

UNIVERSITE DE LIMOGES

FACULTE DES SCIENCES ET TECHNIQUES
Laboratoire Xlim/C2S2/ESTE (UMR CNRS 6172)

MEMOIRE

Pour obtenir

LE DIPLOME D'HABILITATION A DIRIGER DES RECHERCHES

Présenté par

VAHID MEGHDADI

Maître de Conférences à l'Université de Limoges - ENSIL

*Contribution à l'étude des systèmes de communications à antennes multiples : synchronisation,
et au décodage analogique : implantation*

Soutenue le 11 décembre 2006 devant le jury composé de :

M. Dominique MEIZEL, Professeur à l'ENSIL

Président

Mme Catherine DOUILLARD, Professeur à l'ENST Bretagne

Rapporteur

M. Pierre DUHAMEL, Directeur de Recherches CNRS

Rapporteur

M. Olivier SENTIEYS, Professeur à l'ENSSAT

Rapporteur

M. Jean-Pierre CANCES, Professeur à l'ENSIL

Examinateur

M. Jean-Michel DUMAS Professeur à l'ENSIL

Examinateur

M. Jean-François HELARD, Professeur à l'INSA de Rennes

Examinateur

Avant-Propos

J'ai obtenu ma maîtrise (*Bachelor of science*) en 1989 et mon master (*Master of Science*) en 1992, les deux en télécommunications de l'Université de Sharif, Téhéran, Iran. J'ai ensuite joint l'Université de Mashad, Iran où j'enseignais mathématique, télécommunication et circuit. Participant au concours national iranien du ministère de l'éducation supérieure, j'ai réussi à obtenir une bourse d'étude pour effectuer une thèse à l'étranger. J'ai choisi la France comme destination et été admis par l'Université de Limoges. J'y ai débuté ma thèse en septembre 1994 sous la direction du Professeur Jean-Michel DUMAS et avec le co-encadrement du Monsieur Jean-Pierre CANCES. Après avoir obtenu mon diplôme de thèse en février 1998, avoir fait un post-doctorat au sein du même laboratoire (UMOP/GESTE) et avoir obtenu la qualification pour le poste de maître de conférences, j'ai rejoint l'ENSIL en tant que maître de conférences en septembre 2000.

En enseignement, je m'occupe des cours de microprocesseur, VHDL, théorie de l'information et de codage, traitement d'image, communication analogique et traitement de signal.

En recherche, je fais partie des premiers chercheurs de notre équipe où j'ai pu participer activement à son élan. En 2003, notre équipe a été labellisé CNRS et en 2005, nous avons fusionné avec l'IRCOM pour former le laboratoire de recherche Xlim.

J'ai été le co-responsable scientifique de plusieurs contrats de recherche avec France Télécom, Eutelsat, Alcatel, Sodielec, et récemment CNES.

J'essaye dans ce document de résumer quelques unes de mes activités phares depuis ma thèse.

Remerciements

Je tiens à remercier deux personnes qui m'ont soutenu dans tous les instants depuis ma venue en France en 1994. La première est Monsieur le Professeur Jean Michel DUMAS qui m'a fait confiance en me proposant une thèse sur un sujet très intéressant en 1994. Il m'a soutenu pendant toute la durée de ma thèse et au delà et m'a guidé dans toutes mes démarches professionnelles. C'est quelqu'un pour qui j'ai beaucoup de respect et sur qui j'ai pu compter.

La deuxième personne est bien sûr Monsieur le Professeur Jean-Pierre CANCES qui était pour moi non seulement le co-encadrant de thèse, mais un ami sincère qui m'a toujours encouragé et soutenu. Ses capacités scientifiques mêlées avec sa bonne humeur en font un moteur de recherche indispensable. Je le remercie et je tiens à lui dédier ce mémoire.

Je tiens aussi à remercier les doctorants avec qui j'ai eu beaucoup de plaisir à travailler. Je pense à G.R. Mohammad Khani, E. Jadas-Hécart, L. Azizi, S. Jamalullil, R. Zahabi, A. Saemi et G. Ferré.

Je remercie également mes collègues de l'ENSIL et plus particulièrement ceux de la filière Electronique et Télécommunication (ELT) d'avoir créé une bonne ambiance et avec qui j'ai le plaisir de partager mes journées. Je pense à Anne, Christelle, Marie Sandrine, Valérie, Richard, Stéphane,

Enfin, je tiens à remercier ma femme Mitra sur qui je me suis appuyé toujours et qui m'encourage et me soutient quoi qu'il arrive.

Vahid Meghdadi

PRESENTATION

Ce mémoire est rédigé en trois chapitres. Le premier chapitre est le résultat des travaux de recherche effectués sur le codage Temps Espace en Bloc. Ce travail a été effectué avec deux doctorants Mlle Leila Azizi et M. Golam Reza Mohammad Khani.

Le deuxième chapitre traite la synchronisation dans les systèmes MIMO-OFDM. Ces travaux sont effectués dans le cadre d'un contrat de recherche avec France Télécom R&D qui finance une thèse. Le doctorant avec qui je travaille sur ce sujet et M. Amir Saemi qui soutiendra sa thèse en fin 2007.

Le troisième chapitre traite des travaux en cours sur les circuits analogiques pour réaliser les algorithmes de décodage de canal. Ces principes peuvent être utilisés pour implanter d'autres types d'algorithmes tels que FFT, filtre FIR, multiplication matricielle, etc. Le doctorant avec qui je travaille est M. Mohammad Reza Zahabi qui soutiendra sa thèse en début de l'année 2008.

Glossaire

ADSL	Asymmetric Digital Subscriber Line
ASIC	Application Specified Integrated Circuit
CAN	Convertisseur Analogique Numérique
CNA	Convertisseur Analogique Numérique
FFT	Fast Fourier Transform
LAN	Local Area Network
MIMO	Multiple Input Multiple output
ML	Maximum Likelihood
OFDM	Orthogonal Frequency Division Multiplexing
PDP	Power Delay Profile
RIF	Réponse Impulsionnelle Finie
RII	Réponse Impulsionnelle Infinie
RX	Récepteur
SIMO	Single Input Multiple Output
SISO	Single Input Single Output
SNR	Signal to Noise Ratio
TFD	Transformée de Fourier Discrète
TX	Emetteur

Table de matières

CHAPITRE 1 CODES STBC ORTHOGONAUX ET DERIVATION DES BORNES SUPERIEURES PRECISES	1-1
1.1 INTRODUCTION	1-1
1.2 DESCRIPTION DU SYSTEME	1-2
1.2.1 <i>Développement et mise en équation de la matrice du code</i>	1-6
1.3 DECODAGE ML DES CODES STBC ORTHOGONAUX	1-7
1.3.1 <i>Calcul des paramètres de décision</i>	1-9
1.3.2 <i>Algorithme de Décodage</i> :	1-12
1.4 ETUDE ANALYTIQUE (DERIVATION DES BORNES SUPERIEURS DE TEB)	1-13
1.4.1 <i>Canal constant</i>	1-15
1.4.2 <i>Canal à variation aléatoire</i>	1-17
1.4.3 <i>Cas des codes réels</i>	1-23
1.5 RESULTATS DES SIMULATIONS	1-26
1.6 CONCLUSION	1-28
1.7 RÉFÉRENCES	1-30
ANNEXE A : PREUVE DES PARTICULARITES DES MATRICES DE CODES	1-33
ANNEXE B : PREUVE DES PARTICULARITES DES MATRICES H_R^J ET H_I^J	1-35
ANNEXE C : PARTICULARITE DES MATRICES ANTISYMETRIQUES	1-38
ANNEXE D : DENSITE DE PROBABILITE DE GAMMA	1-39
ANNEXE E : REPERE POLAIRE DE DIMENSION M	1-41
CHAPITRE 2 SYNCHRONISATION DANS LES SYSTEMES MIMO-OFDM	2-1
2.1 INTRODUCTION	2-1
2.2 MODULATION OFDM	2-2
2.3 CANAL SELECTIF EN FREQUENCE	2-4
2.4 MIMO-OFDM	2-5
2.5 SYNCHRONISATION EN TEMPS	2-9
2.5.1 <i>Synchronisation fine de temps</i>	2-13
2.6 SYNCHRONISATION EN FREQUENCE	2-15
2.7 ESTIMATION CONJOINTE DU CANAL ET SYNCHRONISATION EN TEMPS	2-17
2.7.1 <i>Performances</i>	2-19
2.8 ESTIMATION CONJOINTE DU CANAL ET SYNCHRONISATION EN TEMPS ET EN FREQUENCE	2-21
2.8.1 <i>Performances</i>	2-25

2.8.2	<i>Estimation fine de la fréquence</i>	2-26
2.8.3	<i>Performances</i>	2-28
2.9	CONCLUSION.....	2-29
ANNEXE I DIAGONALISATION DE LA MATRICE CIRCULANTE		2-30
ANNEXE II APPROXIMATION OPTIMALE		2-34
ANNEXE III BORNE INFERIEURE DE CRAMER-RAO POUR LES SYSTEMES MIMO-OFDM		2-35
REFERENCES		2-37
CHAPITRE 3 DECODAGE ANALOGIQUE		3-1
3.1	INTRODUCTION	3-1
3.2	DECODAGE.....	3-4
3.2.1	<i>LLR</i>	3-4
3.2.2	<i>Loi de détection</i>	3-5
3.2.3	<i>Algorithme de MAP</i>	3-7
3.3	ELEMENTS DU CIRCUIT ANALOGIQUE.....	3-13
3.3.1	<i>Transformation de LLR en probabilité</i>	3-13
3.3.2	<i>Transformation des probabilités en LLR</i>	3-14
3.3.3	<i>Opération Box-Plus</i>	3-15
3.3.4	<i>Miroir de courant</i>	3-16
3.3.5	<i>Winner-Take-All</i>	3-16
3.4	DECODAGE ANALOGIQUE POUR LES CODES EN BLOC.....	3-18
3.4.1	<i>Introduction</i>	3-18
3.4.2	<i>Décodage itératif</i>	3-20
3.4.3	<i>Implantation avec circuit analogique</i>	3-22
3.5	DECODAGE ANALOGIQUE POUR LES CODES EN TREILLIS	3-25
3.5.1	<i>Introduction</i>	3-25
3.5.2	<i>Algorithme de décodage</i>	3-28
3.5.3	<i>Circuit analogique</i>	3-29
3.5.4	<i>Dimensionnement du circuit</i>	3-33
3.6	DECODAGE EN BLOC DES CODES CONVOLUTIFS	3-35
3.7	ASPECT DE LA SIMULATION.....	3-39
3.8	TURBO CODES	3-42
3.8.1	<i>Introduction</i>	3-42
3.8.2	<i>Analogue turbo décodeur</i>	3-43
3.9	AUTRES APPLICATIONS DES METHODES ANALOGIQUES	3-46
3.9.1	<i>Filtre RIF à coefficient constant</i>	3-46
3.9.2	<i>Filtre RII à coefficient constant</i>	3-47

3.9.3	<i>Transformée de Fourier Rapide (FFT)</i>	3-47
3.9.4	<i>Récepteur MIMO analogique</i>	3-48
3.9.5	<i>Filtrage FIR et sur échantillonnage</i>	3-48
3.10	PERSPECTIVES	3-50
ANNEXE I	REALISATION DE TREILLIS.....	3-52
ANNEXE II	PROGRAMME MATLAB POUR UN DECODAGE EN GRAPHE.....	3-55
ANNEXE III	ARTICLE ICEE2006	3-57
ANNEXE IV	ARTICLE PUBLIE ELECTRONIC LETTERS	3-65
ANNEXE V	ARTICLE SOUMIS A ICC 2007.....	3-72
REFERENCE	3-77
CHAPITRE 4	CONCLUSIONS ET PERSPECTIVES.....	4-1

Chapitre 1 Codes STBC Orthogonaux et Dérivation des Bornes Supérieures Précises

1.1 *Introduction*

La diversité de transmission a été largement étudiée comme une méthode efficace pour les canaux à fading [1][2][3][4][5]. Elle est particulièrement intéressante pour sa simplicité relative d'implantation et la faisabilité d'installation d'antennes multiples à la station de base. Le STTC (*Space-Time Trellis Coding*) combine le codage de canal de type TCM avec les techniques appropriées aux antennes multiples à l'émetteur. Dans [6][7][8], les auteurs atteignent une très bonne performance pour 2 à 4 antennes d'émission et pour les canaux slow-fading en s'approchant de la capacité d'outage du canal. Cependant, l'inconvénient de tels codes est la complexité importante de l'algorithme de décodage. Cette complexité augmente de manière exponentielle par rapport à la taille de la constellation utilisée et le nombre d'antennes d'émission.

Pour éviter la complexité prohibitive de STTC, les schémas de la simple diversité de transmission basés sur la conception de matrices orthogonales ont été proposés [9][10][11]. L'utilisation d'un tel schéma de diversité, en combinant avec un code FEC (*Forward Error Correction*) qui peut être de type TCM afin de pouvoir comparer les deux cas, aboutit à une performance presque similaire obtenue par STTC. La construction des matrices orthogonales aboutit aux techniques de codage STBC, qui pourront être planifiées, par exemple, pour la nouvelle génération des systèmes radio-mobiles cellulaires.

Dans ce chapitre, on propose d'abord une structure générale de réception pour le décodage ML des codes STBC. Utilisant les outils algébriques, on propose donc un modèle matriciel particulièrement approprié à la description des systèmes MIMO. Le modèle obtenu a l'avantage de donner un simple algorithme de décodage ML, basé sur un traitement linéaire au récepteur. Grâce à ce modèle matriciel, on va trouver une nouvelle borne supérieure très précise

pour le TEB du code STBC. Ces résultats sont finalement vérifiés avec une simulation Monte-Carlo.

1.2 Description du Système

La Figure 1 illustre un système multi-antennes de type STBC. Il y a donc n antennes d'émission, m antennes de réception et un rendement de k/l qui décrit la redondance du code. Pour un paquet de k symboles d'information en entrée du codeur, on aura l symboles sur chaque antenne d'émission. Ces l symboles peuvent être une combinaison linéaire des symboles d'entrée (x_1, x_2, \dots, x_k) et de leur conjugué. C_t^i représente le symbole émis par l'antenne d'émission i dans le temps t . Le coefficient $\alpha_{i,j}$ est le gain de trajet entre l'antenne d'émission i et l'antenne de réception j ; ces atténuations du canal sont modélisées comme des variables aléatoires gaussiennes complexes et indépendantes avec une variance de 0,5 pour les parties réelle et imaginaire.

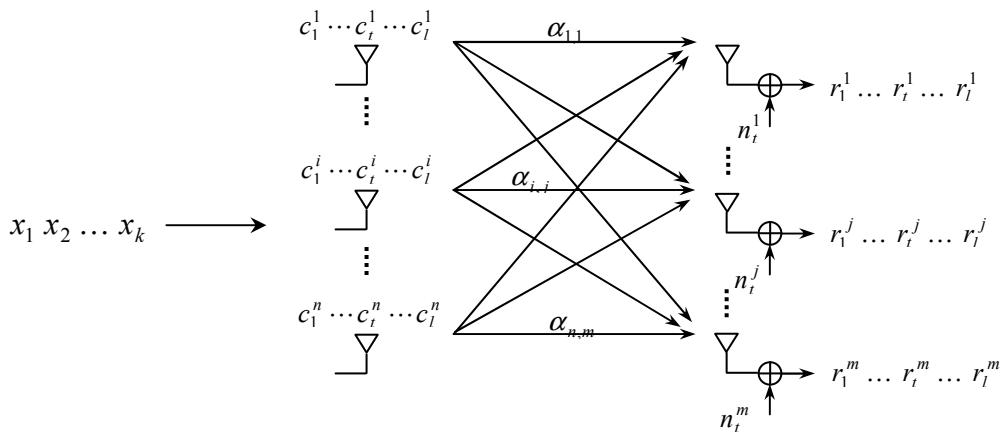


Figure 1 : Modèle du système MIMO-STBC

Dans le temps t , le signal r_t^j reçu par l'antenne j s'écrit sous la forme suivante :

$$r_t^j = \sum_{i=1}^n \alpha_{i,j} c_t^i + n_t^j \quad (1)$$

où n_t^j est le bruit gaussien complexe et additif sur l'antenne j pour le temps t ; les n_t^j sont les variables aléatoires gaussiennes complexes et indépendantes à moyenne nulle et variance $\sigma^2/2 = 1/(2\text{SNR})$ par dimension complexe : $E\left\{ |n_t^j|^2 \right\} = \sigma^2$. Pour obtenir la totalité d'énergie

émise égale à 1, l'énergie moyenne des symboles transmis par chaque antenne est normalisée par $1/n$. C'est à dire que la totalité d'énergie sur toutes les antennes d'émission est égale à 1. Ainsi une comparaison judicieuse par rapport à un système monoantenne est possible.

Afin d'avoir une forme matricielle pour l'équation (1), on définit le vecteur $\mathbf{r}^j = (r_1^j \ \dots \ r_t^j \ \dots \ r_l^j)$ représentant le signal reçu sur l'antenne j . On arrive ainsi à l'équation suivante :

$$\mathbf{r}^j = \underbrace{[\alpha_{1,j} \ \dots \ \alpha_{i,j} \ \dots \ \alpha_{n,j}]}_{\mathbf{h}^j} \underbrace{\begin{bmatrix} c_1^1 & \dots & c_t^1 & \dots & c_l^1 \\ \vdots & & & & \vdots \\ c_1^i & & c_t^i & & c_l^i \\ \vdots & & & & \vdots \\ c_1^n & \dots & c_t^n & \dots & c_l^n \end{bmatrix}}_{\mathbf{C}} + \mathbf{n}^j = \mathbf{h}^j \mathbf{C} + \mathbf{n}^j \quad (2)$$

où le vecteur \mathbf{h}^j contient les gains des trajets reçus par l'antenne j , \mathbf{C} est la matrice de codes en bloc contenant les symboles émis par toutes les antennes et pour tous les temps d'un paquet et $\mathbf{n}^j = (n_1^j \ \dots \ n_t^j \ \dots \ n_l^j)$ est le vecteur de bruit avec la matrice d'auto-corrélation suivante :

$$E\{\mathbf{n}^j \mathbf{n}^j\} = \sigma^2 \mathbf{I}_{l \times l}$$

Maintenant en définissant la matrice \mathbf{r} sous la forme suivante, on trouve une forme matricielle pour tous les signaux reçus par toutes les antennes.

$$\mathbf{r} = \begin{bmatrix} \mathbf{r}^1 \\ \vdots \\ \mathbf{r}^j \\ \vdots \\ \mathbf{r}^m \end{bmatrix} = \underbrace{\begin{bmatrix} \mathbf{h}^1 \\ \vdots \\ \mathbf{h}^j \\ \vdots \\ \mathbf{h}^m \end{bmatrix}}_{\mathbf{H}} \cdot \mathbf{C} + \underbrace{\begin{bmatrix} \mathbf{n}^1 \\ \vdots \\ \mathbf{n}^j \\ \vdots \\ \mathbf{n}^m \end{bmatrix}}_{\mathbf{n}} = \mathbf{H} \cdot \mathbf{C} + \mathbf{n} \quad (3)$$

Supposons que les informations du canal sont disponibles, le récepteur ML minimise la métrique de décision ci-dessous sur tous les mots du code \mathbf{C} :

$$\text{ML : } \underset{\mathbf{C}}{\text{Min}} \quad \Omega = \sum_{j=1}^m \sum_{t=1}^l \left| r_t^j - \sum_{i=1}^n \alpha_{i,j} c_t^i \right|^2 \quad (4)$$

En considérant les définitions matricielles présentées ci-dessus, on peut réécrire la métrique Ω sous la forme suivante :

$$\begin{aligned} \text{Min } \Omega &= \sum_{j=1}^m (\mathbf{r}^j - \mathbf{h}^j \mathbf{C})(\mathbf{r}^j - \mathbf{h}^j \mathbf{C})^\dagger \\ \text{Min } & \sum_{j=1}^m (-\mathbf{r}^j \mathbf{C}^\dagger (\mathbf{h}^j)^\dagger - \mathbf{h}^j \mathbf{C} (\mathbf{r}^j)^\dagger + \mathbf{h}^j \mathbf{C} \mathbf{C}^\dagger (\mathbf{h}^j)^\dagger) \end{aligned} \quad (5)$$

Pour pouvoir simplifier cette relation, il faut connaître le code et ses propriétés. Cependant, il est toujours possible de décoder en calculant la métrique de l'équation (5) pour tous les mots possibles du code et choisir celui qui minimise cette équation. Par exemple, pour une modulation M-aire et pour les séquences de k symboles en entrée du codeur, la métrique ci-dessus est calculée M^k fois. Le code qui donne la métrique minimale parmi toutes ces métriques, est annoncé en tant que la séquence décodée suivant le critère ML.

Codes STBC Orthogonaux

Dans ce paragraphe, on cherche à simplifier l'équation (5) en utilisant une matrice de code orthogonale. On considère deux cas séparément :

- Le cas du code complexe où la constellation des symboles envoyés est définie dans le domaine complexe comme des modulation MPSK ou QAM.
- Le cas du code réel où la constellation des symboles envoyés est définie uniquement sur l'axe réel comme la modulation PAM. C'est en fait un cas particulier des codes complexes.

La matrice de code $\mathbf{C}_{n \times l}$ (complexe en général) est de taille n et le rendement du code est k/l .

Le code est orthogonal si les conditions suivantes sont satisfaites :

- Les éléments de la matrice \mathbf{C} sont les combinaisons linéaires de k symboles d'entrées (x_1, x_2, \dots, x_k) et de leur conjugué.
- $\mathbf{C} \mathbf{C}^\dagger = \mathbf{D}_c$ où \mathbf{D}_c est une matrice diagonale avec les éléments (i,i) de la forme suivante :

$$\left(l_1^i |x_1|^2 + l_2^i |x_2|^2 + \dots + l_k^i |x_k|^2 \right)$$

et les coefficients $l_1^i, l_2^i, \dots, l_k^i$ sont strictement positifs. Il est clair que pour cela, il faut absolument que $l \geq n$. Car le rang de la matrice \mathbf{C} est inférieur ou égal à $\min(n, l)$ et le fait que $\mathbf{C} \mathbf{C}^\dagger$ est une matrice diagonale de taille $n \times n$ avec les éléments de diagonal non nuls, nécessite que le rang de \mathbf{C} soit égal à n et par conséquent $n \leq l$.

Tarokh a montré [11] que s'il existe un code complexe et orthogonal $n \times l$ pour les variables (x_1, x_2, \dots, x_k) , il existera un code complexe et orthogonal \mathbf{C} de la même taille et pour les mêmes variables telle que :

$$\mathbf{C}\mathbf{C}^\dagger = \left(|x_1|^2 + |x_2|^2 + \dots + |x_k|^2 \right) \mathbf{I}_{n \times n}$$

Dans la définition du code orthogonal, on peut donc remplacer la deuxième condition par cette dernière relation.

Pour la première condition et dans la plupart des cas, on peut trouver un code orthogonal dont les éléments sont

$$0, \pm x_1, \pm x_1^*, \pm x_2, \pm x_2^*, \dots, \pm x_k, \pm x_k^*$$

et non pas la combinaison linéaire des x_i et x_i^* . Nous développons ci-dessous le cas particulier du code réel.

Cas des codes réels :

Pour un code réel et orthogonal, \mathbf{C} est une matrice réelle dont les éléments sont les combinaisons linéaires des symboles d'entrées (x_1, x_2, \dots, x_k) . Les x_k sont choisis dans une constellation réelle, ce qui est le cas des modulations PAM. Dans ce cas $\mathbf{C}^\dagger = \mathbf{C}^T$ et pour un code réel et orthogonal, on aura :

$$\mathbf{C}\mathbf{C}^T = \left(x_1^2 + x_2^2 + \dots + x_k^2 \right) \mathbf{I}_{n \times n}$$

Dans la majorité des cas, on peut trouver le code orthogonal réel dont les éléments sont

$$0, \pm x_1, \pm x_2, \dots, \pm x_k$$

Exemples des codes orthogonaux:

Quel que soit le type du code (complexe ou réel), la définition du code orthogonal que l'on vient de présenter n'impose aucune condition sur les éléments de la matrice \mathbf{C} , et cette définition peut comprendre les exemples ci-dessous.

Codes réels : $\begin{pmatrix} x_1 & -x_2 \\ x_2 & x_1 \end{pmatrix} \quad \begin{pmatrix} x_1 & -x_2 & -x_3 & -x_4 \\ x_2 & x_1 & x_4 & -x_3 \\ x_3 & -x_4 & x_1 & x_2 \\ x_4 & x_3 & -x_2 & x_1 \end{pmatrix} \quad \begin{pmatrix} x_1 & -x_2 & -x_3 & -x_4 \\ x_2 & x_1 & x_4 & -x_3 \\ x_3 & -x_4 & x_1 & x_2 \\ x_4 & x_3 & -x_2 & x_1 \end{pmatrix}$

Codes complexes :

$$\begin{pmatrix} x_1 & -x_2^* \\ x_2 & x_1^* \end{pmatrix} \begin{pmatrix} x_1 & -x_2^* & \frac{x_3^*}{\sqrt{2}} & \frac{x_3^*}{\sqrt{2}} \\ x_2 & x_1^* & \frac{x_3^*}{\sqrt{2}} & -\frac{x_3^*}{\sqrt{2}} \\ \frac{x_3}{\sqrt{2}} & \frac{x_3}{\sqrt{2}} & \frac{(-x_1 - x_1^* + x_2 - x_2^*)}{2} & \frac{(x_1 - x_1^* + x_2 + x_2^*)}{2} \end{pmatrix}$$

$$\begin{pmatrix} x_1 & -x_2 & -x_3 & -x_4 & x_1^* & -x_2^* & -x_3^* & -x_4^* \\ x_2 & x_1 & x_4 & -x_3 & x_2^* & x_1^* & x_4^* & -x_3^* \\ x_3 & -x_4 & x_1 & x_2 & x_3^* & -x_4^* & x_1^* & x_2^* \\ x_4 & x_3 & -x_2 & x_1 & x_4^* & x_3^* & -x_2^* & x_1^* \end{pmatrix}$$

1.2.1 Développement et mise en équation de la matrice du code

D'après la définition du code orthogonal, la matrice \mathbf{C} peut s'écrire comme la somme de deux termes. Le premier ne contient que les x_i et le second les x_i^* . La matrice \mathbf{C} peut s'écrire alors sous la forme :

$$\mathbf{C}_{n \times l} = (x_1 \mathbf{A}_1 + \cdots + x_k \mathbf{A}_k) + (x_1^* \mathbf{B}_1 + \cdots + x_k^* \mathbf{B}_k) = \sum_{i=1}^k (x_i \mathbf{A}_i + x_i^* \mathbf{B}_i) \quad (6)$$

où les matrices \mathbf{A}_i et \mathbf{B}_i sont les matrices constantes $n \times l$ et avec les éléments $\{0, \pm 1\}$ (nous avons supposé que les éléments du \mathbf{C} ne sont que des x_i et x_i^* (qui est vrai dans la majorité des cas). Pour vérifier l'orthogonalité, ces matrices doivent satisfaire les conditions suivantes (preuve dans l'annexe A):

$$\begin{cases} \mathbf{A}_i \mathbf{A}_i^\dagger + \mathbf{B}_i \mathbf{B}_i^\dagger = \mathbf{I}_{n \times n} \\ \mathbf{A}_i \mathbf{A}_j^\dagger + \mathbf{B}_j \mathbf{B}_i^\dagger = \mathbf{O}_{n \times n} \\ \mathbf{A}_i \mathbf{B}_i^\dagger = \mathbf{O}_{n \times n} \\ \mathbf{A}_i \mathbf{B}_j^\dagger + \mathbf{A}_j \mathbf{B}_i^\dagger = \mathbf{O}_{n \times n} \end{cases} \quad \text{pour } i, j = 1, \dots, k \quad \text{et } i \neq j \quad (7)$$

Pour les codes orthogonaux réels les matrices $\mathbf{B}_i = \mathbf{0}_{n \times l}$ et les matrices \mathbf{A}_i sont telles que (preuve dans l'annexe A):

$$\begin{cases} \mathbf{A}_i \mathbf{A}_i^T = \mathbf{I}_{n \times n} \\ \mathbf{A}_i \mathbf{A}_j^T = -\mathbf{A}_j \mathbf{A}_i^T \end{cases} \quad \text{pour } i, j = 1, \dots, k \quad \text{et } i \neq j \quad (8)$$

Nous utiliserons ces propriétés des matrices A et B pour le décodage.

1.3 Décodage ML des codes STBC orthogonaux

Pour expliquer l'algorithme de décodage on revient à l'équation (5)

$$\text{Min } \Omega = \underbrace{- \sum_{j=1}^m \left(\mathbf{r}^j \mathbf{C}^\dagger (\mathbf{h}^j)^\dagger + \mathbf{h}^j \mathbf{C} (\mathbf{r}^j)^\dagger \right)}_{\Omega 1} + \underbrace{\sum_{j=1}^m \mathbf{h}^j \mathbf{C} \mathbf{C}^\dagger (\mathbf{h}^j)^\dagger}_{\Omega 2}$$

Dans cette expression Ω est un scalaire réel, les deux premiers termes ($\Omega 1$) sont linéaires par rapport aux symboles envoyés et le troisième terme ($\Omega 2$) est un polynôme de deuxième degré qui contient les termes $|x_i|^2, x_i x_j^*, x_i x_j, x_i^* x_j^*$.

L'idée du code orthogonal est la suivante: si on pouvait minimiser le Ω , symbole par symbole et non pas sur la séquence entière de taille l symboles, l'algorithme de décodage se simplifierait énormément. Si le code \mathbf{C} est de telle sorte que dans Ω_2 , il ne reste que les termes $|x_i|^2$ et non pas les termes croisés $x_i x_j$, la minimisation de Ω (ou Ω_2) sur k symboles envoyés peut être remplacée par la minimisation de chacune des k métriques symbole par symbole. Ceci peut être obtenu si le code est orthogonal. Par la suite, on ne considère que les codes orthogonaux $\mathbf{C}_{n \times l}$ et on se concentre sur le terme Ω_2 .

$$\begin{aligned} \Omega_2 &= \sum_{j=1}^m \mathbf{h}^j \mathbf{C} \mathbf{C}^\dagger (\mathbf{h}^j)^\dagger = \sum_{j=1}^m \mathbf{h}^j \left(\sum_{q=1}^k |x_q|^2 \right) \mathbf{I} (\mathbf{h}^j)^\dagger = \sum_{q=1}^k |x_q|^2 \cdot \sum_{j=1}^m \mathbf{h}^j (\mathbf{h}^j)^\dagger \\ &= \sum_{q=1}^k |x_q|^2 \cdot \sum_{j=1}^m (\alpha_{1,j} \quad \dots \quad \alpha_{n,j}) \cdot \begin{pmatrix} \alpha_{1,j}^* \\ \vdots \\ \alpha_{n,j}^* \end{pmatrix} = \sum_{q=1}^k |x_q|^2 \cdot \left(\sum_{i,j} |\alpha_{i,j}|^2 \right) \end{aligned} \quad (9)$$

Maintenant on passe au terme Ω_1 , étant donné la linéarité de Ω_1 par rapport à la matrice \mathbf{C} dont chaque élément est une combinaison linéaire de x_1, \dots, x_k (et leur conjugué) et en notant que Ω_1 est un scalaire réel, on pourra dire que :

$$\Omega_1 = \sum_{j=1}^m - \left(\mathbf{r}^j \mathbf{C}^\dagger (\mathbf{h}^j)^\dagger + \mathbf{h}^j \mathbf{C} (\mathbf{r}^j)^\dagger \right) = \sum_{q=1}^k - \left(p_q^* x_q + p_q x_q^* \right) \quad (10)$$

où p_q est un scalaire complexe. Nous aborderons le calcul des p_q plus tard et on se concentre sur la métrique à minimiser. Utilisant les équations (9) et (10), on obtient :

$$\Omega = \sum_{q=1}^k \Omega_q \quad (11)$$

où $\Omega_q = \left(\sum_{i,j} |\alpha_{i,j}|^2 \right) |x_q|^2 - (p_q^* x_q + p_q x_q^*)$.

Comme Ω_q ne dépend que de x_q , minimiser Ω est équivalent à minimiser toutes les sous-métriques Ω_q .

$$\min_{(x_1, \dots, x_k)} \Omega \Leftrightarrow \min_{x_q} \Omega_q ; \quad \forall q = 1, \dots, k$$

Grâce à cette propriété, l'algorithme de décodage s'effectuera beaucoup plus simplement : pour une modulation M-aire la complexité est dimensionnée par kM et non plus par M^k . Par la suite, on se concentre sur la minimisation des sous-métriques Ω_q . Utilisant l'équation (11) on peut écrire :

$$\begin{aligned} \min_{x_q} \Omega_q &= \underbrace{\left(\sum_{i,j} |\alpha_{i,j}|^2 \right)}_{A>0} |x_q|^2 - p_q^* x_q - p_q x_q^* = A |x_q|^2 - p_q^* x_q - p_q x_q^* \\ &= A \left(|x_q|^2 - \frac{p_q^*}{A} x_q - \frac{p_q}{A} x_q^* + \frac{|p_q|^2}{A^2} \right) - \frac{|p_q|^2}{A} \\ \min_{x_q} \Omega_q &\equiv \min_{x_q} \left| x_q - \frac{p_q}{A} \right|^2 = |x_q - p'_q|^2 ; \quad \forall q = 1, 2, \dots, k \end{aligned} \quad (12)$$

Dans l'expression (12), A est un scalaire réel qui dépend des gains du canal, et p_q se calcule à partir des signaux reçus par toutes les antennes de réception et les informations du canal. on appelle $p'_q = \frac{p_q}{A}$ le paramètre de décision du symbole q . Ainsi, le décodage se simplifie et consiste à trouver pour chaque symbole, le point de constellation le plus proche (en terme de la distance euclidienne) au paramètre de décision de ce symbole.

Le résultat obtenu nous permet d'avoir non seulement une décision sur les données envoyées, mais aussi une mesure sur la fiabilité de ce choix : plus le paramètre de décision est proche

d'un point de constellation, plus le symbole détecté est fiable. On peut donc envisager des systèmes mixtes TCM-STBC avec un décodage type "turbo".

1.3.1 Calcul des paramètres de décision

Dans ce paragraphe nous calculons les paramètres de décision p_q . On revient à l'équation (10). Dans cette équation la somme est composée de deux termes l'un conjugué de l'autre. On commence par le deuxième terme et on se focalise sur $\mathbf{h}^j \mathbf{C}$. Utilisant l'équation (6) on peut écrire :

$$\begin{aligned}
 \mathbf{h}^j \mathbf{C} &= \mathbf{h}^j \cdot \sum_{i=1}^k (x_i \mathbf{A}_i + x_i^* \mathbf{B}_i) = \sum_{i=1}^k x_i \mathbf{h}^j \mathbf{A}_i + \sum_{i=1}^k x_i^* \mathbf{h}^j \mathbf{B}_i \\
 &= \underbrace{(x_1 \quad \dots \quad x_i \quad \dots \quad x_k)}_{\mathbf{x}} \cdot \underbrace{\begin{pmatrix} \mathbf{h}^j \mathbf{A}_1 \\ \vdots \\ \mathbf{h}^j \mathbf{A}_i \\ \vdots \\ \mathbf{h}^j \mathbf{A}_k \end{pmatrix}}_{\mathbf{H}_R^j} + \underbrace{(x_1^* \quad \dots \quad x_i^* \quad \dots \quad x_k^*)}_{\mathbf{x}^*} \cdot \underbrace{\begin{pmatrix} \mathbf{h}^j \mathbf{B}_1 \\ \vdots \\ \mathbf{h}^j \mathbf{B}_i \\ \vdots \\ \mathbf{h}^j \mathbf{B}_k \end{pmatrix}}_k \quad (13) \\
 \mathbf{h}^j \mathbf{C} &= \mathbf{x} \cdot \mathbf{H}_R^j + \mathbf{x}^* \cdot \mathbf{H}_I^j
 \end{aligned}$$

On remplace $\mathbf{h}^j \mathbf{C}$ ainsi obtenu dans l'équation (10) on obtient :

$$\begin{aligned}
 \sum_{j=1}^m \mathbf{h}^j \mathbf{C} (\mathbf{r}^j)^\dagger &= \sum_{j=1}^m (\mathbf{x} \cdot \mathbf{H}_R^j + \mathbf{x}^* \cdot \mathbf{H}_I^j) (\mathbf{r}^j)^\dagger = \mathbf{x} \cdot \sum_{j=1}^m \mathbf{H}_R^j (\mathbf{r}^j)^\dagger + \mathbf{x}^* \cdot \sum_{j=1}^m \mathbf{H}_I^j (\mathbf{r}^j)^\dagger \quad (14) \\
 &= \mathbf{x} \cdot \mathbf{H}_R \cdot \mathbf{R}^\dagger + \mathbf{x}^* \cdot \mathbf{H}_I \cdot \mathbf{R}^\dagger
 \end{aligned}$$

où $\mathbf{H}_R = (\mathbf{H}_R^1 \quad \dots \quad \mathbf{H}_R^j \quad \dots \quad \mathbf{H}_R^m)_{k \times ml}$, $\mathbf{H}_I = (\mathbf{H}_I^1 \quad \dots \quad \mathbf{H}_I^j \quad \dots \quad \mathbf{H}_I^m)_{k \times ml}$ et $\mathbf{R} = (\mathbf{r}^1 \quad \dots \quad \mathbf{r}^j \quad \dots \quad \mathbf{r}^m)_{l \times ml}$. Nous allons remplacer ce résultat dans l'équation (10) pour arriver à :

$$\begin{aligned}
 \Omega_1 &= \sum_{j=1}^m -(\mathbf{r}^j \mathbf{C}^\dagger (\mathbf{h}^j)^\dagger + \mathbf{h}^j \mathbf{C} (\mathbf{r}^j)^\dagger) = \\
 &= -\left((\mathbf{x} \cdot \mathbf{H}_R \mathbf{R}^\dagger + \mathbf{x}^* \cdot \mathbf{H}_I \mathbf{R}^\dagger)^* + (\mathbf{x} \cdot \mathbf{H}_R \mathbf{R}^\dagger + \mathbf{x}^* \cdot \mathbf{H}_I \mathbf{R}^\dagger) \right) \quad (15) \\
 &= -\left((\mathbf{x}^* \cdot \mathbf{H}_R^* \mathbf{R}^T + \mathbf{x} \cdot \mathbf{H}_I^* \mathbf{R}^T) + (\mathbf{x} \cdot \mathbf{H}_R \mathbf{R}^\dagger + \mathbf{x}^* \cdot \mathbf{H}_I \mathbf{R}^\dagger) \right)
 \end{aligned}$$

$$= - \left(\underbrace{\mathbf{x} \cdot (\mathbf{H}_R \cdot \mathbf{R}^\dagger + \mathbf{H}_I^* \cdot \mathbf{R}^T)}_{\mathbf{P}_{k \times 1}^\dagger} + \underbrace{\mathbf{x}^* \cdot (\mathbf{H}_I \cdot \mathbf{R}^\dagger + \mathbf{H}_R^* \cdot \mathbf{R}^T)}_{\mathbf{P}_{k \times 1}^T} \right) = -(\mathbf{x} \cdot \mathbf{P}^\dagger + \mathbf{x}^* \cdot \mathbf{P}^T)$$

Où

$$\mathbf{P} = \mathbf{R} \cdot \mathbf{H}_R^\dagger + \mathbf{R}^* \cdot \mathbf{H}_I^T \quad (16)$$

En comparant l'équation (15) avec (10), le vecteur $\mathbf{P}' = (p_1 \ \cdots \ p_q \ \cdots \ p_k) / A$ contient les paramètres de décision. Maintenant, nous avons tous les éléments nécessaires pour le décodage d'un code STBC orthogonal sous forme matricielle. Cette présentation est très utile pour une programmation en Matlab. De plus, grâce à cette démonstration matricielle, on peut trouver les paramètres de décision de tous les symboles d'un paquet d'information d'un seul coup (en calculant le vecteur \mathbf{P} d'après les signaux reçus et les informations du canal et du code).

Cas des Codes réels :

On peut déduire l'algorithme de décodage pour les codes réels à partir de celui des codes complexes. Pour ces codes $\mathbf{B}_i = \mathbf{0}_{n \times l}$, $\mathbf{C}^\dagger = \mathbf{C}^T$ et $x^* = x \cdot \Omega_1$ et Ω_2 se mettent alors sous les formes suivantes :

$$\begin{aligned} \Omega_1 &= \sum_{j=1}^m -(\mathbf{r}^j \mathbf{C}^T (\mathbf{h}^j)^\dagger + \mathbf{h}^j \mathbf{C} (\mathbf{r}^j)^\dagger) = \\ &= \sum_{q=1}^k - (p_q^* x_q + p_q x_q) = \sum_{q=1}^k - 2 \operatorname{real}(p_q) \cdot x_q \end{aligned} \quad (17)$$

$$\Omega_2 = \sum_{q=1}^k x_q^2 \cdot \left(\sum_{i,j} |\alpha_{i,j}|^2 \right) \quad (18)$$

Et on peut également trouver les sous-métriques Ω_q comme suivant :

$$\begin{aligned}
 \underset{x_q}{\text{Min}} \quad \Omega_q &= \underbrace{\left(\sum_{i,j} |\alpha_{i,j}|^2 \right)}_{A>0} x_q^2 - 2\text{real}(p_q) \cdot x_q = Ax_q^2 - 2\text{real}(p_q) \cdot x_q \\
 &= A \left(x_q^2 - 2\text{real}(p_q) \cdot x_q + \frac{(\text{real}(p_q))^2}{A^2} \right) - \frac{(\text{real}(p_q))^2}{A} \quad (19) \\
 \underset{x_q}{\text{Min}} \quad \Omega_q &\equiv \underset{x_q}{\text{Min}} \quad \left(x_q - \frac{\text{real}(p_q)}{A} \right)^2
 \end{aligned}$$

Et pour le cas particulier de la modulation MDP2 la prise de décision s'effectue par utilisation d'un seuil :

$$\hat{x}_q = \text{sign} \left(\frac{\text{real}(p_q)}{A} \right) \quad (20)$$

Pour trouver le vecteur \mathbf{P} , on réécrit les expressions de (13) à (15) pour les codes réels. On obtient les matrices équivalentes pour l'antenne de réception j :

$$\mathbf{h}^j \mathbf{C} = \mathbf{h}^j \cdot \sum_{i=1}^k x_i \mathbf{A}_i = \sum_{i=1}^k x_i \mathbf{h}^j \mathbf{A}_i = \underbrace{\begin{pmatrix} x_1 & \cdots & x_i & \cdots & x_k \end{pmatrix}}_{\mathbf{x}} \cdot \underbrace{\begin{pmatrix} \mathbf{h}^j \mathbf{A}_1 \\ \vdots \\ \mathbf{h}^j \mathbf{A}_i \\ \vdots \\ \mathbf{h}^j \mathbf{A}_k \end{pmatrix}}_{\mathbf{H}_R^j} = \mathbf{x} \cdot \mathbf{H}_R^j \quad (21)$$

En utilisant cette forme équivalente, on calcule Ω_1 .

$$\sum_{j=1}^m \mathbf{h}^j \mathbf{C} (\mathbf{r}^j)^\dagger = \sum_{j=1}^m \mathbf{x} \cdot \mathbf{H}_R^j \cdot (\mathbf{r}^j)^\dagger = \mathbf{x} \cdot \sum_{j=1}^m \mathbf{H}_R^j (\mathbf{r}^j)^\dagger = \mathbf{x} \cdot \mathbf{H}_R \cdot \mathbf{R}^\dagger \quad (22)$$

Avec les mêmes définitions pour \mathbf{H}_R et \mathbf{R} .

$$\begin{aligned}
 \Omega_1 &= \sum_{j=1}^m - \left(\mathbf{r}^j \mathbf{C}^\top (\mathbf{h}^j)^\dagger + \mathbf{h}^j \mathbf{C} (\mathbf{r}^j)^\dagger \right) = - \left((\mathbf{x} \cdot \mathbf{H}_R \mathbf{R}^\dagger)^* + (\mathbf{x} \cdot \mathbf{H}_R \mathbf{R}^\dagger) \right) \\
 &= -\mathbf{x} \cdot \underbrace{\left(\mathbf{H}_R^* \mathbf{R}^\top + \mathbf{H}_R \mathbf{R}^\dagger \right)}_{\mathbf{P}_{k \times l}^\top} = -2\mathbf{x} \cdot \text{real}(\mathbf{P}^\top) \quad (23)
 \end{aligned}$$

Pour les codes réels le vecteur \mathbf{P} simplifie alors sous la forme suivante :

$$\mathbf{P} = \mathbf{R} \mathbf{H}_R^\dagger \quad (24)$$

En comparant l'équation (23) avec (17), le vecteur $\mathbf{P}' = \text{real}(\mathbf{p}_1 \ \cdots \ \mathbf{p}_q \ \cdots \ \mathbf{p}_k) / A$ contient les paramètres de décision.

1.3.2 Algorithme de Décodage :

La Figure 2 illustre le principe du décodage ML pour les codes STBC orthogonaux et complexes. Nous développons par la suite l'algorithme de décodage pour les codes complexes ainsi que pour les codes réels.

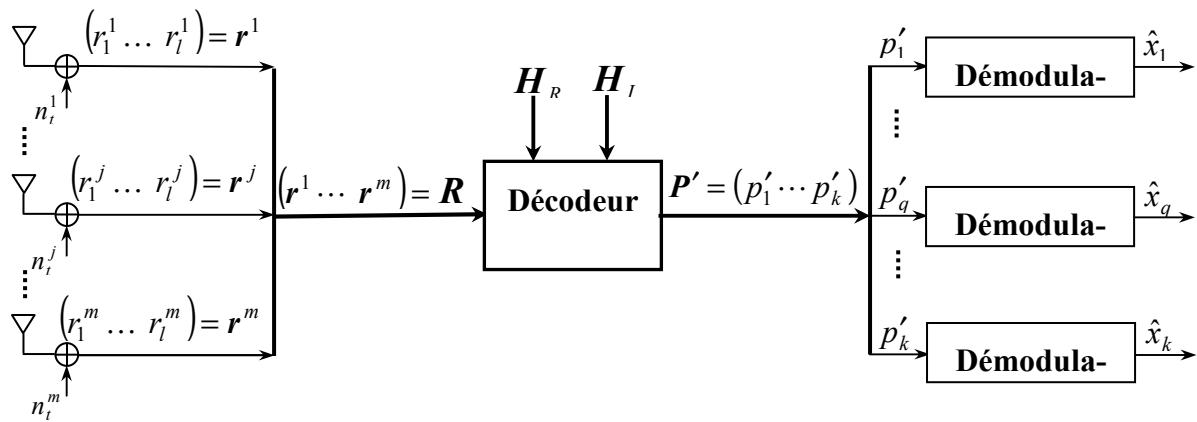


Figure 2 : Structure du décodeur ML pour les codes STBC orthogonaux complexes

Les données du programme :

- Le code, autrement dit les matrices \mathbf{A}_i et \mathbf{B}_i pour $i = 1, \dots, k$.
- Les signaux reçus r_t^j pour $j = 1, \dots, m$; $t = 1, \dots, l$ (Donc le vecteur \mathbf{R}).
- Les informations du canal $\alpha_{i,j}$ pour $\forall i, j$ (Donc les vecteurs \mathbf{h}^j et la matrice \mathbf{H}).

L'algorithme de décodage se résume alors comme suit :

- Trouver les matrices équivalentes \mathbf{H}_R^j et \mathbf{H}_I^j pour $j = 1, \dots, m$ (à partir de l'équation (13)) et par conséquent les matrices \mathbf{H}_R et \mathbf{H}_I (définies dans l'équation (14)).
- Calculer le vecteur \mathbf{P} d'après l'équation (16) et obtenir les paramètres de décision $p'_q = \frac{p_q}{A}$ pour tous les symboles d'un paquet d'information ($q = 1, \dots, k$).
- Prise de décision sur tous les symboles selon l'équation (12).

Cas des codes réels :

Les données du programme :

- Le code, autrement dit les matrices A_i pour $i = 1, \dots, k$.
- Les signaux reçus r_t^j pour $j = 1, \dots, m$; $t = 1, \dots, l$ (Donc le vecteur \mathbf{R}).
- Les informations du canal $\alpha_{i,j}$ pour $\forall i, j$ (Donc les vecteurs \mathbf{h}^j et la matrice \mathbf{H}).

L'algorithme de décodage se résume alors comme suit :

- Trouver les matrices équivalentes \mathbf{H}_R^j pour $j = 1, \dots, m$ (à partir de l'équation (21)) et par conséquent les matrices \mathbf{H}_R (définies dans l'équation (22)).
- Calculer le vecteur \mathbf{P} d'après l'équation (24) et obtenir les paramètres de décision $p'_q = \frac{\text{real}(p_q)}{A}$ pour tous les symboles d'un paquet d'information ($q = 1, \dots, k$).
- Prise de décision sur tous les symboles selon l'équation (19).

1.4 Etude Analytique (Dérivation des Bornes Supérieurs de TEB)

Dans ce paragraphe, on aborde le calcul analytique du TEB pour un décodage ML présenté dans la section précédente. Ce calcul est basé sur des paramètres de décision : le vecteur \mathbf{P} (les équations (16) et (24)). On verra que les paramètres de décision contiennent deux termes, l'un qui est exactement égal au symbole d'information envoyé et l'autre qui correspond au bruit (bruit en sortie du décodeur). Le calcul du TEB nécessite de trouver les statistiques de ce bruit.

Pour commencer, on applique la définition de la matrice \mathbf{R} (utilisée dans l'équation (14)) en utilisant les équations (2) et (13). Avec cette définition, les signaux reçus sur l'ensemble des antennes de réception et dans tous les temps d'un paquet d'information s'écrivent sous la forme suivante :

$$\mathbf{R} = \mathbf{x} \cdot \mathbf{H}_R + \mathbf{x}^* \cdot \mathbf{H}_I + \mathbf{N} \quad (25)$$

où $\mathbf{N} = (\mathbf{n}^1 \quad \dots \quad \mathbf{n}^j \quad \dots \quad \mathbf{n}^m)_{l \times m}$, et \mathbf{n}^j le vecteur de bruit à l'antenne j a déjà été défini dans l'équation (2).

Avant d'appliquer (25) dans l'expression (16), il faut parler de la propriété des matrices \mathbf{H}_R et \mathbf{H}_I (composées des matrices \mathbf{H}_R^j et \mathbf{H}_I^j) qui contiennent les informations du canal ainsi que le code. Pour cela, nous utiliserons les propriétés des matrices de code présentées aux équations (7). En utilisant les définitions des matrices \mathbf{H}_R^j et \mathbf{H}_I^j dans l'équation (13), nous pouvons démontrer que :

$$\mathbf{H}_R^j \cdot (\mathbf{H}_I^j)^\dagger + \left(\mathbf{H}_R^j \cdot (\mathbf{H}_I^j)^\dagger \right)^T = \mathbf{0}_{k \times k} \quad (a)$$

(26)

$$\mathbf{H}_R^j \cdot (\mathbf{H}_R^j)^\dagger + (\mathbf{H}_R^j)^* \cdot (\mathbf{H}_I^j)^T = \sum_{i=1}^n |\alpha_{i,j}|^2 \cdot \mathbf{I}_{k \times k} \quad (b)$$

Les preuves de ces particularités des matrices \mathbf{H}_R^j et \mathbf{H}_I^j se trouvent dans l'annexe B.

L'équation (26)-a signifie que la production matricielle $\left(\mathbf{H}_R^j \cdot (\mathbf{H}_I^j)^\dagger \right)$ est antisymétrique avec des éléments zéros sur la diagonale. La particularité de ce type de matrice est que sa multiplication par un vecteur et son transposé de deux cotés est zéro (l'Annexe C).

