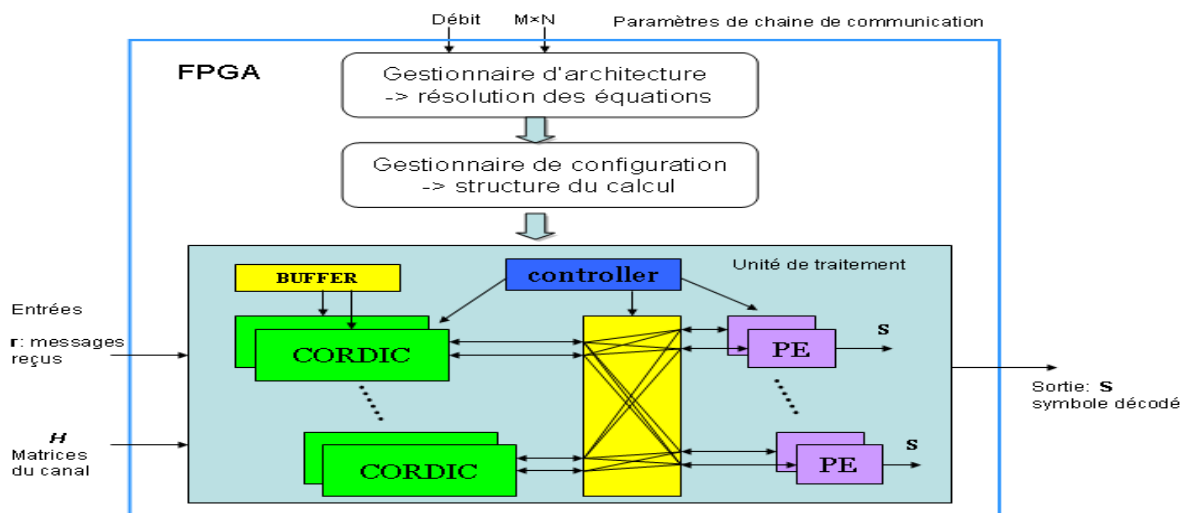


Fig. 15 Flot de gestion de la reconfiguration

FPGA avec reconfiguration dynamique partielle autorise cette approche.

L'architecture reconfigurable dynamiquement que nous proposons se compose d'une part de fonctions de gestion de reconfiguration et d'autre part des fonctions de traitement.

Le gestionnaire d'architecture s'applique à résoudre les équations reliant les caractéristiques de la transmission aux paramètres d'architectures énoncés précédemment. Les données issues de cette analyse sont transmises au gestionnaire de configuration dont la tâche est la construction de l'architecture optimale de traitement [20]. Pour cela le gestionnaire de configuration dispose de ressources matérielles reconfigurables, la reconfiguration matérielle portant sur le nombre d'opérateurs CORDIC implémentés et les chemins de données, ainsi que sur la logique de contrôle.

7 Conclusion

Nous avons réalisé l'implantation du détecteur MIMO basé sur l'algorithme V-BLAST « Square Root ». Cette architecture offre une grande flexibilité vis à vis du nombre d'antennes et des différents débits données. Il est adapté aux différents besoins en utilisant différent nombre d'opérateurs CORDIC.

Notre architecture nous permet d'obtenir des débits considérables, grâce au parallélisme de l'architecture, bien entendu au prix d'une surface utilisée importante du circuit FPGA. Cela fait partie des objectifs de nos travaux futurs de diminuer au maximum la surface utilisée. Les pistes d'étude possibles, sont d'une part un ordonnancement optimal, de manière à ne jamais laissé inactif un opérateur CORDIC et d'autre part de réaliser les modules M4, M5, M6 aussi avec les opérateurs CORDIC. Nous étudierons cette possibilité en

ayant une interprétation dans le domaine fréquentiel de ces modules et donc des processeurs élémentaires PE.

Un autre prolongement de ce travail est de développer une architecture adaptée aux besoins des futurs systèmes de radiocommunication par la reconfiguration dynamique. L'architecture sera définie par bloc permettant une reconfiguration partielle des ressources matérielles afin de minimiser le coût de la reconfiguration.

Références

- [1] Z. Guo and P. Nilsson: A VLSI implementation of MIMO detection for future wireless communications, in Proc. IEEE PIMRC'03, vol. 3, 2003, pp. 2852-2856.
- [2] D. Gesbert, M. Shafi, D. S. Shiu, P. Smith and A. Naguib, "From theory to practice: An overview of MIMO space-time coded wireless systems," IEEE Journal on Selected Areas in Communications, vol. 21, no. 3, pp. 281 – 302, Apr. 2003
- [3] D. Gesbert, J. Akhtar, Breaking the barriers of Shannon's capacity: An overview of MIMO wireless systems," Telektronikk Telenor Journal. Jan. 2002.
- [4] G. J. Foschini and M. J. Gans, "On limits of wireless communications in a fading environment when using multiple antennas," Wireless Personal Communications, vol.6, pp. 311-335, Mar. 1998
- [5] E. Telatar, "Capacity of multi-antenna gaussian channels," AT&T Bell Labs, Tech. Rep., June 1995.
- [6] M. Alamouti, "A simple transmit diversity technique for wireless communications," IEEE Journal on Selected Areas in Communications, vol. 16, no. 8, pp. 1451–1458, Oct 1998.
- [7] V. Tarokh, N. Seshadri and A. R. Calderbank, "Space-time codes for high data rate wireless communication: performance criterion and code construction," IEEE Trans. Inform. Theory, vol. 44, no. 2, pp. 744-765, Mar. 1998
- [8] G. D. Golden, G. J. Foschini, R. A. Valenzuela, P. W. Wolniansky, "Detection algorithm and initial laboratory results using V-BLAST space-time communication architecture," Electronics Letters, vol. 35, no. 1, pp. 14-15, Jan. 1999
- [9] P. W. Wolniansky, G. J. Foschini, G. D. Golden, R. A. Valenzuela, "V-BLAST : An architecture for realizing very high data rates over the rich-scattering channel," invited paper, Proc. ISSSE-98, Pisa, Italy, Sep. 1998
- [10] G. J. FOSCHINI. Layered space-time architecture for wireless communication in a fading environment when using multi-element antennas. *Bell Labs Technical Journal*, pages 41–57, Autumn 1996
- [11] G. D. GOLDEN, C. J. FOSCHINI, R. A. VALENZUELA, and P. W. WOLNIANSKY. Detection algorithm and initial laboratory results using V-BLAST space-time communication architecture. In *Electronic Letters*, volume 35, pages 14–15, January
- [12] M. O. DAMEN, K. ABED-MERAIM, and J. C. BELFIORE. A Generalized Sphere Decoder for Asymmetrical Space-Time Communication Architecture. In *IEE Electronic Letters*, volume 36, January 2000.
- [13] M. O. DAMEN, A. CHKEIF, and J. C. BELFIORE. Lattice Code Decoder for Space-Time Codes. In *IEEE Communications Letters*, volume 4, pages 166–169, May 2000
- [14] W. Zha, S.D. Blostein, "Multiuser Receivers that are Robust to Mismatch," EURASIP Journal on Applied Signal Processing, Special Issue on Multiuser Detection and Blind Estimation, Vol.12, pp.1355-1364, December 2002.
- [15] . A. H. Land and A. G. Doig."An automatic method for solving discrete programming problems." *Econometrica*, 28:497–520, 1960
- [16] P. Willett J. Luo, K. Pattipati and L. Brunel, "Branch-and-bound-based fast optimal algorithm for multiuser detection in synchronous cdma". in, vol. 5:3336–3340, May 2003.
- [17] A. Bhargave, R. J. P. de Figueiredo, and T. Eltoft. A detection algorithm for the V-. BLAST system. In *Proceedings of IEEE Global Telecommunications Conference (Globecom)*, volume 1, pages 494–498, San Antonio (TX), USA, November 2001
- [18] X. Li, H. Huang, A. Lozano, and G. J. Foschini. Reduced-complexity detection algorithms for systems using multi-element arrays. In *Proceedings of IEEE Global Telecommunications Conference (Globecom)*, volume 2, pages 1072–1076, San Francisco (CA), USA, November 27 - December 2000
- [19] B. Hassibi, "An efficient square-root algorithm for BLAST," <http://mars.bell-labs>
- [20] C. M. Rader, "VLSI Systolic Arrays for Adaptive Nulling", *IEEE Sig. Proc. Mag*, Vol. 13, No. 4, pp. 29-49.
- [21] J.P. Delahaye, J. Palicot, P. Leray, "A Hierarchical Modeling Approach in Software Defined Radio System Design", SIPS 2005 , Athènes ,Grèce