A partir de l'équation (26) et en utilisant les définitions des matrices \mathbf{H}_R et \mathbf{H}_I , on peut trouver les expressions semblables pour ces deux matrices :

$$\mathbf{H}_R \cdot \mathbf{H}_I^\dagger + \left(\mathbf{H}_R \cdot \mathbf{H}_I^\dagger \right)^T = \mathbf{0}_{k \times k} \quad (a)$$

(27)

$$\mathbf{H}_R \cdot \mathbf{H}_R^\dagger + \mathbf{H}_I^* \cdot \mathbf{H}_I^T = \sum_{i=1}^n \sum_{j=1}^m |\alpha_{i,j}|^2 \cdot \mathbf{I}_{k \times k} \quad (b)$$

Maintenant, après avoir présenté ces particularités des matrices \mathbf{H}_R et \mathbf{H}_I , on peut se focaliser sur les paramètres de décision (le vecteur \mathbf{P}). Pour cela on remplace le vecteur \mathbf{R} dans l'équation (16) par celui de l'équation (25) et en utilisant les équations (27). On arrive donc à :

$$\begin{aligned} \mathbf{P} &= \mathbf{R} \cdot \mathbf{H}_R^\dagger + \mathbf{R}^* \cdot \mathbf{H}_I^T \\ &= [\mathbf{x} \cdot \mathbf{H}_R + \mathbf{x}^* \cdot \mathbf{H}_I + \mathbf{N}] \cdot \mathbf{H}_R^\dagger + [\mathbf{x}^* \cdot \mathbf{H}_R^* + \mathbf{x} \cdot \mathbf{H}_I^* + \mathbf{N}^*] \cdot \mathbf{H}_I^T \\ &= \mathbf{x} \cdot [\mathbf{H}_R \cdot \mathbf{H}_R^\dagger + \mathbf{H}_I \cdot \mathbf{H}_I^T] + \mathbf{x}^* \cdot [\mathbf{H}_I \cdot \mathbf{H}_R^\dagger + \mathbf{H}_R^* \cdot \mathbf{H}_I^T] + [\mathbf{N} \cdot \mathbf{H}_R^\dagger + \mathbf{N}^* \cdot \mathbf{H}_I^T] \quad (28) \\ &= \left(\sum_{i=1}^m \sum_{j=1}^n |\alpha_{i,j}|^2 \right) \cdot \mathbf{x} + [\mathbf{N} \cdot \mathbf{H}_R^\dagger + \mathbf{N}^* \cdot \mathbf{H}_I^T] \end{aligned}$$

Donc

$$\mathbf{P}' = \mathbf{x} + \frac{1}{\sum_{i,j} |\alpha_{i,j}|^2} [\mathbf{N} \cdot \mathbf{H}_R^\dagger + \mathbf{N}^* \cdot \mathbf{H}_I^T] = \mathbf{x} + \boldsymbol{\eta} \quad (29)$$

où $\boldsymbol{\eta}$ est un vecteur de bruit complexe (et gaussien si les gains du canal sont considérés constants).

$$\boldsymbol{\eta} = \frac{1}{\sum_{i,j} |\alpha_{i,j}|^2} [\mathbf{N} \cdot \mathbf{H}_R^\dagger + \mathbf{N}^* \cdot \mathbf{H}_I^T]$$

Les paramètres de décision que l'on obtient pour le décodeur ML, donnent exactement les symboles d'information contaminés par un bruit additif en sortie de décodeur. On peut donc imaginer que le système est équivalent à un système mono-antenne avec le même type de modulation utilisée dans le système multi-antennes. Par contre, les statistiques du bruit additif équivalent dépendent du canal MIMO.

Afin de trouver une expression analytique pour le TEB, il nous faut calculer les statistiques de ce bruit. On considère dans un premier temps que les gains du canal ($\alpha_{i,j}$) sont constants. Ensuite, on les considère comme les variables aléatoires gaussiennes et complexes, ce qui est généralement le cas dans la pratique.

1.4.1 Canal constant

Pour calculer la matrice d'autocorrélation du vecteur $\boldsymbol{\eta}$, on a besoin des statistiques du vecteur \mathbf{N} . A partir des statistiques des bruits n_i^j , on peut vérifier très facilement que :

$$\begin{aligned} E\{\mathbf{N}^\dagger \mathbf{N}\} &= E\{\mathbf{N}^T \mathbf{N}^*\} = \sigma^2 \mathbf{I}_{ml \times ml} \\ E\{\mathbf{N}^T \mathbf{N}\} &= E\{\mathbf{N}^\dagger \mathbf{N}^*\} = \mathbf{0}_{ml \times ml} \end{aligned} \quad (30)$$

En utilisant les équations (30) et (27-b), on calcule la matrice d'autocorrélation du vecteur bruit $\boldsymbol{\eta}$.

$$E\{\boldsymbol{\eta}^\dagger \boldsymbol{\eta}\} = \frac{1}{\left(\sum_{i,j} |\alpha_{i,j}|^2\right)^2} E\{\left[\mathbf{H}_R \cdot \mathbf{N}^\dagger + \mathbf{H}_I^* \cdot \mathbf{N}^T\right] [\mathbf{N} \cdot \mathbf{H}_R^\dagger + \mathbf{N}^* \cdot \mathbf{H}_I^T]\} \quad (31)$$

$$\begin{aligned}
 &= \frac{1}{\left(\sum_{i,j} |\alpha_{i,j}|^2\right)^2} \left[\mathbf{H}_R \underbrace{E\{N^\dagger N\}}_{\sigma^2 \mathbf{I}_{ml \times ml}} \mathbf{H}_R^\dagger + \mathbf{H}_I^* \underbrace{E\{N^T N^*\}}_{\sigma^2 \mathbf{I}_{ml \times ml}} \mathbf{H}_I^T + \mathbf{H}_R \underbrace{E\{N^\dagger N^*\}}_{\mathbf{0}_{ml \times ml}} \mathbf{H}_I^T + \mathbf{H}_I^* \underbrace{E\{N^T N\}}_{\mathbf{0}_{ml \times ml}} \mathbf{H}_R^\dagger \right] \\
 &= \frac{\sigma^2}{\left(\sum_{i,j} |\alpha_{i,j}|^2\right)^2} [\mathbf{H}_R \mathbf{H}_R^\dagger + \mathbf{H}_I^* \mathbf{H}_I^T] = \frac{\sigma^2}{\sum_{i,j} |\alpha_{i,j}|^2} \mathbf{I}_{k \times k}
 \end{aligned}$$

Cette relation démontre que le bruit du système multi-antennes reste gaussien et indépendant avec une variance atténuee (si $A = \sum_{i,j} |\alpha_{i,j}|^2$ est supérieur à 1, ce qui est souvent le cas).

Comme $\sigma^2 = 1/\text{SNR}$ et que l'énergie de chaque symbole est égale à $1/n$, le rapport équivalent signal à bruit en sortie de décodeur pour chaque symbole se calcule comme suivant.

$$\text{SNR}_q = \frac{1/n}{\sigma^2 / \sum_{i,j} |\alpha_{i,j}|^2} = \left(\sum_{i,j} |\alpha_{i,j}|^2 / n \right) \text{SNR} \quad (32)$$

Le décodeur soft de la Figure 2, calcule les paramètres de décision à partir des sorties des récepteurs et les paramètres de canal. Ces paramètres de décision vont être appliqués au démodulateur qui cherche le symbole le plus proche à ce paramètre dans la constellation envoyée. Cette prise de décision s'effectue séparément pour chaque symbole x_1, x_2, \dots, x_k . Vue l'indépendance des symboles détectés, on se focalise sur la détection d'un symbole pour calculer le taux d'erreur.

Le signal en entrée du démodulateur est composé donc du symbole x_q avec l'énergie $1/n$ et du bruit gaussien et complexe η_q avec la moyenne nulle et la variance $\sigma^2 / \sum_{i,j} |\alpha_{i,j}|^2$. C'est un cas classique qui se trouve dans la littérature [12]-[13] pour lequel il existe des expressions (exacte ou approximatives) pour le taux d'erreur de symbole. Suivant la constellation du signal envoyé, on peut donc donner une expression analytique du taux d'erreur de symbole pour un récepteur ML. C'est une fonction de la distance euclidienne minimale entre les points de constellation d . Nous calculons le taux d'erreur de symbole p_e pour les modulations QPSK, MAQ-16 et MPSK. On définit d'abord les relations suivantes :

$$w = \frac{1}{n} \quad N_0 = \frac{\sigma^2}{\sum_{i,j} |\alpha_{i,j}|^2} \quad p = \frac{1}{2} \operatorname{erfc} \left(\frac{d}{2\sqrt{N_0}} \right) = Q \left(\frac{d}{\sqrt{2N_0}} \right)$$

On obtient donc :

$$\begin{aligned} \text{QPSK : } \quad P_e &= 2p - p^2 \approx 2p \quad ; \quad d = \sqrt{2w} \\ &\Rightarrow P_e = 2Q(\sqrt{SNR_q}) - Q^2(\sqrt{SNR_q}) \approx 2Q(\sqrt{SNR_q}) \end{aligned}$$

$$\text{M-Aire PSK : } P_e \approx 2p \quad ; \quad d = 2\sqrt{w} \sin(\pi/M)$$

$$\Rightarrow P_e \approx 2Q(\sqrt{2SNR_q} \sin(\pi/M))$$

$$\text{QAM-16 : } \quad P_e = 3p - \frac{9}{4}p^2 \approx 3p \quad ; \quad d = 2(\sqrt{10} - \sqrt{2})\sqrt{w}$$

$$\Rightarrow P_e = 3Q(2(\sqrt{5}-1)\sqrt{SNR_q}) - \frac{9}{4}Q^3(2(\sqrt{5}-1)\sqrt{SNR_q}) \approx 3Q(2(\sqrt{5}-1)\sqrt{SNR_q})$$

1.4.2 Canal à variation aléatoire

Maintenant on passe au cas plus réaliste où les gains du canal sont des variables aléatoires. Dans ce cas, pour calculer la matrice d'autocorrélation de bruit, la moyenne s'effectue à la fois sur le bruit et sur les atténuations du canal. Sachant qu'ils sont indépendants, on peut séparer l'opérateur de l'espérance sur chacun des deux variables aléatoires. D'abord, on le fera sur le bruit et ensuite sur les matrices \mathbf{H}_R et \mathbf{H}_I , autrement dit sur les gains $\alpha_{i,j}$.

$$\begin{aligned} E\{\boldsymbol{\eta}^\dagger \boldsymbol{\eta}\} &= E_\alpha \left\{ E_N \left\{ \boldsymbol{\eta}^\dagger \boldsymbol{\eta} \right\} \right\} \\ &= E_\alpha \left\{ \boldsymbol{\sigma}^2 \mathbf{I}_{k \times k} \left/ \left(\sum_{i,j} |\alpha_{i,j}|^2 \right)^2 \right. \right\} = \boldsymbol{\sigma}^2 \mathbf{I}_{k \times k} E_\alpha \left\{ 1 / \sum_{i,j} |\alpha_{i,j}|^2 \right\} \end{aligned} \quad (33)$$

Il nous faut donc l'espérance de l'inverse de $\sum |\alpha_{i,j}|^2$. On sait que les gains du canal $\alpha_{i,j}$ sont les variables aléatoires gaussiens, complexes et indépendantes à variance 0,5 pour chaque partie réelle et imaginaire et à moyenne nulle. Dans ce cas, le module au carré de a_{ij} ($|\alpha_{i,j}|^2$)

suit la loi de Gamma de zéro degré (Gamma(1,0)) avec les paramètres $c = 1$ et $b = 0$ (voir l'Annexe D). La densité de distribution de Gamme est la suivante :

$$y \sim \text{Gamma}(c, N) \Leftrightarrow f_Y(y) = \frac{c^{N+1}}{N!} y^N e^{-cy} U(y)$$

On peut également démontrer que la somme de N variables aléatoire de type Gamme(c,0) suit une densité de probabilité de Gamma de $(N-1)^{\text{ième}}$ degré, Gamma(c,N-1) (Annexe D). Par

conséquent, $z = \sum_{i=1}^n \sum_{j=1}^m |\alpha_{i,j}|^2$ a une densité de probabilité de Gamma(1, $nm - 1$). Le problème

se réduit alors à trouver l'espérance de $E_Z \left\{ \frac{1}{z} \right\}$. On montre dans l'annexe D que cette espérance est égale à $1/nm$. Utilisant ce résultat, l'équation (33) se met sous la forme suivante :

$$E\{\boldsymbol{\eta}^\dagger \boldsymbol{\eta}\} = \sigma^2 E_\alpha \left\{ \frac{1}{z} \right\} = \frac{\sigma^2}{nm} \quad (34)$$

Bien que cette relation donne la matrice d'auto-corrélation du bruit, il ne donne pas sa fonction de distribution. De toute façon le rapport signal à bruit peut être calculé et on remarque une atténuation de la puissance de bruit de mn . Le rapport signal à bruit en sortie de décodeur soft par rapport à un système mono-antenne est donc le suivant.

$$\text{SNR}_q = \frac{1/n}{\sigma^2/mn} = m \times \text{SNR} \quad (35)$$

Nous avons donc réussi à calculer la variance du bruit en sortie du décodeur soft. Mais l'expression de la fonction de distribution de ce bruit s'avère complexe sans quoi, la probabilité d'erreur ne peut pas être estimée. Pour pouvoir débloquer la situation, nous utilisons le théorème de la limite centrale pour le cas où le nombre d'antennes de réception et d'émission (mn) est assez important. En examinant la relation (29), on remarque que le bruit $\boldsymbol{\eta}$ est une somme de ml variables aléatoires. Supposant ml important, on peut modéliser ce bruit avec une distribution gaussienne. Cette condition n'est malheureusement pas vérifiée pour des cas relativement courants où le nombre d'antennes reste assez limité.

Nous utilisons donc la probabilité d'erreur que nous avons calculée pour le cas d'un canal constant. Ensuite, en faisant varier les paramètres du canal, on calcule l'espérance de cette probabilité d'erreur conditionnelle. Autrement dit, on calcule :

$$\bar{P}_e = E_\alpha \{P(e|\alpha)\} \quad (36)$$

où $|\alpha|$ représente l'ensemble des modules de tous les gains complexes de canal : $|\alpha_{i,j}|$ ($i = 1, \dots, n$; $j = 1, \dots, m$) et $P(|\alpha|)$ est la densité de probabilité conjointe de ces derniers. Comme les gains de canal sont gaussiens et indépendants, chaque $|\alpha_{i,j}|$ aura une densité de probabilité de Rayleigh et la densité de probabilité conjointe sera le produit de toutes ces densités de probabilité.

$$\begin{aligned} \alpha_{i,j} &= (\alpha_{i,j}^R, \alpha_{i,j}^I) \sim N(0, 1/2) \Rightarrow P(|\alpha_{i,j}|) = 2|\alpha_{i,j}| e^{-|\alpha_{i,j}|^2} \\ &\Rightarrow P(|\alpha|) = \prod_{i,j} P(|\alpha_{i,j}|) \end{aligned} \quad (37)$$

Utilisant ces deux dernières équations, on va pouvoir calculer P_e .

$$\bar{P}_e = E_\alpha \{P(e|\alpha)\} = \int_{\alpha} P(e|\alpha) P(|\alpha|) d|\alpha| = \int_{\alpha} P(e|\alpha) \prod_{i,j} P(|\alpha_{i,j}|) \prod_{i,j} d|\alpha_{i,j}| \quad (38)$$

Nous allons donc calculer cette probabilité pour un certain nombre de modulations classiques. Ceci se fait en remplaçant le $P(e|\alpha)$ dans la relation ci-dessus par les valeurs trouvées pour le cas du canal constant. D'après nos calculs donnés sur la page 1-17, ces probabilités d'erreur sont des polynômes de $Q(G\sqrt{SNR_q})$ où G n'est qu'un facteur d'échelle. Afin de trouver des expressions simples, on utilise des fonctions approximatives pour la fonction $Q(x)$.[12]-[13] :

$$Q(x) < \frac{1}{2} e^{-\frac{x^2}{2}} \quad (a) \quad (39)$$

$$Q(x) \approx \frac{1}{\sqrt{2\pi x}} e^{-\frac{x^2}{2}} \quad (b)$$

L'expression (39-a) s'utilise afin de trouver une borne supérieure pour le taux d'erreur tandis que l'équation (39-b) est une bonne approximation pour $Q(x)$ (toujours supérieur mais plus proche de $Q(x)$). Afin de trouver une seule expression pour toutes les modulations présentées, on utilise les notations suivantes.

$$P_e = P(e|\alpha) \cong G_1 Q\left(G_2 \sqrt{SNR_q}\right) ; \begin{cases} \text{QPSK : } & G_1 = 2 ; G_2 = 1 \\ \text{M-Aire : } & G_1 = 2 ; G_2 = \sqrt{2} \sin(\pi/M) \\ \text{QAM-16 : } & G_1 = 3 ; G_2 = 2(\sqrt{5} - 1) \end{cases} \quad (40)$$

On peut donc regrouper les équations (32) en utilisant la notation donnée en (40) pour arriver au résultat suivant (dans le quelle l'approximation (39-a) a été utilisée).

$$\begin{aligned} \bar{P}_e &= \int_{|\alpha|} P(e|\alpha) \prod_{i,j} P(|\alpha_{i,j}|) \prod_{i,j} d|\alpha_{i,j}| = \int_{|\alpha|} G_1 Q\left(G_2 \sqrt{SNR_q}\right) \prod_{i,j} \left(2|\alpha_{i,j}| e^{-|\alpha_{i,j}|^2} \right) d|\alpha_{i,j}| \\ &< G_1 \int_{|\alpha|} \frac{1}{2} e^{-\frac{G_2^2 SNR_q}{2}} \prod_{i,j} \left(2|\alpha_{i,j}| e^{-|\alpha_{i,j}|^2} \right) d|\alpha_{i,j}| \\ &< \frac{G_1}{2} \int_{|\alpha|} e^{-G_2^2 SNR \left(\sum |\alpha_{i,j}|^2 / 2n \right)} \prod_{i,j} \left(2|\alpha_{i,j}| e^{-|\alpha_{i,j}|^2} \right) d|\alpha_{i,j}| \\ &< \frac{G_1}{2} \int_{|\alpha|} \prod_{i,j} 2|\alpha_{i,j}| e^{-|\alpha_{i,j}|^2} e^{-G_2^2 SNR |\alpha_{i,j}|^2 / 2n} d|\alpha_{i,j}| \\ &< \frac{G_1}{2} \prod_{i,j} \left(\int_{|\alpha_{i,j}|=0}^{\infty} 2|\alpha_{i,j}| \exp\left(-|\alpha_{i,j}|^2 \left(1 + G_2^2 \frac{SNR}{2n}\right)\right) d|\alpha_{i,j}|\right) \\ &< \frac{G_1}{2} \left(\int_0^{\infty} 2\alpha \exp\left(-\alpha^2 \left(1 + G_2^2 \frac{SNR}{2n}\right)\right) d\alpha \right)^{mn} \\ &< \frac{G_1}{2} \left(\left[-\exp\left(-\alpha^2 \left(1 + G_2^2 \frac{SNR}{2n}\right)\right) \right] \Big/ \left(1 + G_2^2 \frac{SNR}{2n}\right) \right]_0^{\infty} \right)^{mn} \\ &\Rightarrow \bar{P}_e < \frac{G_1}{2 \left(1 + G_2^2 \frac{SNR}{2n}\right)^{mn}} \end{aligned} \quad (41)$$

Cette relation simple donne la borne supérieure du taux d'erreur pour le récepteur ML et pour le code STBC orthogonal et complexe. On remarque que \bar{P}_e diminue de manière exponentielle par rapport au nombre d'antennes.

Pour améliorer l'approximation de la borne supérieure, on recalcule la somme de l'équation (38) en utilisant l'équation l'approximation (39-b) au lieu de (39-a). On a donc :

$$\begin{aligned}
 \bar{P}_e &= \int_{|\alpha|} P(e|\alpha) \prod_{i,j} P(|\alpha_{i,j}|) \prod_{i,j} d|\alpha_{i,j}| = \int_{|\alpha|} G_1 Q\left(G_2 \sqrt{SNR_q}\right) \prod_{i,j} \left(2|\alpha_{i,j}| e^{-|\alpha_{i,j}|^2} \right) d|\alpha_{i,j}| \\
 &= \int_{|\alpha|} \frac{G_1}{\sqrt{2\pi} G_2 \sqrt{SNR_q}} e^{-\frac{G_2^2 SNR_q}{2}} \prod_{i,j} \left(2|\alpha_{i,j}| e^{-|\alpha_{i,j}|^2} \right) d|\alpha_{i,j}| \\
 &= \frac{G_1 \sqrt{n}}{\sqrt{2\pi} G_2 \sqrt{SNR}} \int_{|\alpha|} \frac{1}{\sqrt{\sum_{i,j} |\alpha_{i,j}|^2}} e^{-G_2^2 SNR \left(\sum |\alpha_{i,j}|^2 / 2n \right)} \prod_{i,j} \left(2|\alpha_{i,j}| e^{-|\alpha_{i,j}|^2} \right) d|\alpha_{i,j}| \quad (42) \\
 &= \frac{G_1 \sqrt{n}}{\sqrt{2\pi} G_2 \sqrt{SNR}} \int_{|\alpha|} \frac{1}{\sqrt{\sum_{i,j} |\alpha_{i,j}|^2}} \exp\left(-\left(1 + G_2^2 \frac{SNR}{2n}\right) \sum_{i,j} |\alpha_{i,j}|^2\right) \prod_{i,j} 2|\alpha_{i,j}| d|\alpha_{i,j}|
 \end{aligned}$$

En comparant cette équation avec l'équation (41), on remarque le terme supplémentaire $1/\sqrt{\sum_{i,j} |\alpha_{i,j}|^2}$ qui va largement compliquer le calcul de l'intégrale car il n'est plus possible de le séparer en produit de mn intégrales. Pour résoudre cette intégrale, on se sert d'un changement de repère du cartésien au polaire de dimension mn . Pour cela, et avant tout, on définit les notations suivantes :

$$\begin{aligned}
 M &= mn \quad ; \quad \alpha_d = |\alpha_{i,j}| \quad ; \quad d = 1, 2, \dots, mn \quad ; \quad i = 1, 2, \dots, n \quad ; \quad j = 1, 2, \dots, m \\
 i &= \left\lfloor \frac{d-1}{m} \right\rfloor + 1 \quad , \quad j = (d-1)\%m \quad ; \quad i = (i-1)m + j
 \end{aligned}$$

où $\lfloor y \rfloor$ indique l'entier inférieur le plus proche de y et $\%$ le module de la division de deux entiers.

Avec ces notations, on applique le changement de repère du cartésien au polaire comme suivant :

$$\begin{aligned}
 \alpha_1 &= \rho \cos \varphi_1 \\
 \alpha_i &= \rho \left(\prod_{j=1}^{i-1} \sin \varphi_j \right) \cos \varphi_i \quad ; \quad i = 2, \dots, M-1 \\
 \alpha_M &= \rho \prod_{j=1}^{M-1} \sin \varphi_i
 \end{aligned} \quad (43)$$

La limite de variations de toutes les φ_i est de 0 à $\pi/2$ car les gains α_d sont positifs. On peut vérifier très facilement que $\sum \alpha_d^2 = \rho^2$ et c'est aussi grâce à cela que l'intégrale (42) peut se simplifier. Dans ce changement de repère on peut montrer que [Annexe E] :

$$d\alpha_1 d\alpha_2 \cdots d\alpha_M = \rho^{M-1} (\sin \varphi_1)^{M-2} \cdots (\sin \varphi_i)^{M-1-i} \cdots \sin \varphi_{M-2} d\rho d\varphi_1 d\varphi_2 \cdots d\varphi_{M-1} \quad (44)$$

Maintenant en appliquant ce changement du repère l'intégrale (42) se réécrit sous la forme suivante.

$$\begin{aligned} \bar{P}_e &= \frac{G_1 \sqrt{n}}{\sqrt{2\pi} G_2 \sqrt{SNR}} \int_{|\alpha|} \frac{1}{\sqrt{\sum_{i,j} |\alpha_{i,j}|^2}} \exp\left(-\left(1+G_2^2 \frac{SNR}{2n}\right) \sum_{i,j} |\alpha_{i,j}|^2\right) \prod_{i,j} 2|\alpha_{i,j}| d|\alpha_{i,j}| \\ &= \frac{2^{mn} G_1 \sqrt{n}}{\sqrt{2\pi} G_2 \sqrt{SNR}} \int_{\rho, \varphi} \frac{1}{\rho} e^{-\left(1+G_2^2 \frac{SNR}{2n}\right) \rho^2} \left(\rho^M \prod_{i=1}^{M-1} (\sin \varphi_i)^{M-i} \cos \varphi_i \right) \left(\rho^{M-1} d\rho \prod_{i=1}^{M-1} (\sin \varphi_i)^{M-1-i} d\varphi_i \right) \\ &= \frac{2^{mn} G_1 \sqrt{n}}{\sqrt{2\pi} G_2 \sqrt{SNR}} \int_{\rho=0}^{\infty} \rho^{2M-2} e^{-\left(1+G_2^2 \frac{SNR}{2n}\right) \rho^2} d\rho \prod_{i=1}^{M-1} \int_{\varphi=0}^{\pi/2} (\sin \varphi_i)^{2M-2i-1} \cos \varphi_i d\varphi_i \quad (45) \end{aligned}$$

Dans cette expression, les intégrales concernant les φ_i se calculent facilement alors que pour l'intégrale concernant ρ , on utilise la relation suivante (preuve dans l'annexe E).

$$\int_0^{\infty} \rho^{2K} e^{-\rho^2} d\rho = \frac{(2K)!}{K! 2^{2K+1}} \sqrt{\pi} \quad (46)$$

Pour pouvoir utiliser cette relation, on applique le changement de variable suivant :

$$\rho' = \sqrt{1+G_2^2 \frac{SNR}{2n}} \rho \Rightarrow d\rho' = \sqrt{1+G_2^2 \frac{SNR}{2n}} d\rho'$$

On arrive finalement à :

$$\begin{aligned} \bar{P}_e &\equiv \frac{2^{mn} G_1 \sqrt{n}}{\sqrt{2\pi} G_2 \sqrt{SNR}} \sqrt{\left(1+G_2^2 \frac{SNR}{2n}\right)^{2M-1}} \int_{\rho=0}^{\infty} \rho^{2mn-2} e^{-\rho^2} d\rho \prod_{i=1}^{mn-1} \left[\frac{(\sin \varphi_i)^{2mn-2i}}{2mn-2i} \right]_0^{\frac{\pi}{2}} \\ &= \frac{2^{mn} G_1 \sqrt{n}}{\sqrt{2\pi} G_2 \sqrt{SNR}} \sqrt{\left(1+G_2^2 \frac{SNR}{2n}\right)^{2mn-1}} \left(\frac{(2mn-2)! \sqrt{\pi}}{(mn-1)! 2^{2mn-1}} \right) \prod_{j=1}^{mn-1} \left(\frac{1}{2j} \right) \\ &= \frac{G_1}{\left(1+G_2^2 \frac{SNR}{2n}\right)^{mn}} \sqrt{\frac{1+G_2^2 \frac{SNR}{2n}}{G_2^2 \frac{SNR}{2n}} \left(\frac{(2mn-2)!}{((mn-1)!)^2 2^{2mn-1}} \right)} \quad (47) \end{aligned}$$

Cette relation simple donne une borne supérieure très proche de la probabilité d'erreur du système STBC.

En comparant cette expression avec celle donnée en (41), on remarque le terme commun $G_1 / (1 + G_2^2 \text{SNR} / 2n)^{mn}$ qui diminue de manière exponentielle par rapport au nombre d'antennes (mn). Le terme $\sqrt{\left(1 + G_2^2 \frac{\text{SNR}}{2n}\right) / G_2^2 \frac{\text{SNR}}{2n}}$ s'approche à un pour les rapports SNR assez importants. En revanche pour les SNR très faibles, il devient trop important, ce qui peut se voir aussi par l'approximation (39-b) qui n'est plus une bonne approximation de fonction Q . Cela veut dire que pour les SNR faibles, la précision de cette borne va chuter et on préfère utiliser la borne de l'équation (41). Pour les grands SNR, on peut encore simplifier l'équation (47) pour arriver à :

$$\bar{P}_e \equiv \frac{G_1}{\left(1 + G_2^2 \frac{\text{SNR}}{2n}\right)^{mn}} \left(\frac{(2mn-2)!}{((mn-1)!)^2 2^{2mn-1}} \right) \quad (48)$$

1.4.3 Cas des codes réels

Dans ce paragraphe nous essayons de simplifier les résultats obtenus pour le cas d'une constellation réelle. En utilisant les équations (2) et (21) et la définition de la matrice \mathbf{R} (voir page 1-9), les signaux reçus sur toutes les antennes de réception et pour tous les temps d'un paquet s'écrivent sous la forme suivante :

$$\mathbf{R} = \mathbf{x} \cdot \mathbf{H}_R + \mathbf{N} \quad (49)$$

avec la même définition du vecteur \mathbf{N} donnée par l'équation (25). En utilisant les propriétés de la matrices du code réel (les équations (8)) et la définition des matrices \mathbf{H}_R^j dans l'équation (21), nous pouvons démontrer que (ref. Annexe B) :

$$\text{real}(\mathbf{H}_R^j \cdot (\mathbf{H}_R^j)^\dagger) = \sum_{i=1}^n |\alpha_{i,j}|^2 \cdot \mathbf{I}_{k \times k} \quad (50)$$

Et à partir de cette dernière et la définition des matrices \mathbf{H}_R , on arrive à :

$$\text{real}(\mathbf{H}_R \cdot (\mathbf{H}_R)^\dagger) = (\mathbf{H}_R \mathbf{H}_R^\dagger + \mathbf{H}_R^* \mathbf{H}_R^T) / 2 = \sum_{i=1}^n \sum_{j=1}^m |\alpha_{i,j}|^2 \cdot \mathbf{I}_{k \times k} \quad (51)$$

Maintenant on se concentre sur les paramètres de décision et on remplace le vecteur \mathbf{R} dans l'équation (24) par celui de (49) et en utilisant l'équation (51) on arrive à :

$$\mathbf{P} = \mathbf{R} \cdot \mathbf{H}_R^\dagger = [\mathbf{x} \cdot \mathbf{H}_R + \mathbf{N}] \cdot \mathbf{H}_R^\dagger = \mathbf{x} \cdot [\mathbf{H}_R \cdot \mathbf{H}_R^\dagger] + \mathbf{N} \cdot \mathbf{H}_R^\dagger \quad (52)$$

Donc

$$\mathbf{P}' = \frac{\text{real}(\mathbf{P})}{\sum_{i,j} |\alpha_{i,j}|^2} = \mathbf{x} + \frac{1}{\sum_{i,j} |\alpha_{i,j}|^2} \text{real}(\mathbf{N} \cdot \mathbf{H}_R^\dagger) = \mathbf{x} + \boldsymbol{\eta} \quad (53)$$

où

$$\boldsymbol{\eta} = \frac{1}{\sum_{i,j} |\alpha_{i,j}|^2} \text{real}(\mathbf{N} \cdot \mathbf{H}_R^\dagger) = \frac{1}{2 \sum_{i,j} |\alpha_{i,j}|^2} (\mathbf{N} \cdot \mathbf{H}_R^\dagger + \mathbf{N}^* \cdot \mathbf{H}_R^T)$$

En ce qui concerne les statistiques de bruit additif du canal \mathbf{N} , l'équation (30) est toujours valable. On va alors calculer la matrice d'autocorrélation de vecteur de bruit réel $\boldsymbol{\eta}$, d'abord pour le cas où les gains du canal seraient constants.

$$\begin{aligned} E\{\boldsymbol{\eta}^\dagger \boldsymbol{\eta}\} &= \frac{1}{4 \left(\sum_{i,j} |\alpha_{i,j}|^2 \right)^2} E\{ [\mathbf{H}_R \cdot \mathbf{N}^\dagger + \mathbf{H}_R^* \cdot \mathbf{N}^T] [\mathbf{N} \cdot \mathbf{H}_R^\dagger + \mathbf{N}^* \cdot \mathbf{H}_R^T] \} \quad (54) \\ &= \frac{1}{4 \left(\sum_{i,j} |\alpha_{i,j}|^2 \right)^2} \left[H_R \underbrace{E\{N^* N\}}_{\sigma^2 I_{ml \times ml}} H_R^\dagger + H_R^* \underbrace{E\{N^T N^*\}}_{\sigma^2 I_{ml \times ml}} H_R^T + H_R \underbrace{E\{N^* N^*\}}_{0_{ml \times ml}} H_R^T + H_R^* \underbrace{E\{N^T N\}}_{0_{ml \times ml}} H_R^\dagger \right] \\ &= \frac{\sigma^2}{4 \left(\sum_{i,j} |\alpha_{i,j}|^2 \right)^2} [H_R H_R^\dagger + H_R^* H_R^T] = \frac{\sigma^2}{2 \sum_{i,j} |\alpha_{i,j}|^2} I_{k \times k} \end{aligned}$$

Donc l'énergie de chaque élément de bruit est identique à l'énergie de bruit additif du système multi-antennes divisée par 2A. Le rapport signal à bruit en sortie de décodeur soft pour chaque symbole sera alors :

$$\text{SNR}_k = \frac{1/n}{\sigma^2 / 2 \sum_{i,j} |\alpha_{i,j}|^2} = \left(\frac{2 \sum_{i,j} |\alpha_{i,j}|^2}{n} \right) \text{SNR} \quad (55)$$

Alors pour le cas de la modulation PAM et pour un canal aux coefficients constants, on peut écrire :

$$w = \frac{1}{n} \frac{N_0}{2} = \frac{\sigma^2}{2 \sum_{i,j} |\alpha_{i,j}|^2} \quad p = \frac{1}{2} \operatorname{erfc} \left(\frac{d}{2\sqrt{N_0}} \right) = Q \left(\frac{d}{\sqrt{2N_0}} \right)$$

$$\text{BPSK :} \quad P_e = p \quad ; \quad d = 2\sqrt{w} \quad \Rightarrow \quad P_e = Q \left(\sqrt{SNR_q} \right)$$

$$\text{M-Aire PAM :} \quad P_e = \frac{2(M-1)}{M} p \quad ; \quad d = \frac{4}{M} \sqrt{w} \quad \Rightarrow \quad P_e = \frac{2(M-1)}{M} Q \left(\sqrt{\frac{2}{M} SNR_q} \right)$$

Dans le cas où les gains du canal sont considérés aléatoires, on trouve une expression semblable à l'équation (33).

$$E \{ \boldsymbol{\eta}^\dagger \boldsymbol{\eta} \} = E_\alpha \left\{ E_N \left\{ \boldsymbol{\eta}^\dagger \boldsymbol{\eta} \right\} \right\} = E_\alpha \left\{ \sigma^2 \mathbf{I}_{k \times k} \middle/ 2 \left(\sum_{i,j} |\alpha_{i,j}|^2 \right)^2 \right\} = \frac{\sigma^2}{2} \mathbf{I}_{k \times k} \overbrace{E_\alpha \left\{ 1 \middle/ \sum_{i,j} |\alpha_{i,j}|^2 \right\}}^{1/nm} \quad (56)$$

$$E \{ \boldsymbol{\eta}^\dagger \boldsymbol{\eta} \} = \frac{\sigma^2}{2nm}$$

Alors, l'énergie du bruit équivalent et la même que l'énergie de bruit du système MIMO mais atténuee de $2mn$. Le rapport signal à bruit en sortie du décodeur soft se calcule alors :

$$\text{SNR}_q = \frac{1/n}{\sigma^2 / 2mn} = 2m \times \text{SNR} \quad (57)$$

Mais comme on a vu pour le cas des codes complexes, cette expression n'est pas directement exploitable car le bruit $\boldsymbol{\eta}$ n'est plus gaussian. On peut donc calculer la probabilité d'erreur de symbole comme on a fait pour le cas des codes complexes. C'est-à-dire en calculant pour le cas du canal constant et en évaluant son espérance sur la distribution de probabilité des coefficients du canal. Sachant que les équations sont identiques à ceux que nous avons écrites pour le cas complexe à condition de redéfinir les paramètres G_1 et G_2 .

$$P_e = P(e|\alpha) \cong G_1 Q \left(G_2 \sqrt{SNR_q} \right) ; \begin{cases} \text{BPSK :} & G_1 = 1 ; G_2 = 1 \\ \text{PAM :} & G_1 = \frac{2(M-1)}{M} ; G_2 = \frac{2}{M} \end{cases} \quad (58)$$

Dans ce cas les équations (41) et (47) sont toujours utilisables.

1.5 Résultats des Simulations

Afin d'éprouver la précision des approximations que l'on a trouvées dans la section précédente (équations (41) et (47)), on utilise deux codes STBC orthogonaux donnés par la suite et en faisant varier le nombre d'antennes de réception. Le premier code, connu sous le nom du code d'Alamouti [9], est un code très simple (2×2), avec deux antennes d'émission et un rendement de code égal à 1. Dans le deuxième code, nous avons un code plus compliqué (4×8), avec quatre antennes d'émission et un rendement de code 0,5. Les matrices de code pour ces deux exemples sont les suivantes :

$$C_1 : \begin{pmatrix} x_1 & -x_2 \\ x_2 & x_1^* \end{pmatrix} \quad C_2 : \begin{pmatrix} x_1 & -x_2 & -x_3 & -x_4 & x_1^* & -x_2^* & -x_3^* & -x_4^* \\ x_2 & x_1 & x_4 & -x_3 & x_2^* & x_1^* & x_4^* & -x_3^* \\ x_3 & -x_4 & x_1 & x_2 & x_3^* & -x_4^* & x_1^* & x_2^* \\ x_4 & x_3 & -x_2 & x_1 & x_4^* & x_3^* & -x_2^* & x_1^* \end{pmatrix}$$

Une série de simulations a été faite pour ces deux cas dont les résultats sont présentés par la suite.

La Figure 3 présente la performance du code Alamouti (C_1) en comparant nos bornes supérieures avec les résultats d'une simulation Monte-Carlo. La simulation a été faite pour différents nombres d'antennes de réception.

Le même principe a été effectué pour la Figure 4 mais utilisant le code C_2 . Il faut bien noter que le nombre d'antennes d'émission est fixé par la matrice du code, ce qui n'est pas le cas pour le nombre d'antennes de réception. La précision de la borne supérieure améliorée (équation (47)) est telle que la différence avec les résultats des simulations Monte-Carlo est inférieure de 0,2 dB pour les forts SNRs.

Figure 5 trace la probabilité d'erreur de symbole en fonction du nombre d'antennes m de réception pour un SNR donné (8 dB). L'influence du nombre d'antennes de réception est clairement démontrée et la précision de nos bornes est également validée.

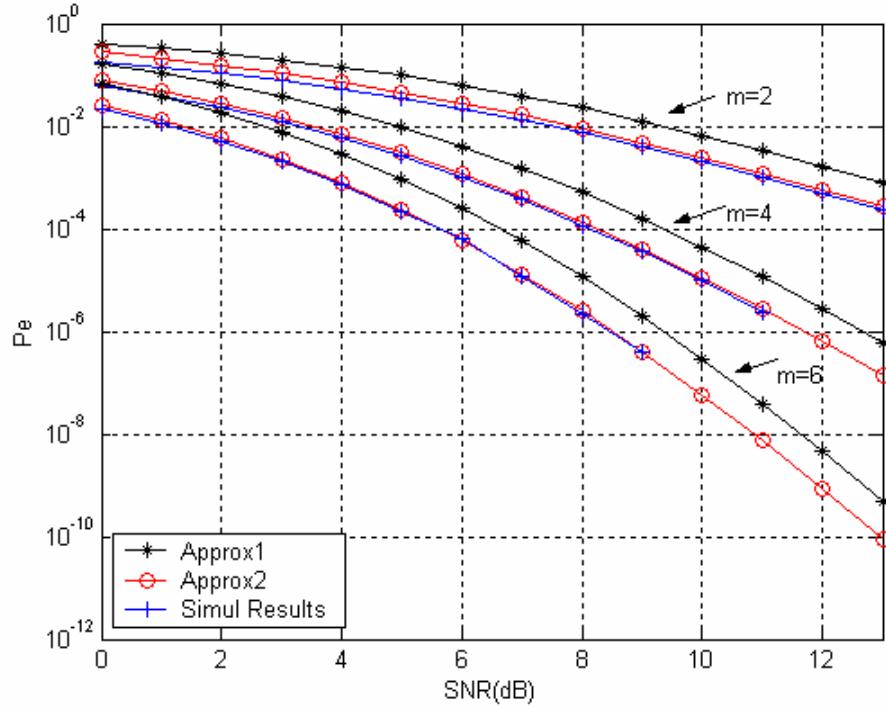


Figure 3 : Probabilité d'erreur de symbole en fonction du SNR (E_s/N_0) pour le code C1

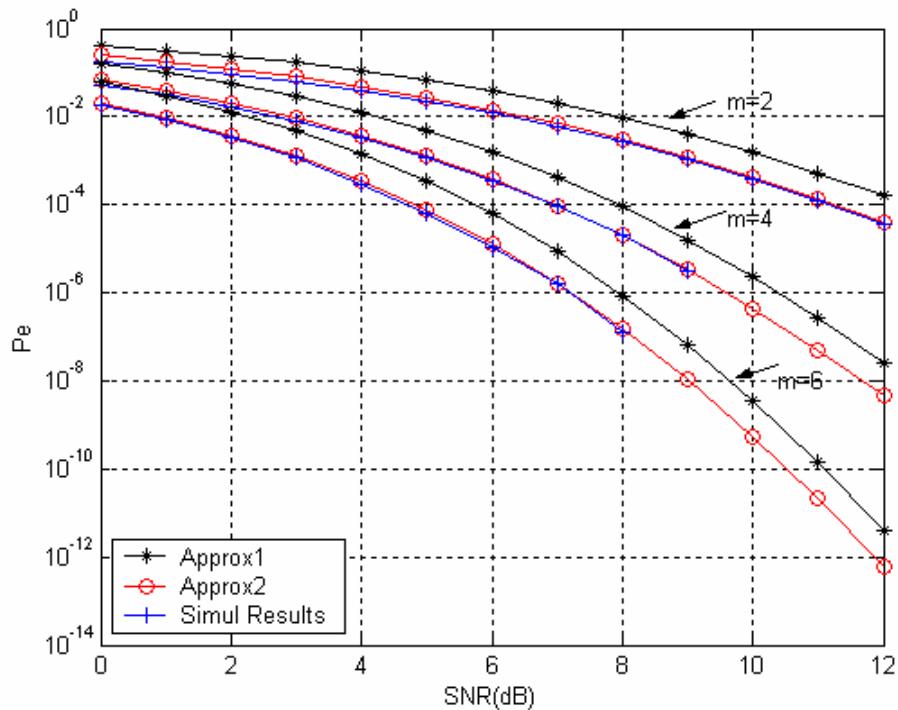


Figure 4 : Probabilité d'erreur de symbole en fonction du SNR (E_s/N_0) pour le code C2

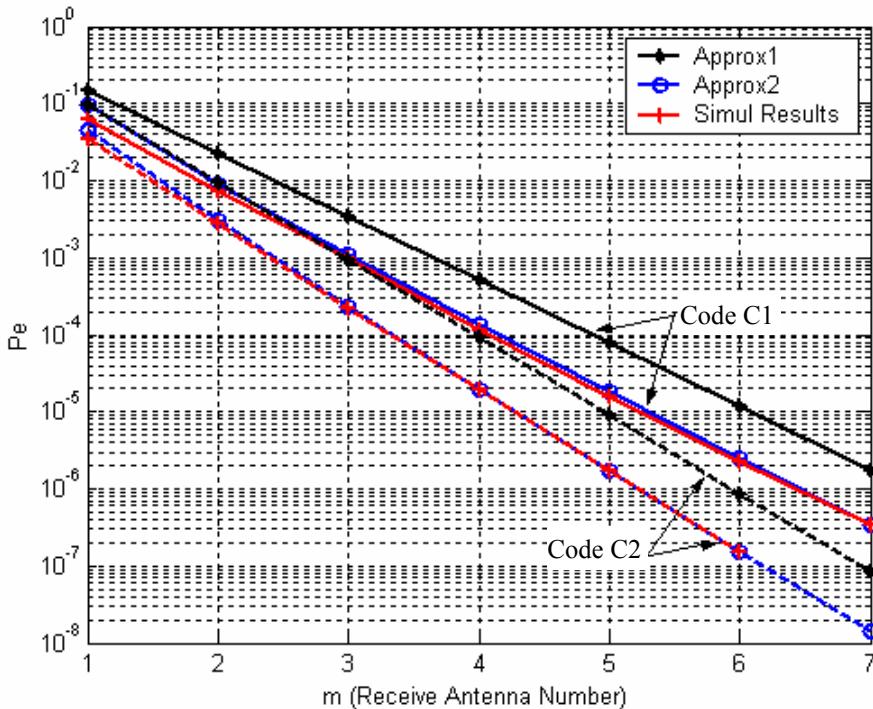


Figure 5 : Probabilité d'erreur de symbole en fonction du nombre d'antennes de réception pour SNR=8dB

1.6 Conclusion

Les principes des codes STBC étant bien développé dans la littérature, l'objectif de ce travail était de trouver, dans un premier temps, une démonstration systématique matricielle pour le décodeur ML des codes STBC orthogonaux (complexes ou réels). Cette forme systématique a aboutit à un algorithme de décodage très facile à implanter. De plus, nous n'imposons aucune limite sur le nombre d'antennes d'émission, de réception ou rendement de code. La seule condition est que le code doit être orthogonal.

Nous avons considéré le code STBC orthogonal typique, avec n antennes d'émission, m antennes de réception et un rendement de k/l . Ce code peut être représenté par une matrice $C_{n \times l}$ qui satisfait l'orthogonalité définie dans la section 1.2.

Ensuite, nous avons fait une étude analytique sur la probabilité d'erreur du décodeur ML pour les codes STBC orthogonaux. Pour le cas du canal aux gains constants, nous avons trouvé les valeurs exactes de la probabilité d'erreur pour certaines modulations. Pour le cas plus réaliste

où les gains du canal seraient aléatoires, et à partir du taux d'erreur du canal constant, nous avons obtenu deux bornes supérieures pour la probabilité d'erreur symbole. Nous avons ensuite comparé ces deux bornes et vu l'utilité de chacune.

Ces travaux ont donné l'objet à des communications internationnelles :

1. G.R. Mohammad-Khani, J.P. Cances, V. Meghdadi, J.M. Dumas, "Derivation of accurate upperbounds for STBC maximum likelihood decoding", IEE Proceedings Circuits, Devices and Systems, Volume 153, Issue 4, August 2006 Page(s):307 - 314
2. G.R. Mohammad Khani, J.P. Cances and V. Meghdadi, "Maximum likelihood decoding rules for STBC: Generalized framework for detection and derivation of accurate upper bounds", Annals of telecommunications, Vol. 59, n°9-10, September-October 2004
3. J.P. Cances, V. Meghdadi, M.J. Syed and G.R. Mohammad-Khani, "Accurate Upper-bounds Derivation for Maximum Likelihood STBC Decoder", 2nd IEEE International Symposium on Image/Video Communications over fixed and mobile networks, 7 - 9 July 2004, Brest, France
4. G.R. Mohammad-Khani, V. Meghdadi and J.P. Cances, "Maximum Likelihood Decoding Rules for STBC : Generalized Framework for Detection and Derivation of Accurate Upperbounds", ICC, 2004, Volume 5, Page(s):2578 - 2583 Vol.5, 20-24 June 2004, Paris

1.7 Références

- [1] N. Balaban and J. Salz, "Dual diversity combining and equalization in digital cellular mobile radio", *IEEE Trans. Veh. Technol.*, vol. 40, pp. 342-354, May 1991.
- [2] G. J. Foschini Jr. and M. J. Gans, "On limits of wireless communication in a fading environment when using multiple antennas", *Wireless Personnal Commun.*, Mar. 1998.
- [3] J. C. Guey, M. P. Fitz, M. R. Bell, and W. Y. Kuo, "Signal design for transmitter diversity wireless communication systems over Rayleigh fading channels", in Proc. VTC'96, 1996, pp. 136-140.
- [4] N. Seshadri and J. H. Winters, "Two signaling schemes for improving the error performance of frequency-division-duplex (FDD) transmission systems using transmitter antenna diversity", *Int. J. Wireless Inform. Networks*, vol. 1, n° 1, 1994.
- [5] E. Telatar, "Capacity of multi-antenna Gaussian channels", AT&T-Bell Laboratories Internal Tehv. Memo., June 1995.
- [6] V. Tarokh, N. Seshadri and A. R. Calderbank, "Space-time codes for high data rate wireless communications : Performance analysis and code construction", *IEEE trans. Inform. Theory*, vol. 44, pp. 744-765, Mar. 1998.
- [7] V. Tarokh, A. Naguib, N. Seshadri and A. R. Calderbank, "Space-time codes for high data rate wireless communications : Performance criteria in the presence of channel estimation errors, mobility and multipaths", *IEEE trans. Commun.*, vol. 47, pp. 199-207, Feb. 1999.
- [8] S. Baro, G. Bauch and A. Hansmann, "Improved codes for space-time trellis-coded modulation", *IEEE Commun. Lett.*, vol. 4, pp. 20-22, Jan. 2000.
- [9] S. M. Alamouti, "A simple transmitter diversity scheme for wireless communications", *IEEE J. Select. Areas. Commun.*, vol. 16, pp. 1451-1458, Oct. 1998.
- [10] V. Tarokh, H. Jafarkhani and A. R. Calderbank, "Space-Time Block Codes from Orthogonal Designs", *IEEE trans. Inform Theory*, vol. 45, pp. 1456-1467, July 1999.

- [11] V. Tarokh, H. Jafarkhani and A. R. Calderbank, "Space-Time Block Coding for wireless communications : Performance results", IEEE J. Select. Areas Commun., vol. 17, pp. 451-460, Mar. 1999.
- [12] S. Benedetto and E. Biglieri, Principles of Digital Transmission With Wireless Applications, Plenum Publishers, 1999.
- [13] J. G. Proakis, Digital Communications, Third Edition, McGraw-Hill, Inc., 1995.
- [14] Papoulis, Probability, Random Variable, and Stochastic Processes,

Annexe A : Preuve des particularités des matrices de codes

En appliquant l'équation (6) dans la définition de l'orthogonalité de code on aura :

$$\begin{aligned}
 \mathbf{C}\mathbf{C}^\dagger &= \left(\sum_{i=1}^k (x_i \mathbf{A}_i + x_i^* \mathbf{B}_i) \right) \left(\sum_{j=1}^k (x_j \mathbf{A}_j + x_j^* \mathbf{B}_j) \right)^\dagger \\
 &= \left(\sum_{i=1}^k (x_i \mathbf{A}_i + x_i^* \mathbf{B}_i) \right) \left(\sum_{i=1}^k (x_i^* \mathbf{A}_i^\dagger + x_i \mathbf{B}_i^\dagger) \right) \\
 &= \sum_{i=1}^k \sum_{j=1}^k (x_i x_j^* \mathbf{A}_i \mathbf{A}_j^\dagger + x_i x_j \mathbf{A}_i \mathbf{B}_j^\dagger + x_i^* x_j^* \mathbf{B}_i \mathbf{A}_j^\dagger + x_i^* x_j \mathbf{B}_i \mathbf{B}_j^\dagger) \\
 &= \sum_{i=1}^k |x_i|^2 (\mathbf{A}_i \mathbf{A}_i^\dagger + \mathbf{B}_i \mathbf{B}_i^\dagger) + \sum_{i=1}^k x_i^2 (\mathbf{A}_i \mathbf{B}_i^\dagger) + \sum_{i=1}^k (x_i^*)^2 (\mathbf{B}_i \mathbf{A}_i^\dagger) \\
 &\quad + \sum_{i=1}^k \sum_{j=i+1}^k (x_i x_j^* (\mathbf{A}_i \mathbf{A}_j^\dagger + \mathbf{B}_j \mathbf{B}_i^\dagger) + x_i^* x_j (\mathbf{A}_j \mathbf{A}_i^\dagger + \mathbf{B}_i \mathbf{B}_j^\dagger)) \\
 &\quad + \sum_{i=1}^k \sum_{j=i+1}^k (x_i x_j (\mathbf{A}_i \mathbf{B}_j^\dagger + \mathbf{A}_j \mathbf{B}_i^\dagger) + x_i^* x_j^* (\mathbf{B}_i \mathbf{A}_j^\dagger + \mathbf{B}_j \mathbf{A}_i^\dagger))
 \end{aligned}$$

Pour que les matrices \mathbf{A}_i et \mathbf{B}_i satisfassent l'orthogonalité de code ($\mathbf{C}\mathbf{C}^\dagger = \sum_{i=1}^k |x_i|^2 \mathbf{I}_{n \times n}$) pour n'importe quels symboles d'informations (x_1, x_2, \dots, x_k) , les coefficients de x_i^2 , $x_i x_j^*$ et $x_i x_j$ doivent être égaux à zéros et les coefficients de $|x_i|^2$ égaux à la matrice unité. Donc les matrices \mathbf{A}_i et \mathbf{B}_i doivent satisfaire l'équation (7).

1.7.1.1 Cas des codes réels :

On applique encore la matrice de codes dans la définition de l'orthogonalité :

$$\begin{aligned}
 \mathbf{C}\mathbf{C}^T &= \left(\sum_{i=1}^k x_i \mathbf{A}_i \right) \left(\sum_{j=1}^k x_j \mathbf{A}_j \right)^T = \left(\sum_{i=1}^k x_i \mathbf{A}_i \right) \left(\sum_{i=1}^k x_i \mathbf{A}_i^T \right) \\
 &= \sum_{i=1}^k \sum_{j=1}^k (x_i x_j \mathbf{A}_i \mathbf{A}_j^T) = \sum_{i=1}^k (x_i^2 \mathbf{A}_i \mathbf{A}_i^T) + \sum_{i=1}^k \sum_{j=i+1}^k (x_i x_j (\mathbf{A}_i \mathbf{A}_j^T + \mathbf{A}_j \mathbf{A}_i^T))
 \end{aligned}$$

Pour que les matrices A_i satisfassent l'orthogonalité de code ($\mathbf{C}\mathbf{C}^T = \sum_{i=1}^k x_i^2 \mathbf{I}_{n \times n}$) pour n'importe quels symboles d'informations (x_1, x_2, \dots, x_k) , les coefficients de $x_i x_j$ soient égaux à zéros et les coefficients de x_i^2 égaux à la matrice unité. Donc les matrices A_i doivent satisfaire l'équation (8).

Annexe B : Preuve des particularités des matrices H_R^j et H_I^j

D'abord on va montrer que la matrice $H_R^j \cdot (H_I^j)^\dagger$ est antisymétrique (et avec les éléments zéros sur le diagonal principal)

$$H_R^j \cdot (H_I^j)^\dagger = \begin{pmatrix} \mathbf{h}^j \mathbf{A}_1 \\ \vdots \\ \mathbf{h}^j \mathbf{A}_i \\ \vdots \\ \mathbf{h}^j \mathbf{A}_k \end{pmatrix}_{k \times l} \cdot \begin{pmatrix} \mathbf{h}^j \mathbf{B}_1 \\ \vdots \\ \mathbf{h}^j \mathbf{B}_q \\ \vdots \\ \mathbf{h}^j \mathbf{B}_k \end{pmatrix}_{l \times k}^\dagger = \begin{pmatrix} \mathbf{h}^j \mathbf{A}_1 \\ \vdots \\ \mathbf{h}^j \mathbf{A}_i \\ \vdots \\ \mathbf{h}^j \mathbf{A}_k \end{pmatrix}_{k \times l} \cdot \begin{pmatrix} \mathbf{B}_1^\dagger (\mathbf{h}^j)^\dagger & \cdots & \mathbf{B}_q^\dagger (\mathbf{h}^j)^\dagger & \cdots & \mathbf{B}_k^\dagger (\mathbf{h}^j)^\dagger \end{pmatrix}_{l \times k}$$

Donc l'élément (i,q) de cette production matricielle est égale à : $\mathbf{h}^j \mathbf{A}_i \cdot \mathbf{B}_q^\dagger (\mathbf{h}^j)^\dagger$, et d'après l'équation (7) on peut apercevoir que $H_R^j \cdot (H_I^j)^\dagger$ est antisymétrique.

Maintenant on passe à l'équation (26-b).

$$H_R^j (H_R^j)^\dagger + (H_I^j)^* (H_I^j)^\dagger = \begin{pmatrix} \mathbf{h}^j \mathbf{A}_1 \\ \vdots \\ \mathbf{h}^j \mathbf{A}_i \\ \vdots \\ \mathbf{h}^j \mathbf{A}_k \end{pmatrix} \left(\mathbf{A}_1^\dagger (\mathbf{h}^j)^\dagger \cdots \mathbf{A}_k^\dagger (\mathbf{h}^j)^\dagger \right) + \begin{pmatrix} (\mathbf{h}^j)^* \mathbf{B}_1^* \\ \vdots \\ (\mathbf{h}^j)^* \mathbf{B}_i^* \\ \vdots \\ (\mathbf{h}^j)^* \mathbf{B}_k^* \end{pmatrix} \left(\mathbf{B}_1^T (\mathbf{h}^j)^T \cdots \mathbf{B}_k^T (\mathbf{h}^j)^T \right)$$

Donc d'après l'équation (7) l'élément (i,i) de la matrice résultat est :

$$\begin{aligned} \mathbf{h}^j \mathbf{A}_i \mathbf{A}_i^\dagger (\mathbf{h}^j)^\dagger + \overbrace{(\mathbf{h}^j)^* \mathbf{B}_i^* \mathbf{B}_i^T (\mathbf{h}^j)^T}^{scalaire réel} &= \mathbf{h}^j \mathbf{A}_i \mathbf{A}_i^\dagger (\mathbf{h}^j)^\dagger + \mathbf{h}^j \mathbf{B}_i \mathbf{B}_i^\dagger (\mathbf{h}^j)^\dagger \\ \mathbf{h}^j (\mathbf{A}_i \mathbf{A}_i^\dagger + \mathbf{B}_i \mathbf{B}_i^\dagger) (\mathbf{h}^j)^\dagger &= \mathbf{h}^j \cdot \mathbf{I}_{n \times n} \cdot (\mathbf{h}^j)^\dagger = \mathbf{h}^j \cdot (\mathbf{h}^j)^\dagger = \sum_{i=1}^n |\alpha_{i,j}|^2 \end{aligned}$$

Et l'élément (i,q) sera comme suivant :

$$\begin{aligned}
 \mathbf{h}^j \mathbf{A}_i \mathbf{A}_q^\dagger (\mathbf{h}^j)^\dagger + \overbrace{(\mathbf{h}^j)^* \mathbf{B}_i^* \mathbf{B}_q^\top (\mathbf{h}^j)^\top}^{\text{scalaire réel}} &= \mathbf{h}^j \mathbf{A}_i \mathbf{A}_q^\dagger (\mathbf{h}^j)^\dagger + \mathbf{h}^j \mathbf{B}_q \mathbf{B}_i^\dagger (\mathbf{h}^j)^\dagger & i \neq q \\
 &= \mathbf{h}^j (\mathbf{A}_i \mathbf{A}_q^\dagger + \mathbf{B}_q \mathbf{B}_i^\dagger) (\mathbf{h}^j)^\dagger = \mathbf{h}^j \cdot \mathbf{0}_{n \times n} (\mathbf{h}^j)^\dagger = 0
 \end{aligned}$$

$$\text{Donc } \mathbf{H}_R^j (\mathbf{H}_R^j)^\dagger + (\mathbf{H}_I^j)^* (\mathbf{H}_I^j)^\top = \sum_{i=1}^n |\alpha_{i,j}|^2 \mathbf{I}_{k \times k}$$

1.7.1.2 Cas des codes réels :

D'abord on va montrer que la matrice $\mathbf{H}_R^j (\mathbf{H}_R^j)^\dagger$ est antisymétrique (mais avec $\sum_i |\alpha_{i,j}|^2$ sur le diagonal principal).

$$\mathbf{H}_R^j (\mathbf{H}_R^j)^\dagger = \begin{pmatrix} \mathbf{h}^j \mathbf{A}_1 \\ \vdots \\ \mathbf{h}^j \mathbf{A}_i \\ \vdots \\ \mathbf{h}^j \mathbf{A}_k \end{pmatrix}_{k \times l} \cdot \begin{pmatrix} \mathbf{h}^j \mathbf{A}_1 \\ \vdots \\ \mathbf{h}^j \mathbf{A}_q \\ \vdots \\ \mathbf{h}^j \mathbf{A}_k \end{pmatrix}_{l \times k}^\dagger = \begin{pmatrix} \mathbf{h}^j \mathbf{A}_1 \\ \vdots \\ \mathbf{h}^j \mathbf{A}_i \\ \vdots \\ \mathbf{h}^j \mathbf{A}_k \end{pmatrix}_{k \times l} \cdot \begin{pmatrix} \mathbf{A}_1^\dagger (\mathbf{h}^j)^\dagger & \cdots & \mathbf{A}_q^\dagger (\mathbf{h}^j)^\dagger & \cdots & \mathbf{A}_k^\dagger (\mathbf{h}^j)^\dagger \end{pmatrix}_{l \times k}$$

Donc d'après l'équation (8), l'élément diagonal (i,i) de matrice résultat est :

$$\mathbf{h}^j \mathbf{A}_i \cdot \mathbf{A}_i^\dagger (\mathbf{h}^j)^\dagger = \mathbf{h}^j \cdot \mathbf{I}_{n \times n} (\mathbf{h}^j)^\dagger = \mathbf{h}^j \cdot (\mathbf{h}^j)^\dagger = \sum_{i=1}^n |\alpha_{i,j}|^2$$

On peut également vérifier, encore en utilisant l'équation (8), que la somme de l'élément (i,q) et (q,i) est égale à zéro (Antisymétrie).

$$\mathbf{h}^j \mathbf{A}_i \cdot \mathbf{A}_q^\dagger (\mathbf{h}^j)^\dagger + \mathbf{h}^j \mathbf{A}_q \cdot \mathbf{A}_i^\dagger (\mathbf{h}^j)^\dagger = \mathbf{h}^j \cdot \underbrace{(\mathbf{A}_i \cdot \mathbf{A}_q^\dagger + \mathbf{A}_q \cdot \mathbf{A}_i^\dagger)}_{\mathbf{0}_{n \times n}} (\mathbf{h}^j)^\dagger = 0$$

Donc la matrice $\mathbf{H}_{RR} = \mathbf{H}_R^j (\mathbf{H}_R^j)^\dagger$ est antisymétrique avec $\sum_i |\alpha_{i,j}|^2$ sur le diagonal.

D'ailleurs cette matrice est hermitien, donc à part les éléments diagonaux, les autres ont des valeurs imaginaires, ce qui peut être trouvé très facilement. Pour cela, on montre l'élément (i,q) de la matrice \mathbf{H}_{RR} par $h_{RR}(i,q)$.

D'après l'herméticité $\Rightarrow h_{RR}(q,i) = h_{RR}^*(i,q)$

$$i \neq q \Rightarrow h_{RR}^*(i, q) = -h_{RR}(i, q) \Rightarrow \text{real}(h_{RR}(i, q)) = 0$$

$$\text{D'après l'anti-symétrie} \Rightarrow h_{RR}(q, i) = -h_{RR}(i, q)$$

Donc l'élément non diagonaux sont imaginaires et par conséquent.

$$\text{real}(\mathbf{H}_R \cdot (\mathbf{H}_R)^\dagger) = \sum_{i=1}^n \sum_{j=1}^m |\alpha_{i,j}|^2 \cdot \mathbf{I}_{k \times k}$$

Annexe C : Particularité des matrices antisymétriques

Dans cette annexe on présente la définition et propriétés des matrices antisymétriques. Une matrice antisymétrique est une matrice carrée qui satisfait la condition suivante :

$$\mathbf{G}_{n \times n} : \text{Antisymétrique} \Leftrightarrow \mathbf{G}^T = -\mathbf{G}$$

Il est claire que les éléments de diagonales sont égaux à zéros. Certaines propriétés de ces matrices sont les suivantes :

1. $\mathbf{G}_{n \times n}$: Antisymétrique $\Leftrightarrow \mathbf{G}^T, \mathbf{G}^\dagger, \mathbf{G}^* : \text{Antisymétrique}$
2. $\mathbf{G}_{n \times n}$ et $\mathbf{J}_{n \times n}$: Antisymétrique $\Rightarrow \mathbf{G} \pm \mathbf{J} : \text{Antisymétrique}$
3. Si on multiplie une matrice antisymétrique par une matrice quelconque d'un coté et par sa transposé de cette matrice de l'autre coté, le résultat sera encore une matrice antisymétrique.

$$\mathbf{G}_{n \times n} : \text{Antisymétrique} ; \mathbf{Q}_{m \times m} = \mathbf{S}_{m \times n} \mathbf{G}_{n \times n} \mathbf{S}_{n \times m}^T \Rightarrow \mathbf{Q}_{m \times m} : \text{Antisymétrique}$$

Il suffit juste de montrer que $\mathbf{Q}^T = -\mathbf{Q}$.

$$\mathbf{Q} = \mathbf{S} \cdot \mathbf{G} \cdot \mathbf{S}^T \Rightarrow \mathbf{Q}^T = \mathbf{S} \cdot \mathbf{G}^T \cdot \mathbf{S}^T = -\mathbf{S} \cdot \mathbf{G} \cdot \mathbf{S}^T = -\mathbf{Q}$$

Si \mathbf{S} est un vecteur de taille $1 \times n$, \mathbf{Q} sera scalaire et $\mathbf{Q}^T = \mathbf{Q}$, d'ailleurs $\mathbf{Q}^T = -\mathbf{Q}$, donc \mathbf{Q} sera égale à zéro.

Annexe D : Densité de probabilité de Gamma

La densité de Gamma se définit comme suivante [14] :

$$y \sim \text{Gamma}(c, b) \Leftrightarrow f_Y(y) = \frac{c^{b+1}}{\Gamma(b+1)} y^b e^{-cy} U(y)$$

où des constants $b \geq 0$ et $c > 0$ sont des paramètres de la densité et $\Gamma(b+1)$ est connu sous le nom de la fonction de Gamma.

$$\Gamma(b+1) = \int_0^\infty x^b e^{-x} dx$$

On peut montrer que pour cette densité les moments m_k s'écrivent sous la forme suivante [14] :

$$m_k = E\{y^k\} = \int_0^\infty y^k \frac{c^{b+1}}{\Gamma(b+1)} y^b e^{-cy} dy = \frac{c^{b+1}}{\Gamma(b+1)} \int_0^\infty y^{b+k} e^{-cy} dy$$

Maintenant avec un changement de variable, pour $k \geq -b$ on aura :

$$m_k = E\{y^k\} = \frac{c^{b+1}}{\Gamma(b+1) c^{b+k+1}} \int_0^\infty x^{b+k} e^{-x} dx = \frac{\Gamma(b+k+1) c^k}{\Gamma(b+1)}$$

Si $b = n$ est un entier, $\Gamma(n+1) = n!$ et on appelle cette densité de probabilité, la densité Gamma de $n^{\text{ième}}$ degré.

$$y \sim \text{Gamma}(c, n) \Leftrightarrow f_Y(y) = \frac{c^{n+1}}{n!} y^n e^{-cy} U(y)$$

Donc pour une densité Gamma(c, n), les moments m_k seront comme suivant :

$$m_k = E\{y^k\} = \frac{(b+k)!}{b!} c^k \quad \text{pour } k \geq -b$$

Maintenant on montre que si x est une variable gaussian complexe avec la moyenne zéro et la variance $\sigma^2/2$ pour chaque partie réelle et imaginaire (ces deux-là sont indépendantes), $|x|^2$ aura une densité de Gamma de zéro degré.

$$x_R \sim N\left(0, \sigma^2/2\right); \quad x_I \sim N\left(0, \sigma^2/2\right); \quad x_R \text{ et } x_I \text{ indépendants} \quad \Rightarrow \quad y = x_R^2 + x_I^2 \sim \text{Gamma}(\sigma^2, 0)$$

Preuve :

$$y = x_R^2 + x_I^2 \quad \theta = \arctg\left(\frac{x_I}{x_R}\right) \Rightarrow f_{Y,\Theta}(y, \theta) = \frac{f_{X_R, X_I}(x_R, x_I)}{\left| \begin{array}{cc} 2x_R & 2x_I \\ x_I & x_R \\ \hline x_R^2 + x_I^2 & x_R^2 + x_I^2 \end{array} \right|} = \frac{1}{4\pi(\sigma^2/2)} e^{-\frac{x_R^2+x_I^2}{2(\sigma^2/2)}} = \frac{1}{2\pi\sigma^2} e^{-\frac{y}{\sigma^2}}$$

$$f_Y(y) = \int f_{Y,\Theta}(y, \theta) d\theta \Rightarrow f_Y(y) = \frac{1}{\sigma^2} e^{-\frac{y}{\sigma^2}}$$

Maintenant on montre par récurrence que la somme de n variables aléatoires indépendantes de type $\text{Gamma}(c, 0)$ sera une variable aléatoire de type $\text{Gamma}(c, n-1)$.

On suppose que $x_i \sim \text{Gamma}(c, 0)$; pour $i = 1, \dots, n$.

Pour $i = 1$: $y_1 = x_1 \Rightarrow y_1 \sim \text{Gamma}(c, 0)$.

Si pour $i = n-1$: $y_{n-1} = \sum_{j=1}^{n-1} x_j \sim \text{Gamma}(c, n-2)$

On va montrer que pour $i = n$: $y_{n-1} + x_n = y_n \sim \text{Gamma}(c, n-1)$.

Preuve : $y = y_{n-1}$; $x = x_n$; $z = y_n = x + y$

$$y \sim \text{Gamma}(c, n-2) \quad x \sim \text{Gamma}(c, 0) \quad y \text{ et } x \text{ indépendants} \Rightarrow f_{X,Y}(x, y) = \frac{c^{n-1}}{(n-2)!} y^{n-2} e^{-cy} c e^{-cx} U(x) U(y)$$

$$f_{X,Y}(x, y) = \frac{c^{n-1}}{(n-2)!} y^{n-2} e^{-c(x+y)} U(x) U(y) \quad f_Z(z) = \int_{-\infty}^{+\infty} f_{X,Y}(z-y, y) dy$$

$$f_Z(z) = \int_0^z \frac{c^n}{(n-2)!} y^{n-2} e^{-cz} dy = \frac{c^n}{(n-2)!} e^{-cz} \int_0^z y^{n-2} dy = \frac{c^n}{(n-1)!} z^{n-1} e^{-cz} U(z)$$

Annexe E : Repère polaire de dimension M

On va montrer que dans un changement de repère du cartésien (y_1, y_2, \dots, y_M) au polaire de M dimension $(\rho, \varphi_1, \varphi_2, \dots, \varphi_{M-1})$ comme suivant :

$$y_1 = \rho \cos \varphi_1 \quad ; \quad y_M = \rho \prod_{j=1}^{M-1} \sin \varphi_i$$

$$y_i = \rho \left(\prod_{j=1}^{i-1} \sin \varphi_j \right) \cos \varphi_i \quad ; \quad i = 1, \dots, M-1$$

On aura :

$$dy_1 dy_2 \cdots dy_M = \rho^{M-1} (\sin \varphi_1)^{M-2} \cdots (\sin \varphi_i)^{M-1-i} \cdots \sin \varphi_{M-2} d\rho d\varphi_1 d\varphi_2 \cdots d\varphi_{M-1}$$

Pour cela on doit développer la formule suivante.

$$dy_1 dy_2 \cdots dy_M = \begin{vmatrix} \frac{\partial y_1}{\partial \rho} & \frac{\partial y_1}{\partial \varphi_1} & \cdots & \frac{\partial y_1}{\partial \varphi_{M-1}} \\ \frac{\partial y_2}{\partial \rho} & \frac{\partial y_2}{\partial \varphi_1} & \cdots & \frac{\partial y_2}{\partial \varphi_{M-1}} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial y_M}{\partial \rho} & \frac{\partial y_M}{\partial \varphi_1} & \cdots & \frac{\partial y_M}{\partial \varphi_{M-1}} \end{vmatrix} d\rho d\varphi_1 d\varphi_2 \cdots d\varphi_{M-1}$$

$$\Delta_M(\rho, \varphi_1, \dots, \varphi_{M-1})$$

Donc on va montrer par récurrence que $\Delta_M(\rho, \varphi_1, \dots, \varphi_{M-1}) = \rho^{M-1} \prod_{i=1}^{M-1} (\sin \varphi_i)^{M-1-i}$.

Pour $i = 2$: $dy_1 dy_2 = \rho d\rho d\varphi_1 \Rightarrow \Delta_2(\rho, \varphi_1) = \begin{vmatrix} \cos \varphi_1 & -\rho \sin \varphi_1 \\ \sin \varphi_1 & \rho \cos \varphi_1 \end{vmatrix} = \rho$

Pour $i = M-1$: $\Delta_{M-1}(\rho, \varphi_2, \dots, \varphi_{M-1}) = \rho^{M-2} \prod_{i=2}^{M-1} (\sin \varphi_i)^{M-1-i}$

Pour $i = M \geq 2$:

$$\Delta_M = \begin{vmatrix} \cos \varphi_1 & -\rho \sin \varphi_1 & 0 & \cdots & \cdots & \cdots & \cdots & \cdots & 0 \\ \sin \varphi_1 \cos \varphi_2 & \rho \cos \varphi_1 \cos \varphi_2 & -\rho \sin \varphi_1 \sin \varphi_2 & 0 & \cdots & \cdots & \cdots & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & & & & \vdots \\ \prod_{j=1}^{i-1} \sin \varphi_j \cos \varphi_i & \rho \cos \varphi_1 \prod_{j=2}^{i-1} \sin \varphi_j \cos \varphi_i & \cdots & \cdots & \rho \cos \varphi_k \prod_{\substack{j=1 \\ j \neq k}}^{i-1} \sin \varphi_j \cos \varphi_i & \cdots & -\rho \prod_{j=1}^i \sin \varphi_j & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \ddots & & & \vdots \\ \prod_{j=1}^{M-2} \sin \varphi_j \cos \varphi_{M-1} & \rho \cos \varphi_1 \prod_{j=2}^{M-2} \sin \varphi_j \cos \varphi_{M-1} & & & \rho \cos \varphi_k \prod_{\substack{j=1 \\ j \neq k}}^{M-2} \sin \varphi_j \cos \varphi_{M-1} & \cdots & \cdots & \cdots & -\rho \prod_{j=1}^{M-1} \sin \varphi_j \\ \prod_{j=1}^{M-1} \sin \varphi_j & \rho \cos \varphi_1 \prod_{j=2}^{M-1} \sin \varphi_j & \cdots & \cdots & \rho \cos \varphi_k \prod_{\substack{j=1 \\ j \neq k}}^{M-1} \sin \varphi_j & \cdots & \cdots & \cdots & \rho \cos \varphi_{M-1} \prod_{j=1}^{M-1} \sin \varphi_j \end{vmatrix}$$

Pour calculer ce déterminant on le développe sur la première ligne. Dans les déterminants de $\Delta_{1,1}$ et $\Delta_{1,2}$, il existe respectivement des termes de $\rho \cos \varphi_1$ et $\sin \varphi_1$ à factoriser et après ces factorisations, on arrive à une matrice identique.

$$\Delta_M = \cos \varphi_1 \Delta_{1,1} + \rho \sin \varphi_1 \Delta_{1,2} = \rho (\cos \varphi_1)^2 |\Lambda| + \rho (\sin \varphi_1)^2 |\Lambda| = \rho |\Lambda|$$

où

$$|\Lambda| = \begin{vmatrix} \cos \varphi_2 & -\rho \sin \varphi_1 \sin \varphi_2 & 0 & \cdots & \cdots & \cdots & \cdots & \cdots & 0 \\ \vdots & \vdots & & & & & & & \vdots \\ \prod_{j=2}^{i-1} \sin \varphi_j \cos \varphi_i & \cdots & \cdots & \rho \cos \varphi_k \prod_{\substack{j=1 \\ j \neq k}}^{i-1} \sin \varphi_j \cos \varphi_i & \cdots & -\rho \prod_{j=1}^i \sin \varphi_j & 0 & \cdots & 0 \\ \vdots & \vdots & & \vdots & \ddots & & & & \vdots \\ \prod_{j=2}^{M-2} \sin \varphi_j \cos \varphi_{M-1} & & & \rho \cos \varphi_k \prod_{\substack{j=1 \\ j \neq k}}^{M-2} \sin \varphi_j \cos \varphi_{M-1} & & & & & -\rho \prod_{j=1}^{M-1} \sin \varphi_j \\ \prod_{j=2}^{M-1} \sin \varphi_j & \cdots & \cdots & \rho \cos \varphi_k \prod_{\substack{j=1 \\ j \neq k}}^{M-1} \sin \varphi_j & \cdots & \cdots & \cdots & \cdots & \rho \cos \varphi_{M-1} \prod_{j=1}^{M-1} \sin \varphi_j \end{vmatrix}$$

Et dans ce déterminant dans toutes les colonnes (à part le premier), il existe un terme de $\sin \varphi_1$ à factoriser et après cette factorisation on arrive à une matrice de la même structure que Δ_M , mais avec la dimension $M-1$ et par rapport aux variables $(\rho, \varphi_2, \dots, \varphi_{M-1})$.

$$|\Lambda| = (\sin \varphi_1)^{M-2} \begin{vmatrix} \cos \varphi_2 & -\rho \sin \varphi_2 & 0 & \cdots & \cdots & \cdots & \cdots & \cdots & 0 \\ \vdots & \vdots & & & & & & & \vdots \\ \prod_{j=2}^{i-1} \sin \varphi_j \cos \varphi_i & \cdots & \cdots & \rho \cos \varphi_k \prod_{\substack{j=2 \\ j \neq k}}^{i-1} \sin \varphi_j \cos \varphi_i & \cdots & -\rho \prod_{j=2}^i \sin \varphi_j & 0 & \cdots & 0 \\ \vdots & \vdots & & \vdots & \ddots & & & & \vdots \\ \prod_{j=2}^{M-2} \sin \varphi_j \cos \varphi_{M-1} & & & \rho \cos \varphi_k \prod_{\substack{j=2 \\ j \neq k}}^{M-2} \sin \varphi_j \cos \varphi_{M-1} & & & & & -\rho \prod_{j=2}^{M-1} \sin \varphi_j \\ \prod_{j=2}^{M-1} \sin \varphi_j & \cdots & \cdots & \rho \cos \varphi_k \prod_{\substack{j=2 \\ j \neq k}}^{M-1} \sin \varphi_j & \cdots & \cdots & \cdots & \cdots & \rho \cos \varphi_{M-1} \prod_{j=2}^{M-1} \sin \varphi_j \end{vmatrix}$$

$$|\Lambda| = (\sin \varphi_1)^{M-2} \Delta_{M-1}(\rho, \varphi_2, \dots, \varphi_{M-1})$$

On peut donc réécrire Δ_M sous la forme suivante.

$$\begin{cases} \Delta_M = \rho |\Lambda| = \rho (\sin \varphi_1)^{M-2} \Delta_{M-1}(\rho, \varphi_2, \dots, \varphi_{M-1}) \\ \Delta_{M-1}(\rho, \varphi_2, \dots, \varphi_{M-1}) = \rho^{M-2} \prod_{i=2}^{M-1} (\sin \varphi_i)^{M-1-i} \end{cases} \Rightarrow \Delta_M(\rho, \varphi_1, \dots, \varphi_{M-1}) = \rho^{M-1} \prod_{i=1}^{M-1} (\sin \varphi_i)^{M-1-i}$$

On a finalement trouvé $\Delta_M(\rho, \varphi_1, \dots, \varphi_{M-1})$, et on peut maintenant utiliser le repère polaire en

remplaçant $dy_1 dy_2 \cdots dy_M$ par $\rho^{M-1} \prod_{j=1}^{M-1} (\sin \varphi_i)^{M-1-i} \cdot d\rho d\varphi_1 d\varphi_2 \cdots d\varphi_{M-1}$.

Preuve de l'équation (46) : $I = \int_0^\infty \rho^{2K} e^{-\rho^2} d\rho = \frac{(2K)!}{K! 2^{2K+1}} \sqrt{\pi}$

Avec un changement de variable $\rho = y/\sqrt{2}$ on aura :

$$I = \frac{1}{2^K \sqrt{2}} \int_0^\infty y^{2K} e^{-y^2/2} dy = \frac{1}{2^K \sqrt{2}} I_K$$

Donc on va montrer par récurrence que :

$$I_K = \int_0^\infty y^{2K} e^{-y^2/2} dy = \frac{(2K)!}{K! 2^{K+1}} \sqrt{2\pi}$$

Pour $i = 0$: $I_0 = \int_0^\infty e^{-y^2/2} dy = \frac{\sqrt{2\pi}}{2}$

Pour $i = K - 1$, nous avons $I_{K-1} = \int_0^\infty y^{2K-2} e^{-y^2/2} dy = \frac{(2K-2)!}{(K-1)! 2^K} \sqrt{2\pi}$.

Pour $i = K \geq 1$:

$$I_K = \int_0^\infty y^{2K} e^{-y^2/2} dy = \int_0^\infty \underbrace{(y^{2K-1})}_u \underbrace{ye^{-y^2/2} dy}_{dv} = \left[-y^{2K-1} e^{-y^2/2} \right]_0^\infty + (2K-1) \int_0^\infty y^{2K-2} e^{-y^2/2} dy$$

$$I_K = (2K-1)I_{K-1} = (2K-1) \frac{(2K-2)!}{(K-1)! 2^K} \sqrt{2\pi} = \frac{2K}{2K} \frac{(2K-1)!}{(K-1)! 2^K} \sqrt{2\pi} = \frac{(2K)!}{K! 2^{K+1}} \sqrt{2\pi}$$

Chapitre 2 Synchronisation dans les systèmes MIMO-OFDM

2.1 *Introduction*

Les canaux multi trajets, appelés aussi des canaux sélectifs en fréquence, sont des canaux réalistes pour les systèmes de télécommunication mobile. L'utilisation de la modulation OFDM permet de simplifier la tâche d'identification de canal et de l'égaliser. En effet, un canal multi trajets se transforme en plusieurs canaux mono trajets et non sélectifs en fréquence. C'est la raison pour laquelle dans les environnements "*indoor*" ou dans les milieux riches en trajets multiples, ce type de modulation est largement utilisé.

Les tâches à effectuer dans un récepteur numérique sont les suivantes :

- détecter le début d'un paquet
- identifier le canal
- compenser l'erreur de fréquence due à la fréquence doppler ou/et à la différence de fréquence des oscillateurs locaux
- détecter les symboles envoyés

Le mode de transmission que nous allons étudier dans ce rapport est le mode paquet où chaque paquet est précédé d'une séquence d'apprentissage (préambule) qui sert à aider le récepteur à se synchroniser sur le paquet arrivé.

Après un aperçu succinct des systèmes OFDM et puis MIMO-OFDM¹, nous présenterons, dans le paragraphe 2.5, la synchronisation en temps en présence d'une erreur de fréquence. Nous présenterons la méthode la plus utilisée donnée dans la littérature. Le problème lié à

¹ *Multiple Input Multiple Output*

cette méthode est que la fonction d'autocorrélation présente un plateau au lieu d'un pic pointu. Ce pic aurait été utilisé pour donner le temps exact d'arrivée d'un symbole OFDM. Nous avons proposé [1] de remplacer la fonction d'autocorrélation par une fonction particulière qui donne une estimation précise de début du symbole OFDM. Un algorithme simple a été aussi développé pour la conception des séquences d'apprentissage à envoyer par les antennes d'émission.

Le paragraphe 2.6 traite l'estimation de l'offset de fréquence connaissant le timing. Utilisant notre méthode de timing exact, nous pouvons nous rapprocher de la borne de Cramer-Rao mieux qu'avec les méthodes précédentes exposées dans la littérature.

Dans le paragraphe 2.7, nous développons l'estimation conjointe du canal avec la synchronisation de temps en utilisant la méthode de maximum de vraisemblance¹. Ici, nous avons supposé que l'erreur de fréquence était nulle.

Dans le paragraphe 2.8, nous introduisons une erreur de fréquence et nous proposons par la suite, une estimation conjointe de canal, de fréquence offset et de temps. Cette méthode est limitée, par construction, en résolution de fréquence, nous proposons une méthode itérative afin d'améliorer notre estimée de fréquence. La simulation montre qu'avec cette méthode nous atteignons la borne de Cramer-Rao au terme de quelques itérations.

2.2 Modulation OFDM

La modulation OFDM a été sélectionnée dans la norme IEEE 802.11 pour les réseaux locaux (LAN), pour les communications mobiles multimédia, pour la Télévision Numérique Terrestre (TNT), pour l'ADSL, et considérée comme le candidat potentiel pour être utilisé dans la quatrième génération des portables. La Figure 2-1 présente le modulateur OFDM qui applique l'inverse d'une transformée de Fourier discrète (TFD⁻¹) sur les symboles avant de les transmettre dans le canal.



Figure 2-1-Modulateur OFDM

¹ Maximum Likelihood ou ML

Outre les interprétations fréquentielles dues à l'utilisation de TFD^{-1} que l'on peut donner sur ce schéma, nous remarquons qu'une modulation OFDM n'est qu'une projection d'un vecteur dans un espace de dimension N vers le même espace utilisant une matrice de transformation : la matrice de TFD^{-1} présentée ci-dessous :

$$\mathbf{Q}^H = \begin{bmatrix} 1 & 1 & 1 & \dots & 1 \\ 1 & w^{-1} & w^{-2} & \dots & w^{-(n-1)} \\ 1 & w^{-2} & w^{-4} & \dots & w^{-2(n-1)} \\ \vdots & \vdots & \vdots & & \vdots \\ 1 & w^{-(n-1)} & w^{-2(n-1)} & \dots & w^{-(n-1)(n-1)} \end{bmatrix} \quad (59)$$

avec $w = e^{-j\frac{2\pi}{n}}$. Alors, au lieu d'envoyer le vecteur dont les éléments sont des symboles à émettre, on envoie les coordonnées du vecteur résultant. Pouvant diagonaliser une matrice circulante, cette transformation linéaire est devenue bien plus populaire que d'autres types de transformation et permet de décorrélérer les symboles dans un environnement multi trajets (voir Annexe I). Soit le vecteur \mathbf{S} le vecteur des symboles d'information à envoyer : $\mathbf{S} = [s_0, s_1, \dots, s_{N-1}]^T$, la sortie de la TFD^{-1} peut s'écrire sous la forme suivante :

$$\mathbf{C} = \mathbf{Q}^H \mathbf{S} \quad (60)$$

L'interprétation fréquentielle de cette relation est que le symbole s_i à envoyer est modulé par $\exp(j2\pi in/N)$. Nous avons donc N porteuses avec un espacement fréquentiel $2\pi/N$ entre deux porteuses successives pour envoyer pendant NT_s le symbole s_i . Les sorties de ces N modulateurs sont ensuite additionnées. Il y a donc N porteuses chacune modulée par une séquence de symboles sortant du bloc convertisseur série - parallèle.

Avant d'envoyer ce signal sur un canal multi trajets, on y ajoute un préfixe cyclique et on le passe à travers le filtre d'émission. Le premier rôle du préfixe cyclique est l'ajout d'un intervalle de garde pour que les deux symboles successifs, chacun de taille N échantillons que l'on appelle symbole OFDM, ne se chevauchent pas. En fait, sans cet intervalle, chaque symbole est affecté par le symbole précédent à cause du canal multi trajets. Le deuxième rôle de ce préfixe **cyclique** est de transformer une convolution linéaire entre le signal émis et le canal, modélisé par un filtre FIR, en une convolution circulaire de taille N . On remarque que cette convolution circulaire nous permet de multiplier directement la TFD du canal par les symboles d'information avant effectuer la TFD^{-1} . L'ajout du préfixe circulaire peut être présenté par la multiplication matricielle suivante:

$$\tilde{\mathbf{C}} = \Theta \cdot \mathbf{C} = \begin{bmatrix} \mathbf{0}_{N_g \times (N-N_g)} | \mathbf{I}_{N_g \times N_g} \\ \mathbf{I}_{N \times N} \end{bmatrix}_{(N+N_g) \times N} \begin{bmatrix} \mathbf{c}_0 \\ \mathbf{c}_1 \\ \dots \\ \mathbf{c}_{N-1} \end{bmatrix}_{N \times 1} = \begin{bmatrix} \mathbf{c}_{N-N_g} \\ \dots \\ \mathbf{c}_{N-1} \\ \mathbf{c}_0 \\ \mathbf{c}_1 \\ \dots \\ \mathbf{c}_{N-1} \end{bmatrix}_{(N+N_g) \times 1} \quad (61)$$

où N_g est la taille de l'intervalle de garde. Le vecteur à envoyer peut s'écrire alors :

$$\tilde{\mathbf{C}} = \Theta \mathbf{Q}^H \mathbf{S} \quad (62)$$

2.3 Canal sélectif en fréquence

Le canal est un canal multi trajets, donc sélectif en fréquence, et modélisé par un filtre RIF de taille L en supposant que la taille du préfixe N_g est supérieur à L . Les coefficients du canal sont des variables aléatoires complexes de moyenne nulle et de variance $\sigma(l)^2$ où l'indice l indique le $l^{\text{ième}}$ trajet.

Le canal de transmission se caractérise par le vecteur $\mathbf{h} = [h_0, h_1, \dots, h_{L-1}]$. Ne considérant pas pour le moment le bruit additif gaussien, la sortie du canal à l'instant n est le résultat de la convolution de la séquence \mathbf{h} avec le signal émis :

$$y(n) = \mathbf{h}^* [x_{n-L+1}, \dots, x_{n-1}, x_n] = [h_{L-1}, h_{L-2}, \dots, h_0] \cdot [x_{n-L+1}, \dots, x_{n-1}, x_n]^T \quad (63)$$

où x_i est l'échantillon envoyé à l'instant i . Pour une séquence de sortie de taille $N+N_g$ on peut écrire :

$$\mathbf{Y}' = \mathbf{H}' \cdot \mathbf{X}_{(N+N_g) \times (N+N_g+L-1)} \quad (64)$$

Avec

$$\mathbf{H}' = \begin{bmatrix} h_{L-1} & h_{L-2} & \dots & h_0 & 0 & 0 & 0 \\ 0 & h_{L-1} & h_{L-2} & \dots & h_0 & 0 & 0 \\ 0 & \dots & \dots & \dots & \dots & \dots & \dots \\ 0 & \dots & 0 & h_{L-1} & h_{L-2} & \dots & h_0 \end{bmatrix}_{(N+N_g) \times (N+N_g+L-1)} \quad (65)$$

En prenant \mathbf{Y} une séquence de taille N et synchronisée avec la fin du symbole, et profitant du préfixe cyclique disponible dans le message envoyé, l'équation (64) peut s'écrire sous la forme :

$$\mathbf{Y} = \mathbf{H}_{N \times N} \mathbf{C}_{N \times 1} \quad (66)$$

où \mathbf{H} est la matrice circulante suivante :

$$\mathbf{H} = \begin{bmatrix} h_0 & 0 & \cdots & 0 & h_{L-1} & h_{L-2} & \cdots & h_1 \\ h_1 & h_0 & 0 & \cdots & 0 & h_{L-1} & \cdots & h_2 \\ \cdots & \cdots \\ 0 & \cdots & 0 & h_{L-1} & h_{L-2} & h_{L-3} & \cdots & h_0 \end{bmatrix}_{N \times N} \quad (67)$$

La séquence reçue est donc :

$$\mathbf{r} = \mathbf{H}_{N \times N} \mathbf{C}_{N \times 1} + \mathbf{v}_{N \times 1} \quad (68)$$

où \mathbf{v} est la séquence du bruit additif complexe avec la matrice de covariance $\sigma^2 I_{N \times N}$.

2.4 MIMO-OFDM

La généralisation des systèmes SISO (*single Input Single Output*) en MIMO se fait de différentes façons. Ces différentes méthodes ont un point commun, c'est que le symbole OFDM est précédé par un préfixe cyclique. Par exemple, le système proposé par Zelst [2] est celui de Figure 2-2 et Figure 2-3. Sur ces schémas, chaque antenne envoie un signal OFDM indépendamment des autres antennes d'émission.

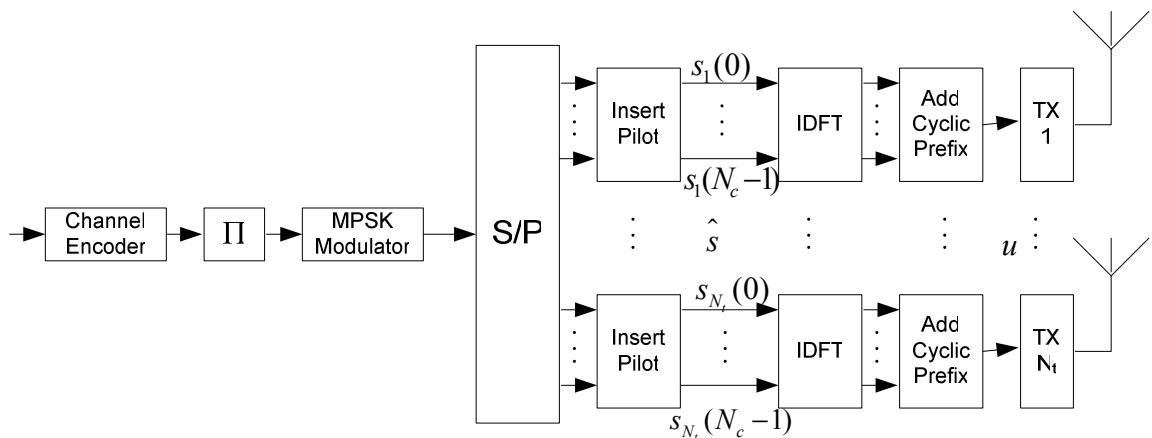


Figure 2-2- Emetteur MIMO-OFDM

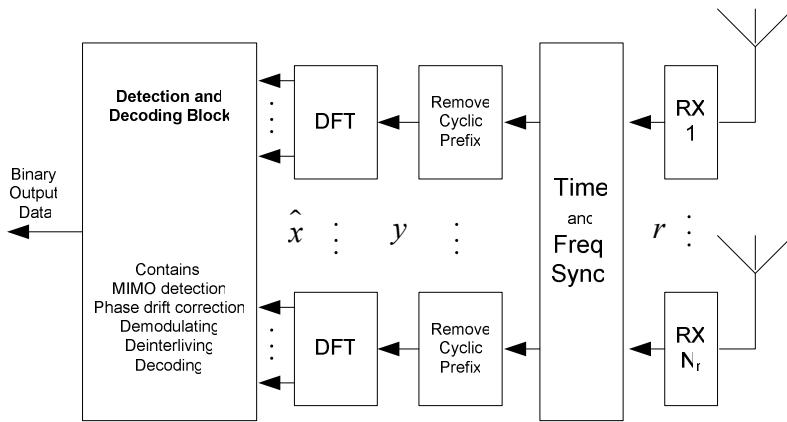


Figure 2-3- Récepteur MIMO-OFDM

Pour une synchronisation de temps, nous travaillons sur les signaux après IDFT en émission et avant DFT en réception dans le domaine temporel. La Figure 2-4 présente le schéma simplifié que nous allons détailler.

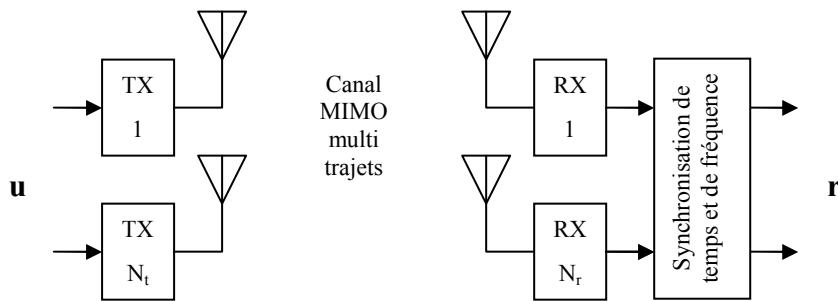


Figure 2-4- Schéma simplifié du domaine temporel du système MIMO-OFDM

Soit N_t le nombre d'antennes d'émission et N_r le nombre d'antennes de réception. L'émetteur envoie à l'instant discret τ le vecteur colonne $\mathbf{u}(\tau)$ de taille N_t . Le récepteur reçoit le vecteur colonne $\mathbf{r}(\tau)$ de taille N_r .

Le canal est considéré comme un canal de Rayleigh sélectif en fréquence et invariant sur un paquet (quasi statique). Le canal entre l'antenne d'émission p et l'antenne de réception q est considéré comme un filtre RIF à L coefficients complexes $h_{qp}(l)$ avec $l=\{0, \dots, L-1\}$. On suppose que ces coefficients sont des variables aléatoires gaussiennes indépendantes à moyenne nulle avec la variance $P(l)/2$ par dimension et que la puissance totale est normalisée à 1. L'ensemble de $P(l)$, $l=\{0, \dots, L-1\}$ est appelé PDP¹.

¹ Power delay profile

Soit $s_p(t)$, le signal en bande de base envoyé par la $p^{ième}$ antenne d'émission. Le signal reçu à l'instant t par l'antenne de réception q s'écrit :

$$r_q(t) = \sum_{p=1}^{N_t} \int_{-\infty}^{+\infty} s_p(t-u) h_{qp}(u) du + v_q(t) \quad (69)$$

où $v_q(t)$ est un bruit additif gaussien et indépendant des autres antennes de réception. Le signal reçu est échantillonné à l'instant $t=kT_s+\varepsilon_0 T_s$ où T_s est la période d'échantillonnage, égale à $T_{s-OFDM}/(N+N_g)$ et $0 \leq \varepsilon_0 < 1$ représente l'erreur fractionnée sur l'instant d'échantillonnage. Ce signal s'écrit alors sous la forme :

$$r_q(kT_s + \varepsilon_0 T_s) = \sum_{p=1}^{N_t} \int_{-\infty}^{+\infty} s_p(kT_s - u') h_{qp}(u') du' + v_q(kT_s + \varepsilon_0 T_s) \quad (70)$$

où $u' = u - \varepsilon_0 T_s$. Comme il est démontré dans [3], si la largeur de bande de canal B_h vérifie l'inégalité $B_h < \frac{1}{T_s} - B_s$ où B_s représente la largeur de bande du signal transmis, la relation (70) et en remplaçant l'intégrale par une somme, peut être parfaitement représentée par l'équation ci-dessous :

$$r_{q,k} = \sum_{p=1}^{N_t} \sum_{m=-\infty}^{\infty} s_p(kT_s - mT_s) h_{qp}(mT_s) + v_{q,k} \quad (71)$$

où $r_{q,k} \triangleq r_q(kT_s + \varepsilon_0 T_s)$, $v_{q,k} \triangleq v_q(kT_s + \varepsilon_0 T_s)$. Cette condition est normalement vérifiée car nous supposons la présence d'un filtre en racine de cosinus surélevé en émission et en réception. Le canal équivalent étant la mise en cascade de ces trois filtres et en sachant que la réponse impulsionnelle du filtre en cosinus surélevé est nulle pour des multiples de T_s , les coefficients du canal équivalent sont les mêmes que les coefficients du canal multi trajets. Ce canal multi trajets est supposé quasi statique et ne varie pas pendant un paquet mais varie d'un paquet à l'autre de manière indépendante.

Le vecteur colonne d'observation $\mathbf{r}_{q,n}$ de taille N_{ob} contient les échantillons du signal reçu par l'antenne q pour les instants de n à $n+N_{ob}-1$:

$$\mathbf{r}_{q,n} = \begin{bmatrix} r_{q,n} & r_{q,n+1} & \dots & r_{q,n+N_{ob}-1} \end{bmatrix}^T_{N_{ob} \times 1} \quad (72)$$

Nous nous mettons dans le cas où nous sommes en train de recevoir le préambule et nous prenons pour la référence de temps, le début du préambule comme présenté sur la Figure 2-5¹.

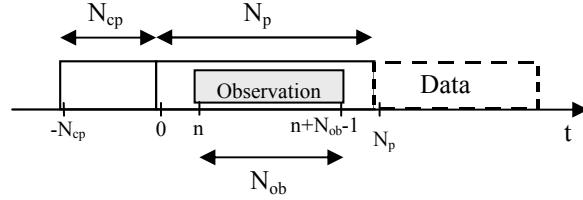


Figure 2-5 Fenêtre d'observation de taille N_{ob}

Sachant que la séquence envoyée par le $p^{ième}$ TX est $[c_{p,0} \ c_{p,1} \ \dots \ c_{p,N_p-1}]$, la séquence avec le préfixe cyclique sera : $[c_{p,N_p-N_{cp}} \ \dots \ c_{p,N_p-1} \ c_{p,0} \ \dots \ c_{p,N_p-1}]$. Pour que la totalité de l'observation soit dans le préambule, la condition $-N_{cp} + L - 1 \leq n \leq N_p - N_{ob}$ est nécessaire. Dans ce cas le vecteur reçu est :

$$\mathbf{r}_{q,n} = \sum_{p=1}^{N_t} \mathbf{C}_{p,n} \mathbf{h}_{qp} + \mathbf{v}_{q,n} \quad (73)$$

où

$$\mathbf{C}_{p,n} = \begin{bmatrix} c_{p,mod(n,N_p)} & c_{p,mod(n-1,N_p)} & \dots & c_{p,mod(n-L+1,N_p)} \\ c_{p,mod(n+1,N_p)} & c_{p,mod(n,N_p)} & \dots & c_{p,mod(n-L+2,N_p)} \\ \vdots & \vdots & \ddots & \vdots \\ c_{p,mod(n+N_{ob}-1,N_p)} & c_{p,mod(n+N_{ob}-2,N_p)} & \dots & c_{p,mod(n-L+N_{ob},N_p)} \end{bmatrix}_{N_{ob} \times L} \quad (74)$$

et le vecteur \mathbf{h}_{qp} représente le canal entre l'antenne TX p et l'antenne RX q :

$$\mathbf{h}_{qp} = [h_{qp}(0) \ h_{qp}(1) \ \dots \ h_{qp}(L-1)]^T_{L \times 1} \quad (75)$$

et $\mathbf{v}_{q,n}$ est un vecteur de bruit avec la matrice de covariance $\sigma_v^2 \mathbf{I}_{N_{ob}}$. Nous formons le vecteur colonne \mathbf{r}_n comme étant le vecteur des séquences reçues par toutes les antennes RX et pendant la durée de l'observation sous la forme :

$$\mathbf{r}_n = [\mathbf{r}_{1,n}^T \ \mathbf{r}_{2,n}^T \ \dots \ \mathbf{r}_{N_r,n}^T]^T_{N_{ob}N_r \times 1} \quad (76)$$

¹ Dans ce rapport, les notations N_g et N_{cp} sont équivalentes, ainsi que N et N_p .

Nous formons le vecteur colonne $\mathbf{h}_{q,:} = \begin{bmatrix} \mathbf{h}_{q1}^T & \mathbf{h}_{q2}^T & \dots & \mathbf{h}_{qN_t}^T \end{bmatrix}^T_{LN_t \times 1}$ regroupant les coefficients du canal aboutissant à l'antenne RX q et le vecteur colonne regroupant tous les coefficients du canal :

$$\mathbf{h} = \begin{bmatrix} \mathbf{h}_{:,1}^T & \mathbf{h}_{:,2}^T & \dots & \mathbf{h}_{:,N_r}^T \end{bmatrix}^T_{LN_t N_r \times 1} \quad (77)$$

Utilisant la définition du produit de Kronecker de deux matrices, on peut écrire la grande séquence reçue \mathbf{r}_n comme :

$$\mathbf{r}_n = (\mathbf{I}_{N_r} \otimes \mathbf{C}_n) \mathbf{h} + \mathbf{v}_n \quad (78)$$

Avec

$$\begin{aligned} \mathbf{C}_n &= \begin{bmatrix} \mathbf{C}_{1,n} & \mathbf{C}_{2,n} & \dots & \mathbf{C}_{N_t,n} \end{bmatrix}_{N_{ob} \times LN_t} \\ \mathbf{v}_n &= \begin{bmatrix} \mathbf{v}_{1,n}^T & \mathbf{v}_{2,n}^T & \dots & \mathbf{v}_{N_r,n}^T \end{bmatrix}^T_{N_{ob} N_r \times 1} \end{aligned} \quad (79)$$

Nous utiliserons ces relations pour la synchronisation de temps et de canal. Le but sera donc dans une fenêtre d'observation, de déclarer si le préambule est détecté et dans ce cas, d'estimer la valeur de n . Dans un premier temps, nous parlerons du problème de la synchronisation de temps à base de corrélation, sans nous occuper de l'estimation de canal. En suite, une méthode de maximum de vraisemblance sera proposée pour une estimation conjointe de canal en même temps que le "*timing*".

2.5 Synchronisation en temps

Le problème de la synchronisation de temps dans un système MIMO-OFDM consiste à la détection de début de paquet. Comme nous l'avons déjà évoqué, chaque paquet est précédé d'une séquence d'apprentissage appelé aussi préambule. Le rôle du dispositif de la synchronisation est de détecter avec précision la présence de cette séquence. Sachant que nous sommes dans le cadre des systèmes MIMO, le signal reçu est la superposition de toutes les séquences envoyées par toutes les antennes d'émission. De plus, des répliques déphasées et atténuees de ces séquences sont aussi superposées.

Un des premiers articles traitant ce sujet fut celui de Mody [4] qui est une extension simple de l'article de Schmidl [5] proposé pour les systèmes SISO-OFDM. En effet, les préambules qui sont choisis ont les propriétés suivantes :

- Les préambules de toutes les antennes d'émissions sont orthogonaux par paires
- Chacun des préambules est orthogonal à des versions décalées de lui-même et aussi à des versions décalées des préambules des autres antennes¹
- L'orthogonalité décalée est vérifiée au moins pour un décalage inférieur à la taille du canal.

Comme il est proposé par la norme IEEE802.11 pour les systèmes SISO-OFDM, le préambule est construit par la répétition de deux séquences auxquelles on ajoute un préfixe cyclique comme présentée sur la Figure 2-7.

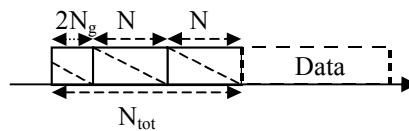


Figure 2-6- Préambule pour les canaux sélectifs en fréquence en présence d'erreur de fréquence

Cette idée facilite la tâche de la synchronisation dans le cas où une erreur de fréquence existe. Certains articles ([2] par exemple) utilisent la séquence proposée par Suehiro [6] pour les systèmes SISO. Par contre, la généralisation de cette séquence à des systèmes MIMO n'est pas une affaire simple (voir par exemple la référence [7]).

Zelst [2] rapporte l'implantation d'un récepteur MIMO-OFDM pour les canaux multi trajets. La solution simple pour les préambules qu'il propose afin d'obtenir les propriétés ci-dessus est d'allumer successivement les émetteurs pour n'avoir qu'un canal SISO (ou SIMO) à la fois. La Figure 2-7 schématise la forme de ces préambules. La taille totale de chaque préambule est deux fois la taille d'un symbole OFDM : $2(N+N_g)$. L'orthogonalité décalée est donc assurée en utilisant les séquences Suehiro pour chaque antenne et en éteignant toutes les antennes sauf une, cette orthogonalité décalée est aussi vérifiée entre antennes.

¹ *Shift orthogonality*

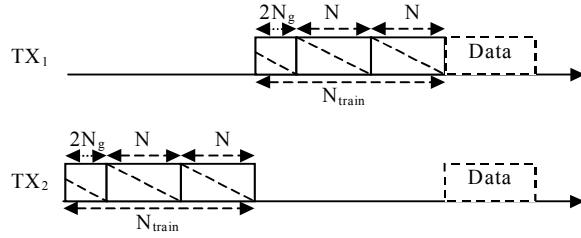


Figure 2-7- Les préambules orthogonaux pour deux antennes d'émission

Ces séquences seront utilisées non seulement pour la synchronisation en temps, mais également pour la synchronisation en fréquence et de symbole et pour l'estimation de canal multi trajets entre chaque paire d'antennes d'émission et de réception.

Pour la détection des préambules, Zelst [2] utilise la même méthode que les systèmes SISO : calcul de la corrélation du signal reçu avec le même signal décalé de N . C'est-à-dire qu'il exploite la forme périodique des préambules. C'est le même principe que [5] et [8][9]-[14] initialement donnés pour les système SISO-OFDM. En effet, l'algorithme de Zelst est une extension de [10] qui donne selon [15] une meilleure performance que [5] et [13].

Nous définissons la matrice $\mathbf{H}(l)$ avec l'élément (q,p) comme étant égal à $h_{qp}(l)$. Avec ce modèle le vecteur reçu $\mathbf{r}(\tau)$ à l'instant τ s'écrit :

$$\mathbf{r}(\tau) = \sum_{l=0}^{L-1} \mathbf{H}(l) \mathbf{u}(\tau - l) + \mathbf{v}(\tau) \quad (80)$$

où $\mathbf{v}(\tau)$ représente un bruit complexe additif gaussien à moyenne nulle et à variance $\sigma_v^2/2$ par dimension. Les éléments du vecteur $\mathbf{u}(\tau)$ sont supposés des variables aléatoires non corrélées à moyenne nulle et à variance σ_u^2 . Le rapport signal à bruit SNR pour chaque antenne de réception est :

$$\rho = N_t \frac{\sigma_u^2}{\sigma_v^2} \quad (81)$$

Afin de pouvoir comparer les résultats avec ceux d'un système SISO, nous gardons la même puissance totale, ce qui revient à diviser la puissance de chaque antenne par N_t .

Si on ajoute une erreur de fréquence due à l'imprécision des oscillateurs locaux ou à l'effet Doppler, l'équation (81) s'écrit sous la forme :

$$\mathbf{r}(\tau) = e^{j2\pi\Delta f\tau} \sum_{l=0}^{L-1} \mathbf{H}(l) \mathbf{u}(\tau-l) + \mathbf{v}(\tau) \quad (82)$$

où Δf est l'erreur de fréquence.

Calculant la corrélation et moyennant sur toutes les antennes RX, la fonction d'auto corrélation peut s'écrire sous la forme :

$$R_p(\tau) = \sum_{q=1}^{N_r} \left| \sum_{i=0}^{N-1} r_q(\tau-i) r_q^*(\tau-N-i) \right| \quad (83)$$

Sachant qu'à chaque instant un seul TX est actif, nous pouvons accumuler ce résultat sur les préambules de toutes les antennes d'émission pour obtenir :

$$R_r(\tau) = \sum_{p=1}^{N_r} \sum_{q=1}^{N_r} \left| \sum_{i=0}^{N-1} r_q(\tau-i-(p-1)N_{train}) r_q^*(\tau-N-i-(p-1)N_{train}) \right| \quad (84)$$

La Figure 2-8 présente le résultat pour un rapport signal à bruit de 6 dB avec une erreur de fréquence de $0,1/N_c$ où N_c est le nombre de porteuses. La simulation montre que la performance obtenue est quasiment indépendante du tirage de canal et de l'erreur de fréquence. En étudiant le comportement de cette courbe, on remarque que le pic représente le temps d'arrivée du paquet.

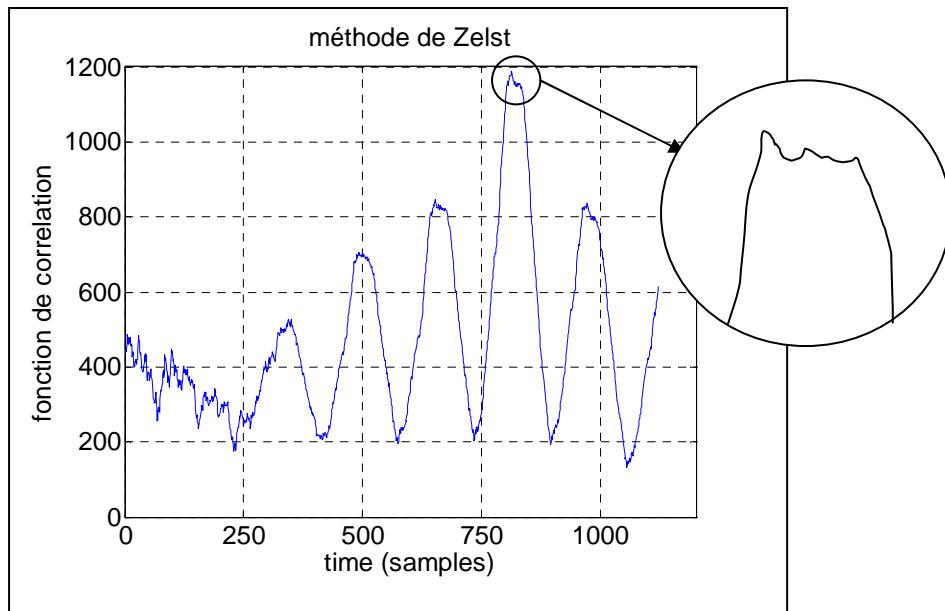


Figure 2-8- Résultat obtenu à partir de l'équation (84) pour un système 4x4 sur un canal sélectif en fréquence (SNR=6dB).

Cependant, il est difficile de détecter de manière exacte le pic de cette fonction car idéalement, le pic est un plateau de taille approximative de $2N_g$ (la taille du préfixe cyclique du préambule).

2.5.1 Synchronisation fine de temps

Pour remédier au problème du plateau, nous avons proposé une nouvelle fonction de corrélation [1]. Le problème de la méthode conventionnelle ci-dessus provient de l'utilisation directe de la fonction d'autocorrélation. En présence d'une erreur de fréquence, le calcul de la corrélation du signal reçu avec le préambule, connu par le récepteur, ne peut pas produire un pic. Nous utilisons donc une astuce pour résoudre ce problème, basée sur une détection différentielle. Ainsi, l'erreur de fréquence n'influence plus notre estimation. Cette nouvelle fonction est :

$$y_q(\tau) = \sum_{i=1}^{N_{train}-1} r_q^*(\tau+1-i) c_p(N_{train}-i) r_q(\tau-i) c_p^*(N_{train}-i-1) \quad (85)$$

où le N_{train} est présenté sur la Figure 2-7. Dans cette équation c_p représente le préambule de taille N_{train} . Dans cet algorithme nous n'avons plus besoin d'un préambule périodique avec un préfixe cyclique. On peut donc écrire le signal reçu pour l'antenne q supposant (pour simplicité de l'écriture) que le temps correspondant au début du préambule est $\tau=0$. Alors, quand le $p^{ième}$ émetteur est allumé, le signal reçu par l'antenne q est :

$$r_q(\tau) = c_p(\tau) h_{qp}^1 e^{j2\pi\tau\Delta f T_s} + v_q(\tau) \quad (86)$$

Pour simplifier cette équation, nous n'avons considéré que le premier coefficient du canal. En effet, en choisissant un préambule orthogonal décalé suivant une certaine condition que l'on verra par la suite, les termes supplémentaires seront supprimés et l'équation ci-dessus reste correcte. Remplaçant (86) dans (85), nous obtenons :

$$y_q(\tau) \Big|_{\tau=N_{train}-1} = y_q^{\max} = \sum_{i=1}^{N_{train}-1} |c_p(N_{train}-i)|^2 |c_p(N_{train}-i-1)|^2 |h_{qp}^1|^2 e^{-j2\pi\Delta f T_s} + v'_q(N_{train}-1) \quad (87)$$

La fonction $y(\tau)$ présente cette fois-ci un pic très pointu à l'arrivée du paquet indépendamment de l'erreur de fréquence et de la phase du canal (remarquez que τ n'apparaît plus dans la phase de l'exponentiel). Pour profiter de toutes les antennes de réception et tous les préambules envoyés, on construit finalement la fonction suivante :

$$R_r(\tau) = \sum_{p=1}^{N_t} y^p(\tau - (p-1)N_{train}) = \sum_{p=1}^{N_t} \sum_{q=1}^{N_r} y_q(\tau - (p-1)N_{train}) \quad (88)$$

Des simulations montrent effectivement la robustesse de la méthode par rapport à la variation du canal multi trajets et l'erreur de fréquence. La Figure 2-9 présente et compare le résultat obtenu par rapport à l'utilisation de la méthode conventionnelle.

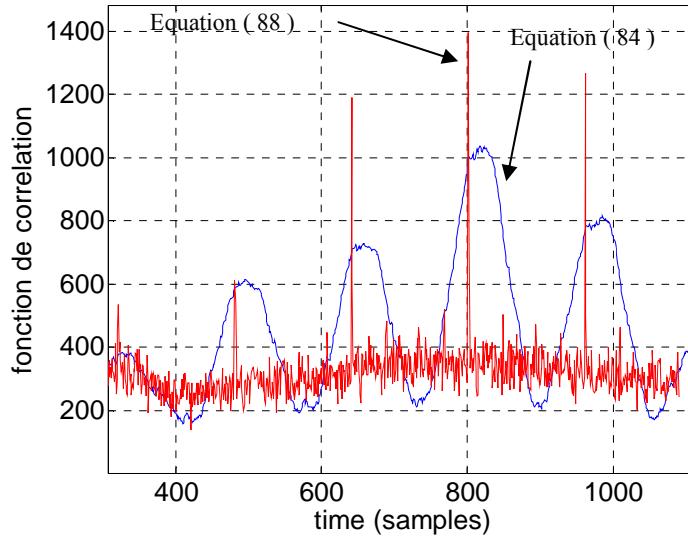


Figure 2-9- Utilisation de la méthode de l'équation (88) pour un système 4×4 et un canal sélectif en fréquence (SNR=6 dB)

En étudiant l'équation (85), on remarque que la condition d'orthogonalité décalée n'est plus nécessaire sur la séquence du préambule elle même. En effet, la fonction proposée n'est pas le calcul d'une simple corrélation entre la séquence reçue et le préambule. Dans ce cas, c'est la condition ci-dessous qui devient importante :

$$\sum_{\substack{n=0 \\ m \neq 0}}^{N-1} c'(n)c'^*(((n+m) \bmod N_{train})) = 0 \quad (89)$$

où $c'(n) = c(n)c^*(n-1)$. On peut utiliser le code proposé par Frank et Zadoff [16] qui présente une bonne propriété de la corrélation périodique. La condition de l'équation ci-dessus garantit un pic pour les canaux multi trajets. On applique alors le code de Frank-Zadoff à c' et non pas à c . Pour obtenir le préambule à partir de la séquence c' , nous proposons l'algorithme récursif suivant. On fixe d'abord $c(0)=1$, et ensuite par itération on obtient :

$$c(i) = \frac{c'^*(i)}{c^*(i-1)} \quad (90)$$

On peut vérifier facilement qu'en remplaçant l'équation (90) dans l'équation (89), on obtient la condition voulue.

2.6 Synchronisation en fréquence

Nous supposons toujours un système OFDM-MIMO avec un canal sélectif en fréquence. On a démontré que le signal reçu peut s'écrire sous la forme de l'équation (82). Le but de la synchronisation de fréquence est d'estimer la valeur de Δf à partir de la séquence reçue $\mathbf{r}(\tau)$. Dans la littérature, Moose [8] montre comment l'erreur de fréquence peut être estimée pour un système SISO-OFDM. Cette méthode a été améliorée par Schmidl [5]. Schenk a généralisé cette idée pour les systèmes MIMO-OFDM afin de calculer l'erreur de fréquence. L'idée est de calculer la moyenne de $r_q(\tau)r_q^*(\tau-N)$ sur le préambule et pour toutes les antennes RX et à partir du point où un pic dans la fonction d'auto corrélation se présente (équation (84)). Ce principe est schématisé sur la Figure 2-10.

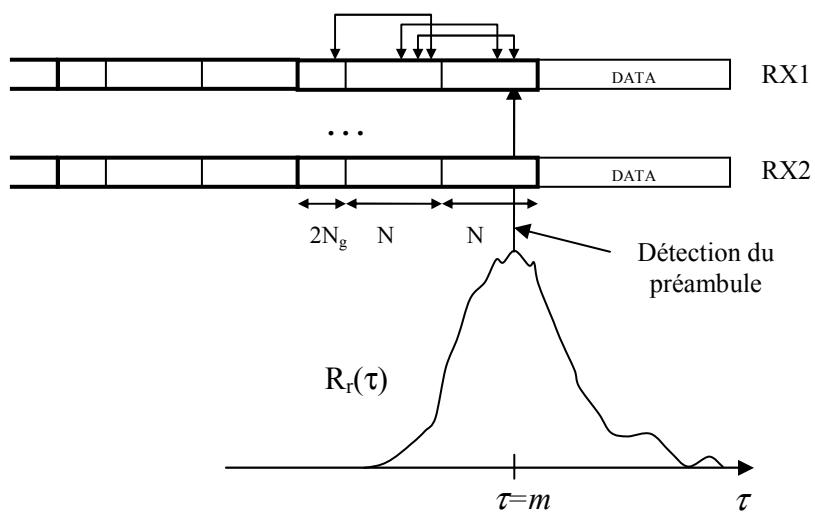


Figure 2-10- Fenêtre pour l'estimation de fréquence

Sachant que le préambule est répété, pendant N valeurs, le résultat de la multiplication reste le même. On calcule donc la somme pour diminuer le bruit. La fonction obtenue est alors la suivante :

$$\Lambda(m) = \sum_{q=1}^{N_r} \sum_{\tau=m-N+1}^m r_q(\tau)r_q^*(\tau-N) \quad (91)$$

L'interprétation de cette somme, est que nous superposons les mêmes vecteurs complexes bruités et que la phase du vecteur résultant est proportionnelle à l'erreur de fréquence. Ainsi on peut écrire :

$$\Delta f_{est} = \frac{\theta_{est}}{2\pi N T_s} = \frac{f_s \angle \Lambda(m)}{2\pi N} \quad (92)$$

où f_s indique la fréquence d'échantillonnage.

Utilisant notre algorithme de l'équation (88) pour détecter le préambule plus précisément, nous pouvons utiliser la totalité de la séquence d'apprentissage de taille N_{train} qui donne une meilleure estimation de fréquence. Dans ce cas, la relation de l'estimateur est la suivante :

$$\Lambda(m) = \sum_{q=1}^{N_r} \sum_{\tau=m-N-2N_g+1}^m r_q(\tau) r_q^*(\tau-N) \quad (93)$$

où le point m est donné par l'estimateur de temps fin. La Figure 2-11 compare la variance de notre estimateur par rapport au résultat obtenu dans la littérature.

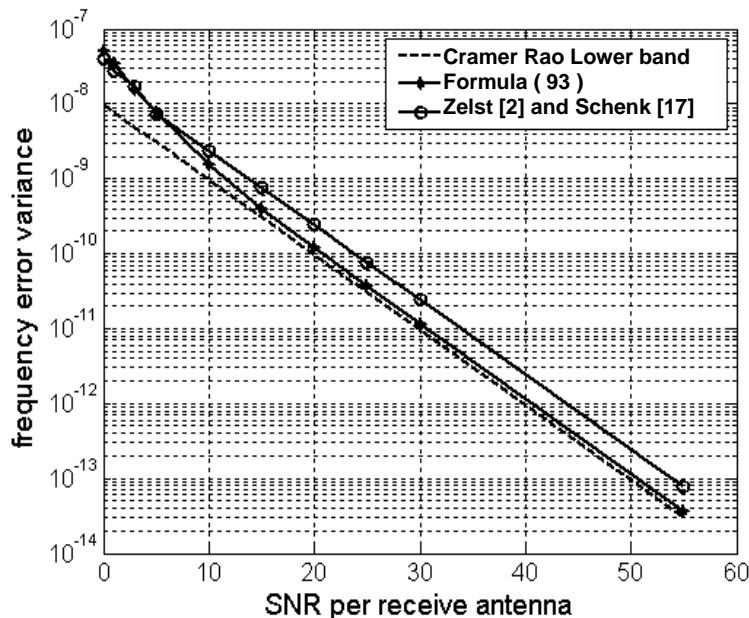


Figure 2-11- Variance de l'estimateur de fréquence pour un système 4×4

Pour pouvoir éprouver l'estimateur proposé, on compare la variance avec la borne inférieure de Cramer-Rao. D'après l'article de Rife [18] pour un système SISO la borne est donnée par la relation :

$$\text{var}(\theta) \geq \frac{1}{SNR \times N} \quad (94)$$

où N est le nombre d'observations. Ici, nous avons N_{train} échantillons disponibles que nous remplaçons dans l'équation (94). Ensuite, utilisant l'équation (92) la borne de la variance de notre estimateur de fréquence se calcule :

$$\text{var}[\Delta f_{est} T_s] \geq \frac{1}{4\pi^2 N^2 SNR \times N_r N_{train}} \quad (95)$$

Cette borne est aussi tracée sur la Figure 2-11 et démontre que notre estimateur de fréquence est très proche de l'estimateur optimal.

2.7 Estimation conjointe du canal et synchronisation en temps

Nous partons de l'équation (78) qui donne l'ensemble des échantillons pour une durée de N et pour toutes les antennes de réception. Le canal étant additif, gaussien et indépendant, nous pouvons écrire la fonction de distribution de probabilité de la séquence reçue conditionnelle à la réalisation du canal et à l'instant n , comme suit :

$$p(\mathbf{r}_n | n, \mathbf{h}) = \frac{1}{(\pi \sigma_v^2)^{N_{ob} N_r}} \exp \left\{ -\frac{\|\mathbf{r}_n - (\mathbf{I}_{N_r} \otimes \mathbf{C}_n) \mathbf{h}\|^2}{\sigma_v^2} \right\} \quad (96)$$

Cette fonction doit être maximisée en faisant varier n et \mathbf{h} (la méthode de maximum de vraisemblance ou ML). On suppose que cette fenêtre d'observation vérifie la condition sur n de la Figure 2-5 et que la totalité de la fenêtre est entièrement contenue dans la partie préambule reçue (préfixe cyclique compris). C'est-à-dire que la détection grossière de début de paquet a été déjà effectuée et que l'erreur de fréquence a été corrigée.

La maximisation de la relation (96) est équivalente à la minimisation de la métrique ci-dessous :

$$J(\mathbf{r}_n | n, \mathbf{h}) = (\mathbf{r}_n - (\mathbf{I}_{N_r} \otimes \mathbf{C}_n) \mathbf{h})^H (\mathbf{r}_n - (\mathbf{I}_{N_r} \otimes \mathbf{C}_n) \mathbf{h}) \quad (97)$$

On fixe d'abord la valeur de n pour obtenir une métrique à une seule variable et on minimise cette métrique en faisant varier \mathbf{h} . D'après [19], la solution ML est la suivante (voir Annexe II pour la preuve) :

$$\hat{\mathbf{h}} = [(\mathbf{I}_{N_r} \otimes \mathbf{C}_n)^H (\mathbf{I}_{N_r} \otimes \mathbf{C}_n)]^{-1} (\mathbf{I}_{N_r} \otimes \mathbf{C}_n)^H \mathbf{r}_n \quad (98)$$

Pour obtenir une solution unique, le nombre des points d'observation doit être supérieur ou égal au nombre des inconnus : $N_{ob}N_r \geq LN_tN_r \Rightarrow N_{ob} \geq LN_t$. On remplace donc l'équation (98) dans (97). En simplifiant le résultat et en supprimant les termes non pertinents, la fonction à minimiser, en faisant varier cette fois n , sera la suivante :

$$\Psi(n) = \mathbf{r}_n^H (\mathbf{I}_{N_r} \otimes \mathbf{C}_n) [(\mathbf{I}_{N_r} \otimes \mathbf{C}_n)^H (\mathbf{I}_{N_r} \otimes \mathbf{C}_n)]^{-1} (\mathbf{I}_{N_r} \otimes \mathbf{C}_n)^H \mathbf{r}_n \quad (99)$$

Utilisant les propriétés concernant le produit de Kronecker $(A \otimes B)^{-1} = A^{-1} \otimes B^{-1}$, $(A \otimes B)^H = A^H \otimes B^H$ et $(A \otimes B)(C \otimes D) = (AC) \otimes (BD)$ la relation ci-dessus se simplifie en :

$$\Psi(n) = \mathbf{r}_n^H (\mathbf{I}_{N_r} \otimes \mathbf{C}_n (\mathbf{C}_n^H \mathbf{C}_n)^{-1} \mathbf{C}_n^H) \mathbf{r}_n = \sum_{q=1}^{Nr} \mathbf{r}_{q,n}^H \mathbf{C}_n (\mathbf{C}_n^H \mathbf{C}_n)^{-1} \mathbf{C}_n^H \mathbf{r}_{q,n} \quad (100)$$

où $\mathbf{r}_{q,n}$ est la séquence de l'observation de l'antenne de réception q . L'algorithme de synchronisation consiste donc à tester cette relation pour différentes valeurs de n et sélectionner le n qui maximise $\Psi(n)$ ¹:

$$\hat{n} = \arg \max_n \Psi(n)$$

Il est à remarquer que dans cette méthode, nous n'avons pas imposé le préfixe cyclique, ni la répétition du préambule. Pour le choix de la taille de la fenêtre d'observation N_{ob} il est à remarquer que $n \in [-N_{cp} + L, N_p - N_{ob}]$ tout en sachant que $L < N_{cp}$. Si N_{ob} est important, l'intervalle de variation de n se réduit. En conséquence, il nous faut une estimation grossière plus précise avant de lancer notre algorithme ML. D'un autre côté, la précision de l'estimateur s'améliore car il y a plus d'observations. Alors, il existe un compromis sur le choix de N_{ob} . En terme de complexité de l'algorithme, si on la mesure en terme de nombre de multiplications total, il nous faut donc $N_r(N_p + N_{cp} - L - N_{ob} + 1)(N_{ob}^2 + N)$ multiplications. Cette estimation suppose que les matrices $\mathbf{C}_n (\mathbf{C}_n^H \mathbf{C}_n)^{-1} \mathbf{C}_n^H$ pour différentes valeurs possibles de n ont été déjà calculées et stockées dans la mémoire du récepteur.

¹ Cette idée est publiée sous forme d'article à l'EUSIPCO2006.

2.7.1 Performances

Pour évaluer la performance de notre détecteur de paquet, nous utilisons le critère suivant. On calcule la probabilité que le n estimé (\hat{n}) soit égal à la valeur exacte de n , ce que l'on appelle la probabilité de détection :

$$P_{\text{det}} = \Pr\{\hat{n} = n_{\text{réel}}\} \quad (101)$$

De plus, nous introduisons comme dans [9] un autre critère qui consiste à évaluer combien nous sommes loin du point optimal. C'est la probabilité d'échec définie par la probabilité que le n estimé soit au moins m échantillons plus loin que le n réel :

$$P_{\text{échec}}(m) = \Pr\{|\hat{n} - n_{\text{réel}}| > m\} \quad (102)$$

Nous avons effectué une série de simulations Monte-Carlo pour déterminer ces facteurs de mérite. Nous fixons les mêmes paramètres que [9], c'est-à-dire que la taille d'un symbole OFDM est de 64 avec un préfixe cyclique de taille 8. Le PDP du canal est exponentiel de taille 8 et causal avec 3dB d'atténuation par coefficient :

$$PDP = \frac{\left[1 \quad \frac{1}{2} \quad \dots \quad \frac{1}{2^7} \right]}{\sum_{i=0}^7 2^{-i}} \quad (103)$$

Contrairement à [9] qui s'occupait des systèmes SISO, nous utilisons un système MIMO 4×4. Les préambules sont générés aléatoirement pour chaque antenne TX. La taille du vecteur d'observation est de 64 et le résultat est obtenu sur 10^5 paquets. La Figure 2-12 présente la probabilité d'échec suivant le critère de l'équation (102) pour $m=5$ et $m=15$. Nous avons aussi présenté le résultat de [9] pour comparaison. Bien que nous avons tendance à penser qu'un système MIMO est toujours plus performant qu'un système SISO, quand il s'agit de la synchronisation, ce n'est pas aussi évident. En effet, nous recevons un signal bien bruité dû à l'interférence entre plusieurs antennes envoyant chacune un signal différent. L'effet de diversité attendu dans les récepteurs MIMO n'existe pas et en plus, un bien plus grand nombre de coefficients de canal doivent être estimés. Cependant les courbes obtenues montrent une meilleure performance par rapport à un système SISO. Ceci peut être justifié par le fait que même si ici le nombre de coefficients inconnus du canal augmente de la même manière que les données observées, il faut remarquer que la fréquence d'offset et le timing restent identiques pour tous les trajets. Ceci est justifié car les antennes TX (et RX) sont assez proches par rapport à la

différence de trajets multiples entre chaque paire d'antennes TX-RX, ce qui rend négligeable le retard en fonction des paires d'antennes.

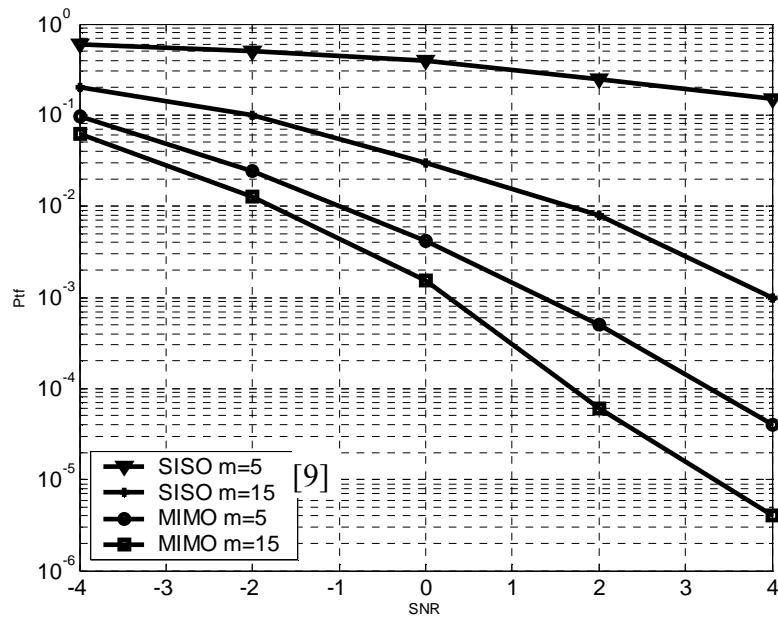


Figure 2-12- Probabilité d'échec pour un système MIMO-OFDM 4×4

Pour voir l'effet du nombre d'antennes dans la performance de l'algorithme d'estimation, une série de simulations a été effectuée. La Figure 2-13 présente le résultat. Comme on aurait pu deviner, la performance du système 2×2 est plus mauvaise que celle du 1×2.

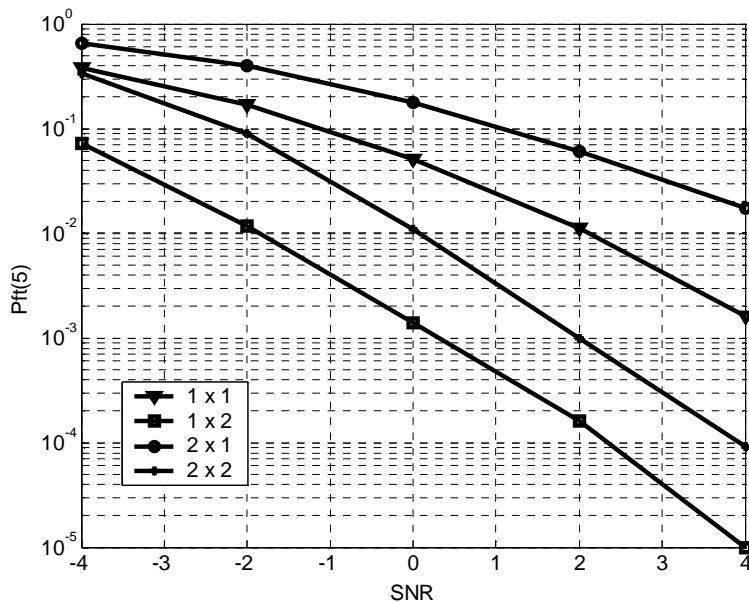


Figure 2-13- Effet de la taille du système MIMO sur la probabilité d'échec

Nous avons aussi évalué la probabilité de détection exacte définie par l'équation (101) pour différentes valeurs de SNR (voir Figure 2-14)

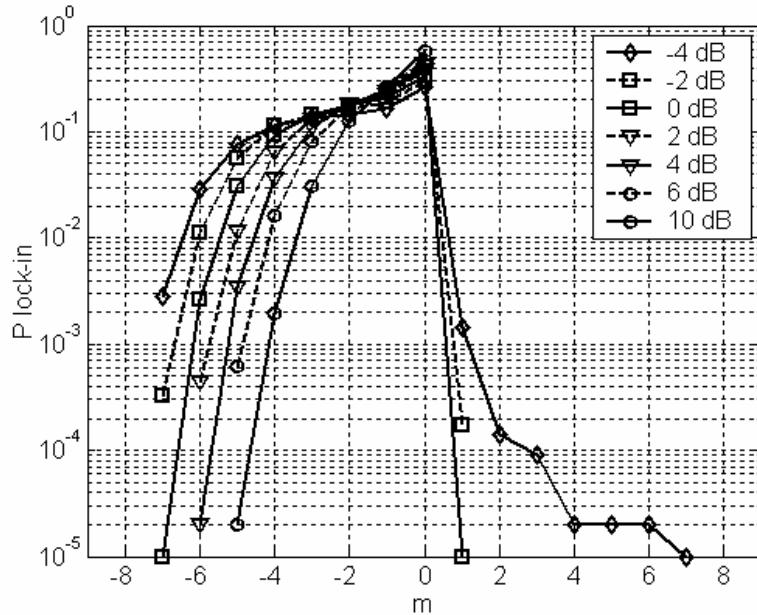


Figure 2-14- Probabilité de détection pour un système MIMO-OFDM 4×4

2.8 **Estimation conjointe du canal et synchronisation en temps et en fréquence**

Dans le paragraphe précédent, nous avons supposé que l'erreur de fréquence n'existe pas. Ici, nous nous plaçons dans le cas où un décalage de fréquence entre les émetteurs et les récepteurs existe. Ceci est dû soit à l'imprécision des oscillateurs soit à une fréquence Doppler.

La méthode que nous proposons est inspirée de l'article de Y. Sun [20] où l'offset de fréquence est estimé dans un canal sélectif en fréquence et puis le canal est identifié. Sun suppose que la synchronisation de temps a déjà été parfaitement réussie. Ici, nous considérons que le timing reste à faire et nous l'intégrons dans notre métrique à minimiser. Contrairement à Sun qui pouvait traiter le problème dans le domaine fréquentiel (après FFT), nous ne le pouvons pas car celui-ci nécessite une synchronisation de temps parfaite. L'algorithme que nous proposons fonctionne donc dans le domaine temporel (avant FFT).

La mise en équation du problème est la suivante. On commence par modifier l'équation (73) en y ajoutant le décalage fréquentiel comme suit :

$$\mathbf{r}_{q,n} = \mathbf{E}_n \sum_{p=1}^{N_t} \mathbf{C}_{p,n} \mathbf{h}_{qp} + \mathbf{v}_{q,n} \quad (104)$$

Dans cette équation \mathbf{E}_n représente la rotation de phase due à l'offset de fréquence noté par ε . La matrice \mathbf{E}_n est construite par la relation ci-dessous :

$$\mathbf{E}_n = \text{diag} \left[e^{j2\pi n \varepsilon} \quad e^{j2\pi(n+1)\varepsilon} \quad \dots \quad e^{j2\pi(n+N_{ob}-1)\varepsilon} \right]_{N_{ob} \times N_{ob}} \quad (105)$$

L'indice n n'ajoute qu'une phase constante qui peut être absorbée dans les coefficients du canal. On peut donc, sans perte de généralité, supprimer l'indice n et définir la matrice \mathbf{E} comme :

$$\mathbf{E} = \left[\mathbf{I}_{N_r} \otimes \text{diag} \left[1 \quad e^{j2\pi \varepsilon} \quad \dots \quad e^{j2\pi(N_{ob}-1)\varepsilon} \right]_{N_{ob} \times N_{ob}} \right]_{N_{ob}N_r \times N_{ob}N_r} \quad (106)$$

Maintenant pour annuler l'effet de l'offset de fréquence, il suffit de multiplier le signal reçu par le hermitien de \mathbf{E} et la relation (96) se transforme en

$$p(\mathbf{r}_n | n, \mathbf{h}) = \frac{1}{(\pi \sigma_v^2)^{N_{ob}N_r}} \exp \left\{ - \frac{\|\mathbf{E}^H \mathbf{r}_n - (\mathbf{I}_{N_r} \otimes \mathbf{C}_0) \mathbf{h}\|^2}{\sigma_v^2} \right\} \quad (107)$$

où \mathbf{C}_0 s'obtient en mettant $n=0$ dans l'équation (79). Pour maximiser cette équation, il suffit de minimiser la métrique suivante :

$$J(\mathbf{r}_n | n, \mathbf{h}) = (\mathbf{E}^H \mathbf{r}_n - (\mathbf{I}_{N_r} \otimes \mathbf{C}_0) \mathbf{h})^H (\mathbf{E}^H \mathbf{r}_n - (\mathbf{I}_{N_r} \otimes \mathbf{C}_0) \mathbf{h}) \quad (108)$$

Pour avoir une réponse unique, il faut que $N_{ob}N_r \geq LN_rN_r$ et comme nous l'avons fait dans le paragraphe 2.7, le \mathbf{h} qui minimise cette équation est :

$$\hat{\mathbf{h}} = [(\mathbf{I}_{N_r} \otimes \mathbf{C}_0)^H (\mathbf{I}_{N_r} \otimes \mathbf{C}_0)]^{-1} (\mathbf{I}_{N_r} \otimes \mathbf{C}_0)^H \mathbf{E}^H \mathbf{r}_n \quad (109)$$

En remplaçant l'équation (109) dans (108) et en ignorant les termes non pertinents, on arrive à l'équation suivante qu'il faut maximiser en faisant varier le n et le ε .

$$\Psi(n, \varepsilon) = \mathbf{r}_n^H \mathbf{E} (\mathbf{I}_{N_r} \otimes \mathbf{C}_0) [(\mathbf{I}_{N_r} \otimes \mathbf{C}_0)^H (\mathbf{I}_{N_r} \otimes \mathbf{C}_0)]^{-1} (\mathbf{I}_{N_r} \otimes \mathbf{C}_0)^H \mathbf{E}^H \mathbf{r}_n \quad (110)$$

Utilisant une fois de plus les propriétés concernant le produit de Kronecker $(A \otimes B)^{-1} = A^{-1} \otimes B^{-1}$, $(A \otimes B)^H = A^H \otimes B^H$ et $(A \otimes B)(C \otimes D) = (AC) \otimes (BD)$ la relation ci-dessus se simplifie en :

$$\Psi(n, \varepsilon) = \mathbf{r}_n^H \mathbf{E} (\mathbf{I}_{N_r} \otimes \mathbf{C}_0 (\mathbf{C}_0^H \mathbf{C}_0)^{-1} \mathbf{C}_0^H) \mathbf{E}^H \mathbf{r}_n = \sum_{q=1}^{N_r} \mathbf{r}_{q,n}^H \mathbf{E}_n \mathbf{C}_0 (\mathbf{C}_0^H \mathbf{C}_0)^{-1} \mathbf{C}_0^H \mathbf{E}_n^H \mathbf{r}_{q,n} \quad (111)$$

Ainsi, le temps de début de paquet et la fréquence de l'offset se calculent :

$$(\hat{n}, \hat{\varepsilon}) = \arg \max_{n, \varepsilon} \Psi(n, \varepsilon) \quad (112)$$

Nous allons d'abord résoudre ce problème pour ε et ensuite pour n . Afin de simplifier cette relation, on remplace la matrice constante $\mathbf{C}_0(\mathbf{C}_0^H \mathbf{C}_0)^{-1} \mathbf{C}_0^H$ par \mathbf{S} . La relation (111) se transforme en

$$\Psi(n, \varepsilon) = \sum_{m=1}^{N_r} \mathbf{r}_{m,n}^H \mathbf{E}_n \mathbf{S} \mathbf{E}_n^H \mathbf{r}_{m,n} = \sum_{m=1}^{N_r} \sum_{p=0}^{N_{ob}-1} r_{m,n+p}^* \sum_{q=0}^{N_{ob}-1} [s]_{p,q} r_{m,n+q} e^{-j2\pi\varepsilon(q-p)} \quad (113)$$

où $[s]_{p,q}$ et $p, q \in \{0, 1, \dots, N_{ob}-1\}$ désigne l'élément de la ligne p et de la colonne q de la matrice \mathbf{S} . On peut développer cette équation sous la forme suivante :

$$\begin{aligned} \Psi(n, \varepsilon) = & \sum_{m=1}^{N_r} \left\{ \sum_{p=0}^{N_{ob}-1} r_{m,n+p}^* r_{m,n+p} [s]_{p,p} + \right. \\ & \sum_{p=0}^{N_{ob}-2} \sum_{q=p+1}^{N_{ob}-1} r_{m,n+p}^* r_{m,n+q} e^{-j2\pi\varepsilon(q-p)} [s]_{p,q} + \\ & \left. \sum_{q=0}^{N_{ob}-2} \sum_{p=q+1}^{N_{ob}-1} r_{m,n+p}^* r_{m,n+q} e^{-j2\pi\varepsilon(q-p)} [s]_{p,q} \right\} \end{aligned} \quad (114)$$

En sachant que la matrice \mathbf{S} définie ci-dessus est hermitienne, cette équation se simplifie en :

$$\Psi(n, \varepsilon) = \sum_{m=1}^{N_r} \left\{ \sum_{p=0}^{N_{ob}-1} |r_{m,n+p}|^2 [s]_{p,p} + 2 \operatorname{Re} \left(\sum_{p=0}^{N_{ob}-2} \sum_{q=p+1}^{N_{ob}-1} r_{m,n+p}^* r_{m,n+q} e^{-j2\pi\varepsilon(q-p)} [s]_{p,q} \right) \right\} \quad (115)$$

Le premier terme n'étant pas une fonction de ε , nous maximisons le second terme. Remplaçant $q-p$ par z , la nouvelle métrique est la suivante :

$$\Psi'(n, \varepsilon) = \operatorname{Re} \left(\sum_{m=1}^{N_r} \sum_{z=1}^{N_{ob}-1} \sum_{p=0}^{N_{ob}-z-1} [s]_{p,p+z} r_{m,n+p}^* r_{m,n+p+z} e^{-j2\pi\varepsilon z} \right) \quad (116)$$

On peut donc écrire

$$\Psi'(n, \varepsilon) = \operatorname{Re} \left(\sum_{z=1}^{N_{ob}-1} r_z' e^{-j2\pi\varepsilon z} \right) = \operatorname{Re}(R(n, \varepsilon)) \quad (117)$$

où

$$r_z' = \sum_{m=1}^{N_r} \sum_{p=0}^{N_{ob}-z-1} [s]_{p,p+z} r_{m,n+p}^* r_{m,n+p+z} \quad (118)$$

et que R peut être calculé par l'algorithme de FFT. On peut donc résumer l'algorithme de synchronisation. Dans chaque fenêtre d'observation où la valeur de n est fixée (voir la Figure 2-5) et en utilisant l'équation (118) on calcule la séquence r'_z . On y ajoute assez de zéros pour augmenter la résolution de la FFT et on forme la fonction $\Psi'(n, \varepsilon)$ utilisant l'équation (117). La fréquence pour laquelle cette fonction est maximale détermine notre estimation de fréquence. On remplace cette valeur de ε dans l'équation (113) pour calculer $\Psi(n, \varepsilon)$. C'est une valeur scalaire qui varie en faisant avancer la fenêtre d'observation. Quand cette fonction dépasse un seuil (à régler dans la pratique) le début du préambule peut être détecté en cherchant dans un petit voisinage du point en question. La Figure 2-15 présente une forme typique de cette fonction pour un SNR assez faible (2 dB).

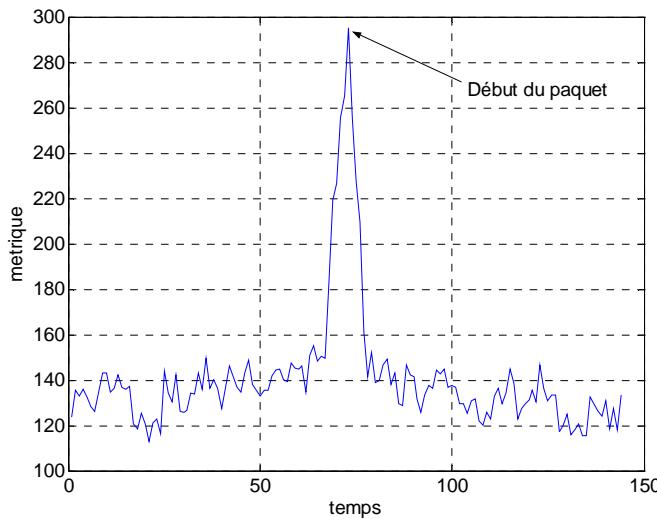


Figure 2-15- Relation (113) avec un SNR de 2 dB pour un système 4×4

Nous remarquons les propriétés suivantes pour la méthode proposée :

- On n'a pas besoin de deux préambules répétés ce qui était le cas dans les méthodes traditionnelles
- La résolution de la FFT peut être améliorée en insérant des zéros à la fin de la séquence r'_z
- Nous avons supposé que $|\varepsilon| < 0.5$. C'est-à-dire que les erreurs plus importantes sont censées être corrigées par un circuit de contrôle automatique de fréquence [21].
- Une fois la fréquence estimée, le canal peut être identifié utilisant l'équation (109).

Nous allons éprouver la performance de cette méthode par une série de simulation.

2.8.1 Performances

Pour évaluer la performance de notre algorithme nous utilisons les mêmes critères que le paragraphe 2.7.1 et le même scénario que [9]. C'est-à-dire que la taille d'un symbole OFDM est fixée à 64 échantillons, plus un préfixe cyclique de taille 8, avec un canal dont le PDP est donné en (103). Contrairement au [9] qui ne considérait que des systèmes SISO, nous supposons un système MIMO 4×4 . Dans un premier temps, nous souhaitons définir la résolution nécessaire de la FFT. Pour cette fin, nous évaluons la probabilité d'échec pour $m=5$ (voir l'équation (102)) quand l'erreur de fréquence n'a pas été compensée. Le résultat est donné sur la Figure 2-16. La motivation pour ce calcul est que nous voulions voir jusqu'à quelle précision il fallait compenser la fréquence d'offset. D'après les courbes, une erreur de fréquence résiduelle inférieure à 0.002 ne joue pas grandement sur la détection du début de paquet. Cette précision demande donc une FFT de taille minimale de 512.

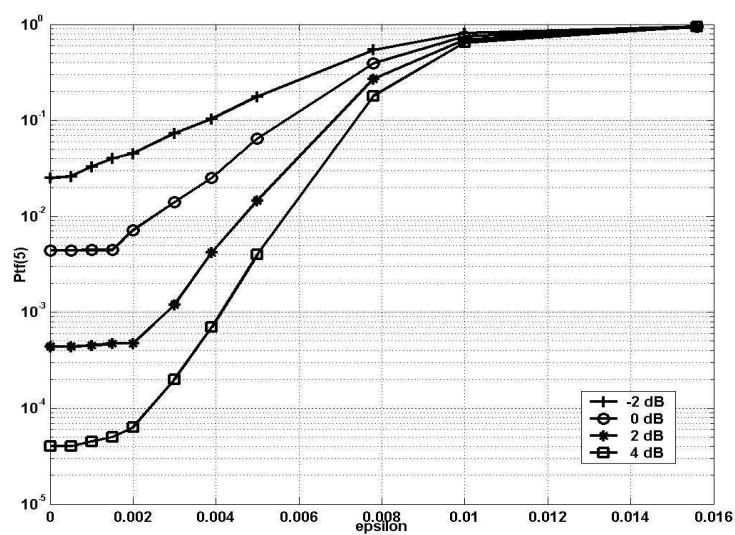


Figure 2-16- Probabilité d'échec de synchronisation de temps en fonction de l'erreur résiduelle de fréquence

Une fois la fréquence compensée, nous lançons une série de simulations pour le système complet utilisant l'algorithme proposé. On obtient les mêmes courbes que la Figure 2-12 qui démontre que l'ajout de la fréquence d'offset ne dégrade pas notre estimation de timing.

Malheureusement il n'y a pas d'étude similaire dans la littérature pour les systèmes MIMO pour que l'on puisse comparer le résultat.

2.8.2 Estimation fine de la fréquence

Utilisant la méthode donnée au paragraphe précédent, la variance de l'estimateur est limitée par la résolution de la FFT. Par contre, on sait qu'une fois la synchronisation faite, l'erreur de fréquence (ε) résiduelle est inférieure à 0,002 et que le timing est effectué avec précision. Ceci est le résultat de la Figure 2-16. C'est-à-dire qu'améliorer l'estimée de la fréquence ne va plus améliorer la performance de la synchronisation de temps. Dans ce paragraphe, nous supposons que le début du paquet est donc connu et nous tenterons d'améliorer l'estimation de fréquence.

Nous supposons dans un premier temps que l'algorithme précédent nous a fourni une estimation correcte du canal \mathbf{h}_0 . Connaissant le canal, nous pouvons utiliser les mêmes principes que les système SISO pour calculer la fréquence d'offset : corrélation entre la séquence reçue et la séquence attendue. La séquence attendue peut être réécrite sous la forme suivante :

$$\sum_{m=1}^{N_r} \mathbf{E}_0 \mathbf{C}_0 \mathbf{h}_{:,m} = \sum_{m=1}^{N_r} \sum_{s=0}^{N_{ob}-1} e^{j2\pi\varepsilon s} \mathbf{C}_0(s,:) \mathbf{h}_{:,m} \quad (119)$$

où \mathbf{C}_0 et $\mathbf{h}_{:,m}$ ont été définis au paragraphe 2.4 et $\mathbf{C}_0(s,:)$ est la ligne s de la matrice \mathbf{C}_0 . Dans cette équation n est remplacé par zéro car nous avons supposé que la synchronisation de temps a été effectuée avec succès.

La corrélation entre cette séquence attendue et la séquence effectivement reçue est à maximiser en faisant varier la fréquence d'offset inconnue :

$$\hat{\varepsilon} = \arg \max_{\varepsilon} \left| \sum_{m=1}^{N_r} \sum_{s=0}^{N_{ob}-1} \mathbf{h}_{:,m}^H \mathbf{C}_0^H(s,:) \mathbf{r}_{m,0}(s) e^{-j2\pi\varepsilon s} \right|^2 \quad (120)$$

Pour faire ressortir la dépendance en ε , on définit

$$y(s) = \sum_{m=1}^{N_r} \mathbf{h}_{:,m}^H \mathbf{C}_0^H(s,:) \mathbf{r}_{m,0}(s) \quad (121)$$

Remplaçant (121) dans (120), l'estimation de fréquence est :

$$\hat{\varepsilon} = \arg \max_{\varepsilon} \left| \sum_{s=0}^{N_{ob}-1} y(s) e^{-j2\pi\varepsilon s} \right|^2 \quad (122)$$

En n'écrivant que les termes dépendant en ε on peut écrire de manière équivalente :

$$\hat{\varepsilon} = \arg \max_{\varepsilon} \operatorname{Re} \left\{ \sum_{s=1}^{N_{ob}-1} y''(s) e^{-j2\pi\varepsilon s} \right\} \quad (123)$$

où

$$r''(s) = \sum_{p=0}^{N_{ob}-1} y(p)y^*(p+s) \quad (124)$$

La relation (123) étant une fonction scalaire de ε , on peut la maximiser en mettant la dérivée par rapport à ε égale à zéro. On obtient donc :

$$\operatorname{Im} \left\{ \sum_{s=1}^{N_{ob}-1} s r''(s) e^{-j2\pi\varepsilon s} \right\} = \sum_{s=1}^{N_{ob}-1} s |r''(s)| \sin(\arg(r''(s)) - 2\pi\varepsilon s) = 0 \quad (125)$$

Nous souhaitons démontrer que l'argument de sinus est quasiment nul. Pour cela, nous démontrons que la phase de $r''(s)$ est en moyenne égale à $2\pi\varepsilon s$. Remplaçant (121) dans (124) et aussi $r_{m,0}(s)$ par le signal attendu plus bruit, et en faisant sortir le bruit du sigma, on obtient la relation suivante :

$$r''(s) = \sum_{p=0}^{N_{ob}-1-s} \sum_{m=1}^{N_r} \left| C_0(p,:) h_{:,m} \right|^2 \left| C_0(p+s,:) h_{:,m} \right|^2 e^{j2\pi\varepsilon s} + bruit \quad (126)$$

Alors, à fort signal à bruit ou quand la taille de la fenêtre d'observation est assez longue, la phase de $r''(s)$ est quasiment égale à $2\pi\varepsilon s$. Alors, nous pouvons utiliser l'approximation $\sin(x) \approx x$ dans l'équation (125) et obtenir facilement l'approximation de la fréquence d'offset :

$$\hat{\varepsilon} = \frac{1}{2\pi} \frac{\sum_{s=1}^{N_{ob}-1} s |r''(s)| \arg(r''(s))}{\sum_{s=1}^{N_{ob}-1} s^2 |r''(s)|} \quad (127)$$

Pour éviter l'ambiguïté de phase, il ne faut pas que $|\arg(r''(s))|$ dépasse π . Ceci va limiter la plage de variation sur ε qui doit être inférieure à $1/2N_{ob}$. Nous pouvons donc résumer l'algorithme de la synchronisation comme suit.

D'abord, utilisant la méthode expliquée au paragraphe 2.8 nous calculons le début de paquet, les coefficients du canal et la fréquence d'offset. On peut donc compenser l'erreur de fréquence en multipliant la séquence reçue par la conjuguée de \mathbf{E}_0 . L'erreur de fréquence à ce moment là est inférieure à $1/N_{FFT}$. Utilisant ces valeurs estimées, on peut appliquer la méthode de ce paragraphe pour estimer la fréquence d'erreur résiduelle et ensuite recalculer les coeffi-

lients du canal utilisant la relation (109). Cette procédure peut être itérée pour améliorer au fur et à mesure l'estimation du canal et l'offset fréquentiel¹.

2.8.3 Performances

Pour évaluer l'amélioration apportée avec cette méthode itérative, une série de simulations a été effectuée. En simulation, un système 2×2 a été considéré et une erreur de fréquence égale à 0,1 a été fixée. Le PDP du canal est la même que l'équation (103), la taille de préambule est de 64 avec un préfixe cyclique de taille 8 comme le paragraphe 2.8.1. Une FFT de taille 512 a été utilisée. Le résultat est présenté sur la Figure 2-17. La première courbe est le résultat de [20] où le traitement se fait dans le domaine fréquentiel (après FFT) et où le timing est considéré parfait. La précision de la méthode est limitée par la résolution de la FFT. La deuxième courbe est la première itération de la méthode proposée. On remarque que nous arrivons rapidement à la limite imposée par la résolution de FFT. Les courbes suivantes montrent que la qualité de notre estimateur s'améliore au fur et à mesure des itérations. Pour pouvoir évaluer la qualité de notre système, il nous a fallu calculer la borne de Cramer-Rao. Cette limite a été déjà calculée pour les systèmes SISO-OFDM [22]- [23]. Nous avons étendu cette borne pour les systèmes MIMO-OFDM et le résultat est le suivant (voir Annexe III pour la preuve) :

$$CRB(\mathcal{E}) = \frac{\sigma_v^2}{8\pi^2} \left[\sum_{m=1}^{N_r} \mathbf{h}_{:m} \mathbf{C}_0^H \mathbf{D} \Theta \mathbf{D} \mathbf{C}_0 \mathbf{h}_{:m}^T \right]^{-1} \quad (128)$$

où Θ est une matrice constante définie à partir de la séquence émise $\Theta = I_{N_{obs}} - \mathbf{C}_0 (\mathbf{C}_0^H \mathbf{C}_0)^{-1} \mathbf{C}_0^H$ et $\mathbf{D} = \text{diag}(0, 1, \dots, N_{obs} - 1)$. La relation (128) se calcule pour chaque réalisation du canal MIMO. On peut donc calculer l'espérance de cette borne par rapport aux variables aléatoires \mathbf{h} pour obtenir la borne moyenne de Cramer-Rao. Le résultat a été présenté en tant que courbe de référence sur la Figure 2-17.

¹ Nous avons présenté cette idée dans la conférence PIMRC 2006

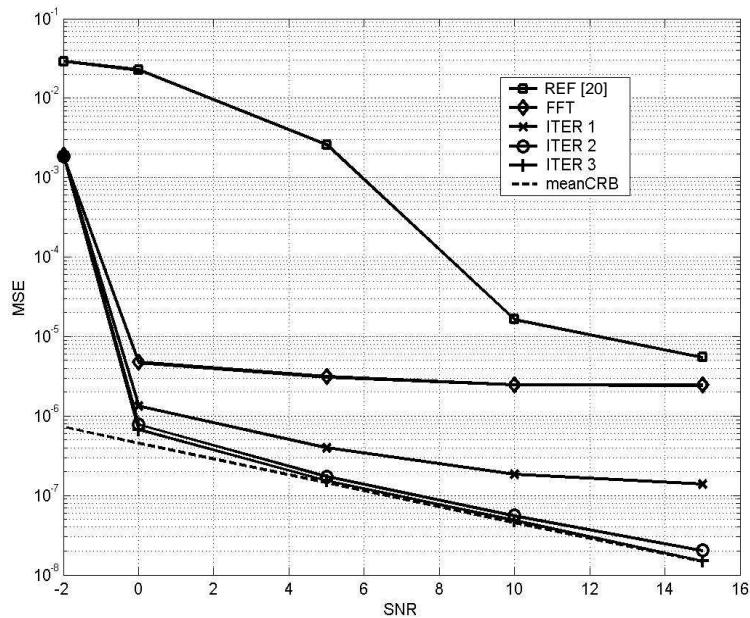


Figure 2-17 Performance de la méthode itérative pour un système MIMO-OFDM 2×2

En étudiant cette figure, on remarque que la variance de notre estimateur atteint cette borne minimale au bout de quelques itérations.

2.9 Conclusion

Dans ce chapitre, nous avons présenté les méthodes classiques de synchronisation de temps de symbole et de fréquence. Nous avons vu qu'il était possible d'effectuer la synchronisation conjointe de temps avec l'identification de canal, et de temps et de fréquence d'offset avec l'identification de canal. Pour pouvoir estimer la performance de nos systèmes, nous avons calculé la borne inférieure de Cramer-Rao de manière analytique, ce qui nous a permis de remarquer que nous sommes capables de l'atteindre en quelques itérations.

Annexe I Diagonalisation de la matrice circulante

On dit que la matrice $n \times n$ \mathbf{C} est une matrice circulante si \mathbf{C} a la forme

$$\mathbf{C} = \begin{bmatrix} c_0 & c_{n-1} & c_{n-2} & \cdots & c_1 \\ c_1 & c_0 & c_{n-1} & \cdots & c_2 \\ c_2 & c_1 & c_0 & \cdots & c_3 \\ \vdots & \vdots & \vdots & & \vdots \\ c_{n-1} & c_{n-2} & c_{n-3} & \cdots & c_0 \end{bmatrix} \quad (1)$$

\mathbf{C} est une matrice Toeplitz (chaque élément de la matrice $c_{i,j}$ ne dépend que de $i-j$). Chaque colonne est obtenue en faisant un décalage circulaire vers le bas de la colonne précédente. La matrice \mathbf{C} est complètement définie par sa première colonne. On peut écrire \mathbf{C} comme étant

$$\mathbf{C} = [\mathbf{c} \ \mathbf{R} \cdot \mathbf{c} \ \mathbf{R}^2 \cdot \mathbf{c} \ \cdots \ \mathbf{R}^{n-1} \cdot \mathbf{c}] \quad (2)$$

où

$$\mathbf{c} = [c_0 \ c_1 \ \cdots \ c_{n-1}]^T$$

$$\mathbf{R} = [e_2 \ e_3 \ \cdots \ e_n \ e_1]$$

et e_k la $k^{\text{ième}}$ colonne de la matrice identité.

A partir de cette écriture, plusieurs propriétés peuvent être déduites. L'une d'elles est très utile pour calculer le produit entre un vecteur et une matrice. La matrice circulante peut être diagonalisée par la matrice de Fourier unitaire, \mathbf{Q}

$$\mathbf{Q}_n = \begin{bmatrix} 1 & 1 & 1 & \cdots & 1 \\ 1 & w^1 & w^2 & \cdots & w^{(n-1)} \\ 1 & w^2 & w^4 & \cdots & w^{2(n-1)} \\ \vdots & \vdots & \vdots & & \vdots \\ 1 & w^{(n-1)} & w^{2(n-1)} & \cdots & w^{(n-1)(n-1)} \end{bmatrix} \quad (3)$$

$$\text{avec } w = e^{-j\frac{2\pi}{n}}.$$

Théorème

Si C est une matrice circulante, alors elle peut être diagonalisée

$$C = Q^H \Delta Q \quad (4)$$

avec $\Delta = \text{diag}(F \cdot c)$ une matrice diagonale.

Le produit entre le vecteur x et la matrice DFT Q donne la transformée de Fourier de x . La multiplication par les matrices de FFT/IFFT permet de diagonaliser la matrice C .

Pour le prouver, on a besoin de deux lemmes.

Lemme 1

Une matrice circulante est un polynôme en fonction de l'opérateur de décalage R .

$$C = c_0 I + c_1 R + \dots + c_{n-1} R^{n-1} \quad (5)$$

Preuve du lemme 1

En comparant les colonnes j de (5) et sachant que $R^k e_j = e_{(j+k) \bmod n}$, on a

$$\begin{aligned} (c_0 I + c_1 R + \dots + c_{n-1} R^{n-1}) \cdot e_j &= c_0 e_j + c_1 e_{j+1} + \dots + c_{n-j} e_n + c_{n-j+1} e_1 + \dots + c_{n-1} e_{j-1} \\ &= R^{j-1} \cdot c \\ &= C \cdot e_j \end{aligned} \quad (6)$$

Ceci implique que la colonne j de la matrice polynomiale est égale à la colonne j de la matrice C . Comme ceci est vrai pour tout j , le lemme 1 est vérifié.

A noter que la matrice R est aussi circulante. Le théorème énoncé devrait être vrai pour cette matrice spéciale. On montre ci après que la matrice de Fourier diagonalise R et avec le lemme 1, on montre que Q diagonalise les matrices circulantes de façon plus générale

Lemme 2

$$Q \cdot R = D \cdot Q \quad (7)$$

avec

$$D = \text{diag}(1, w, w^2, \dots, w^{n-1}) \quad (8)$$

Preuve du lemme 2

On prouve le lemme précédent en comparant le $kj^{\text{ième}}$ élément de $\mathbf{Q} \cdot \mathbf{R}$ avec le $kj^{\text{ième}}$ élément de $\mathbf{D} \cdot \mathbf{Q}$.

$$\begin{aligned} [\mathbf{Q} \cdot \mathbf{R}]_{kj} &= \frac{1}{\sqrt{n}} \begin{bmatrix} 1 & w^{(k-1)} & w^{2(k-1)} & \dots & w^{(n-1)(k-1)} \end{bmatrix} \cdot e_{j+k} \\ &= \frac{1}{\sqrt{n}} w^{j(k-1)} \end{aligned} \quad (9)$$

$$\begin{aligned} [\mathbf{D} \cdot \mathbf{Q}]_{kj} &= w^{(k-1)} e_k^T \frac{1}{\sqrt{n}} \begin{bmatrix} 1 & w^{(j-1)} & w^{2(j-1)} & \dots & w^{(n-1)(j-1)} \end{bmatrix}^T \\ &= \frac{1}{\sqrt{n}} w^{(k-1)} w^{(k-1)(j-1)} \\ &= \frac{1}{\sqrt{n}} w^{(k-1)j} \end{aligned} \quad (10)$$

Comme k et j sont arbitraires, la relation (7) est donc validée.

On a donc suffisamment de résultats pour prouver le théorème énoncé.

Preuve du théorème

On commence par montrer que $\mathbf{Q} \cdot \mathbf{C} \cdot \mathbf{Q}^H$ est diagonale. En remplaçant \mathbf{C} par la matrice polynomiale (5), et $\mathbf{Q} \cdot \mathbf{R} \cdot \mathbf{Q}^H = \mathbf{D}$ obtenue à partir de la lemme 2, on a

$$\begin{aligned} \mathbf{Q} \cdot \mathbf{C} \cdot \mathbf{Q}^H &= \mathbf{Q} \cdot \left(\sum_{k=0}^{n-1} c_k \mathbf{R}^k \right) \cdot \mathbf{Q}^H \\ &= \sum_{k=0}^{n-1} c_k \mathbf{Q} \cdot \mathbf{R}^k \cdot \mathbf{Q}^H \\ &= \sum_{k=0}^{n-1} c_k (\mathbf{Q} \cdot \mathbf{R} \cdot \mathbf{Q}^H)^k \\ &= \sum_{k=0}^{n-1} c_k \cdot \mathbf{D}^k \\ &= P(\mathbf{D}) \end{aligned} \quad (11)$$

avec $P(\mathbf{z}) = c_0 + c_1 \mathbf{z} + \dots + c_{n-1} \mathbf{z}^{n-1}$. Comme \mathbf{D} est une matrice diagonale, \mathbf{D}^k l'est aussi.

$$\mathbf{D}^k = \text{diag} \left(1, w^k, w^{2k}, \dots, w^{(n-1)k} \right) \quad (12)$$

$P(\mathbf{D})$ est donc une matrice diagonale.

Il reste à montrer que $P(\mathbf{D}) = \text{diag}(\mathbf{Q} \cdot \mathbf{c})$. Utilisant (12), l'élément k dans $P(\mathbf{D})$ peut être déterminé

$$\begin{aligned}
 [P(\mathbf{D})]_{kk} &= [c_0 \mathbf{I} + c_1 \mathbf{D} + \cdots + c_{n-1} \mathbf{D}^{n-1}]_{kk} \\
 &= c_0 + c_1 w^{k-1} + \cdots + c_{n-1} w^{(k-1)(n-1)} \\
 &= [\mathbf{Q} \cdot \mathbf{c}]_k
 \end{aligned} \tag{13}$$

La relation (13) prouve le théorème.

Annexe II Approximation optimale

Supposons que 'on souhaite résoudre l'équation $AXB=C$ où une solution exacte n'existe pas. On cherche donc à trouver le X qui minimise la quantité $\|AXB - C\|$.

Théorème

Le X qui minimise la quantité $\|AXB - C\|$ est $\hat{X} = A^\dagger C B^\dagger$ où A^\dagger est le pseudo inverse de A , défini comme :

$$A^\dagger = (A^H A)^{-1} A$$

Nous appliquons ce théorème dans le cas de l'équation simplifiée $AX=C$ où X et C sont des vecteurs colonnes. Le X qui minimise $\|AX - C\|$ est donc $\hat{X} = A^\dagger C$.

C'est la même forme que la relation (97) avec $C = r_n$, $A = I_{N_r} \otimes C_n$, $X = h$.

Le h qui minimise donc cette équation est :

$$\hat{h} = [(I_{N_r} \otimes C_n)^H (I_{N_r} \otimes C_n)]^{-1} (I_{N_r} \otimes C_n)^H r_n$$

Annexe III Borne inférieure de Cramer-Rao pour les systèmes MIMO-OFDM

Afin d'obtenir la borne inférieur de Cramer-Rao pour les systèmes MIMO, on se base sur les travaux publiés de Stocia [22] et de Morelli [23].

A partir de l'équation (78) et en tenant compte de l'erreur de fréquence, la séquence reçue pour toutes les antennes de réception peut s'écrire :

$$\mathbf{r}_0 = (\mathbf{I}_{N_r} \otimes \mathbf{E}_0)(\mathbf{I}_{N_r} \otimes \mathbf{C}_0)\mathbf{h} + \mathbf{v}_0$$

Avec les dimensions $(\mathbf{E}_0)_{N_{ob} \times N_{ob}}$, $(\mathbf{C}_0)_{N_{ob} \times LN_r}$ et $\mathbf{h}_{LN_r \times 1}$.

Cette équation a la même forme que l'équation 1 de[22] développée pour un système SISO. La borne Cramer-Rao est donc calculée et donnée par l'équation 11 de la même référence. En adaptant les notations et en tenant compte des produits Kronecker dus aux multi antennes la borne moyenne de Cramer-Rao est la suivante :

$$CRB(\varepsilon) = \frac{\sigma_v^2}{8\pi^2} \left[\mathbf{h}^H (\mathbf{I}_{N_r} \otimes \mathbf{C}_0)^H (\mathbf{I}_{N_r} \otimes \mathbf{D}) \Theta' (\mathbf{I}_{N_r} \otimes \mathbf{D}) (\mathbf{I}_{N_r} \otimes \mathbf{C}_0) \mathbf{h} \right]^{-1}$$

où

$$\Theta' = (\mathbf{I}_{N_r} \otimes \mathbf{I}_{N_{ob}}) - (\mathbf{I}_{N_r} \otimes \mathbf{C}_0) \left((\mathbf{I}_{N_r} \otimes \mathbf{C}_0)^H (\mathbf{I}_{N_r} \otimes \mathbf{C}_0) \right)^{-1} (\mathbf{I}_{N_r} \otimes \mathbf{C}_0)^H$$

$$\mathbf{D} = \text{diag}(0, 1, \dots, N_{ob} - 1)$$

Utilisant quelques propriétés du produit Kronecker : $(\mathbf{A} \otimes \mathbf{B})^{-1} = \mathbf{A}^{-1} \otimes \mathbf{B}^{-1}$, $(\mathbf{A} \otimes \mathbf{B})^H = \mathbf{A}^H \otimes \mathbf{B}^H$ et $(\mathbf{A} \otimes \mathbf{B})(\mathbf{C} \otimes \mathbf{D}) = \mathbf{A}\mathbf{C} \otimes \mathbf{B}\mathbf{D}$, on arrive à simplifier ce résultat :

$$CRB(\varepsilon) = \frac{\sigma_v^2}{8\pi^2} \left[\mathbf{h}^H \left(\mathbf{I}_{N_r} \otimes \left(\mathbf{C}_0^H \mathbf{D} (\mathbf{I}_{N_r} - \mathbf{C}_0 (\mathbf{C}_0^H \mathbf{C}_0)^{-1} \mathbf{C}_0^H) \mathbf{D} \mathbf{C}_0) \right) \mathbf{h} \right]^{-1}$$

En supprimant le produit Kronecker, cette équation peut être écrite sous la forme suivante

$$CRB(\varepsilon) = \frac{\sigma_v^2}{8\pi^2} \left[\sum_{m=1}^{N_r} \mathbf{h}_{:,m} \mathbf{C}_0^H \mathbf{D} \Theta \mathbf{D} \mathbf{C}_0 \mathbf{h}_{:,m}^T \right]^{-1}$$

Où Θ est une matrice constante définie comme :

$$\Theta = \mathbf{I}_{N_{ob}} - \mathbf{C}_0 (\mathbf{C}_0^H \mathbf{C}_0)^{-1} \mathbf{C}_0^H$$

Références

- [1] V. Meghdadi; A. Saemi, J.P. Cances, M.J. Syed, G. Ferre and J.M. Dumas, "Improving frequency synchronization by means of new correlation criteria for fine time synchronization in MIMO system" ELMAR, 2005. 47th International Symposium, 8-10 June 2005 Page(s):283 - 286
- [2] A. van Zelst and T. C. W. Schenk, "Implementation of a MIMO OFDM- based Wireless LAN System," IEEE Trans. on Signal Processing, vol. 52, no. 2, pp. 483–494, Feb. 2004.
- [3] H. Meyr, M. Oerder and A. Polydoros, "On sampling rate, analog prefiltering, and sufficient statistics for digital receivers," IEEE Trans. Commun., vol. 42, pp. 3208-3213, Dec. 1994.
- [4] N. Mody and G. L. Stüber, "Synchronization for MIMO OFDM systems," in *Proc. IEEE Global Commun. Conf.*, vol. 1, Nov. 2001, pp. 509-513.
- [5] M. Schmidl and D. C. Cox, "Robust frequency and timing synchronization for OFDM," IEEE Trans. Commun., vol. 45, pp. 1613-1621, Dec. 1997.
- [6] Suehiro N. and Hatori M., "Modulatable Orthogonal Sequences and their Application to SSMA Systems", Trans. on Inform. Theory, 34, 93–100, 1998.
- [7] C Fragouli, N. Al-Dhahir, W. Turin "Reduced-complexity training schemes for multiple-antenna broadband transmissions", Wireless Communications and Networking Conference, WCNC2002. Volume: 1 , 17-21 March 2002 Pages:78 - 83 vol.1
- [8] P.Moose, "A technique for orthogonal frequency division multiplexing frequency offset correction," IEEE Trans. Commun., vol. 42, pp. 2908-2914, Oct. 1994.
- [9] S. H. Müller-Weinfurtner, "On the optimality of metrics for coarse frame synchronization in OFDM: A comparison," in Proc. PIMRC , Sept. 1998, pp. 533-537.

- [10] J.J. van de Beek, M. Sandell, and P.O. Borjesson, "Timing and frequency synchronization in OFDM system using the cyclic prefix", In Proceedings of International Symposium on Synchronization, pp. 16-19, Essen,Germany, December 1995.
- [11] Speth M., Classen F., Meyr H. "Frame synchronization of OFDM systems in frequency selective fading channels", Vehicular Technology Conference, 1997 IEEE 47th, Vol 3, pp.1807-1811, May 1997
- [12] Speth M, Fechtel S A, et al. "Optimum receiver design for OFDM-based broadband transmission .part II. A case study." IEEE Trans. Communication, vol. 49(4): pp 571-578, Apr. 2001
- [13] T. Keller and L. Hanzo, "Orthogonal frequency division multiplex synchronization techniques for wireless local area networks," in Proc. PIMRC Taipei, Taiwan R.O.C., 1996, pp. 963-967.
- [14] Pierre R. Chevillat, Dietrich Maiwald, and Gottfried Ungerboeck, "Rapid Training of a Voiceband Data-Modem Receiver Employing an Equalizer with Fractional-T Spaced Coefficients," IEEE Trans. on Commun., vol. 35, no. 9, pp. 869-876, 1987
- [15] K. Shi and E. Serpedin, "Robust Coarse Frame and Carrier Synchronization for OFDM Systems: A New Metric and Performance Evaluation" IEEE Trans. on Wireless Communications, vol. 3, no. 3, pp. 1271-1284, July 2004.
- [16] R.L. Frank and S.A. Zadoff, "Phase Shift Pulse Codes With Good Periodic Correlation Properties", IRE Trans. on Information Theory, vol. IT-8, pp. 381-382. 1962.
- [17] T.C.W. Schenk and A. van Zelst, "Frequency Synchronization for MIMO OFDM Wireless LAN Systems," Proc. IEEE Vehicular Technology Conference (VTC Fall 2003), paper 05D-03, Oct. 2003.
- [18] D. C. Rife and R. R. Boorstyn, "Single-Tone Parameter Estimation From Discrete-Time Observations", IEEE Trans. Inform. Theory, vol. IT-20, pp. 591-598, Sept. 1974.
- [19] S.M. Kay, Fundamentals of Statistical Signal Processing. Prentice-Hall: Upper Saddle River, NJ 1993.
- [20] Y. Sun, Z. Xiong, and X. Wang, "EM-based iterative receiver design with carrier frequency offset estimation for MIMO OFDM systems," IEEE Transactions on Communications, 53(4), pp.581-586, April 2005

- [21] S.A. Fechtel, "OFDM carrier and sampling frequency synchronization and its performance on stationary and mobile channels," IEEE Trans. Consum. Electron., vol. 46, pp 438-441, Aug. 2000
- [22] P. Stoica and O Besson, "Training sequence design for frequency offset and frequency-selective channel estimation", IEEE Trans Communications, vol 51, 1910-1917, 2003.
- [23] M. Morelli and U. Mengali, "Carrier-frequency estimation for transmissions over selective channels," IEEE trans. Commun., vol. 48, pp.1580-1589, Sept 2000

Chapitre 3 Décodage Analogique

3.1 *Introduction*

Considérons un circuit analogique complexe à plusieurs entrées auxquelles nous avons connecté des sources de courant constant ou de tension constante. Le circuit, s'il n'est pas instable, va se stabiliser à un état donné où on peut mesurer la tension de chaque noeud ou le courant de chaque branche. C'est-à-dire que le circuit a su résoudre un ensemble d'équations non linéaires où les entrées et la topologie du circuit définissaient l'ensemble d'équations et un certain nombre de courants (ou de tensions) étaient les inconnus qui ont été calculés par le circuit. Le temps de calcul nécessaire pour résoudre cet ensemble d'équations non linéaires, est le temps d'établissement du circuit.

La question que l'on se pose est "comment peut-on proposer une topologie de circuit pour résoudre un problème qui s'avère complexe en terme d'implantation de l'algorithme de calcul".

Dans cette partie du rapport, nous essayons de répondre à cette question essentielle pour résoudre le problème de décodage de canal. C'est-à-dire que nous regardons les algorithmes de décodage et la façon dont ce problème doit être traité pour ensuite proposer des circuits analogiques de décodage. La résolution sera finalement un processus parallèle (au sens utilisé dans un circuit avec plusieurs processeurs), ce qui se fait de manière naturelle dans un circuit analogique. Ensuite, nous étudierons les circuits analogiques à base de transistors qui peuvent donner des équations ressemblant à celles qui existent dans les algorithmes de décodage classiques.

Les motivations pour explorer l'implantation analogique sont nombreuses.

- **La consommation** : D'abord, un circuit numérique a besoin de N fils pour véhiculer une seule information. La tension des ces fils varie entre la valeur la plus positive et la

valeur la plus négative ('0' ou '1' logique). Dans un circuit analogique, chaque information est présentée par un seul fil dont la tension ne varie pas en général sur la totalité de l'alimentation. De plus, les convertisseurs analogiques numériques (CAN), très gourmands en puissance, sont remplacés par des éléments échantillonneurs bloqueurs (*sample and hold*). Par exemple, un CAN à 6 bits opérant à 1,3 GEch/sec peut consommer 545 mW avec une alimentation de 3,3 volts [1].

- **La vitesse** : Dans les implantations numériques, la vitesse des CAN est un élément limitatif. De plus, le délai de traitement est proportionnel à la taille du paquet à traiter et au nombre d'itération à effectuer. Dans une implantation analogique, les CAN n'existent plus et vu la petite taille des blocs, un parallélisme massif est possible. De l'autre côté, l'implantation asynchrone nous permet de nous débarrasser des itérations cadencées par une horloge.

- **La surface** : Suivant le degré du parallélisme on peut obtenir un circuit beaucoup plus petit que son équivalent numérique.

Pour donner quelques exemples pratiques d'implantation des décodeurs analogiques rapportés dans la littérature, un décodeur Hamming (8,4,4) réalisé avec la technologie 0,25 μ m ne consomme que 55 μ w (140 nJ par bit d'information) [2], un décodeur Reed-Muller (16,5,8) réalisé avec la technologie 0,18 ne consomme que 55 μ w (11 nJ par bit d'information) [2], et un décodeur turbo de rendement 1/3 pour la norme UMTS réalisé avec la technologie 0,35 μ m ne consomme que moins de 10mw (moins de 2 nJ par bit d'information décodé) [21]. Si on effectue une implantation numérique toute parallèle de ce décodeur, d'après les calculs de Lustenberger et Loeliger, une puissance consommée supérieure à 11,5 W est obtenue [30].

Nous essayons de faire en sorte que ce rapport soit autosuffisant, c'est la raison pour laquelle nous commençons par un petit rappel sur le principe de LLR¹. Ensuite, la théorie des graphes sera présentée. On constate que les graphes peuvent être appliqués au code en bloc aussi bien qu'au code en treillis. Puis, nous présenterons la similitude entre quelques circuits à transistors et les équations obtenues dans les décodeurs ML ou MAP. Un certain nombre de blocs de base seront présentés dont nous nous servons dans les paragraphes suivants où nous nous occupons de décodage de type Viterbi et puis MAP. Nous étudierons l'utilité de décodage analogique pour le cas des turbo-codes (produit ou convolutif) et finalement une perspective pour

¹ Log Likelihood Ratio

3.1- Introduction

cette nouvelle voie sera donnée afin de résoudre les problèmes courants des récepteurs numériques.

3.2 Décodage

3.2.1 LLR

Nous considérons le canal équivalent en regroupant le canal physique, les filtres d'émission et de réception, les dispositifs de la synchronisation et d'égalisation du canal etc. pour arriver finalement à l'expression ci-dessous :

$$y = ax + n \quad (129)$$

où x est le signal envoyé par l'émetteur, n est un bruit gaussien à variance σ^2 et a est l'atténuation du canal. Nous nous mettons dans le cas d'une transmission binaire où $x \in \{+1, -1\}$ et les signaux sont réels. Dans ce cas la probabilité à posteriori sur x conditionnellement à la réception de y peut s'écrire :

$$p(x|y) = \frac{p(y|x)p(x)}{p(y)} \quad (130)$$

La probabilité de recevoir y à condition que x a été envoyé $P(y|x)$, supposant un canal gaussien, est la suivante :

$$p(y|x) = \frac{1}{\sqrt{2\pi}\sigma} \exp\left(\frac{-(y-ax)^2}{2\sigma^2}\right) \quad (131)$$

A partir de ces deux équations on peut définir le LLR du bit x connaissant la sortie du canal y comme suit :

$$L(x|y) = \ln \frac{P(x=+1|y)}{P(x=-1|y)} \quad (132)$$

En développant cette équation et en utilisant la loi bayésienne on obtient :

$$L(x|y) = \ln \frac{p(y|x=+1)}{p(y|x=-1)} + \ln \frac{P(x=+1)}{P(x=-1)} \quad (133)$$

En remplaçant l'équation (131) dans (133) on obtient :

$$L(x|y) = L_c y + L(x) \quad (134)$$

où

$$L(x) = \ln \frac{P(x=+1)}{P(x=-1)}, \quad L_c = \frac{2a}{\sigma^2} = 4a \frac{E_s}{N_0} \quad (135)$$

$L(x)$ constitue l'information à priori disponible sur le bit. Si cette information n'est pas disponible, ce qui se passe dans le premier passe d'un récepteur turbo, on suppose les symboles equiprobables et on initialise les $L(x)$ à zéro. Par contre, dans les itérations suivantes d'un décodeur turbo où chaque bit d'information est codé deux fois, chacun des décodeurs fournit son jugement sur le bit en question en tant qu'une information à priori au second décodeur. L_c détermine l'état du canal et s'appelle CSI¹.

Nous pouvons calculer la probabilité à posteriori sur x connaissant la sortie du canal. Cette probabilité est donnée ci-dessous :

$$\begin{aligned} P(x=1|y) &= \frac{p(y|x=1)P(x=1)}{p(y)} = \frac{1}{1 + \exp\left(-\frac{2ay}{\sigma^2}\right)} = \frac{1}{1 + \exp(-L_c y)} \\ &= \frac{\exp\left(\frac{L_c y}{2}\right)}{\exp\left(\frac{L_c y}{2}\right) + \exp\left(-\frac{L_c y}{2}\right)} \end{aligned} \quad (136)$$

Et de la même manière

$$P(x=-1|y) = \frac{\exp(-L_c y)}{1 + \exp(-L_c y)} = \frac{\exp\left(-\frac{L_c y}{2}\right)}{\exp\left(\frac{L_c y}{2}\right) + \exp\left(-\frac{L_c y}{2}\right)} \quad (137)$$

3.2.2 Loi de détection

Ayant sondé la sortie du canal, il existe deux catégories pour pondérer la décision sur le bit envoyé : ML et MAP. En ce référant au schéma de la Figure 3-1, le but de la détection est d'estimer les informations envoyées (la séquence \mathbf{u}) en examinant la séquence \mathbf{y} reçue.

¹ Channel State Information

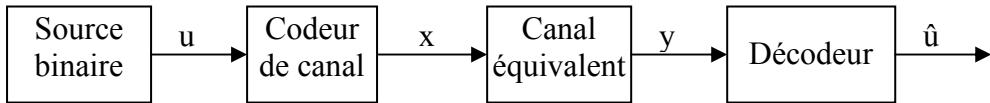


Figure 3-1- Système de transmission numérique

Sachant que les données envoyées dans chaque paquet sont corrélées, un détecteur MAP est celui qui maximise la probabilité de \mathbf{u} conditionnellement à la réception de \mathbf{y} où cette fois ci, \mathbf{u} est le vecteur des bits d'information émis et \mathbf{y} est le vecteur d'observation ou la séquence reçue. On peut donc écrire :

$$\hat{\mathbf{u}}_{MAP}(\mathbf{y}) = \arg \max_{\mathbf{u}} \{P(\mathbf{u} | \mathbf{y})\} \quad (138)$$

La méthode la plus utilisée pour calculer la probabilité à posteriori pour un code en treillis est l'algorithme de BCJR [29]. On peut aussi écrire cette équation sous la forme suivante :

$$\hat{\mathbf{u}}_{MAP}(\mathbf{y}) = \arg \max_{\mathbf{u}} \left\{ \frac{P(\mathbf{u}, \mathbf{y})}{p_Y(\mathbf{y})} \right\} = \arg \max_{\mathbf{u}} \{P(\mathbf{u}, \mathbf{y})\} \quad (139)$$

Si aucune information à priori n'est disponible sur les bits d'information, les probabilités sur différentes réalisations de \mathbf{u} sont égales et on peut écrire le critère de maximum de vraisemblance qui sera le même que le critère MAP :

$$\hat{\mathbf{u}}_{ML}(\mathbf{y}) = \arg \max_{\mathbf{u}} \{P(\mathbf{u}, \mathbf{y})\} = \arg \max_{\mathbf{u}} \{P(\mathbf{y} | \mathbf{u})p_U(\mathbf{u})\} = \arg \max_{\mathbf{u}} \{P(\mathbf{y} | \mathbf{u})\} \quad (140)$$

La méthode la plus utilisée pour calculer les bits les plus vraisemblables pour un code en treillis utilisant le critère ML est l'algorithme de Viterbi. Dans le cas où l'information à priori n'est pas disponible, ces deux méthodes sont équivalentes.

Puisque la relation entre les \mathbf{u} et les \mathbf{x} est injective, connaître \mathbf{u} ou \mathbf{x} est équivalent. Or le canal est supposé être gaussien et sans mémoire. On en déduit que :

$$\hat{\mathbf{u}}_{MAP}(\mathbf{y}) \leftrightarrow \arg \max_{\mathbf{x}} \{P(\mathbf{x} | \mathbf{y})\} = \arg \max_{\mathbf{x} \in C} \left\{ \prod_{i=1}^n P(x_i | y_i) \right\} \quad (141)$$

$$\hat{\mathbf{u}}_{ML}(\mathbf{y}) \leftrightarrow \arg \max_{\mathbf{x}} \{P(\mathbf{y} | \mathbf{x})\} = \arg \max_{\mathbf{x} \in C} \left\{ \prod_{i=1}^n P(y_i | x_i) \right\} \quad (142)$$

où n est la taille totale de la séquence codée reçue. Comme ces deux relations le montrent, on maximise la probabilité qu'une séquence valide du code soit envoyée. C'est-à-dire que le résultat est un code (ou une séquence) valide. C'est la caractéristique d'un décodeur de type Viterbi. On peut ne pas forcer la séquence reçue d'être une séquence valide mais maximiser la

3.2- Décodage

probabilité à posteriori (ou la vraisemblance) pour chaque bit d'information. Ceci peut aboutir à un meilleur TEB. On peut mettre en relation cette idée pour le $k^{\text{ième}}$ bit d'information comme suit :

$$\hat{u}_{kMAP}(y) = \arg \max_{u_k} \{P(u_k | y)\} = \arg \max_{u_k} \left\{ \sum_{u \in U, u_k = \hat{u}_k} P(u | y) \right\} \quad (143)$$

$$\hat{u}_{kML}(y) = \arg \max_{u_k} \{P(y | u_k)\} = \arg \max_{u_k} \left\{ \sum_{u \in U, u_k = \hat{u}_k} P(y | u) \right\} \quad (144)$$

Dans cette méthode, pour décider sur un bit, les probabilités de toutes les séquences valides donnant le bit en question égal à \hat{u}_k (0 ou 1) sont accumulées. Ceci va bien sûr donner un algorithme plus complexe à mettre en œuvre. On peut toujours utiliser l'injection et le fait que le canal est sans mémoire pour écrire les équations suivantes :

$$\hat{u}_{kMAP}(y) = \max_{u_k} \left\{ \sum_{u \in U, u_k = \hat{u}_k} \prod_{i=1}^n P(x_i | y_i) \right\} \quad (145)$$

$$\hat{u}_{kML}(y) = \max_{u_k} \left\{ \sum_{u \in U, u_k = \hat{u}_k} \prod_{i=1}^n P(y_i | x_i) \right\} \quad (146)$$

3.2.3 Algorithme de MAP

3.2.3.1 Codes en treillis

L'algorithme de MAP étant traité suffisamment dans la littérature [3], nous n'allons pas le détailler ici. Nous nous contentons de donner uniquement le résultat qui en découle. Supposons un code en treillis avec M_s états possibles schématisé sur la Figure 3-2.

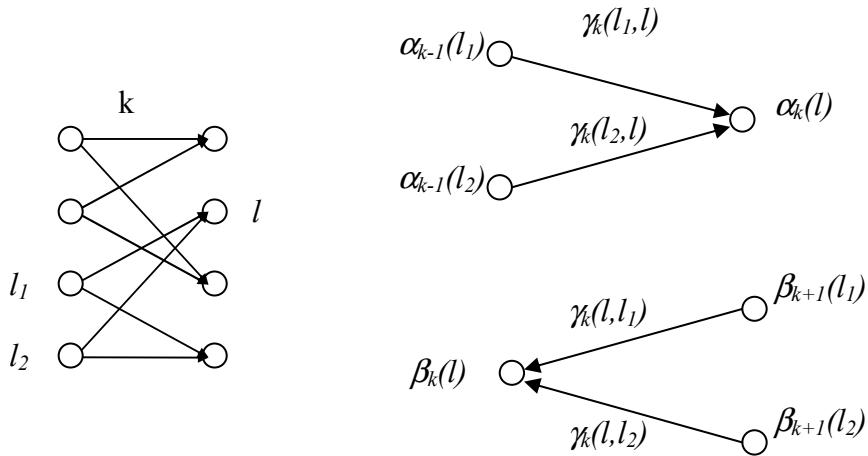


Figure 3-2- Calcul des alpha et des beta dans un treillis

Sur ce schéma la variable $\alpha_k(l)$ est définie comme étant la probabilité conjointe de se trouver dans l'état l à l'instant k et la séquence reçue de l'instant zéro jusqu'à l'instant $k-1$:

$$\alpha_k(l) = P(s_k = l, y_0, y_1, \dots, y_{k-1}) \quad (147)$$

Les valeurs de α peuvent être calculées pour tous les états et pour tous les temps utilisant la relation récursive suivante :

$$\alpha_k(l) = \sum_{l'} \alpha_{k-1}(l') \gamma_k(l', l) \quad (148)$$

La somme est à effectuer sur toutes les branches reliant un état précédent à l'instant $k-1$ vers l'état l à l'instant k . C'est une récursivité "en avant". On suppose que l'état de départ est l'état zéro, ce qui implique que $\alpha_0(0)=1$ et $\alpha_0(i)=0$ pour $i \neq 0$. On définit aussi la probabilité conditionnelle de recevoir le signal reçu de l'instant $k+1$ jusqu'à la fin à condition d'être dans l'état l à l'instant k :

$$\beta_k(l) = P(y_{k+1}, \dots, y_n | s_k = l) \quad (149)$$

Ici aussi, on peut obtenir la relation récursive ci-dessous pour calculer le β pour tous les états et pour tous les temps :

$$\beta_k(l) = \sum_{l'} \beta_{k+1}(l') \gamma_k(l, l') \quad (150)$$

où la somme est à effectuer sur toutes les branches reliant l'état l à l'instant k vers l'état l' à l'instant $k+1$. On suppose que l'état final n'est pas connu a priori ce qui nous oblige à initialiser les $\beta_n(i)$ à $1/M_s$ pour tout i . Dans certain cas, on préfère terminer le treillis à un état connu,

souvent zéro. Dans ce cas, $\beta_0(0)=1$ et $\beta_0(i)=0$ pour $i \neq 0$. La probabilité $\gamma_k(l, l')$, supposant un bruit additif gaussien, est donnée par l'équation suivante :

$$\gamma_k(l', l) = P(u_k) \exp \left(-\frac{\sum_j (x_k^j - y_k^j)^2}{2\sigma^2} \right) \quad (151)$$

où x_k^j représente le $j^{\text{ième}}$ signal pour le $k^{\text{ième}}$ bit d'information envoyé. Par exemple pour un code convolutif du rendement 0,5 j varie de 0 à 1 car pour chaque bit d'information, deux bits codés sont envoyés sur le canal. Le LLR sur le bit u_k peut être maintenant calculé avec la relation :

$$L(u_k | y) = \log \frac{\sum_{l, l'=0}^{M_s-1} \alpha_{k-1}(l') \gamma_k^1(l, l') \beta_k(l)}{\sum_{l, l'=0}^{M_s-1} \alpha_{k-1}(l') \gamma_k^0(l, l') \beta_k(l)} \quad (152)$$

La somme dans le numérateur est à effectuer sur les branches où un bit 1 serait décodé, et la somme dans le dénominateur pour les branches où un bit 0 serait décodé. Cela explique la signification des notations γ_k^0 et γ_k^1 .

Algorithme log-map

Sachant que les probabilités dans l'équation (152) sont extrêmement petites, une programmation traditionnelle résulte en erreurs "under flow". Pour éviter ce problème il est préférable de travailler avec le logarithme des probabilités. Dans ce cas, les α , β et γ sont remplacés par leur logarithme respectifs $\bar{\alpha}$, $\bar{\beta}$ et $\bar{\gamma}$:

$$\bar{\alpha}_k(l) = \log \alpha_k(l), \bar{\beta}_k(l) = \log \beta_k(l), \bar{\gamma}_k^i(l, l') = \log \gamma_k^i(l, l') \quad (153)$$

$$\bar{\alpha}_k(l) = \log \sum_{l'=0}^{M_s-1} e^{\bar{\alpha}_{k-1}(l') + \bar{\gamma}_k(l', l)}, \bar{\alpha}_0(0) = 0, \bar{\alpha}_0(i) = -\infty \text{ pour } i \neq 0$$

$$\bar{\beta}_k(l) = \log \sum_{l'=0}^{M_s-1} e^{\bar{\beta}_{k+1}(l') + \bar{\gamma}_k(l, l')}, \bar{\beta}_0(0) = 0, \bar{\beta}_0(i) = -\infty \text{ pour } i \neq 0$$

On a supposé que le treillis se ferme à l'état zéro. Ainsi, le LLR sur le bit u_k se calcule par :

$$L(u_k | y) = \log \frac{\sum_{l, l'=0}^{M_s-1} e^{\bar{\alpha}_{k-1}(l') + \bar{\gamma}_k^1(l, l') + \bar{\beta}_k(l)}}{\sum_{l, l'=0}^{M_s-1} e^{\bar{\alpha}_{k-1}(l') + \bar{\gamma}_k^0(l, l') + \bar{\beta}_k(l)}} \quad (154)$$

Max-log-map

En comparant (152) et (154), on remarque que les multiplications sont remplacés par des additions. Le problème est la fonction exponentielle. On peut simplifier ce résultat en faisant une approximation qui simplifiera énormément la réalisation mais qui ne dégrade pas pour autant la performance en terme de TEB :

$$\log(e^{\delta_1} + e^{\delta_2} + \dots + e^{\delta_n}) \approx \max_i(\delta_i)$$

Dans ce cas, la relation (154) se transformera en :

$$L(u_k | y) = \max_{l,l'} \left[\bar{\alpha}_{k-1}(l') + \bar{\gamma}_k^1(l, l') + \bar{\beta}_k(l) \right] - \max_{l,l'} \left[\bar{\alpha}_{k-1}(l') + \bar{\gamma}_k^0(l, l') + \bar{\beta}_k(l) \right] \quad (155)$$

Une réalisation sans multiplicateur est donc possible. Ainsi, pour chaque état du treillis un bloc comme celui de la Figure 3-3 est à réaliser pour calculer les α et un bloc comme celui de la Figure 3-4 pour calculer les β . Et finalement à chaque étage, on réalise l'équation (155).

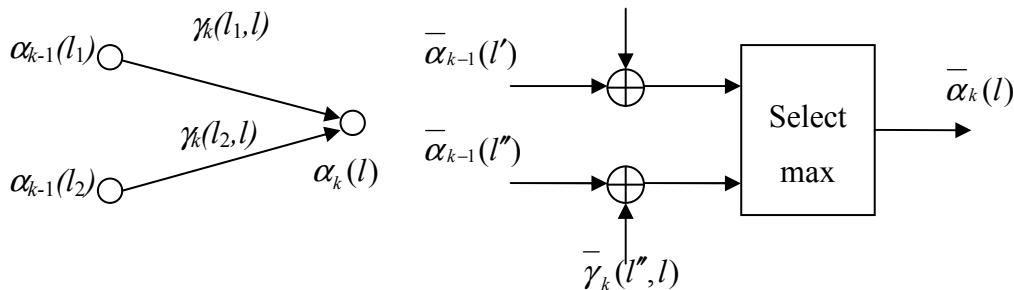


Figure 3-3- Circuit du calcul des alpha

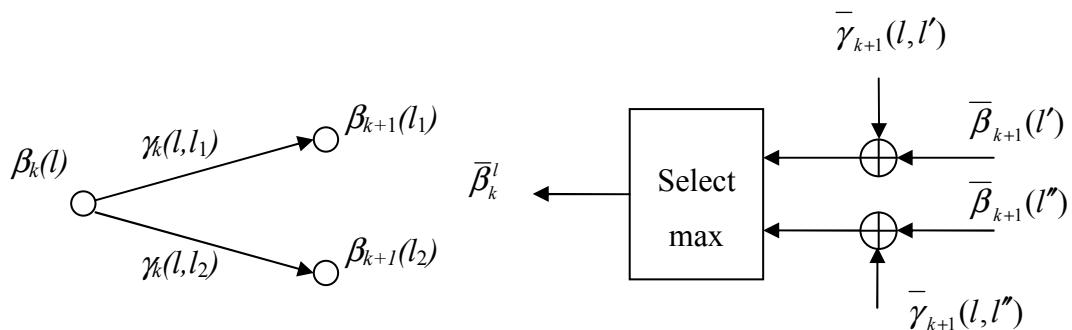


Figure 3-4- Circuit du calcul des beta

3.2.3.2 Code en bloc

Le code en bloc va affecter à une séquence de taille k une séquence de taille N . Les mots du code engendrent donc un sous espace de dimension k dans un espace de dimension N . Chaque coordonnée du mot de code est un élément binaire obtenu par une combinaison linéaire des bits d'information. Si les bits codés sont présentés par x_i , et pour un code linéaire, on peut écrire :

$$\sum_{i=1}^N \oplus x_i = x_1 \oplus x_2 \oplus \dots \oplus x_n = 0$$

où l'opération d'addition est modulo 2. Cette équation nous aide à reconnaître le bit j connaissant les autres bits.

$$x_j = \sum_{i=1, i \neq j}^N \oplus x_i$$

Imaginons que maintenant les x_i passent à travers un canal bruité et nous n'observons que la sortie de ce canal : $y_k = ax_k + n_k$. Nous ne disposons plus d'informations binaires mais les probabilités à posteriori sur chaque bit en connaissant la sortie du canal : $L(x_k|y_k)$ (voir équation (133) pour le cas où l'information à priori est nulle). Maintenant, que peut-on dire sur le bit j quand on connaît la sortie du canal pour les autres bits ? Il est démontré [4] que l'on peut calculer le LLR sur le bit j connaissant les autres LLR par la relation :

$$L_e(x_j) = 2 \tanh^{-1} \prod_{i=1, i \neq j}^N \tanh(L(x_i|Y)/2) \quad (156)$$

L'indice "e" (pour extrinsèque) est pour insister que c'est une information extrinsèque venant des bits autres que le bit concerné. Cette opération a été schématisée par l'opérateur "box-plus" défini par Hagenauer [5] comme présente la Figure 3-5.

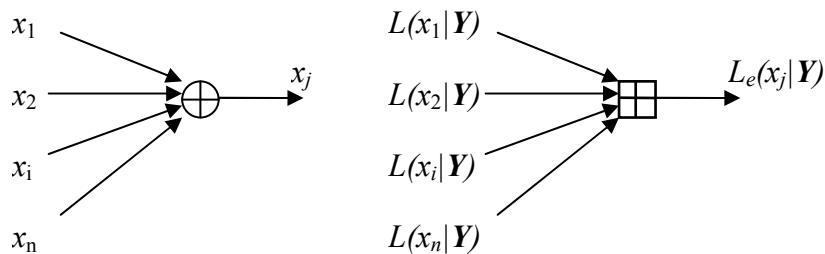


Figure 3-5- Opérateur box-plus

Par exemple pour trois bits avec $L(x_2|Y)=-0.3$ et $L(x_3|Y)=-5.5$, l'information extrinsèque que nous avons sur le bit x_1 est la suivante :

$$L_e(x_1|Y) = 2\tanh^{-1}[\tanh(-0.3/2).\tanh(-5.5/2)] \approx 0.3 = \text{sign}(-0.3).\text{sign}(-5.5).\min(0.3, 5.5)$$

Compte tenu des signes, les bits x_2 et x_3 sont plus probablement 1 et donc le bit x_1 est probablement 0, ce qui est conforme à ce que nous attendions.

Après une introduction sur les circuits analogiques utilisés dans ce rapport, nous présenterons dans les paragraphes suivants l'algorithme de décodage MAP pour les codes en bloc utilisant une présentation graphique. On envisage donc un algorithme itératif pour décoder une séquence reçue. Dans un premier temps, on considère chaque bit individuel et on réunit l'avis des autres bits sur ce bit en question, ce qui constitue l'information extrinsèque sur ce bit. Ensuite, cette information extrinsèque plus l'information venant du canal sur le bit en question, constituent le LLR sur le bit. Ces LLR sont à nouveau considérés comme une information utilisée dans les équations de parité, ce qui va créer encore des LLR, et ainsi de suite. Quand on aura défini les graphes, cette procédure se verra beaucoup plus clairement.

3.3 **Eléments du circuit analogique**

Dans les circuits présentés dans ce rapport, là où des caractéristiques exponentielles sont nécessaires, nous utilisons dans nos schémas des transistors bipolaires. Sachant que nous pouvons également utiliser des MOSFET (N canal ou P canal) schématisés sur la Figure 3-6 dans la zone sous seuil (*subthreshold*), c'est-à-dire que la tension de grille-source est inférieure à la tension de seuil. Dans ce cas la relation entre le courant de drain et la tension de grille-source est la suivante :

$$I_d = I_0 e^{V_{gs}/\zeta V_T} \quad \text{où } V_T = KT/q$$

K est la constante de Boltzmann, T est la température absolue et $\zeta > 1$ est un facteur correcteur. Cette relation reste valide pour des tensions drain-source supérieures à 200 mv approximativement. On remarque que le courant de drain est exponentiel par rapport à la tension de grille-source. Ceci ressemble à des transistors bipolaires et c'est la raison pour laquelle là où nous souhaitons obtenir des caractéristiques exponentielles, nous pouvons utiliser soit les MOS dans leurs zones subthreshold soit des transistors bipolaires.

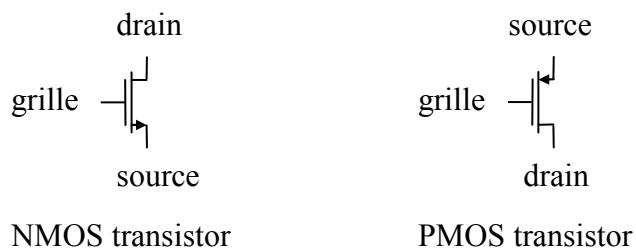


Figure 3-6- Schéma des transistors MOS

Par contre, pour les miroirs de courant nous utilisons de préférence des MOS car le courant de grille est nul, ce qui n'est pas le cas pour des transistors bipolaires.

3.3.1 **Transformation de LLR en probabilité**

Un des circuits nécessaires est celui qui transforme les probabilités en LLR. On considère le circuit ci-dessous :

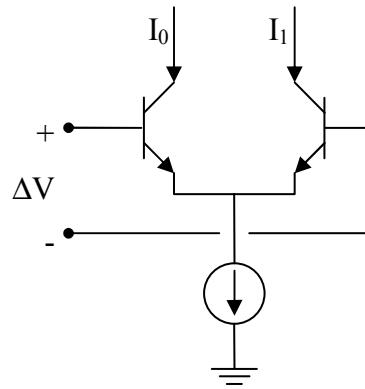


Figure 3-7- Transformation LLR en probabilité

Si on écrit la relation courant - tension pour ce circuit, on arrive à l'équation :

$$\frac{\Delta V}{V_T} = \ln \frac{I_0/I}{I_1/I}$$

En comparant cette équation avec la relation entre LLR et les probabilités données en (132) ou (135), on remarque que $\Delta V/V_T$ peut être $L(x)$ et que $P(X=0)=I_0/I$ et $P(X=1)=I_1/I$. Vu les contraintes du circuit, on remarque que $P(X=0)+P(X=1)=1$. La conclusion est que ce circuit prend $L(X)$ comme une tension en entrée et fournit les probabilités $X=0$ et $X=1$ sous forme de courant.

3.3.2 Transformation des probabilités en LLR

Le but est de récupérer les probabilités $P(X=0)$ et $P(X=1)$ et de fournir le LLR correspondant.

Considérons le circuit ci-dessous :

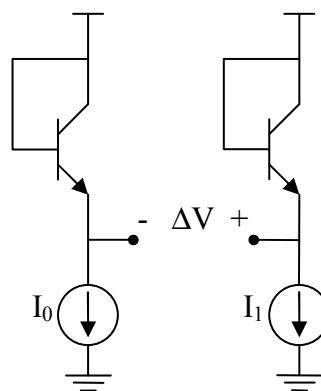


Figure 3-8- Transformation probabilité en LLR

En écrivant la loi de tension on obtient facilement la relation ci-dessous entre les courants I_0 et I_1 et la tension de sortie ΔV :

$$\frac{\Delta V}{V_T} = \ln \frac{I_0}{I_1}$$

En définissant $L(x) = \Delta V/V_T$, $I = I_0 + I_1$, $P(X=0) = I_0/I$ et $P(X=1) = I_1/I$ on arrive donc à donner $L(X)$ comme une tension de sortie en fonction des courants représentant des probabilités.

3.3.3 Opération Box-Plus

Nous avons remarqué dans le paragraphe précédent que la version "soft" d'un additionneur modulo-2 était présentée par l'opérateur Box-Plus. Nous souhaitons présenter le circuit qui prend en entrée $L(X_1)$ et $L(X_2)$ et sort le résultat indiqué en équation (156), à savoir :

$$X = 2 \tanh^{-1} \left[\tanh \frac{X_1}{2} \tanh \frac{X_2}{2} \right]$$

Il n'est pas difficile de montrer que le circuit de Gilbert donné sur la Figure 3-9 calcule effectivement cette expression [6]. Dans ce circuit, $X = \Delta V/V_T$ est le LLR de sortie, et $X_1 = \Delta V_1/V_T$ et $X_2 = \Delta V_2/V_T$ constituent les LLR en entrée. On remarque que les entrées et la sortie sont en mode tension.

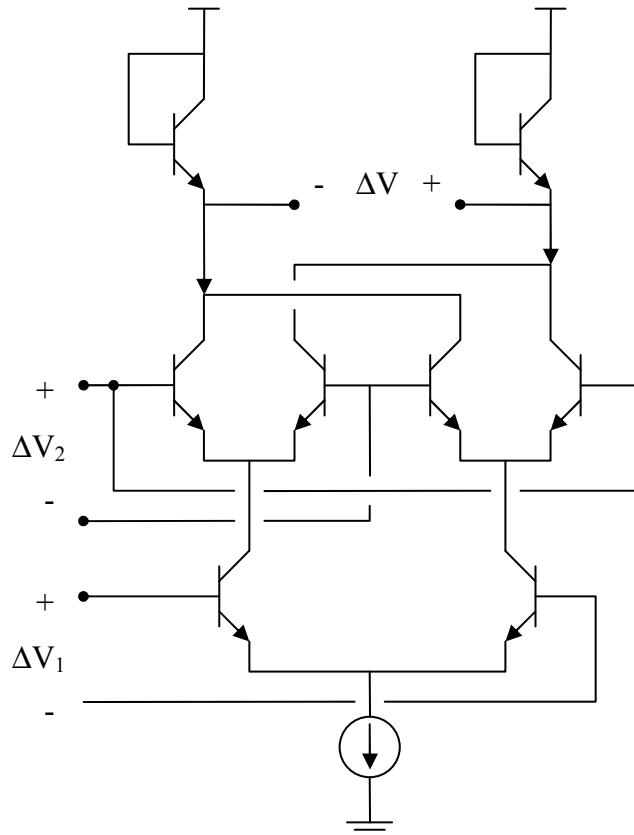


Figure 3-9- Multiplieur Gilbert calculant l'opérateur box-plus

3.3.4 Miroir de courant

Dans beaucoup de cas, nous avons besoin de copier ou dupliquer un courant électrique. Le circuit utilisé est le suivant :

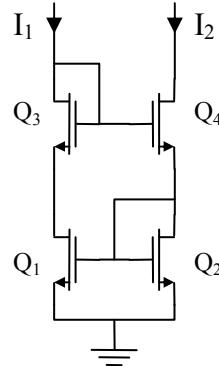


Figure 3-10- Miroir de courant à base de N-CMOS

L'entrée est le courant I_1 et la sortie est le courant $I_2=I_1$. Ce circuit est désensibilisé au changement de la tension drain-source du transistor de sortie $Q4$ grâce à la paire $Q1$ et $Q2$.

3.3.5 Winner-Take-All

Dans certains cas, comme le bloc compare-add-select utilisé dans un décodage de type Viterbi, nous devons donner la valeur maximale de plusieurs courants. Lazzaro [7] en 1989 a proposé un circuit relativement simple capable de comparer N courants et sélectionner le courant maximal (Figure 3-11).

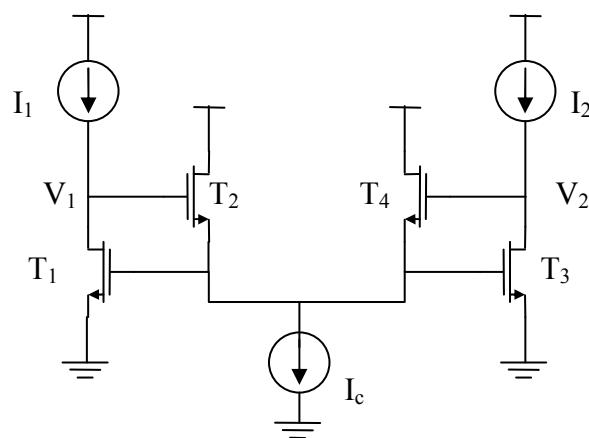


Figure 3-11- Circuit de Winner-Take-All

Si $I_1=I_2$, on a $V_{GS1}=V_{GS3}$ et $I_{T2}=I_{T4}=I_c/2$. Si I_1 augmente et devient $I_1+\delta$, le V_{GS1} sera augmenté. Puisque T_1 et T_3 partagent la même grille et la même source, V_{GS2} sera augmentée et donc le

3.3- Eléments du circuit analogique

transistor T_3 réclame plus de courant. Or, le courant de drain de T_3 est limité à I_2 . Alors la tension de drain de T_3 va baisser pour demander plus de courant. Ceci va faire baisser la tension de grille de T_4 qui va l'entraîner dans sa zone off. Cela signifie que tout le courant I_c passe à travers T_2 : $I_{T2}=I_c$, $I_{T4}=0$. En examinant donc V_1 et V_2 , on peut détecter lequel de ces deux courants (I_1 ou I_2) était plus fort.

En modifiant légèrement ce schéma, on peut donner le circuit nécessaire dans une implantation de Viterbi. Ici, on cherche dans chaque nœud, à donner la métrique cumulée la plus importante, comme présente la Figure 3-12.

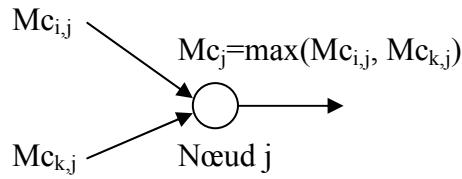


Figure 3-12- Opération de sélection de la valeur maximale dans l'algorithme de Viterbi

Le circuit implantant ce schéma, inspiré des travaux de Baturone [8], est donné ci-dessous :

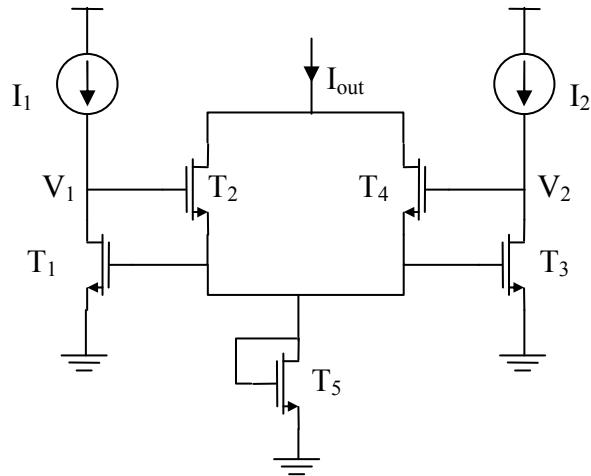


Figure 3-13 Comparaison et sélection

Ici, la source de courant I_c est remplacée par une diode dont le courant est contrôlé par la tension grille-source de T_1 ou T_3 . De plus, la somme de deux courants de T_2 et de T_4 constitue la sortie. Ainsi, vu la description de WTA que nous avons donné ci-dessus, $I_{out}=\max(I_1, I_2)$. On peut aussi enchaîner des étages pour réaliser la relation $I_{out}=\max(I_1, I_2, \dots, I_N)$.

Pour améliorer la précision, comme les miroirs de courant de Wilson, on ajoute deux transistors sur les drains de T_1 et T_3 pour égaliser la tension de drain-source des paires constituant les miroirs de courant comme présenté ci-dessous.

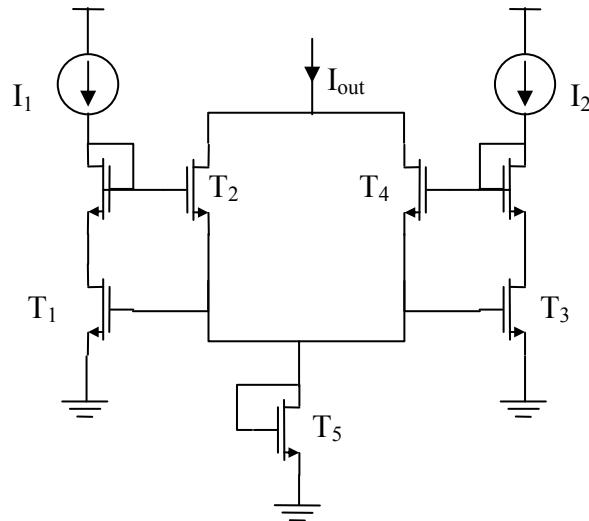


Figure 3-14- Comparaison et sélection amélioré

Nous avons proposé [9]¹ une architecture à base de cette cellule pour implanter l'algorithme de Viterbi. Dans cette architecture, une sortie est aussi prévue pour l'opération de "trace back".

3.4 Décodage analogique pour les codes en bloc

3.4.1 Introduction

Le problème d'ajouter de la redondance à une séquence d'information est de transformer un mot d'information de taille k en un mot de code de taille n ($n > k$). En effet, dans un espace n dimensionnel du code, il n'y a qu'un sous espace de dimension k qui contient les mots valides. Le problème de décodage est alors de trouver un mot valide du code qui aurait été envoyé plus vraisemblablement.

¹ Cet article, publié dans la conférence ICEE2006, est donné dans Annexe III .

3.4- Décodage analogique pour les codes en bloc

Deux critères sont couramment utilisés pour sélectionner le mots du code "le plus proche" de la séquence reçue : ML et MAP.

Le critère de MAP sélectionne le mot valide qui présente plus de probabilité d'occurrence conditionnellement à la séquence observée (voir équation (138)).

Le critère de maximum de vraisemblance vérifie tous les mots susceptibles d'être envoyés et calcule pour chaque cas, la probabilité que la sortie du canal soit égale au signal observé. Il sélectionne ensuite celui qui a donné plus de chance à notre séquence d'observation (voir l'équation (140)).

Si aucune information à priori n'est disponible sur la séquence émise \mathbf{u} , tous les mots à tester seront considérés équiprobables et la seule information disponible est la sortie du canal (l'observation). Dans ce cas les deux critères sont mathématiquement et précisément équivalents.

On peut aussi ne pas chercher à trouver un mot valide du code. Pour chaque bit u_i individuel la décision se fait séparément. Dans ce cas aussi, comme le cas précédent où la séquence complète était estimée, les deux critères de ML et MAP peuvent être utilisés (voir (143) et (144)).

Dans un premier temps, nous expliquons l'utilisation de graphe factoriel pour un décodage des codes en bloc. Soit \mathbf{u} la séquence d'information envoyée de taille k et \mathbf{x} la séquence codée de taille n . Il existe une relation surjective entre \mathbf{u} et \mathbf{x} . Cette relation est définie par la matrice génératrice du code appelée $\mathbf{G}_{k \times n}$ telle que

$$\mathbf{x}_{1 \times n} = \mathbf{u}_{1 \times k} \mathbf{G}_{k \times n}$$

où toutes les opérations sont modulo 2.

A partir de la matrice \mathbf{G} on peut trouver une matrice $\mathbf{H}_{n-k \times n}$, appelée matrice de parité, telle que

$$\mathbf{G}\mathbf{H}^T = \mathbf{0}_{k \times k}$$

Pour tester si une séquence binaire reçue de taille n appartient au code, on calcule $\mathbf{H}\mathbf{x}^T$. Si le résultat, appelé syndrome, est nul, la séquence \mathbf{x} est un mot valide du code. La matrice de parité \mathbf{H} détermine donc $n-k$ équations que le décodeur cherche à satisfaire. Par contre, si le syndrome n'est pas nul, le rôle du décodeur est de trouver un mot valide du code le plus "proche" de ce qu'il a reçu. Le critère peut être MAP ou ML.

Le code en bloc peut être présenté graphiquement par un graphe factoriel.

Exemple : La matrice génératrice d'un code (8,4) qui est une version étendue du code Hamming (7,4) est la suivante :

$$H = \begin{bmatrix} 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 0 & 0 & 0 & 1 & 1 \end{bmatrix}$$

On se rend compte que $HH^T = 0_{4 \times 4}$, ce qui signifie que la matrice $\mathbf{G}=\mathbf{H}$. Si on reçoit une séquence binaire $x=(x_1, \dots, x_8)$, les quatre équations de parité sont les suivantes :

$$\begin{aligned} x_1 \oplus x_2 \oplus x_3 \oplus x_4 &= 0 \\ x_3 \oplus x_4 \oplus x_6 \oplus x_7 &= 0 \\ x_5 \oplus x_6 \oplus x_7 \oplus x_8 &= 0 \\ x_2 \oplus x_3 \oplus x_7 \oplus x_8 &= 0 \end{aligned} \quad (157)$$

Utilisant la théorie de graphe factoriel, on peut schématiser ces équations de parité sous la forme de la Figure 3-15.

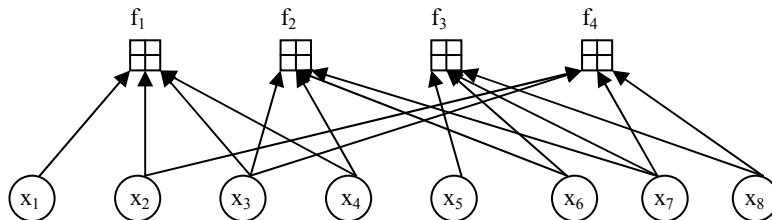


Figure 3-15- Graphe pour le code en bloc (8,4)

Ce graphe est constitué de deux types de nœuds : nœuds de variable et nœud de fonction. Les nœuds de variable contiendront les LLR sur les bits codés et les nœuds de fonction déterminent la relation qui existe entre les différents bits codés.

3.4.2 Décodage itératif

La littérature est assez riche pour la théorie des graphes et le passage des messages entre les nœuds de variable et les nœuds de fonction [10]. Nous allons juste expliquer la façon dont on utilise le schéma pour un décodage sans entrer dans les détails théoriques.

Le but est de donner une estimation de fiabilité (*soft value* ou encore LLR) pour chaque bit conditionnellement à notre observation.

Supposons un canal sans mémoire, on peut écrire que $\mathbf{y} = \mathbf{ax} + \mathbf{n}$ où $\mathbf{x} = (x_1, x_2, \dots, x_n)$ $x_i \in \{-1, +1\}$ est la séquence codée envoyée et $\mathbf{y} = (y_1, y_2, \dots, y_n)$ $y_i \in \mathbb{R}$ est la séquence reçue. A partir de y_i on peut calculer le LLR sur le bit x_i envoyé.

$$L(x_i | \mathbf{Y}) = L_c y_i + L_e(x_i | \mathbf{Y})$$

où $L_e(x)$ est l'information à priori sur le bit x_i et L_c est le CSI (voir équation (134)).

Dans une première itération, la seule information disponible est celle du canal; l'information à priori est nulle. Les nœuds de variable envoient donc l'information disponible $L(x_i | \mathbf{Y}) = L_c y_i$ à tous les nœuds de fonction auxquels ils sont connectés. Cette opération est une demi itération. La deuxième moitié de l'itération concerne l'envoi des messages de retour de la part des nœuds de fonction vers les nœuds de variable.

Le message que le nœud f_i envoie vers le nœud v_j est une fonction de toutes les autres entrées du nœud f_i . En se référant à la Figure 3-16-a, ce message est :

$$\text{Message } f_i \text{ vers } v_j = \mu_{f \rightarrow v}(i, j) = 2 \tanh^{-1} \left(\tanh \frac{e_1}{2} \tanh \frac{e_2}{2} \dots \tanh \frac{e_l}{2} \right)$$

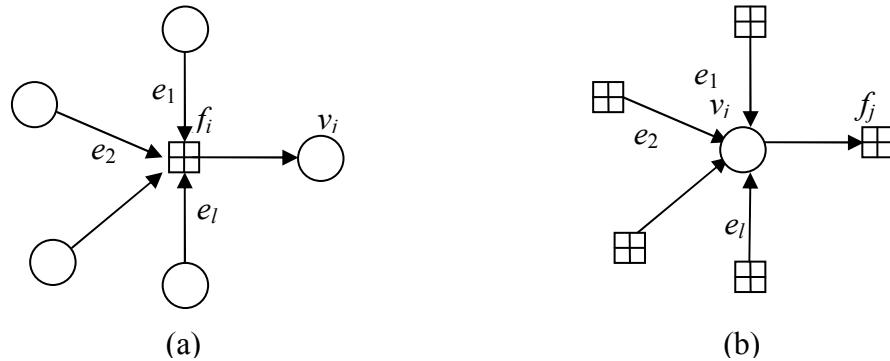


Figure 3-16- Message envoyé (a) d'un nœud de fonction vers un nœud de variable; (b) d'un nœud de variable vers un nœud de fonction

Ainsi la première itération se termine. Dans l'itération suivante, chaque nœud de variable v_i envoie vers le nœud de fonction f_j auquel il est connecté, la somme de toutes les autres entrées à laquelle il ajoute l'information du canal (voir Figure 3-16-b) :

$$\text{Message } v_i \text{ vers } f_j = \mu_{v \rightarrow f}(i, j) = e_1 + e_2 + \dots + e_l + L(x_i | y_i)$$

La première partie de cette équation est en fait, l'avis de tous les autres bits de la séquence reçue sur le bit courant (x_i) (information extrinsèque) et le second terme est l'avis du canal

(information intrinsèque). A l'issue de chaque itération le LLR sur des bits peut être calculé à partir des LLR reçus dans le nœud correspondant. Le LLR sur le bit x_i est donc :

$$LLR(x_i) = \sum_j \mu_{f \rightarrow v}(j, i) + L(x_i | y_i)$$

Le signe de $LLR(x_i)$ donne la décision sur le bit. La condition d'arrêt d'itération est que la séquence \mathbf{x} soit un mot valide du code, c'est-à-dire qu'il satisfait les équations de parité. Un programme Matlab implantant cet algorithme est donné en Annexe II .

3.4.3 **Implantation avec circuit analogique**

Pour réaliser un graphe factoriel tel qu'il est présenté sur la Figure 3-15, deux types de circuit sont nécessaires : le circuit correspondant au nœud de variable et celui de nœud de fonction.

3.4.3.1 *Nœud de variable*

Les nœuds de variable ne sont que des additionneurs. En mode courant, on n'a qu'à connecter les fils pour additionner leur courant. Dans les circuits, pour représenter les LLR on utilise les tensions. Pour travailler en mode courant, il faut alors une conversion tension-courant. On peut également travailler directement en mode tension, dans ce cas les nœuds de variable sont des additionneurs. Le circuit d'additionneur avec les circuits "level shifter" pour ramener les tensions au même niveau est présenté sur la Figure 3-17.

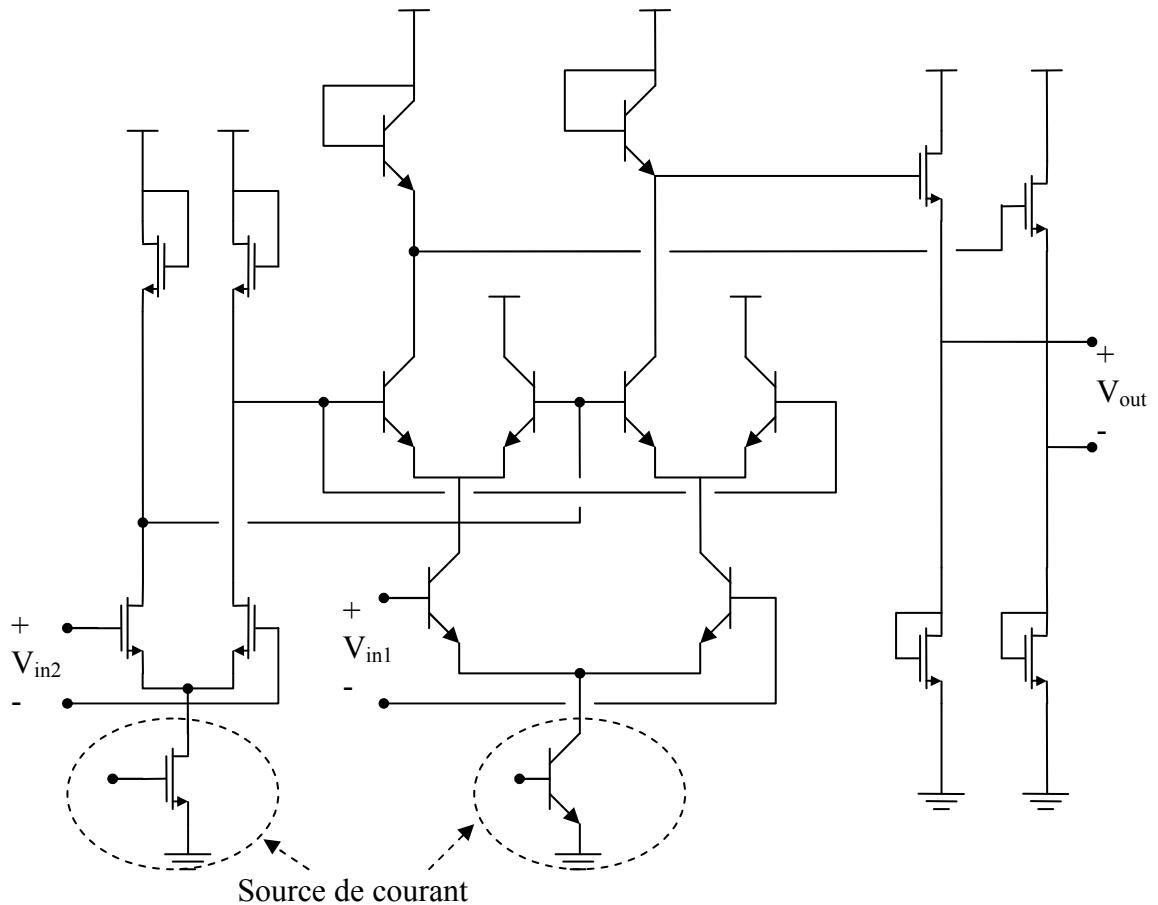


Figure 3-17- Circuit d'un nœud de variable (additionneur) en mode tension

Sachant qu'un nœud de variable à trois entrée-sortie, présenté sur la Figure 3-18, schématisé trois circuits chacun avec deux entrées et une sortie. C'est-à-dire que chaque nœud demande trois circuits comme celui présenté sur la Figure 3-17.

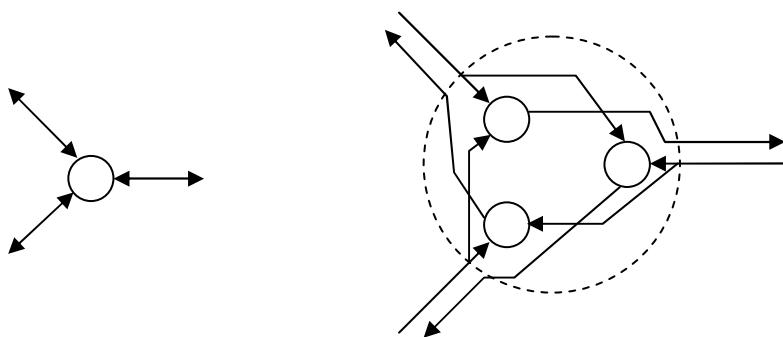


Figure 3-18- Remplacement d'un nœud de variable par trois circuits

Au cas où le nombre d'entrée-sortie est supérieur à 3, il est démontré que l'on peut dupliquer les nœuds, comme présenté sur la Figure 3-19 pour un nœud de variable à 4 entrée-sortie.

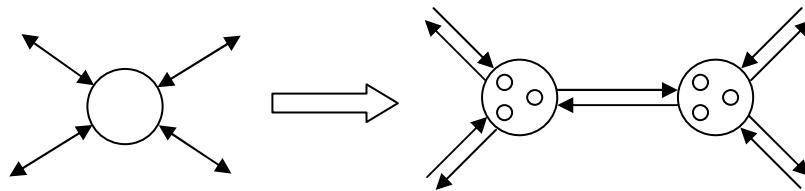


Figure 3-19- Équivalent d'un nœud de variable à 4 branches

3.4.3.2 Nœud de fonction

L'opération effectuée dans un nœud de fonction est l'opération Box-Plus. Le circuit correspondant en mode tension est le circuit de Gilbert donné sur la Figure 3-9 auquel on ajoute deux "level shifters", en entrée et en sortie pour harmoniser toutes les tensions[4], comme présenté sur la Figure 3-17.

En modifiant ce circuit, on peut obtenir l'opération box-plus en mode courant qui opère sur les probabilités. Ce circuit proposé par Loeliger est donné sur la Figure 3-20 [11]. En effet, aux entrées du circuit, une conversion probabilité-LLR est utilisée. Pour obtenir la fonction voulue, les transistors doivent travailler dans leur zone exponentielle : soit bipolaire soit MOS dans la zone sous seuil.

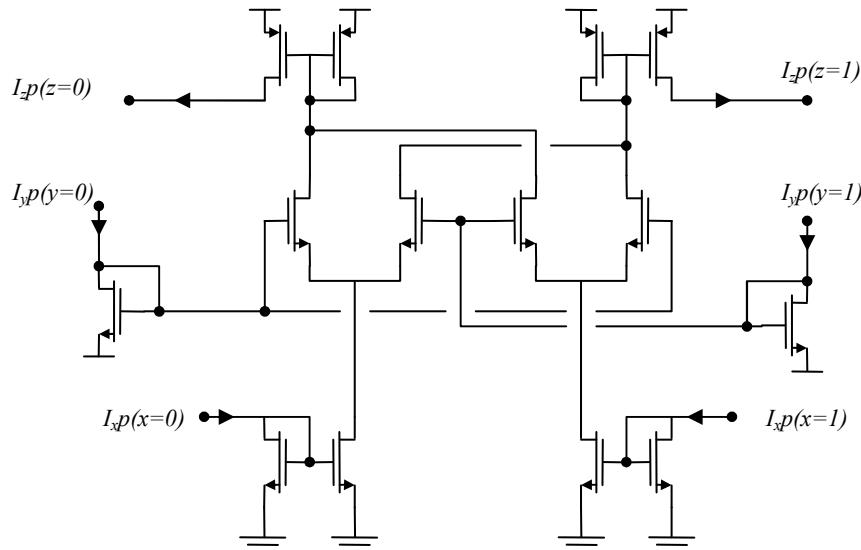


Figure 3-20- Opération box-plus en mode courant

Dans ce circuit les courants normalisés représentent les probabilités tout en sachant qu'une paire différentielle peut transformer les probabilités en LLR et vice versa (voir les paragraphes 3.3.1 et 3.3.2). L'opération directement obtenue utilisant ce circuit est schématisée sur la Figure 3-21.

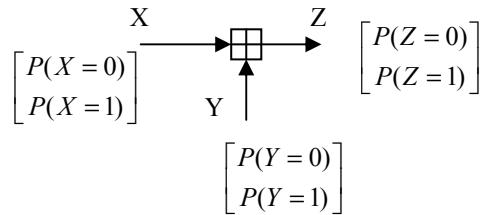


Figure 3-21- Opération box-plus opérant directement sur les probabilités

L'opération nécessaire pour calculer les probabilités sur la variable aléatoire Z est la suivante :

$$\begin{bmatrix} P(Z=0) \\ P(Z=1) \end{bmatrix} = \begin{bmatrix} P(X=0)P(Y=0) + P(X=1)P(Y=1) \\ P(X=0)P(Y=1) + P(X=1)P(Y=0) \end{bmatrix}$$

Avec ce circuit les courants de sortie représentés sur la figure par $I_z P(Z=0)$ et $I_z P(Z=1)$ sont égaux à :

$$\begin{aligned} I_z P(Z=0) &= I_x [P(X=0)P(Y=0) + P(X=1)P(Y=1)] \\ I_z P(Z=1) &= I_x [P(X=0)P(Y=1) + P(X=1)P(Y=0)] \end{aligned}$$

On voit donc que le bloc box-plus peut être fabriqué à partir des transistors en mode tension ou courant. On remarque aussi que le mode courant ne nécessite pas d'étage "level shifter", ce qui était nécessaire pour une implantation en mode tension.

3.5 Décodage analogique pour les codes en treillis

3.5.1 Introduction

Dans le cadre des codes en treillis, les méthodes MAP et ML sont très largement utilisées. On peut trouver des circuits analogiques dans la littérature pour réaliser les papillons d'un treillis dans un décodeur MAP [6][12]. Une implantation analogique est mieux adaptée à des codes convolutifs circulaires (*tailbiting*). C'est un code en treillis où pour chaque séquence valide du code, l'état initial du treillis est identique à l'état final, comme présenté sur la Figure 3-22 avec un exemple d'une séquence valide en trait plein foncé.

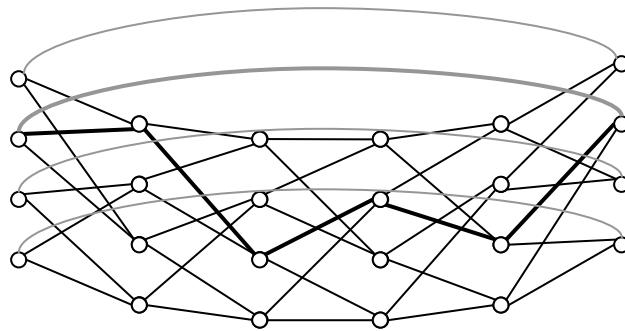


Figure 3-22- Treillis correspondant à "tail biting"

Un code en treillis peut être présenté sous forme de graphe. En réalité, on ne fait que schématiser les relations de l'algorithme MAP données au paragraphe 3.2.3. Le graphe ainsi obtenu est donné sur la Figure 3-23.

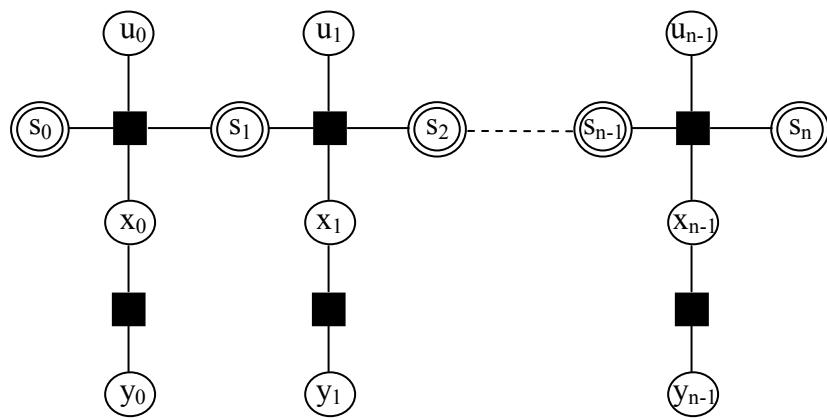


Figure 3-23- Présentation du treillis utilisant les graphes

Si on faisait un agrandissement sur chaque étage, on obtiendrait la Figure 3-24.

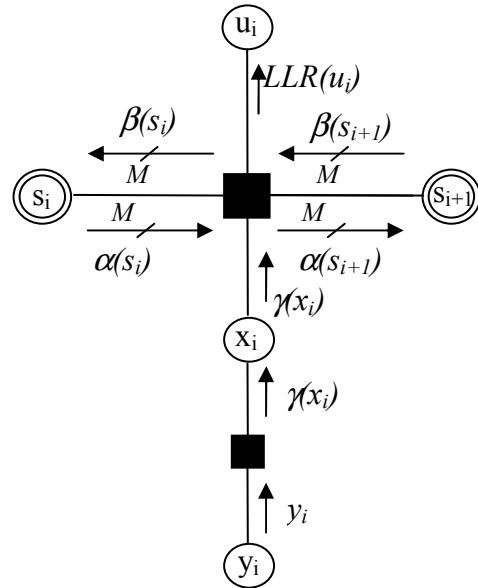


Figure 3-24- Détail des messages échangés entre les nœuds

Les y_i sont les valeurs observées en sortie du canal à partir desquelles on construit les x_i qui représentent les LLR sur les bits codés envoyés. Les messages échangés entre les nœuds sont présentés sur la figure ci-dessus. On remarque que le parcours du graphe ne fait que calculer les alpha et les beta exactement de la même manière que proposaient les formules du paragraphe 3.2.3.1. C'est-à-dire que l'utilisation du graphe ne simplifie point le calcul de l'algorithme MAP mais suggère une implantation circuit tout parallèle. Pour le cas de "tail biting" l'état s_0 et s_n sont les mêmes.

Pour un décodeur MAP, comme il est présenté au paragraphe 3.2.3.1, il nous faut le calcul des alpha et des beta. Ici, puisque le treillis ne commence pas forcément à l'état zéro et ne se termine pas non plus à cet état, il nous faut initialiser les alpha pour le tout premier étage et les beta pour le dernier étage. Il est montré [13] que le vecteur des alpha est le vecteur propre correspondant à la valeur propre la plus importante d'une matrice. Cette matrice est le produit des matrices gamma de tous les étages du treillis complet. Cependant, il se trouve que si on n'initialise pas les alpha et les beta de premier et de dernier étage, et que l'on effectue l'itération correspondant à leur calcul plusieurs fois autour du treillis, ces valeurs vont converger vers les bonnes. Dans une implantation analogique, il suffit de connecter la sortie finale à l'entrée et attendre que le circuit se stabilise. Ainsi, aucun circuit complémentaire n'est nécessaire pour tourner autour du treillis. Le temps de décodage d'un paquet est donc le temps de stabilisation du circuit.

3.5.2 Algorithme de décodage

Pour décoder un code convolutif circulaire (*tailbiting*) [13], on commence par initialiser les $\alpha_0(i)$ et les $\beta_L(i)$ où L représente la profondeur de treillis. Puisque l'état de départ peut être n'importe quel état, les valeurs de départ pour α et β sont toutes initialisées à $1/M$, M étant le nombre d'états. Ensuite, on forme la matrice Γ_t pour tous les étages dont les éléments sont définis comme suit :

$$\Gamma_t(i, j) = \Pr\{s_t = j, y_t | s_{t-1} = i\} \quad t = 1, \dots, L$$

y_t dans cette équation désigne les 2 bits codés reçus disponibles dans chaque étage (le code est supposé au rendement 1/2). Pour un canal sans mémoire cette probabilité se simplifie en :

$$\Gamma_t(i, j) = \Pr\{s_t = j, y_t | s_{t-1} = i\} = \Pr\{y_t | i \rightarrow j\} \Pr\{i \rightarrow j\}$$

La transition de l'état i vers l'état j spécifie les deux bits supposés être envoyés, ce qui nous laisse facilement calculer la probabilité $\Pr\{y_t | i \rightarrow j\}$. La probabilité $\Pr\{i \rightarrow j\}$ s'agit de l'information à priori sur les bits envoyés. Si elle n'est pas disponible, cette probabilité est identique pour toutes les branches disponibles dans un étage de treillis.

Connaissant la matrice Γ_t et le vecteur $\alpha_0 = [\alpha_0, \alpha_1, \dots, \alpha_{M-1}]$, on calcule α_t avec la relation :

$$\alpha_t = \bar{\alpha}_{t-1} \Gamma_t$$

On normalise ce vecteur afin d'obtenir le vecteur $\bar{\alpha}_t$ dont la somme des éléments est égale à 1.

$$\bar{\alpha}_t = \frac{\alpha_t}{\sum_i \alpha_t(i)} \quad t = 1, 2, \dots, L$$

Cette procédure itère tout en sachant que le t est modulo L . Ceci est dû à l'effet "*tailbiting*". En principe, au bout de quelques tours, les valeurs de α se stabilisent.

On utilise le même principe pour le calcul des β . Et à chaque itération, ces valeurs sont normalisées pour obtenir un vecteur dont la somme des éléments est égale à 1 :

$$\begin{aligned} \beta_t &= \Gamma_{t+1} \bar{\beta}_{t+1} \\ \bar{\beta}_t &= \frac{\beta_t}{\sum_j \beta_t(j)} \quad t = 1, 2, \dots, L \end{aligned}$$

Connaissant les α , β et Γ , on peut calculer la probabilité à posteriori sur les bits d'information (voir le paragraphe 3.2.3.1).

3.5.3 Circuit analogique

Deux types de circuit ont été présentés : mode tension [6] et mode courant [11]. Le circuit en mode courant n'ont pas besoin d'utiliser des *level shifter* et sont donc moins complexes, on le présente ici.

Le circuit est en fait une extension de multiplicateur de Gilbert. Le but est de calculer les α et les β dans l'algorithme MAP appliquée à un code en treillis. La Figure 3-25 schématisé ce principe.

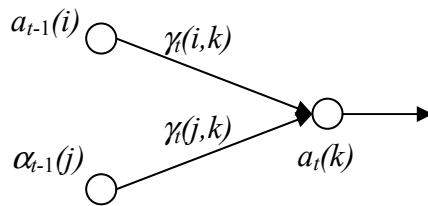


Figure 3-25- Avancement des calculs dans le treillis

Calcul des gamma

Il nous faut calculer les γ à partir des échantillons reçus. Supposant un code de rendement 1/2, pour chaque étage du treillis, on reçoit deux signaux y_t^1, y_t^2 . Utilisant la relation (134) on fabrique les LLR et puis en se servant du circuit présenté au paragraphe 3.3.1, on fabrique les probabilités correspondant à chaque bit codé reçu : $P(y_t^1|x_t^1=0)$, $P(y_t^1|x_t^1=1)$, $P(y_t^2|x_t^2=0)$, $P(y_t^2|x_t^2=1)$. Ensuite on calcule $P(y_t^1, y_t^2 | x_t^1, x_t^2)$. Puisque le canal est sans mémoire, on peut séparer les distributions de probabilité et écrire :

$$P(y_t^1, y_t^2 | x_t^1 = i, x_t^2 = j) = P(y_t^1 | x_t^1 = i) P(y_t^2 | x_t^2 = j) \text{ et } i, j \in \{-1, +1\}$$

Ceci s'implante facilement avec un multiplicateur Gilbert présenté sur la Figure 3-20. La seule différence est qu'ici, les quatre courants sont utilisés et donc, quatre miroirs de courant sont nécessaires pour appliquer ces courants au bloc suivant. On peut schématiser ce circuit par un bloc appelé "A" comme présenté la Figure 3-26

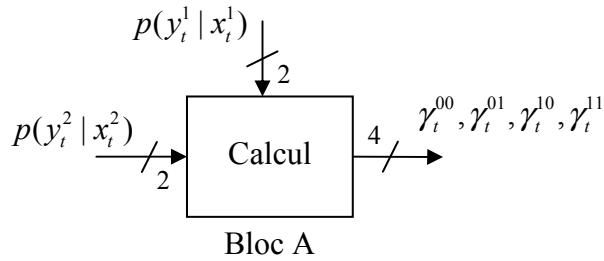


Figure 3-26- Bloc de calcul des gamma

Calcul des alpha

Le calcul des α se fait à partir de l'équation ci-dessous :

$$\alpha_t(k) = \alpha_{t-1}\gamma_t(i, k) + \alpha_{t-1}\gamma_t(j, k)$$

Par exemple pour un code convolutif (7,5), l'étage générique du treillis est celui de la Figure 3-27.

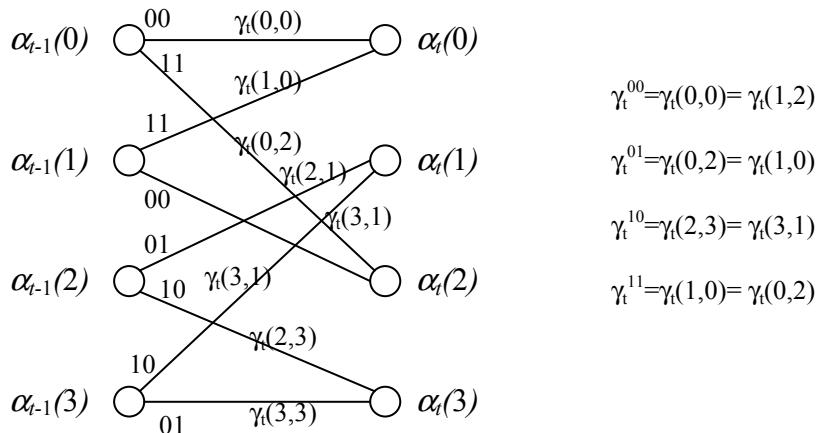


Figure 3-27- Calcul des alpha pour le code (7,5)

Le circuit de calcul des α contient donc 4 sorties correspondant aux α_t de chaque état. Il y a aussi 4 entrées pour les alpha de l'étage précédent et 4 entrées pour fournir les gamma courants. On doit donc fabriquer des multiplieurs pour multiplier toutes les combinaisons d'entrées et ensuite, il faut les additionner deux par deux pour obtenir les sorties.

Le circuit correspondant est une extension du circuit de Gilbert où les sorties des multiplieurs sont additionnées pour former les sorties finales. La Figure 3-28 présente cette réalisation en mode courant.

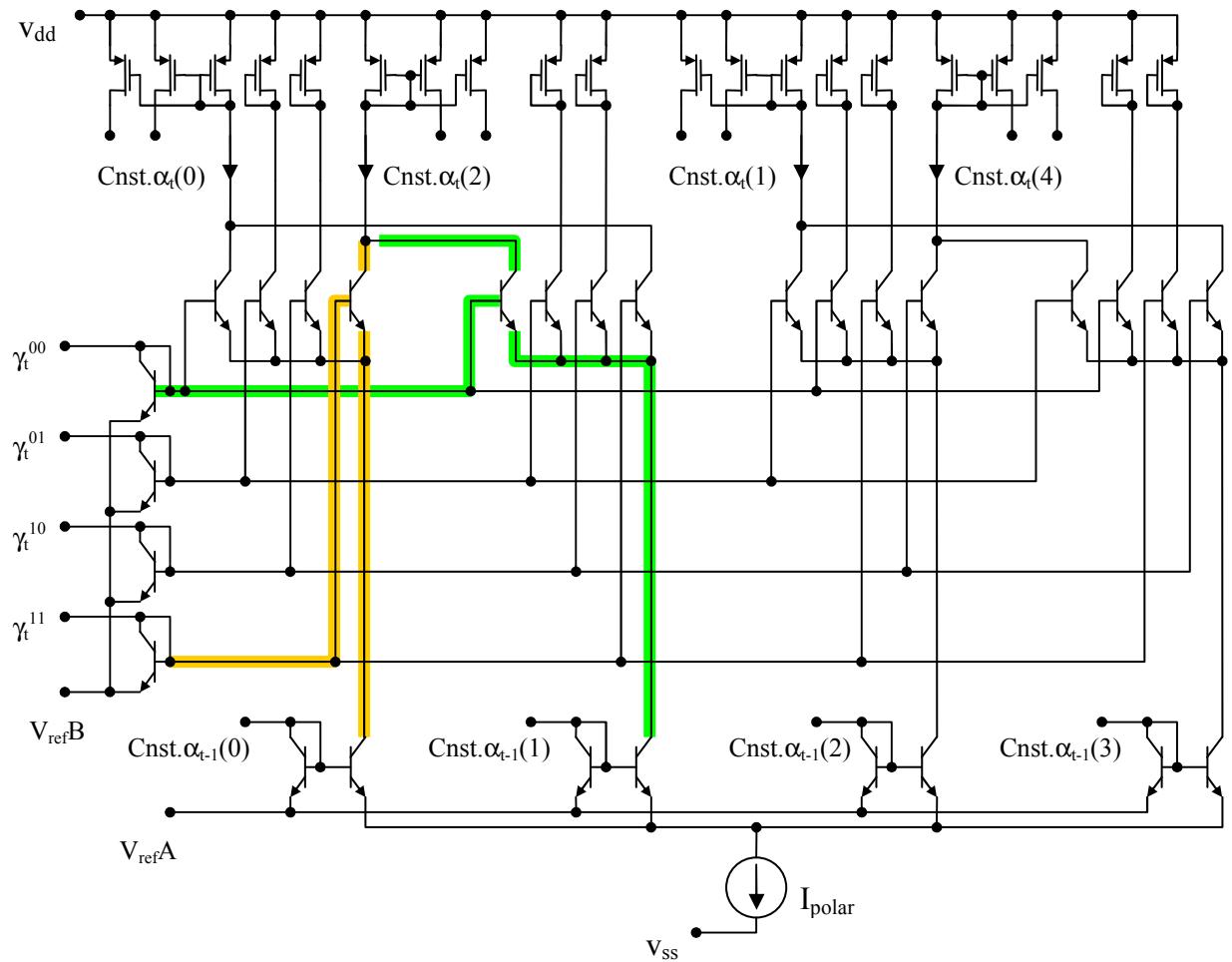


Figure 3-28- Circuit correspondant à chaque étage du treillis

Avec ce circuit, un étage de treillis est parcouru et en le répétant autant de fois qu'il y a d'étages dans le treillis, on calcule les α_t . Le fonctionnement de ce circuit est détaillé dans l'Annexe I. On peut schématiser ce circuit comme présenté sur la Figure 3-29.

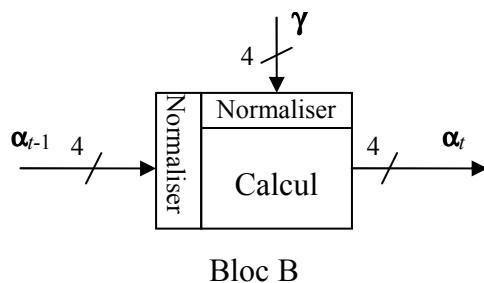


Figure 3-29- Bloc de calcul des alpha

Calcul des beta

Les β sont calculés à partir de l'équation ci-dessous :

$$\beta_t(k) = \beta_{t+1}\gamma_{t+1}(k, i) + \beta_{t+1}\gamma_{t+1}(k, j) \quad (158)$$

Pour partager les multiplicateurs de cet étage et celui qui calculera les probabilités à posteriori sur les bits d'information, le circuit de calcul des β ne fera pas l'addition des deux termes de l'équation (158) et ce rôle est confié à l'étage suivant. C'est-à-dire qu'à l'entrée de chaque circuit, nous avons 8 courants que l'on va additionner deux par deux pour obtenir le vecteur de β . Le circuit ressemblera donc à celui de la Figure 3-28 sauf que 8 courants sont dupliqués pour être appliqués aux étages suivants. On schématise ce circuit par le bloc "C" présenté sur la Figure 3-30.

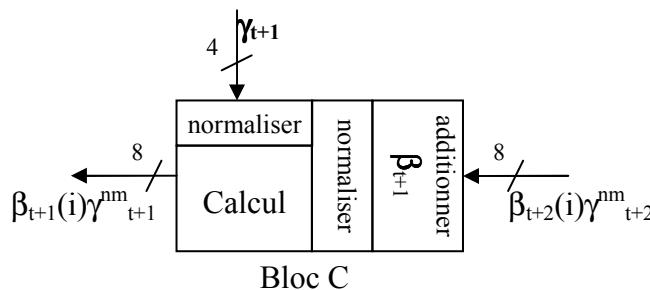


Figure 3-30- Bloc de calcul des beta

Calcul des probabilités à posteriori

Finalement les α et les β sont disponibles, il ne reste plus qu'à calculer les probabilités à posteriori sur les bits d'information. Reprenons l'algorithme BCJR pour ce calcul présenté par l'équation (152). On retrouve les mêmes principes que pour les calculs des α et des β . Le circuit est, une fois encore, celui présenté sur la Figure 3-28. Ici, nous avons 4 entrées pour les α_{t-1} et 8 entrées pour les $\beta_{t+1}(i)\gamma^{nm}_{t+1}$. Il n'y a que deux sorties qui sont en fait $\Pr(u_t = 0 | Y)$ et $\Pr(u_t = 1 | Y)$ où :

$$\Pr(u_t = 0 | Y) = \sum_{\substack{\text{sur les branches} \\ \text{résultant } u_t = 0}} \alpha_{t-1}(i) [\beta_t(j)\gamma_t^{nm}]$$

$$\Pr(u_t = 1 | Y) = \sum_{\substack{\text{sur les branches} \\ \text{résultant } u_t = 1}} \alpha_{t-1}(i) [\beta_t(j)\gamma_t^{nm}]$$

Ce circuit est schématisé par un bloc D présenté sur la Figure 3-31.

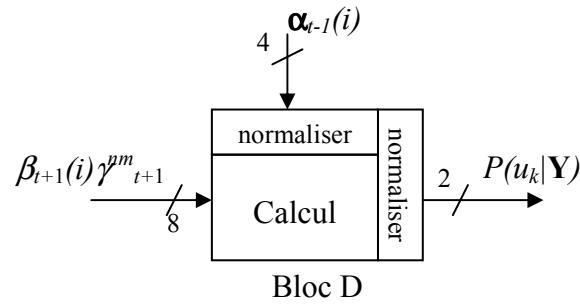


Figure 3-31- Bloc de calcul de la probabilité à posteriori sur les bits d'information

Circuit final

Il faut maintenant enchaîner les blocs pour obtenir le circuit final. La Figure 3-32 présente le résultat.

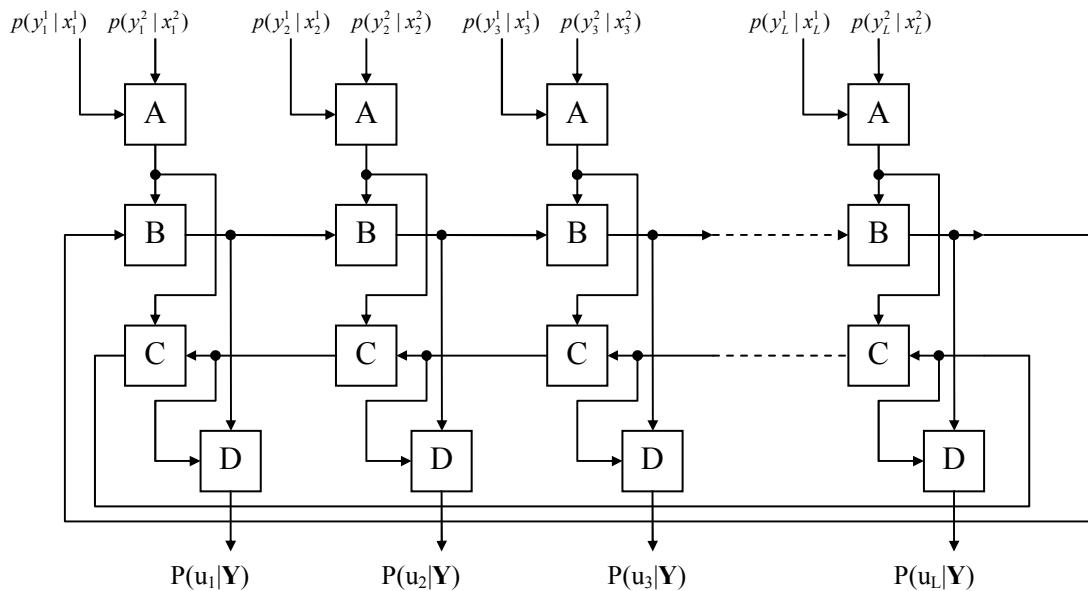


Figure 3-32- Circuit complet avec les connexions entre les blocs

3.5.4 Dimensionnement du circuit

Nous allons donner une estimation de nombre de transistors nécessaires pour fabriquer un décodeur MAP pour un code convolutif circulaire (*tailbiting*) de rendement 1/2 avec M états et une taille de treillis égale à L .

Bloc A

- 8 transistors pour effectuer les multiplications
- 8 transistors pour les miroirs de courant

Total : 16 transistors.

Bloc B

- 4 transistors pour les γ
- M transistors pour les α
- 4M transistors pour les multiplications
- 2M transistors pour les courants non utilisés
- 3M transistors pour les miroirs de courant de M courants de sortie

Total : 10M+4 transistors

Bloc C

C'est comme le bloc B sauf que les 8 courants de sortie sont dupliqués utilisant des miroirs de courant.

- 4 transistors pour les γ
- M transistors pour les β
- 4M transistors pour les multiplications
- 2M transistors pour les courants non utilisés
- 2*2M transistors pour les miroirs de courant de 2M courants de sortie

Total : 11M+4

Bloc D

Ce bloc multiplie un vecteur de taille 2M par un vecteur de taille 4.

- M transistors pour recevoir les α
- 2M transistors pour recevoir les $\beta_{t+1}(i)\gamma^{nm}_{t+1}$
- $2M^2$ transistors pour les multiplications
- 4 transistors pour les miroirs de courant
- $2M^2-2M$ transistors pour les courants non utilisés

Total : $4M^2+M+4$

Total :

3.6- Décodage en bloc des codes convolutifs

Nombre de transistors BJT : $(2M^2 + 13M + 16)L$

Nombre de transistors MOS : $(2M^2 + 9M + 12)L$

Exemple

Le code (5,7) 100L BJT, et 80L MOS.

Le code (15,17) 248L BJT et 121L MOS

Le code (23,35) 736L BJT et 668L MOS

3.6 Décodage en bloc des codes convolutifs¹

Dans la littérature, on voit les cas où on cherche à établir un treillis pour des codes en bloc [15][16]. Une partie de ces travaux consiste à trouver un treillis présentant le code en bloc de manière à minimiser le nombre d'états [14]. Dans la plupart des cas où on arrive à minimiser ces états, le nombre d'états dans chaque étage de treillis varie. Ceci nous empêche d'obtenir un circuit complètement modulaire. Un exemple très connu est celui du code en bloc (8,4) qui est une extension du code Hamming (7,4) (voir l'exemple du paragraphe 3.4). Avec un effort considérable, on trouve pour ce code le treillis donné sur la Figure 3-33 où le "tail biting" est obtenu [17].

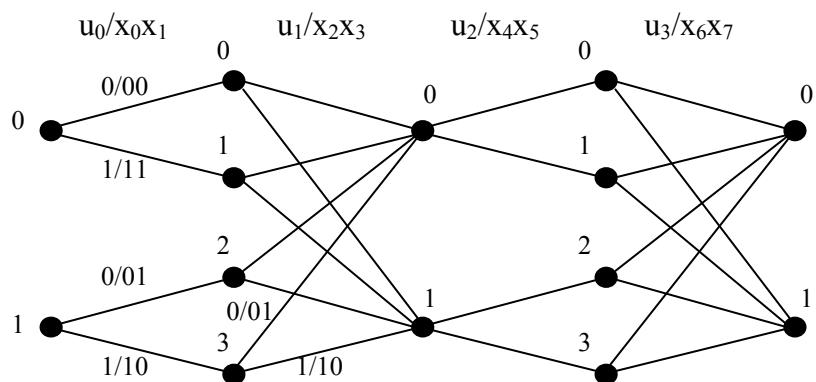


Figure 3-33- treillis minimal à "tail biting"obtenu pour un code en bloc (8,4). Les chemins émergeant et convergeant vers le même état sont valides.

¹ Cette idée, qui va bientôt apparaître dans le journal Elecrtonc Letters, est détaillé dans l'0.

Nous proposons d'utiliser des codes convolutifs à "tail biting" comme un code en bloc. Ainsi, on peut naturellement construire le treillis minimal respectant le "tail biting". Par contre, au lieu de construire le graphe comme on le faisait pour un code en treillis, nous développons le graphe en écrivant les équations de parité pour ce code. Ainsi, on peut obtenir des codes en bloc du rendement 1/2, 1/3 etc pour des tailles du mot du code arbitraires. Nous allons décrire cette procédure pour le code convolutif non systématique (7,5). Le schéma du codeur est le suivant :

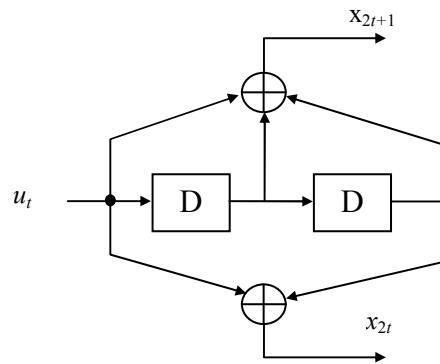


Figure 3-34- Codeur convolutif non systématique (7,5)

Nous prenons par exemple des paquets de taille 8 bits d'information générant des mots du code de taille 16. On obtiendra donc un code en bloc (16,8). Soit $\mathbf{u} = [u_1, u_2, \dots, u_8]$ la séquence d'information, pour obtenir le "tail biting", les valeurs initiales des registres sont $[u_8, u_7]$. Ainsi on peut écrire 16 équations pour les 16 bits codés à partie des 8 bits d'information.

$$\begin{aligned}
 x_1 &= u_1 \oplus u_8 \oplus u_7 \\
 x_2 &= u_1 \oplus u_7 \\
 x_3 &= u_2 \oplus u_1 \oplus u_8 \\
 x_4 &= u_2 \oplus u_8 \\
 &\dots
 \end{aligned} \tag{159}$$

On peut réécrire ces équations sous la forme :

$$\begin{aligned}
 x_1 &= x_2 \oplus u_8 \\
 x_2 &= u_1 \oplus u_7 \\
 x_3 &= x_4 \oplus u_1 \\
 x_4 &= u_2 \oplus u_8 \\
 &\dots
 \end{aligned} \tag{160}$$

On peut écrire la matrice génératrice \mathbf{G} pour ce code utilisant les équations de parité données par la relation (159) :

$$G = \begin{bmatrix} 1 & 1 & 1 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 1 \\ 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 0 \\ 1 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 \end{bmatrix} \quad (161)$$

Nous pouvons construire la matrice de parité H en écrivant les équations de parité en fonction des bits reçus mais aussi en fonction des bits d'information. Ces derniers ressemblent à des états dans une construction du graphe sous forme de treillis constitué des variables cachées dans une chaîne de Markoff. Ainsi, on définit le vecteur des variables $v = [u \ x]$. On définit la matrice H vérifiant l'équation $v \cdot H^T = 0$. Sachant que $x = uG$, on peut écrire :

$$vH^T = [u \ x]H^T = [u \ uG] \begin{bmatrix} H_1^T \\ H_2^T \end{bmatrix} = u \left[H_1^T \oplus GH_2^T \right] = 0_{16 \times 16} \quad (162)$$

La matrice H suivant les équations (160) et vérifiant l'équation (162) est la suivante :

$$H = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}_{16 \times 24} \quad (163)$$

A partir de cette matrice H , on pourra dessiner le graphe correspondant comme présenté la Figure 3-35.

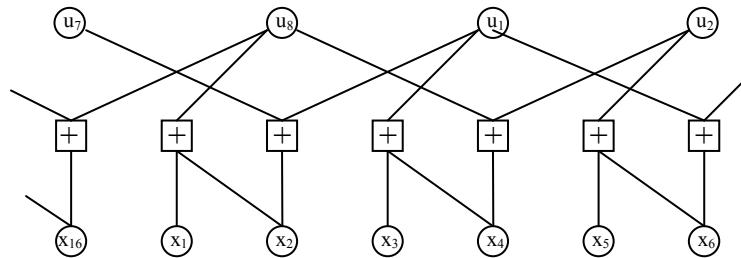


Figure 3-35- Graphe obtenu à partir des équations de parité

Le graphe complet est présenté sur la Figure 3-36.

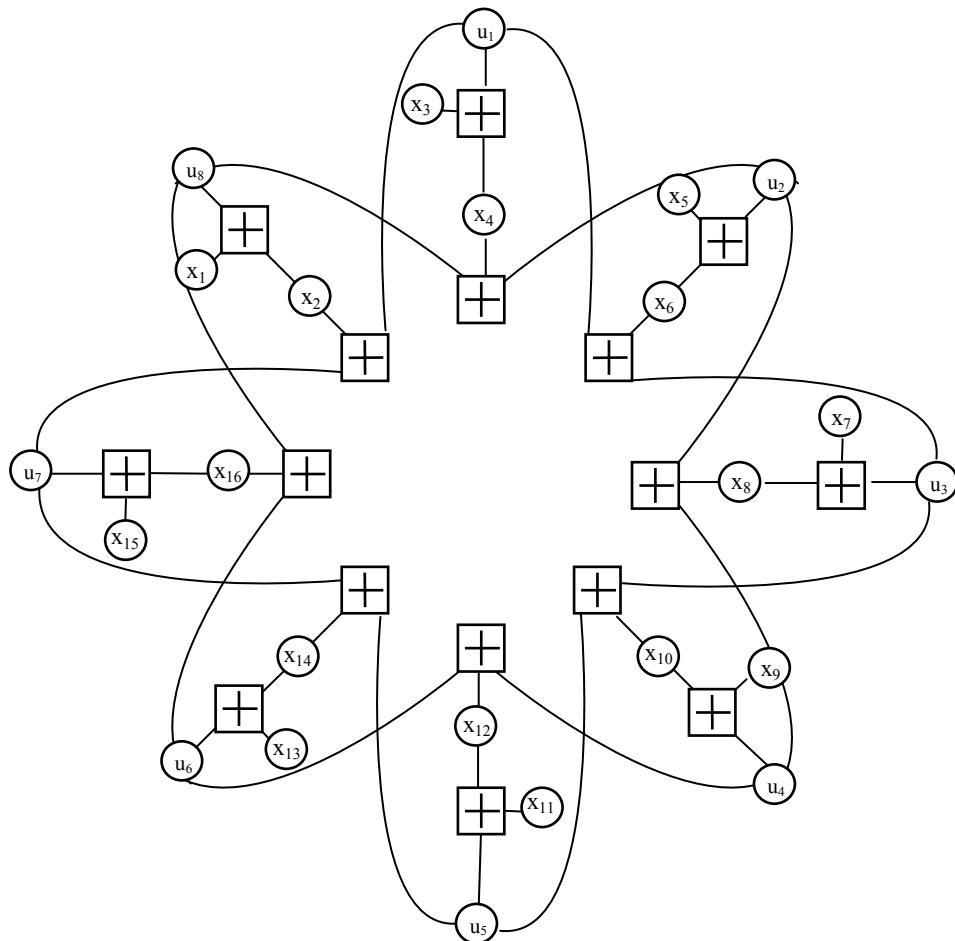


Figure 3-36- Graphe complet où on remarque la nature circulaire du code

En réception, ne disposant pas d'information sur les bits du message (u), on les initialise à zéro. Les x_i sont initialisés par la sortie du canal. Les nœuds u_i , une fois le graphe stabilisé, donnent les probabilités à posteriori sur les bits d'information.

3.7- Aspect de la simulation

Avec cette approche, nous proposons donc une méthode simple et systématique pour un décodage sur graphe des codes convolutifs à "tail biting". Cette architecture se réalise utilisant des circuits analogiques et présente une complexité inférieure à celle proposée dans la littérature qui proposait un graphe directement à partir des équations de MAP.

Une simulation a été effectuée pour éprouver cette architecture dont le résultat est montré sur la Figure 3-37. Cette simulation montre que la dégradation de performance par rapport à une détection MAP optimale est inférieure à 0,4 dB pour les $TEB < 10^{-2}$.

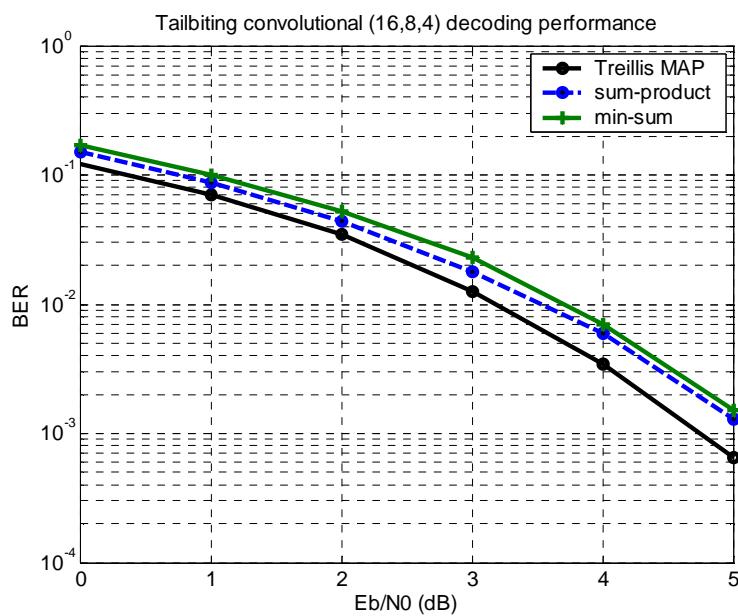


Figure 3-37- Taux d'erreur obtenu utilisant le graphe de la Figure 3-36

3.7 Aspect de la simulation

Un des problèmes majeurs de l'implantation analogique par rapport à son équivalent numérique est le problème de la simulation. Ici, vu le nombre important de transistors utilisés dans un décodeur, une simulation niveau circuit est presque impossible pour déterminer le taux d'erreur d'un décodeur. De plus, vu l'interaction permanente entre les différentes parties du circuit, nous ne sommes pas certain que le circuit arrive à trouver le point optimal. En effet on n'implante pas directement l'algorithme développé mathématiquement et les itérations se font avec la vitesse naturelle du circuit. Il est important de pouvoir simuler un tel circuit en modélisant le comportement de ce circuit et d'essayer de simuler le système complet avec un simulateur système.

Une des premières tentatives est de prendre chaque bloc individuel et de le modéliser comme une fonction à laquelle on ajoute un certain retard. Ce retard est à optimiser bloc par bloc pour approximer le comportement du bloc. Andrews et Divsalar par exemple [18] proposent de remplacer les nœuds de fonction par un circuit d'ordre 1. Puisque chaque nœud implante une fonction avec un comportement temporel qui peut être mesuré ou simulé avec des simulateurs circuits, le modèle simple de la Figure 3-38 peut être envisagé.

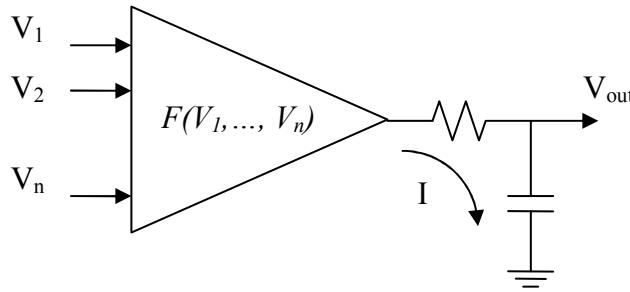


Figure 3-38- Équivalent d'un nœud de fonction pour analyse

On peut donc écrire

$$\frac{dV_{out}}{dt} = \frac{I}{C} = \frac{1}{RC} f(V_1, \dots, V_n) - \frac{1}{RC} V_{out}(t)$$

Pour une simulation numérique on approxime la dérivée pour arriver à

$$\begin{aligned} \frac{dV_{out}}{dt} &\approx \frac{V_{out}(t + \Delta) - V_{out}(t)}{\Delta} \\ V_{out}(t + \Delta) &= (1 - \varepsilon)V_{out} + \varepsilon f(V_1, \dots, V_n) \end{aligned}$$

où $\varepsilon = \Delta / RC = \Delta \tau$. Utilisant cette équation, on peut estimer par la simulation la performance d'un décodeur LDPC ou autre.

Hemati et Banihashemi [19] proposent une méthode mathématique pour trouver la réponse transitoire et la réponse finale d'un circuit analogique. Cette méthode est basée sur la résolution d'équations d'états d'un décodeur utilisant l'approximation d'Euler. En effet, dans un graphe de Tanner, les nœuds de variable et de fonction peuvent être présentés avec les relations ci-dessous :

$$\begin{cases} \text{Variable : } f(Y, R) = X \\ \text{Fonction : } g(X) = Y \end{cases}$$

où R est le vecteur de sorties du canal, X est le vecteur de messages passés aux nœuds de fonction, et Y est le vecteur de sorties des nœuds de fonction. On peut donc écrire :

$$f(g(X), R) = h(X, R) = X \quad (164)$$

Le but recherché est de trouver le vecteur X dans le régime permanent. Ceci est la solution stabilisée de notre graphe qui estime le mot du code envoyé. Dans le même temps, cette équation va donner aussi le régime transitoire qui donne des informations sur la vitesse de convergence de notre décodeur. Les méthodes numériques résolvent cette équation par itération :

$$X_{k+1} = h(X_k, R)$$

L'implantation de cette relation est un circuit synchrone qui avance d'un pas à chaque coup d'horloge. Ici, nous disposons d'un circuit asynchrone (pas d'horloge) et on attend que le circuit se stabilise. La dynamique du système, comme avant, est approximée par un système d'ordre un. La Figure 3-39 présente le modèle proposé par Hemati et al. [19].

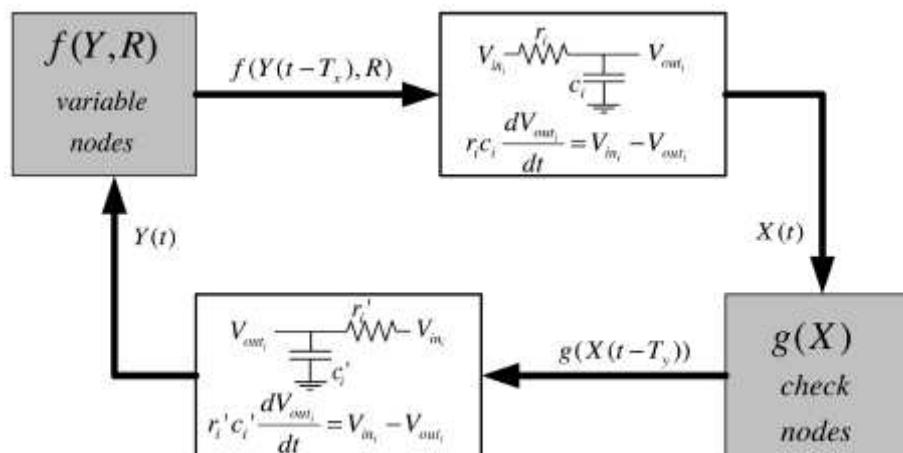


Figure 3-39- Modèle d'un décodeur analogique d'un code en graphe

Sur ce schéma les vecteurs T_x et T_y modélisent les retards dus aux circuits de chaque nœud. Ces retards peuvent être différents suivant le type de nœud (variable ou fonction) et aussi suivant l'ordre du nœud. Pour simplifier, on a supposé que le délai de traitement dans chaque bloc est nettement inférieur au délais de propagation, on suppose donc $T_x=T_y=0$.

Nous nous focalisons sur l'équation (164) en tenant compte de constante de temps d'un tour de boucle pour arriver au modèle simplifié ci-dessous :

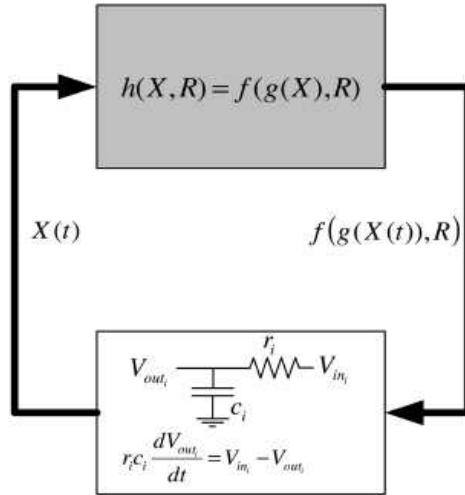


Figure 3-40- Modèle simplifié pour le décodage analogique

Avec cette modélisation l'équation gérant le circuit est la suivante :

$$\tau \dot{X}(t) = h(X(t), R) - X(t) \quad (165)$$

où τ est une matrice diagonale contenant les constantes de temps de différentes liaisons. Utilisant l'approximation d'Euler pour approximer la dérivée, la relation (165) peut être écrite :

$$\begin{aligned} \tau \frac{X(t + \Delta t) - X(t)}{\Delta t} &= h(X(t), R) - X(t) \\ X(t + \Delta t) &= X(t) + \Delta t \tau^{-1} (h(X(t), R) - X(t)) \end{aligned}$$

La simulation montre une bonne concordance entre les valeurs simulées par une simulation circuit (très gourmande en terme de temps de calcul) et les valeurs simulées utilisant cette procédure [19].

3.8 Turbo codes

3.8.1 Introduction

L'implantation numérique des méthodes itératives va soit augmenter la taille du circuit numérique soit diminuer la vitesse proportionnellement au nombre d'itérations. Ceci augmente le délai de décodage de manière linéaire avec la taille du paquet et le nombre d'itérations.

Or, dans l'implantation analogique des décodeurs itératifs, les itérations se font sur le même circuit et de manière asynchrone (continue) en utilisant des contre-réactions. La réponse est disponible dès qu'une certaine stabilité est observée en sorties.

Alors, l'utilisation des systèmes analogiques est bien justifiée dans le cas des turbocodes ou des codes LDPC basés sur graphe.

Jusqu'aujourd'hui peu de circuits turbo analogiques [20], [21] ont été réellement réalisés mais d'ores et déjà on peut constater que la vitesse et la consommation s'améliorent par rapport à des circuits numériques. Cependant, des problèmes techniques et de principe limitent l'utilisation massive de ces décodeurs. Ces problèmes constituent bien sûr un sujet de recherche passionnant qui attire de plus en plus de chercheurs dans le monde.

3.8.2 Analogue turbo décodeur

Dans ce paragraphe nous allons souligner les éléments nécessaires pour une réalisation analogique d'un décodeur turbo. Nous nous limitons aux cas des codes convolutifs récursifs (RSC) avec une concaténation parallèle (PCCC). Le schéma d'un codeur turbo typique est donné sur la Figure 3-41.

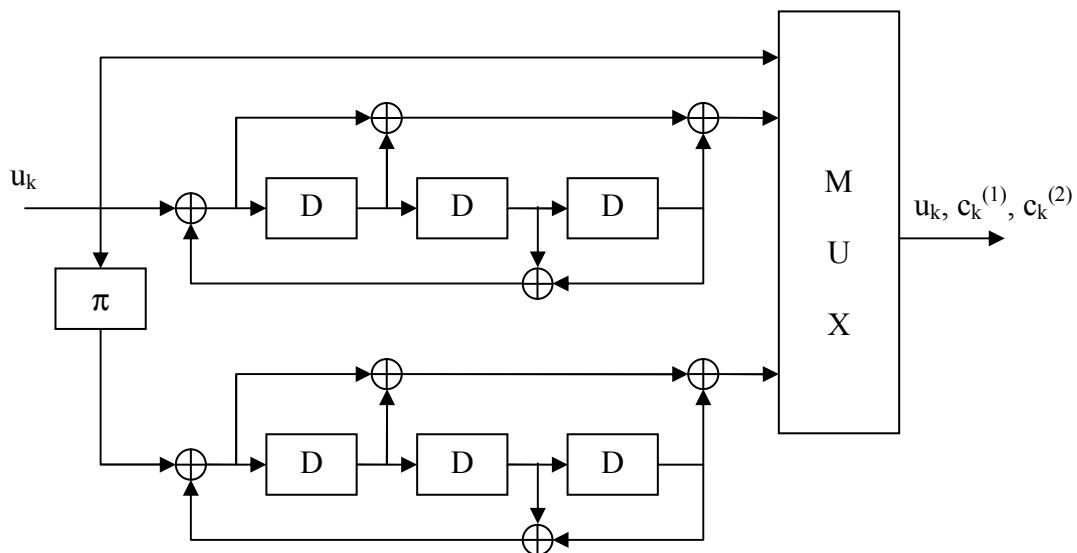


Figure 3-41- Encodeur PCCC avec RSC (15,13)

Ce système de rendement 1/3 est utilisé dans le réseau UMTS 3GPP avec un entrelacement de taille 40. Dans le cas général, on peut terminer le treillis dans un état donné (souvent l'état zéro); l'autre alternative est d'utiliser des codes convolutif tail biting. Ces bits codés passent à travers un canal additif gaussien et arrivent en récepteur. Le rôle du décodeur est de calculer

la probabilité à posteriori sur chaque bit d'information conditionnellement à la séquence entière reçue. Le décodage, basé sur les fameux travaux de C. Berrou [22], est schématisé sur la Figure 3-42.

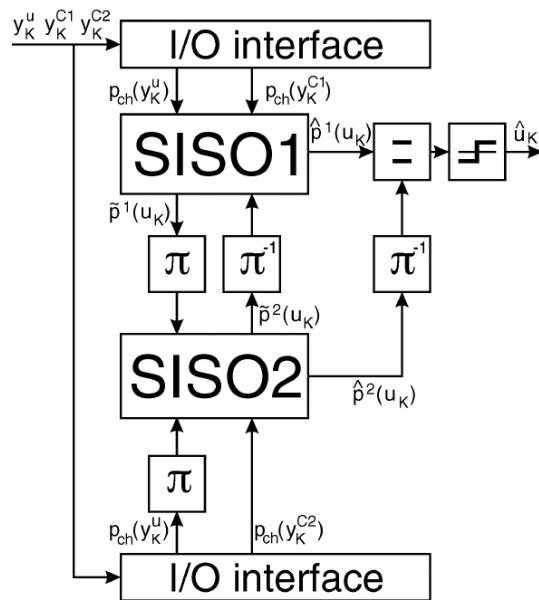


Figure 3-42- Diagramme en bloc d'un décodeur turbo

Contrairement à une implantation numérique, ce système s'itère sans pour autant multiplier le circuit. Les décodeurs SISO se réalisent comme nous l'avons présenté dans le paragraphe 3.5. Les informations extrinsèques, à inter changer entre les décodeurs, sont passées par un entre-laceur qui se réalise de manière figée avec le routage directe sur le VLSI. Une fois le système stabilisé, les probabilités à priori sont utilisées pour calculer les bits les plus probables envoyés par l'émetteur (élément de décision). Les tensions transitoires typiques obtenues avec une simulation circuit sont données sur la Figure 3-43 [21].

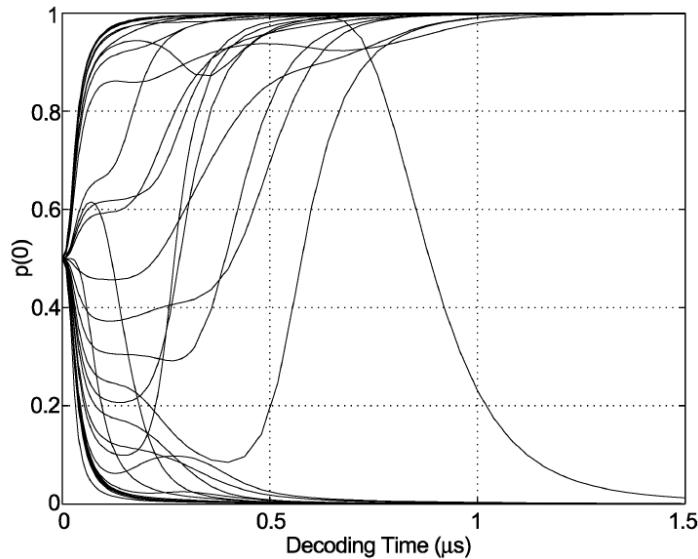


Figure 3-43- Les courants obtenus du décodeur turbo avec une simulation circuit [21]

Avec ce circuit utilisant quelques 70000 transistors travaillant dans la zone subthreshold, une puissance totale de 7,6 à 10,3 mw a été obtenue qui démontre la performance en terme de puissance consommée de décodeurs analogiques.

Dans la littérature, une implantation analogique (sans fabrication réelle) d'un turbo code produit a aussi été rapportée [23] où un code Hamming (7,4) a été utilisé comme les codes constituants. Ce code était considéré comme un petit code LDPC et le décodage est effectué utilisant l'algorithme de somme-produit sur graphe. Bien que ce code soit loin d'être utilisé dans les applications pratiques vu son gain de codage très limité, cette implantation démontre la validité de l'approche analogique.

Dans le but de modularité d'implantation des turbo codes, une méthode récente appelée "slice turbo decoding" consiste à diviser la séquence à coder en plusieurs séquences plus petites et coder chacune de ces séquence avec un code convolutif circulaire (tail biting). Les différents décodeurs MAP de taille réduite pouvant fonctionner en parallèle, une modularité et un parallélisme sont obtenus [25]. A partir de ce principe, M. Arzel et al [24] présentent le circuit analogique correspondant à cette architecture. Les simulations circuits montrent que le délai global obtenu par le décodeur turbo est comparable avec le délai de décodage MAP utilisé seul, ce qui est complètement impossible dans une implantation numérique.

3.9 Autres applications des méthodes analogiques

Le traitement de signal analogique commence à trouver son importance avec des applications de plus en plus nombreuses. Ici, nous listons de manière succincte quelques applications récentes qui découlent de l'approche analogique.

3.9.1 Filtre RIF à coefficient constant

La partie la plus contraignant dans une implantation de filtre RIF concerne les multiplieurs. Un miroir de courant peut être utilisé pour multiplier un courant électrique par une constante. Le circuit simplifié est présenté sur la Figure 3-44. La réalisation se fait tout simplement en jouant sur la largeur du canal dans un transistor MOS.

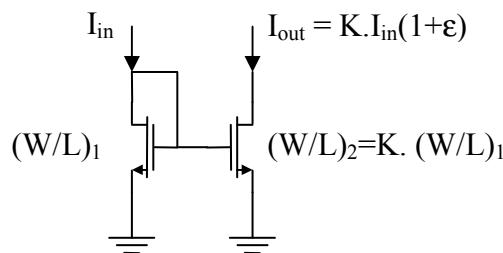


Figure 3-44-Multiplication d'un courant par une constante (ε représente l'effet de mismatch entre les transistors)

Cette multiplication qui s'intègre avec un coût négligeable peut être utilisé dans un nombre important des algorithmes à réaliser. Utilisant cette technique les filtres numériques à coefficient constant peuvent être réalisés très facilement avec un coût très inférieur à leurs équivalents numériques. L'implantation de la somme de convolution se fait avec le diagramme de la Figure 3-45.

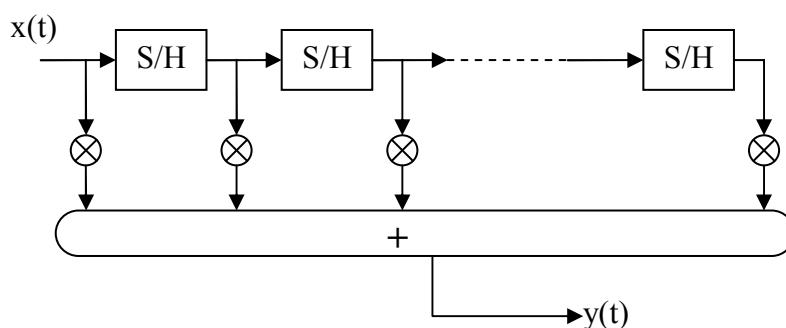


Figure 3-45- Filtre RIF

Les multiplicateurs ne sont que les miroirs de courant et l'addition se réalise en connectant les sorties. Ainsi, une structure parallèle avec une vitesse importante avec peu de transistors implante efficacement le filtre FIR. Le seul inconvénient est que le filtre reste figé dans l'ASIC et on ne peut plus le modifier.

3.9.2 Filtre RII à coefficient constant

Les filtres RII ne sont en général pas implantés avec des circuits numériques travaillant en virgule fixe. Ceci est principalement dû à la limitation de nombre de bits représentant les signaux. Le circuit analogique travaillant directement sur les tensions non quantifiées constitue une alternative intéressante pour la réalisation de ces filtres.

3.9.3 Transformée de Fourier Rapide (FFT)

La FFT est une opération très souvent utilisée dans les applications de traitement de signal. Cette opération, implantée par un circuit numérique ou par un DSP, consomme des ressources importantes. Si on étudie l'architecture d'implantation de cet algorithme présentée sur la Figure 3-46, on remarque qu'un nombre limité de multiplications est à effectuer. Ces multiplications peuvent se réaliser avec des miroirs de courant de la Figure 3-44.

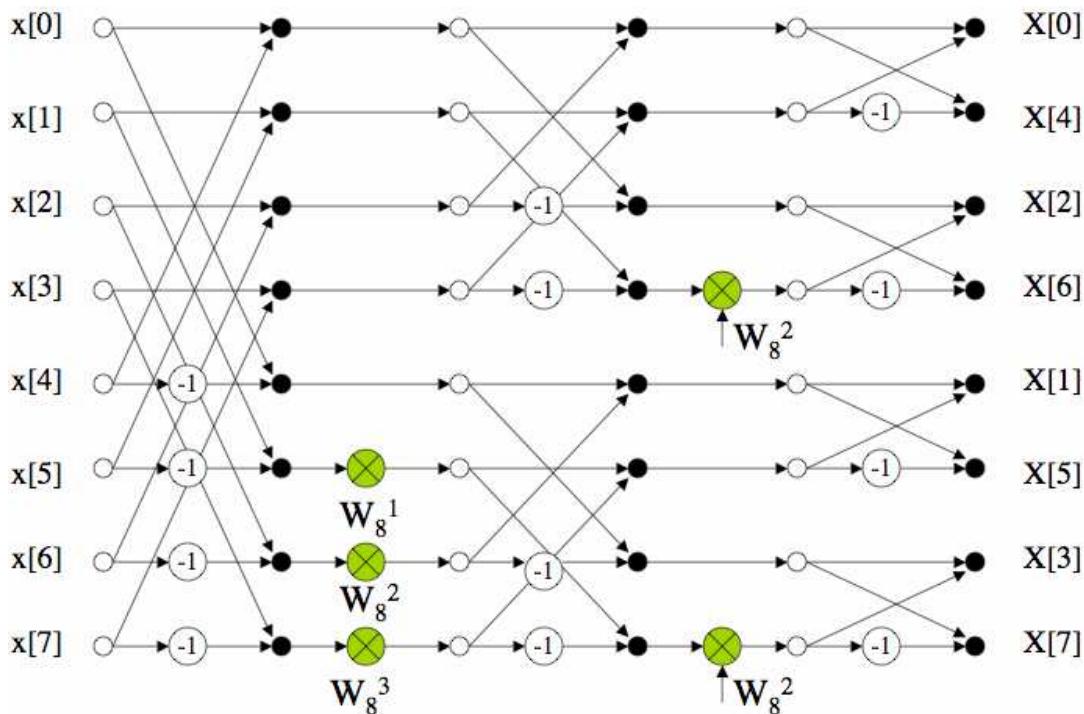


Figure 3-46- FFT de taille 8

Une approche a été récemment proposée pour effectuer des FFT de taille importante tout en réduisant le nombre total des facteurs (W/L) du circuit [26].

3.9.4 Récepteur MIMO analogique

Garrido et al. [27] rapportent une implantation toute analogique d'un détecteur MIMO en utilisant la technologie 0,25 μm SiGe BiCMOS. Ils proposent donc d'incorporer leur circuit dans un décodeur turbo analogique pour pouvoir faire itérer les deux afin de construire un système type Turbo-Blast tout analogique.

3.9.5 Filtrage FIR et sur échantillonnage

Dans ce paragraphe nous présentons une méthode originale pour implanter les filtres RIF adapté au circuit intégré. On commence par écrire la relation de convolution. Soit $x(t)$ le signal reçu à filtrer et $h(t)$ la réponse impulsionnelle du filtre. La sortie est :

$$y(t) = x(t) * h(t) = \int_{t-T}^t x(\tau)h(t-\tau)d\tau$$

Nous avons supposé que le filtre est limité dans le temps dans l'intervalle $(0, T)$. Imaginons que l'on divise la période $(0, T)$ en N intervalles de largeur Δt tel que $T=N\Delta t$. Dans chaque intervalle, nous remplaçons $h(t)$ par $h(k\Delta t)$ où $k \in \{0, 1, \dots, N-1\}$. La relation ci-dessus peut être donc approximée par :

$$y(t) \approx \sum_{m=0}^{N-1} h((N-m-1)\Delta t) \int_{t-T+m\Delta t}^{t-T+(m+1)\Delta t} x(\tau)d\tau \quad (166)$$

Cette relation donne la sortie à l'instant t et pour voir la sortie en $t+\varepsilon$, il faut la recalculer.

Nous proposons un circuit simple adapté à une implantation VLSI dont le schéma simplifié est donné ci-dessous.

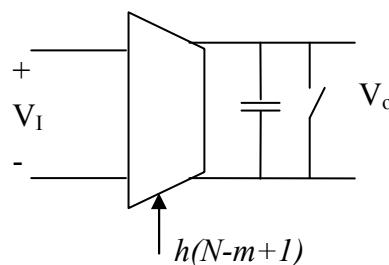


Figure 3-47- Transconductance puis intégrateur

A un instant donné, on court-circuite le condensateur pour initialiser l'intégrale. Ensuite, on charge celui-ci par un courant proportionnel à $h(N-m+1)$ fois le signal d'entrée. Pour suivre l'équation (166), on change le coefficient du filtre à chaque Δt secondes.

Pour une implantation VLSI de l'étage transconductance, on utilise les travaux de Nauta [28] où il obtient une bonne performance en terme de vitesse et de simplicité. Le circuit de transconductance est présenté sur la Figure 3-48 où chaque inverseur est fabriqué avec deux transistors CMOS (voir [28] pour le circuit CMOS des inverseurs).

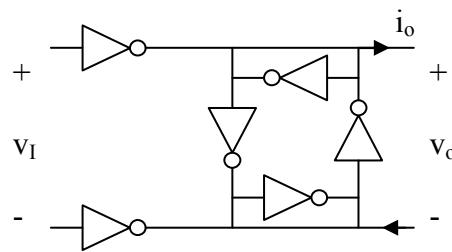


Figure 3-48- Transconductance avec une résistance de sortie "infinie"

Pour faire changer le gain de transconductance, nous proposons d'utiliser un banc de ces éléments chacun contrôlé par un bit du coefficient du filtre. Ainsi, en connectant toutes les sorties, on obtient le courant nécessaire qui chargera le condensateur. Cette architecture suppose donc que les coefficients soient stockés dans une mémoire numérique et qu'ils soient fournis au circuit avec une horloge¹. Le schéma simplifié est donné ci-dessous :

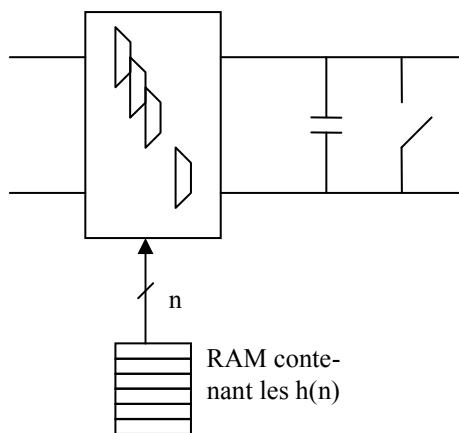


Figure 3-49- Variation du gain en fonction des coefficients du filtre

¹ Cette idée, soumise à la conférence ICC 2007, est donné dans l'Annexe V

On note l'absence d'échantillonneur/bloqueur en entrée du système. De l'autre côté, et pour les applications où un sur échantillonnage est à effectuer sur le signal en entrée, il suffit de multiplier ce circuit et donner un retard de phase au signal qui contrôle la mémoire RAM. Par exemple pour un sur échantillonnage de 4, la Figure 3-50 présente le circuit nécessaire accompagné des signaux d'horloge.

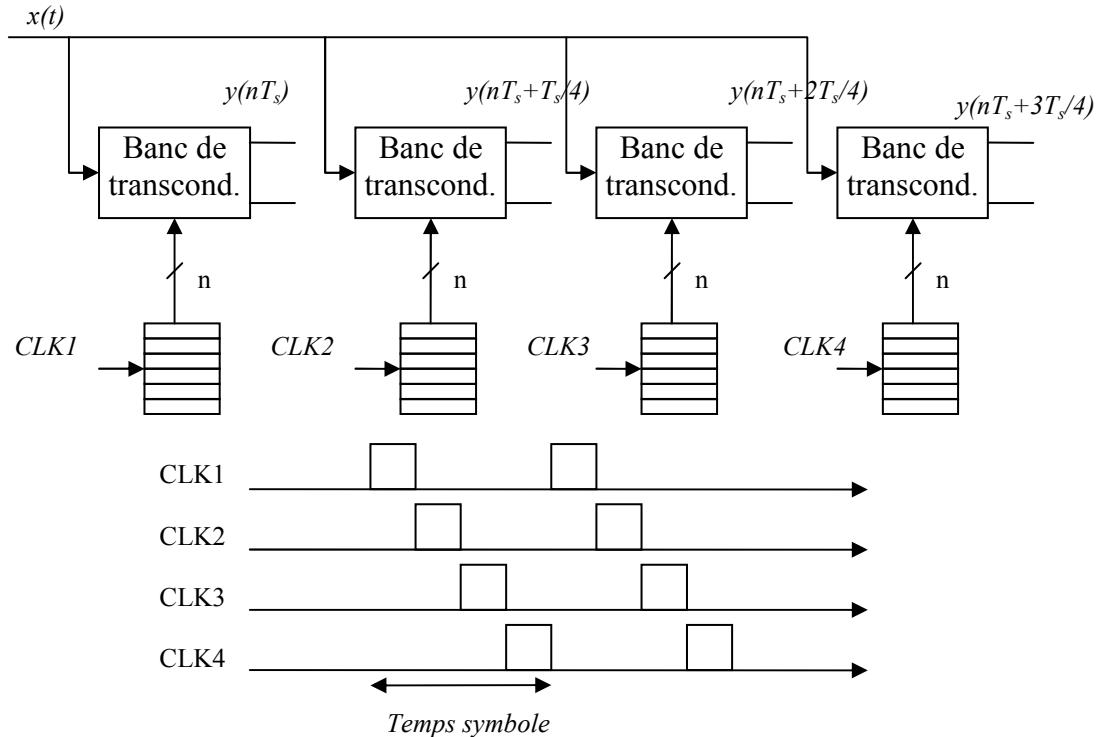


Figure 3-50- Sur échantillonnage de 4

Nous envisageons d'ajouter à ce schéma, le problème d'interpolation qui s'avère nécessaire dans l'étage de synchronisation de temps des récepteurs numériques.

3.10 Perspectives

Dans le souci d'augmenter la taille du code dans une implantation analogique, il est important de pouvoir trouver des architectures modulaires. Un effort doit être fait en terme d'algorithme, de casser le turbo décodeur en plusieurs sous paquets sans pour autant dégrader la performance (un exemple de cette procédure est donné par Arzel [24]).

L'autre point critique est le problème d'entrelacement qui donne très souvent un nombre de piste trop important sur le VLSI aboutissant à un nombre de couches prohibitif. Il faudra donc trouver des entrelaceurs modulaires et programmables.

Le problème de simulation reste encore d'actualité. Bien que nous soyons capables, utilisant la simulation circuit, de visualiser le comportement dynamique du décodeur pour un paquet reçu donné, nous ne pouvons pas en déduire la vitesse de convergence ou le taux d'erreur du décodeur. Il est important donc de développer des modèles réalistes de différents blocs d'un décodeur et des outils de calcul pour évaluer la dynamique d'un décodeur analogique, et finalement pour évaluer sa performance en terme de taux d'erreur.

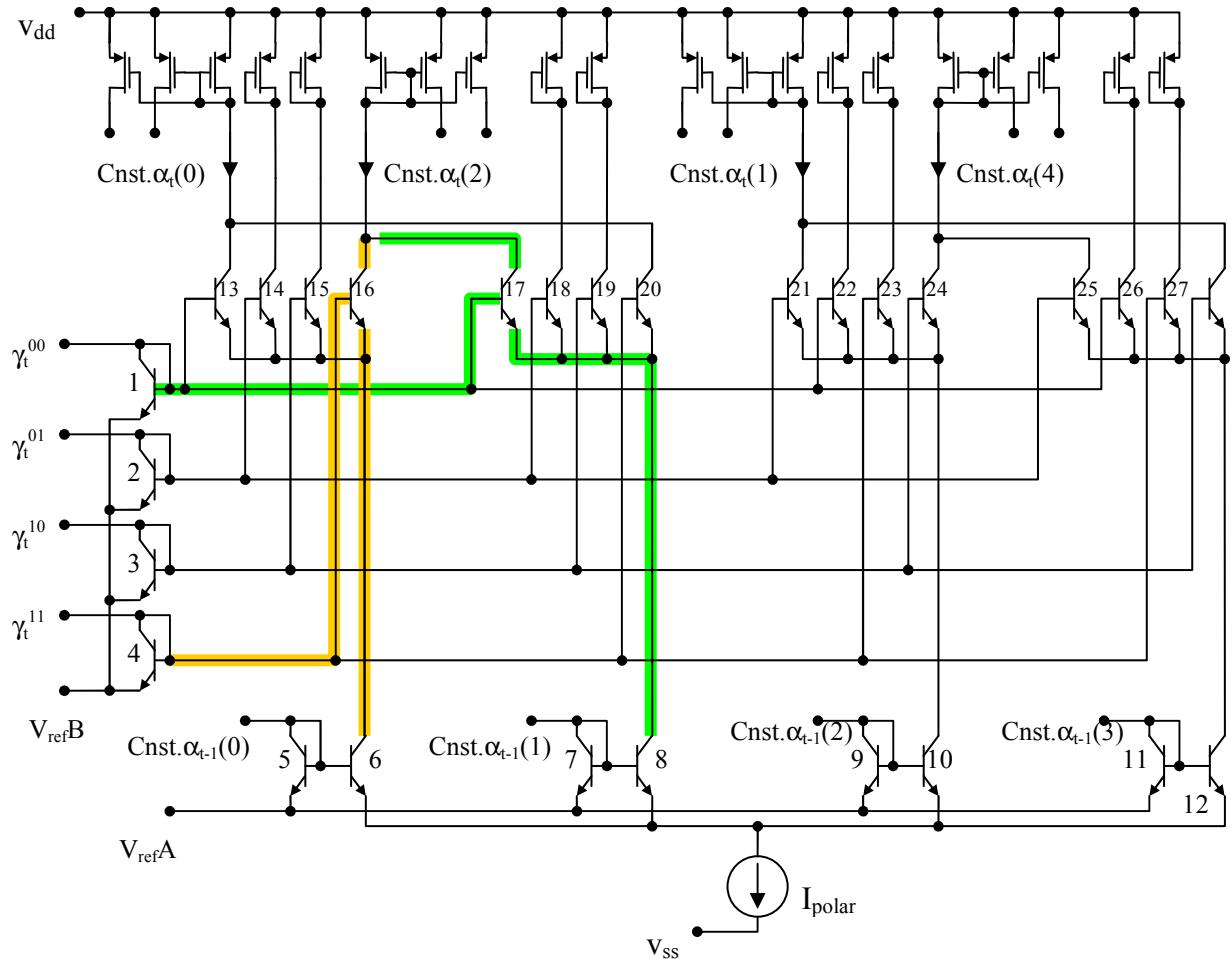
L'effet de mismatch des transistors, dû à la technologie utilisée, doit être considéré pour évaluer la dégradation. Il faut en tenir compte dans la modélisation des composants.

Un gros effort est à fournir pour la conception des échantillonneurs bloqueurs pour réduire leur taille et améliorer leur vitesse. Cela constitue un point de blocage important à étudier par les experts du circuit analogique VLSI.

L'étage de synchronisation est le premier étage dans un récepteur numérique. Pour une implantation toute analogique, il faut développer des algorithmes adaptés à des circuits analogiques. Il en va de même pour l'identification/égalisation du canal.

On voit donc que les méthodes analogiques ont encore beaucoup de chemins à parcourir !

Annexe I Réalisation de treillis



Nous calculons le courant de collecteur des transistors 16 et 17. La somme donne un des courants de sortie qui sera doublé et qui est indiqué sur le schéma par "cnst.α_t(2)".

Nous utilisons la loi de tension de Kirchoff (KVL) sur plusieurs boucles pour arriver aux équations suivantes :

$$\text{Boucle : Tr4, Tr16, Tr13 et Tr1 : } -V_{BE4} + V_{BE16} - V_{BE13} + V_{BE1} = 0$$

$$\text{Boucle : Tr4, Tr16, Tr14 et Tr2 : } -V_{BE4} + V_{BE16} - V_{BE14} + V_{BE2} = 0$$

$$\text{Boucle : Tr4, Tr16, Tr15 et Tr3 : } -V_{BE4} + V_{BE16} - V_{BE15} + V_{BE3} = 0$$

A partir de ces équations, la forme exponentielle existant entre la tension BE et le courant de collecteur, on peut écrire :

$$\frac{I_1}{I_4} = \frac{I_{13}}{I_{16}} \quad , \quad \frac{I_2}{I_4} = \frac{I_{14}}{I_{16}} \quad , \quad \frac{I_3}{I_4} = \frac{I_{15}}{I_{16}} \text{ et on sait que } \frac{I_1}{I_4} = \frac{I_1}{I_4}.$$

En additionnant le tout on a :

$$\frac{I_1 + I_2 + I_3 + I_4}{I_4} = \frac{I_{13} + I_{14} + I_{15} + I_{16}}{I_{16}}$$

On en déduit :

$$\frac{I_{ref\gamma}}{I_4} = \frac{I_6}{I_{16}} \quad \Rightarrow \quad I_{16} = \frac{I_4 \cdot I_6}{I_{ref\gamma}}$$

De l'autre côté, on peut écrire le KVL pour les boucles suivantes :

$$\text{Boucle Tr5, Tr6, Tr8 et Tr7 : } -V_{BE5} + V_{BE6} - V_{BE8} + V_{BE7} = 0$$

$$\text{Boucle Tr5, Tr6, Tr9 et Tr10 : } -V_{BE5} + V_{BE6} - V_{BE9} + V_{BE10} = 0$$

$$\text{Boucle Tr5, Tr6, Tr11 et Tr12 : } -V_{BE5} + V_{BE6} - V_{BE11} + V_{BE12} = 0$$

Et de la même manière :

$$\frac{I_6}{I_5} = \frac{I_8}{I_7} = \frac{I_{10}}{I_9} = \frac{I_{12}}{I_{11}} \quad \Rightarrow \quad \frac{I_6}{I_5} (I_5 + I_7 + I_9 + I_{11}) = I_{polar}$$

On arrive donc à

$$I_6 = \frac{I_{polar}}{I_{ref\alpha}} I_5$$

Le courant du transistor 16 est donc :

$$I_{16} = I_{polar} \frac{I_5}{I_{ref\alpha}} \frac{I_4}{I_{ref\gamma}}$$

C'est utile de remarquer que :

$$\sum_{n=5,7,9,11} \frac{I_n}{I_{ref\alpha}} = 1 \quad \text{et} \quad \sum_{n=1,2,3,4} \frac{I_n}{I_{ref\gamma}} = 1$$

C'est-à-dire qu'une normalisation des α et des γ sont faite directement dans ce circuit.

On peut donc calculer de la même manière le courant I_{17} :

$$I_{17} = I_{polar} \frac{I_7}{I_{ref\alpha}} \frac{I_1}{I_{ref\gamma}}$$

Le courant final de sortie sera donc la somme $I_{16} + I_{17}$. Ce courant est appliqué à un étage de double miroir de courant puisque chaque courant doit être appliqué à deux états de l'étage suivant (voir le treillis de la Figure 3-27).

Annexe II Programme matlab pour un décodage en graphe

Dans cette annexe un programme matlab est fourni pour un décodage sur graphe.

```
function [APP,code, flag]=graph(H,x,n_iter);
% Decoding on Graph. Major inputs to the
% program are parity check matrix H and observation vector x.
% Output APP is an approximation for a posteriori probability of code
% bits. Maximum number of iteration is n_iter but if a valid codeword is
% detected, it exit from the loop and set the flag to true.

% M is a vector containing the messages from variable nodes to function
% nodes. N with the same size as M contains messages from function nodes to
% variable nodes. method of indexing for M (or N) deserves more
% explanation. Each "1" in H corresponds to one edge and consequently to
% two messages. we indexes edges by counting "1" from first column of H,
% then second column and so on.
% M. ZAHABI 20/02/2006
% ENSIL/XLIM University of Limoges France

[F,V]=size(H);
[i, j]=find(H==1);
M=x(j); %initialize messages M (V=>F)
flag=false;
for t=1:n_iter,
    for f=1:F,
        temp(f)=prod(tanh(M(i==f)/2));
    end;
    N=2*atanh(temp(i) ./tanh(M/2)); %update messages N (F=>V)
    for v=1:V,
        temp(v)=sum(N(j==v)) + x(v);
    end;
    M=temp(j)-N; %update messages M (V=>F)
    code=(sign(temp)+1)/2;
    APP=temp;
    if ~any(rem(H*code',2)), %check for a valid code
        flag=true;
        break;
    end
end;
end;
```

Un programme de démonstration utilisant cette fonction est donné ci-dessous.

```
%----- Channel and Code definition -----
EbN0vector=[-9 -6 -3 -2 -1 0 1 2 3 4 5];
H=[1 1 1 0 1 0 0 0;
   1 1 0 1 0 1 0 0;
   1 0 1 1 0 0 1 0
```

```

0 1 1 1 0 0 0 1];          %(8,4,4) Hamming code
G=gen2par(H);               %Generator matrix from H
Block_Size=4;
Rate=0.5;
SNR=EbN0vector+10*log10(Rate)-10*log10(0.5);
EbN0_lin=10.^0.1*EbN0vector;
Es=1;
Eb=Es/Rate;                 %bit energy
N0=Eb./EbN0_lin;
Lc=4./N0;                   %channel state information
%----- Sum-Product Algorithm -----
BER=[];
for k=1:length(EbN0vector),
    n=0;
    ber=0;
    while ber<1000
        n=n+1;
        msg = randint(1,Block_Size,2);          % Random data
        code = rem(msg*G,2);                   %coding by G matrix
        code=code*2-1;                         %modulation
        ncode = awgn(code,SNR(k));            % Add noise.
        [APP,decode, flag]=graph(H,Lc(k)*ncode,10);
        ber=ber+sum(abs(decode(1:4)-msg )); %calculate the number of errors
    end;
    BER=[BER ber/(Block_Size*n)];
end;
BER_Graph=BER;
semilogy(EbN0vector,BER_Graph);grid

```

Annexe III Article ICEE2006

L'article ci-dessous a été présenté dans la conférence ICEE (14th Iranian Conference on Electrical Engineering).

Analog CMOS Kernel for ML Decoding of Convolutional Codes

M.R. Zahabi, V. Meghdadi, J.P. Cances, A. Saemi, J.M. Dumas, B. Barelaud

Xlim – UMR CNRS 6172 , Departement C²S² , University of Limoges

Tel (+33) 555 42 36 70

zahabi@ensil.unilim.fr

Fax (+33) 555 42 36 80

meghdadi@ensil.unilim.fr

Abstract: Decoding of convolutional codes are computationally demanding especially with large code words. It is argued that analog implementations of such decoders can be much more efficient than its digital counterpart. Viterbi Algorithm (VA) is an efficient method to achieve a Maximum Likelihood (ML) solution for convolutional codes. Implementation of Add-Compare-Select which is the kernel of digital realization of the algorithm is still challenging because its effect on overall speed and power consumption of decoder. In this paper a current-based analog CMOS circuit is introduced and demonstrated by its transient and DC simulation results, assuming a 0.35 μ m CMOS technology. Two main source of errors due to practical limitation of the proposed circuit are also considered. Behavioral simulation shows that performance degradation with respect to standard VA is negligible.

Keywords: Analog CMOS, Viterbi, WTA, ML Convolutional Decoding

1 Introduction

The implementation of (VA) to obtain high throughput has been always of great interest due to its application in convolutional decoders, trellis code demodulation, turbo-codes, etc. Many digital architectures, adapted to FPGA or digital VLSI circuits, have been proposed to improve the bit rate [1,2,3]. Recently, the higher throughput with less power consumption seems to be possible using analog VLSI circuit [4,5,6,7]. The idea is to replace the digital add-compare-select (ACS) with its analog equivalent.

Several ideas have been proposed for conditioning of analog signals, for example, finding the maximum (minimum) of two or more currents(voltages) [8]. It is possible to employ these ideas for realization of VA. The advantages are omitting analog to digital converter, saving quantization error as well as a better speed/ power trade-off. Current mode circuits are often preferred because summation of two signal is equivalent to tying their corresponding wires. Nevertheless replicating a current needs current mirror. New MOS technology features very short channel length and

hence good speed. On the other hand effect of channel modulation makes them far from being a good current source. The situation is still worse in p-channel MOS assuming a typical n-Well process. So care must be applied in the design of such circuits. In this work two Winner-Take-All (WTA) circuits are combined so that it can realize the principal part of VA. The output current of a WTA is the maximum of its input currents. A subtle change in standard VA is done to adapt the real circuit requirements. Section (2) provides a brief review of VA. Section (3) details conditioning of input signal (observation). Reformulating the algorithm in order to adapt to practical consideration is discussed in part (4). In part (5) the designed circuitry is introduced and characterized by its response via circuit level simulation. Part (6) demonstrate bit error rate of the proposed circuits based on behavioral simulation. Finally part (7) will conclude this paper.

2 ML Decoding of Convolutional Codes

Briefly, using some tail bits, a convolutional encoder can be considered as a block code where a sequence of k information bits is transformed to a new sequence of size n using a finite state machine with N_s states. The coded sequence $c_t = (c_{1,t}, c_{2,t}, \dots, c_{n,t})$ then is applied to a modulator (mapper) and is sent through communication channel. After mapping, namely BPSK, the input channel symbols are denoted by x_t and the channel output by $y_t = x_t + n_t$ assuming a memoryless additive Gaussian noise channel. Generalization to the case of flat fading channel is trivial.

The task of the receiver is to find the most likely coded sequence $\hat{C} = \{c_t | t=1,2,3,\dots\}$ from observations $Y = \{y_t | t=1,2,3,\dots\}$ which can be formally written as:

$$\hat{C} = \underset{C}{\operatorname{argmax}} P(Y|C)$$

and since the channel is memoryless then:

$$\hat{C} = \underset{c_1, c_2, \dots}{\operatorname{argmax}} \sum_t \log(P(y_t | c_t)) \quad (1)$$

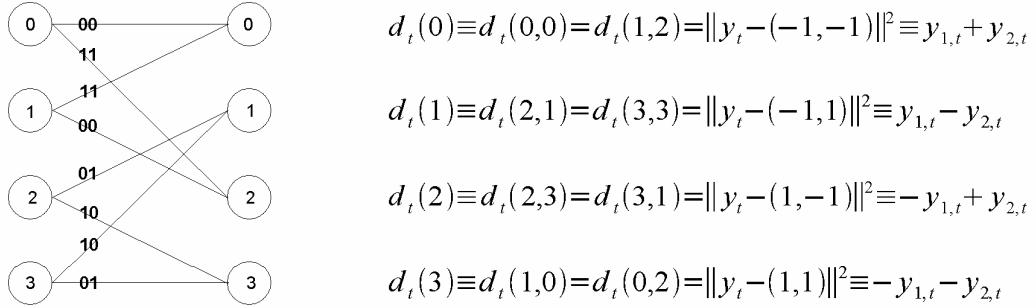


Figure 1: (a) A part of trellis for a generator $(1, 1+D+D^2/1+D^2)$ and corresponding code words c_t ,
(b) calculation of the branch metrics for each edge

Since all sequences of size n are not valid code sequences, the VA presents an efficient method to use ML algorithm only for valid code sequences. Furthermore, using VA, the exhaustive search is replaced by much more efficient iterative method. Using trellis representation, $\log(P(y_t|c_t))$ is called branch metric and when it accumulates over a specific path, it is called corresponding path metric.

Figure (1) shows some details of VA for a systematic recursive (7,5) convolutional code that during this article is kept as a generic code for simulations. The branch metrics d_t are calculated based on observation y_t and corresponding c_t for each edge in the trellis diagram. Assuming Gaussian noise, maximizing $\log P(y_t|c_t)$ is equivalent to minimizing $\|y_t - c_t\|^2$. At each state, branch metrics are added to path metrics of corresponding previous stage based on the structure of trellis, following by a comparison and selecting the smallest competitors. The new path metrics are passed to following stages. Denoting path metric of state k at time t as $D_t(k)$, This operation is summarized as follow:

$$D_t(k) = \min \left\{ \begin{array}{l} D_{t-1}(i) + d_t(i, k) \\ D_{t-1}(j) + d_t(j, k) \end{array} \right\} \quad (2)$$

Furthermore, for decoding of information bits, the surviving path should be stored for each state and used in trace back operation:

$$v_t(k) = \operatorname{argmin} \left\{ \begin{array}{l} D_{t-1}(i) + d_t(i, k) \\ D_{t-1}(j) + d_t(j, k) \end{array} \right\} \quad (3)$$

3 Conditioning of Input Signal

Several circuits were considered in order to obtain a **Max** or **Min** function with good speed and low

complexity. The later feature is more important in the system with a large number of states N_S . Among plenty of such circuits, it seems that the WTA in [8] still outperforms other ones in the sense of robustness, speed and complexity. Assuming a current mode WTA with **Max** functionality, we define a new variable λ in order to replace **Min** in equation (2) with **Max** as well as converting bipolar (voltage) variables to unipolar current variables as:

$$\lambda_t(i) = I_{ref} - \alpha f(d_t(i)) \quad (4)$$

Here $d_t(i)$ represents the distance between observation y_t and x_i for a particular $c_t=i$. The fact that adding (subtracting) a constant to all path metrics has no effect on the algorithm, we can expand $d_t(i)$ and eliminate the constant terms in order to achieve the linear equations as shown in figure (1b). Thus $d_t(i)$ will be a bipolar (voltage) variables. $f(\cdot)$ in equation (4) is a limiting function defined by:

$$f(x) = \begin{cases} LB & x < LB \\ x & LB < x < UB \\ UB & x > UB \end{cases} \quad (5)$$

where LB and UB are lower band and upper band respectively. The bands must be set properly so that overall Bit-Error-Rate (BER) remain almost unchanged. Recall that in a typical competing path, most branch metrics are minimal, we can argue that the bands must span over -2 (volt) with sufficient margin that depend directly on the noise variance. α is a constant with dimension of $(\text{ohm})^{-1}$. Finally I_{ref} is chosen so that λ remains always positive. It must be mentioned that equation (4) can almost be realize by just one MOS as a transconductance converter and a current conveyor or [9].

4 Adapted Algorithm

In the algorithm there are three parameters, I_P , I_N and I_{th} , that should be set properly as will be described below.

I) Set initial path metrics to

$$M_0 = (I_P, 0, 0, \dots, 0)$$

II) Set $t=1$ and calculate branch metrics corresponding to each edge in trellis diagram according to equation (4) and the trellis (codes):

$$\lambda_t(i, j) \quad \forall i, j \in \{0, 1, \dots, N_s - 1\}$$

III) For each t (and all k) update path metrics by

$$M_t(k) = \begin{cases} G & \text{if } M_{t-1}(i) < I_{th} \\ G - I_N & \text{otherwise} \end{cases} \quad (6)$$

where the intermediate variable G is:

$$G = \text{MAX} \left(\begin{array}{l} M_{t-1}(i) + \lambda_t(i, k) \\ M_{t-1}(j) + \lambda_t(j, k) \end{array} \right)$$

Knowing that $0 < \lambda < \lambda_{max} = \alpha(UB - LB)$, selecting $I_P > \lambda_{max}$ ensure that starting state is zero. In fact I_P propagates through the algorithm and determines the current that the circuit works around it. So it must be chosen so that the circuit has the best speed and precision. In our design it is found by simulation to be $50\mu\text{A}$.

It is clear that path metrics continue to increase in each iteration and so normalization is inevitable. This is done by subtracting I_N from all metrics at the specified condition as given in equation (6). Delay of normalization doesn't accumulate on the delay of metric calculation because two WTAs work in parallel. This is why M_{t-1} is sensed instead of M_t . It is expected that metrics have variation between $I_{th} - I_N$ and I_{th} . Again they are

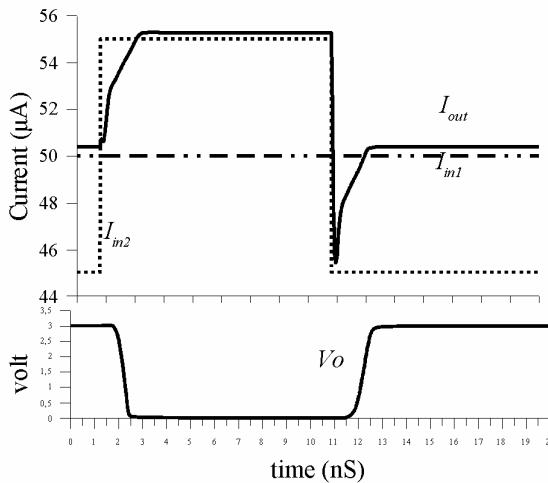


Figure 3: (left) Typical transient response of WTA-MAX in figure (2) , (right) its DC characteristic

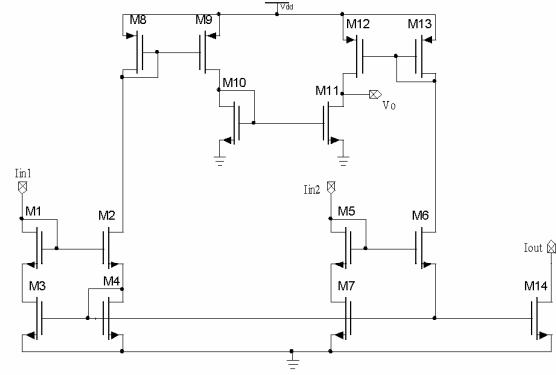


Figure 2: WTA-MAX circuit for realization of **MAX** and **ARGMAX** function.

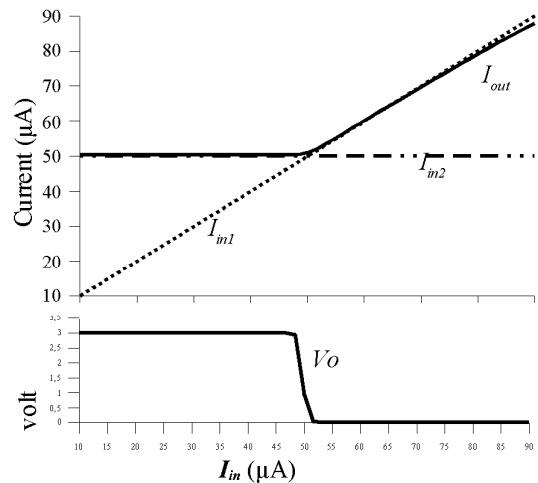
set to the best operating range of the circuit. While practically possible, there is no need to check all path metrics to perform normalization. Dynamic range of path metrics, Δ_{MAX} , is limited due to pruning of surviving paths [10] according to the following equation:

$$\Delta_{MAX} < \lambda_{max} \log_2(N_s) \quad (7)$$

So it will be sufficient to check one metric (say, path metric of state zero) and set $I_N < \Delta_{MAX}$ to ensure all other path metrics remain positive.

5 ANALOG DECODER CIRCUIT KERNEL

In this section we present a CMOS realization of above algorithm and its functionality by means of transient and DC responses. All simulations are based on AMS0.35μ process with BSIM3v3 level 53 model for MOS which were performed in Cadence™. Figure (2) and (3) show WTA-MAX circuit and its response during static and dynamic regime respectively.



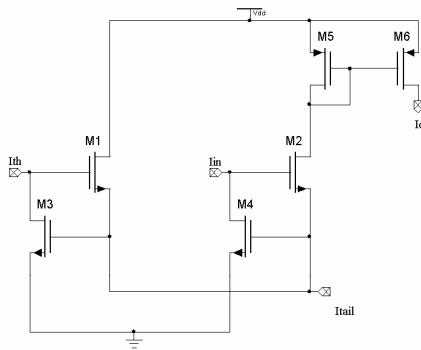


Figure 4: WTA-N circuit which implement a current comparator

From figure (3) it can be recognized that the circuit settle down completely after 1.8nS and with refer to DC response, static error is negligible. Maximum error take place when two inputs are the same. In this case, figure (3) estimates an error of about 2%. At higher current, output curve bends due to effective mobility reduction. Fortunately it is far from operating point. Finally V_o which will be connected to surviving path memory (or state register in register exchange decoding mode) represents very sharp variation. Normalization of path metrics required a current comparator. Fortunately the original WTA discussed in [8] have a desired characteristic. The circuit (WTA-N) and its response are shown in figure (4) and figure (5) respectively. In obtaining the responses, a current source of $I_N=20\mu A$ was connected to port I_{tail} in figure (4). So when current I_{in} become greater than I_{th} , output current changes from zero to I_N . Figure (6) shows complete circuit as kernel of decoder. It consist of WTA-MAX, WTA-N and some current mirror for duplication of the currents. Channel length is augmented for p-channel transistors in this figure to mitigate the effect of channel length modulation. The drawback is a little speed loss which is inevitable in CMOS tech-

nology. All transistors have $W/L=1.2\mu/0.35\mu$, except those indicated in table (1). Note also that for a code with N_s states we need the same number of WTA-MAX, but just one WTA-N will be sufficient as discussed theoretically in part (4).

Moreover in figure (6), transistor M1 is just half of a current mirror. Depending on the code, we need replication of its current by sharing V_{GS} with other transistors. For example for the code defined in figure (1), two edges leave each state so we need two copies of M1's current.

Figure (7) shows the effect of connecting WTA-N to WTA-MAX in the complete circuit of figure (6). Here a step change of $20\mu A$ was expected in WTA-MAX output, but it turns back to $45\mu A$, because I_N had been set to $15\mu A$. For a complete characterizing the CSN circuit, refer to figure (8), in which a family curves are produced by parametric sweep on I_{in1} and I_{in2} . The response is like of figure (3) except an $I_N=20\mu A$ abrupt change at the threshold current of $I_{th}=50\mu A$.

6 Behavioral simulation results

In order to estimate the bit error rate of the proposed decoder, among several source of errors we consider two secondary effects which may impact overall performance of the analog Viterbi decoder. Conditioning of branch metrics is the first one that reduces the dynamic range of metrics. Second effect is the non ideal **Max** function due to its soft transition as depicted in figure (3). First issue is simply modeled by imposing equation (5) in VA. Figure (9) shows that a tighter range results in a worst performance and vise versa. To include the non ideal **Max** function we add a nonlinear function $g(\cdot)$ to ideal **Max** function:

$$\max^*(x, y) = \max(x, y) + g(x-y) \quad (8)$$

where $f(x)$ defined as:

$$g(x) = a e^{-b|x|} - \lambda |x| \quad (8)$$

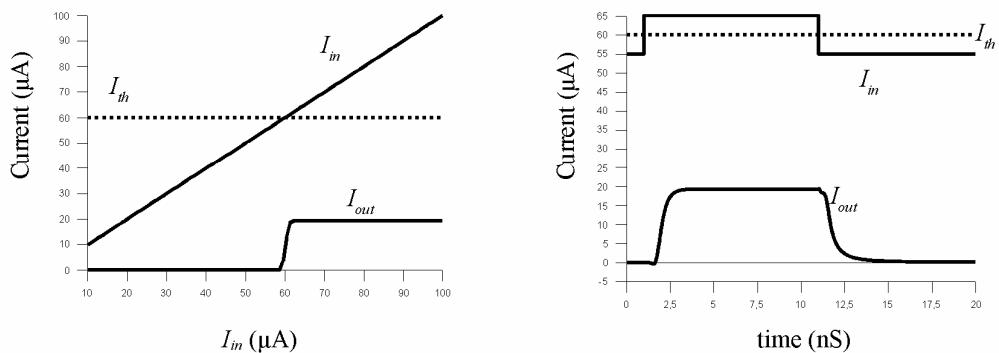


Figure 5: Typical dynamic and static response of WTA-N

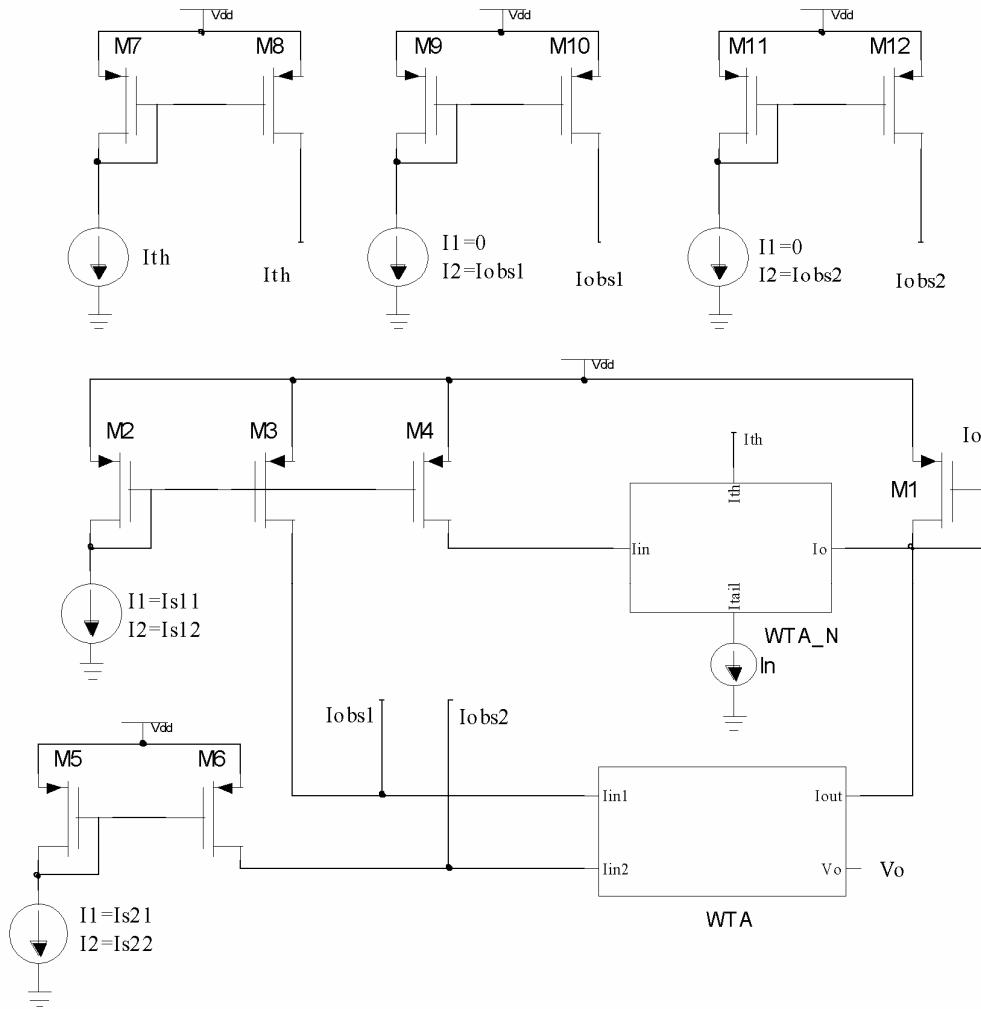


Figure 6: Complete circuit for Compare, Select and Normalization (CSN) of two metrics

The constants a , b and λ are obtained from figure (3). For our case, a least squares fitting results in $a=0.65$, $b=0.9$ and $\lambda=1/30$. Figure (10) shows effect of this model on bit error rate where a dynamic range of $UB-LB=4$ (volt) has been assumed.

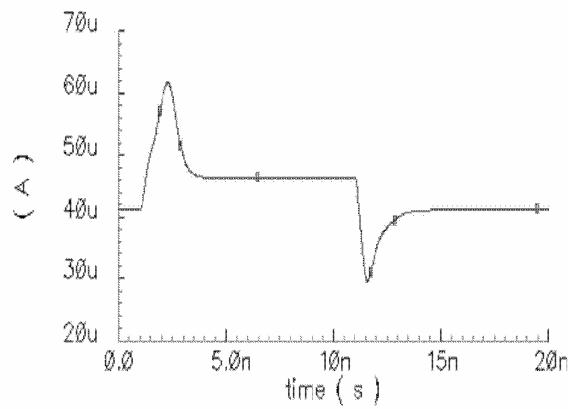


Figure 7: Typical transient response of complete circuit

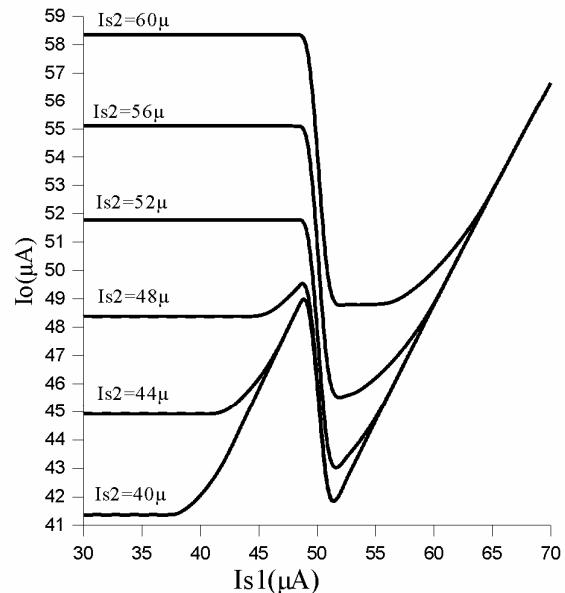


Figure 8: DC characteristic of CSN circuit

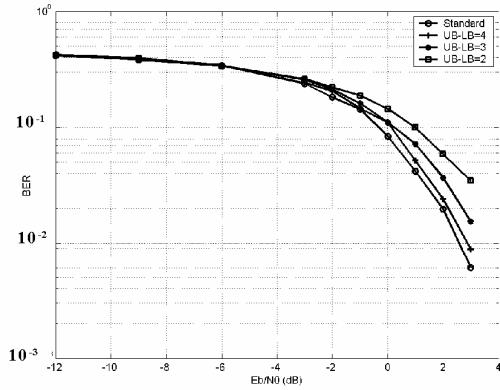


figure 9: Effect of metric's range on bit error rate

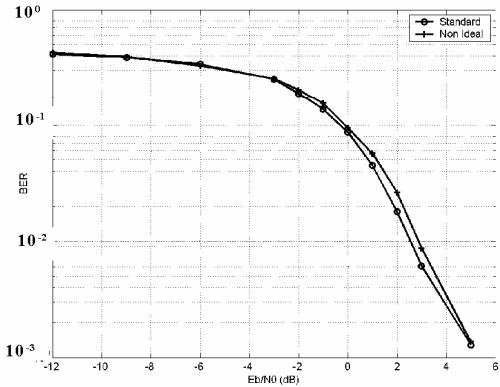


Figure 10: compression of standard decoding with practical one due to Non ideal Max function as well as limiting the dynamic range of metrics.

7 Conclusion

In this paper an analog circuit is proposed to replace Add-Compare-Select of Viterbi decoder. We showed that the proposed circuit is rather simple to implement. Circuit level simulation shows that its settling time is in the order of nanoseconds. Two practical limitations due to non ideal behavior of **Max** function and range limiting of inputs voltages, are considered during a behavioral simulation that reveals a negligible bit error rate degradation.

Nevertheless other part of a Viterbi decoder, say trace back, still remains unchanged. It is evident that with this analog circuit we don't need A/D converters in the input, but instead we need current switches (S/H) to feed output current of WTA back to its inputs in a sequential manner.

Acknowledgments

The authors would like to express their sincere thanks to Mr. S. Darfeuille for his valuable help on using Cadence and Mr. Y. Zouine for his collaboration during preparing this article.

Table 1: Transistor dimensions (W/L)

WTA-MAX	M8-M11	3.6μm/0.35μm
WTA-N	M5,M6	1.2μm/1μm
CSN	M1-M12	1.2μm/1μm

References

- [1]P. J. Black and T. H.-Y. Meng, "A 1-Gb/s, four-state, sliding block Viterbi decoder," *IEEE Journal of Solid-State Circuits*, vol. 32, no. 6, pp. 797–805, June 1997.
- [2]J. Ou and M.K. Prasanna, "Time and energy efficient Viterbi decoding using FPGAs," *IEEE International Conference on Acoustics, Speech, and Signal Processing. (ICASSP '05)*, vol. 5, pp. V/33-v/36, March 2005.
- [3]Jun Tang and K.K. Parhi, "Viterbi decoder for high-speed ultra-wideband communication systems," *IEEE International Conference on Acoustics, Speech, and Signal Processing. vol. 5*, pp. V/37-v/40, March 2005.
- [4]Sung-Woo Choi and Sang-Sung Choi, "200Mbps Viterbi decoder for UWB," *The 7th International Conference on Advanced Communication Technology, 2005, ICACT 2005*.
- [5]H.Kim and H. Son and T. Roska and L.O. Chua, "High-Performance Viterbi Decoder With Circularly Connected 2-D CNN Unilateral Cell Array," *IEEE Transactions on Circuits and Systems*, vol 52, pp.2208 – 2218, Issue 10, Oct. 2005
- [6]Lin Dong and Song Wentao and Liu Xingzhao and Luo Hanwen and Xu Youyun and Zhang Wenjun, "Neural networks based parallel Viterbi decoder by hybrid design," *Fifth World Congress on Intelligent Control and Automation, vol.3*, pp.1923-1926, June 2004.
- [7]J. Hagenauer, "Decoding of Binary Codes with Analog Networks," *Proc.1998 Info. Theory Wksp.*, San Diego, CA, Feb. 8-1 1, 1998, pp. 13-14.
- [8]J. Lazzaro *et al*, "Winner-take-all networks of order N complexity", in *Proc. 1988 IEEE Conf. On Neural Information Processing - Natural and Synthetic*, Denver, 1988.
- [9]Analog VLSI Signal Processing: Why, Where and How? *Journal of VLSI Signal Processing*, Vol.8
- [10]A. P. Hekstra, "An alternative to metric rescaling in Viterbi decoders," *IEEE Transactions on Communications*, vol. 37, no. 11, pp. 1220–1222, Nov. 1989.

Annexe IV Article publié Electronic Letters¹

Analog Decoding of Tail-Biting Convolutional Codes Based on Tanner Graph

Mohammad Reza Zahabi, Vahid Meghdadi and Jean-Pierre Cances

The proposed scheme is to decode a tail-biting convolutional code based on its Tanner graph, which is traditionally done using forward-backward MAP algorithm. Therefore, decoding may be performed using standard sum-product algorithm. With respect to decoding based on trellis, all variables in a Tanner graph are binary that may lead to complexity reduction. We further use min-sum algorithm in order to decrease the analog circuit complexity. Simulation shows that there is no significant degradation in comparison with more complex traditional method.

Introduction: Convolutional codes have the advantage of simple encoding with respect to block codes. Moreover using the concept of tail-biting [1], there is no need to add overhead to close the trellis in a specific state. So the code rate is independent of packet size. The code performance can be improved by increasing the constraint length causing more complex decoding circuit without changing the code rate and/or frame length. That is why we use convolutional tail-biting codes for small packets instead of traditional block codes. Decoding of convolutional codes is often based on trellis representation of the code in which an intermediate hidden state variable is defined. The number of states (N_S) for a trellis is a function of the constraint length (v) : $N_S=2^{v-1}$. This introduced hidden variable is not binary for practical codes where v is greater than 1.

Analog decoding on the other hand is a powerful method to implement the decoder based on either trellis or (Tanner) graph [2][3][4]. While decoding on trellis is optimal in the sense of bit error rate, the complexity is rather high especially in the case of trellis with large number of states. An extension of Gilbert multiplier may be employed to handle the non-binary state variable [5]. Decoding of block codes may be accomplished using a Tanner graph, in which all variables are binary. In this configuration, any function node is realized with a box-plus [2] operation. More simplification may be achieved using min-sum algorithm in which Winner-Take-all (WTA) replaces the box-plus [6] operation. The advantage is that the exponential characteristics of bipolar transistor or sub-threshold CMOS is no longer necessary. So the decoder can be implemented with standard CMOS technology resulting in a very faster time response and lower fabrication costs with respect to sub-threshold CMOS and BiCMOS technology respectively. We propose in this letter a straight forward method to build a Tanner graph from a tail-biting convolutional code for relatively small packets in order to obtain sim-

¹Vol. 42, number 20, September 2006

ple encoding and faster response using CMOS technology. We show that this approach will lead to simple implantation of corresponding analog decoder with no significant degradation in bit-error-rate. The challenging problem of decoding speed is also addressed with a current mode WTA-MIN CMOS circuit that exhibits very fast and precision response but these results are not mentioned in this letter.

Tanner graph construction: In this paper, non-systematic non-recursive convolutional code is considered. A classical example of the code (7,5) is presented in Figure 1. An inherent problem for decoding of non-systematic codes is the lack of systematic information bits in the receiver side. The proposed idea is to define a hidden variable for the systematic bit in a Tanner graph and to eliminate the state variable that is used traditionally. Refer to Figure 1 we can write the following equations:

$$\begin{aligned} z_k &= u_{k-1}, \quad w_k = u_{k-2} \\ c_{2k+1} + u_k + w_k &= 0, \quad c_{2k} + c_{2k+1} + u_{k-1} = 0 \end{aligned}$$

A Tanner graph based on above equations is drawn in Figure 2 in which we define the variable nodes $u_i, i = \{1, \dots, 8\}$ as hidden nodes for information bits. It is clear that this graph is not unique and one may obtain different graphs by manipulation of the equations and introducing other intermediate variable(s). In this figure $c_i, i = \{1, \dots, 16\}$ are obtained from communication channel but there are not any primary information on u_k . Message passing algorithm runs with zero LLR values for u_i and after some iterations, they converge to their final values, i.e. *a posteriori* probabilities on the information bits. Parity check matrix shows there is no small loop in the structure and so good convergence to final solution is expected.

Analog implementation: From the implementation point of view, a Tanner graph with binary variables is attractive, because of its lower complexity and also its ability of realization by min-sum algorithm. We will show in the next section that the performance loss for decoding on binary Tanner graph with respect to standard trellis approach is negligible. The basic cells for realization of a trellis section and box-plus operation can be found in [2], [3], and [7]. Figure 3 is the proposed WTA-MIN cell as building block for realization of min-sum algorithm. This cell works well in strong inversion and so may be utilized for high speed applications. For a brief discussion about the operation of the circuit, notice that there are two incomplete Wilson current mirrors, comprised of M1-M2-M3 and M4-M5-M6. Each set may be completed with a fourth transistor M7 which is shared between them. The set with larger input current take M7 and forms a complete Wilson current mirror. For example if $I_{in\ 1} > I_{in\ 2}$ then M1, M2, M3 and M7 form a current mirror and the currents following into drains of M3 and M6 will be $I_{in\ 1}$ and zero respectively. Structure of Wilson current mirror guarantees precise replication of current despite finite MOS output resistance. Kirchhoff's current law (KCL) at the output node (I_{OUT}) gives the output current which is the minimum of input currents.

For a complexity comparison between trellis based decoding and proposed approaches, the number of required transistors can be estimated. For trellis implementation we use the circuits mentioned in reference [7].

N_s = number of states in trellis

n = number of different branches in a trellis section

$(n+3)N_s+n$ = number of required transistors per information bit for calculation of alphas (forward recursion). We need the same number of transistors for Betas (backward recursion).

$(2n+1)N_s+n+6+4N_s^2$ = number of required transistors per information bit for calculation of LLRs.

Thus, for above example, a total of 150 transistors per information bit is required.

As a second approach, direct realization of the graph in Figure 2 is considered where equality and function nodes are to be implemented. A branch of this graph is redrawn in Figure 4(a) and the details of implementation are depicted in Figure 4(b). Again refer to [7] for a 2-inputs-and-1-output equal node (or function node) 12 transistors are needed. On the other hand, three identical cells are to be implemented for the node of order 3. In Figure 4 there are two nodes of degree 3 and two nodes of degree 2. Total number of cells per information bit is then 10. So $10*12=120$ transistors per information bits are required. Thus second approach saves a 20 percent reduction in the number of transistors.

Third method is based on approximation of box-plus operation as:

$$\text{Box-plus}(X, Y) = 2 \tanh^{-1}(\tanh(X/2) \tanh(Y/2)) \approx \text{sgn}(X) \text{ sgn}(Y) \min(|X|, |Y|)$$

Using the circuit in Figure 3 at function nodes and some current mirrors for duplication of current, we get 85 transistors per information bits. Also notice that unidirectional current is used to represent the LLR. So we need an extra quantity to handle the sign of LLR. (there is other method for example differential-current-based approach). A total number of 120-170 transistors can be estimated per information bit. Table 1 shows the result including their corresponding technology.

Table 1: complexity comparison between the three approaches

Tail-biting trellis decoder	150 BiCMOS or sub-threshold CMOS
Direct tanner graph decoder	120 BiCMOS or sub-threshold CMOS
Tanner graph using min-sum	120~170 Standard CMOS

Simulations: In order to have a criterion on efficiency of the algorithm, we run three different simulations in Figure 5. The best one is based on trellis representation of the code with large number of iteration to guarantee a perfect result for forward and backward variables. Two other curves are based on sum-product and min-sum algorithms respectively. Degradation at $BER > 10^{-2}$ is not more than 0.2 dB.

Reference

- [1] H. H. Ma and J. K. Wolf, “On tail biting convolutional codes,” *IEEE Trans. Commun.*, vol. COM-34, pp. 104–111, Feb. 1986.
- [2] Hagenauer, J.; Moerz, M.; Schaefer, A.: Analog Decoders and Receivers for High Speed Applications.- In: Proc. 2002 International Zurich Seminar on Broadband Communications, Zurich, Switzerland, February 2002, p. 3.1 - 3.8.
- [3] Lustenberger, F., Helfenstein, M., Loeliger, H., Tarköy, F., and Moschytz, G.: ‘An analog VLSI decoding technique for digital codes’. Proc. IEEE ISCAS 1999, Orlando, FL, USA, 1999, Vol. 2, pp. 424–427
- [4] Winstead, C., Dai, J., Yu, S., Meyers, C., Harrison, R.R., and Schlegel, C.:‘CMOS analog MAP decoder for (8,4) Hamming code’, *IEEE J. Solid-State Circuits*, 2004, 39, (1), pp. 122–131
- [5] Lustenberger, F., Helfenstein, M., Loeliger, H.-A., Tarköy, F., and Moschytz, G. S. All-analog decoder for a binary (18,9,5) tail-biting trellis code. In Proceedings of the European Solid-State Circuits Conference (Sept. 1999), pp. 362-365.
- [6] S. Hemati and A. H. Banihashemi, “Iterative Decoding in Analog CMOS,” in Proceedings of the 13th ACM Great Lakes Symposium on VLSI, ACM GLSVLSI 2003, Washington D.C., USA, pp. 15-20, April 27-29, 2003.
- [7] Jie Dai, “Design Metodology for Analog VLSI Implementations of Error Control Decoders,” PhD Thesis, Electrical and Computer Engineering, University of Utah, 2002.

Authors’ affiliations:

Mohammad Reza Zahabi, Vahid Meghdadi and Jean-Pierre Cances, XLIM/University of Limoges, ENSIL, Parc ESTER Technopole, 87068 Limoges, France

Figure captions:

Figure 1- *non-recursive and non-systematic Convolutional encoder*

Figure 2- *graph representation of block convolutional codes of Figure 1 for a frame length of eight.*

Figure 3- *WTA-MIN schematics as basic block for implementing min-sum algorithm.*

Figure 4- (a) *a typical branch of Tanner graph in Figure 2* (b) *details for realization of the graph. C1 and C2 are constants obtained from channel.*

Figure 5- *Bit-Error-Rate versus E_b/N_0 for three different decoding schemes*

Figure 1

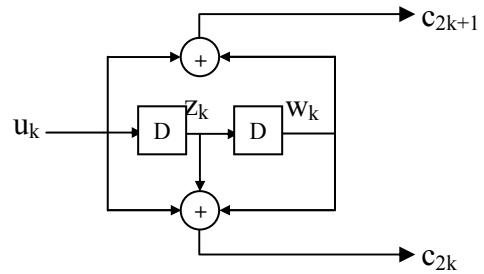


Figure 2

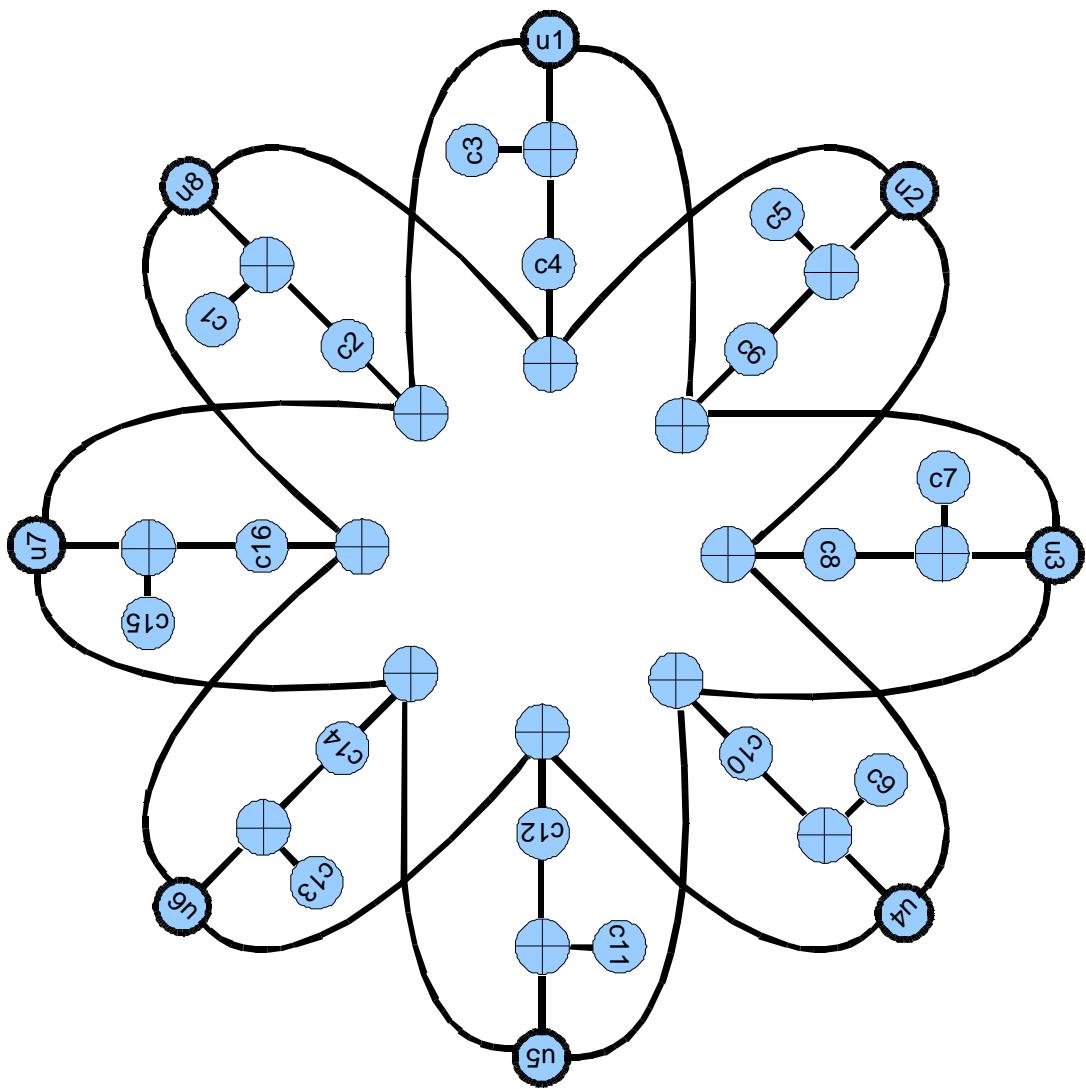


Figure 3

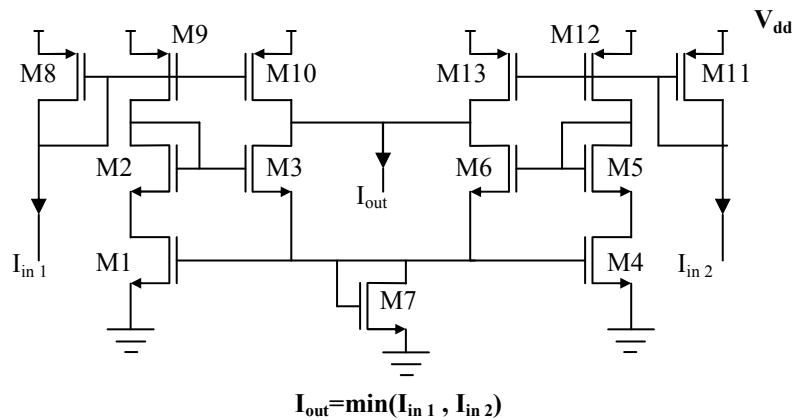


Figure 4

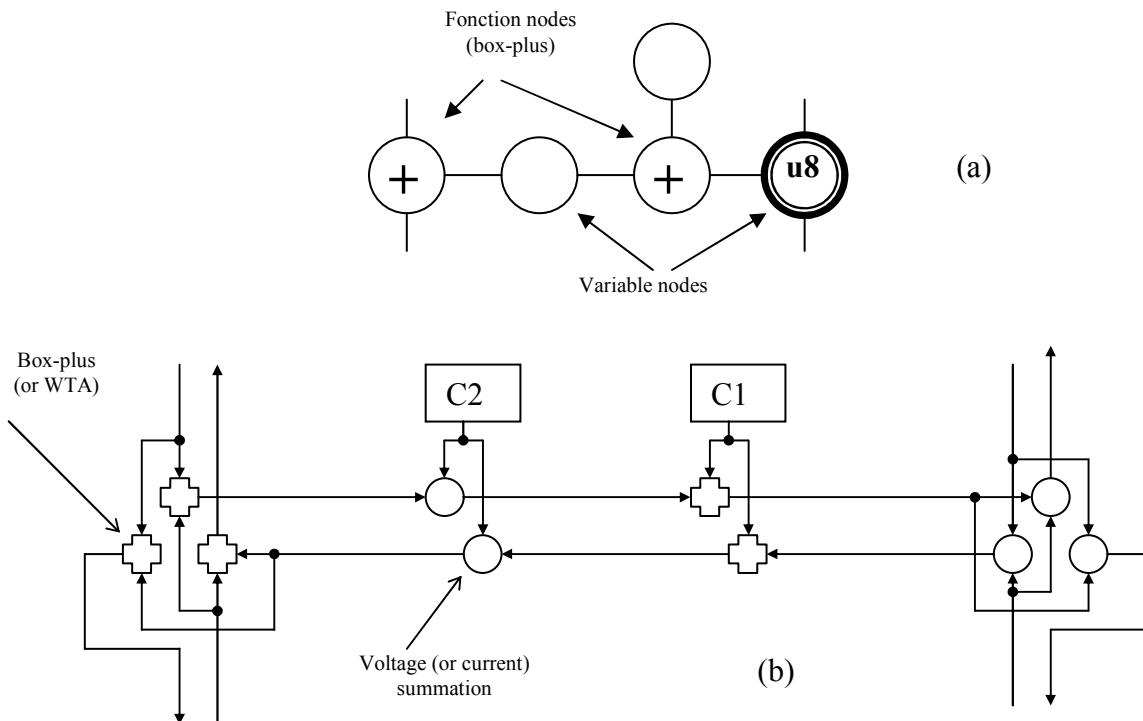
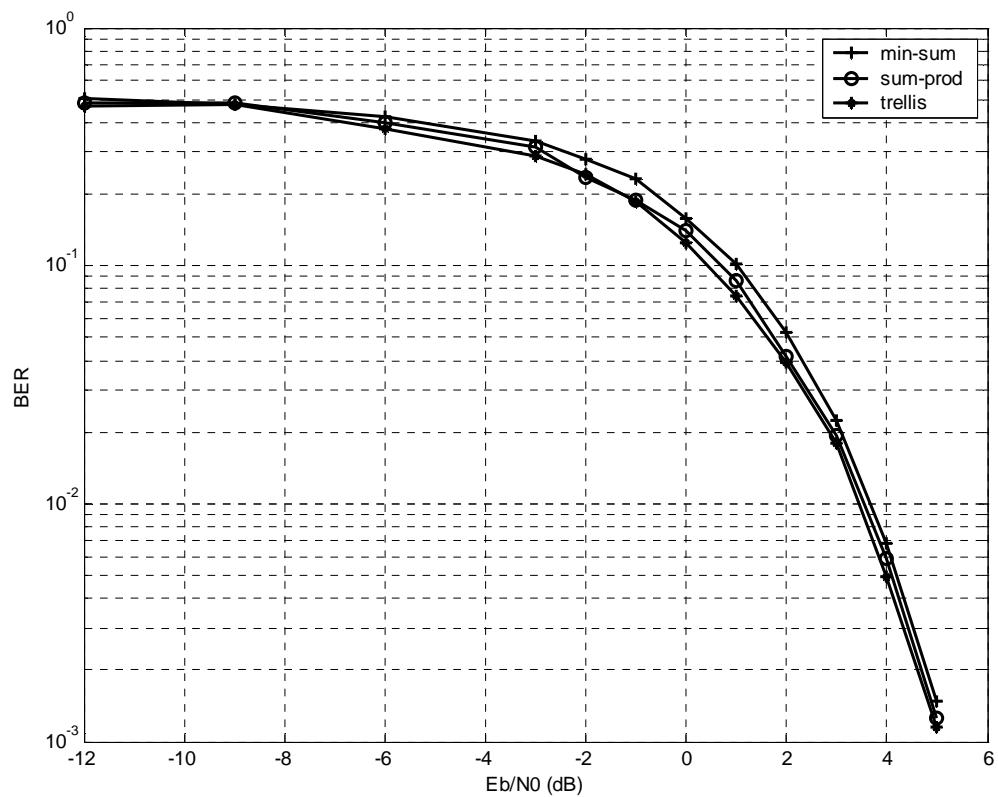


Figure 5



Annexe V Article soumis à ICC 2007

High-frequency Programmable Mixed-Signal FIR Filter

Mohammad Reza Zahabi, Vahid Meghdadi, *Member, IEEE* Jean-Pierre Cances, *Member, IEEE*

Abstract - In this paper we exploit high-frequency Gm-C integrators to realize a finite impulse response (FIR) filter. Analog input signal is partially integrated by multi-gain transconductors and capacitors while the voltage developed on the capacitors are sampled as outputs. A digital memory is used to hold filter coefficients enabling us to modify the filter characteristics in real time. The filter structure is based on MAC (multiplier-accumulator) that is implemented using CMOS transistor. Simple structure of the proposed CMOS MAC guarantees small chip area and low-power consumption when compared with traditional all-digital approach. The ability to change filter parameters on the fly is desirable in many applications such as adaptive filtering. Moreover this special MAC cell with serial analog input has potential applications in synchronization and equalization in digital communication systems.

Index term - MAC, FIR filter implementation, Gm-C, mixed-signal design.

I. INTRODUCTION

Since the works of Haganauer [1] and Loliger[2] leading to analog implementation of analog decodes, the analog realization of digital receiver front end has become a very hot issue. In fact, it is shown that an iterative channel decoder such as turbo-decoder or LDPC decoder using belief propagation algorithm can be advantageously replaced by analog circuit. So, massive degree of parallelization can be obtained due the small size of multipliers and other building blocks of decoders. However, in a practical receiver, there are some other devices before decoders that should be realized with analog CMOS transistors to take advantage of analog realization. The first and probably most critical block is the root raised cosine filter that operates normally on over sampled received signal. These filters are normally implemented using digital VLSI circuits or DSP processors proceeds by a fast sample and hold (SH) and analog to digital converter (ADC). Analog implementation is an alternative that can results in smaller or less power-consuming circuit. It is possible to implement the multipliers with current mirrors with different channel width. These implementation results in a fixed filter and can not be used where filter characteristics should be modified in real time. Switch-capacitors (SC) and switch-current (SI) are two major approaches for processing of analog sampled data [2][4]. Some authors extend SC and SI

operational amplifiers (OpAmps) used in their structure. SI networks are simpler and totally compatible with digital CMOS process because they don't need linear floating capacitors (two polysilicon layers). Several attempts were done to increase clock rate of such networks, but they need special active elements such as GaAs MESFET [5].

In this paper we propose a new architecture to implement analog FIR filter. In proposed structure, the multiplication and integration is done in analog domain while the filter coefficients are stocked in a digital memory, which enables us to modify them easily.

This paper is organized as follow. In the second section mathematical requirement to approximate a convolution integral is discussed. In the third section a general MAC cell, based on CMOS inverters, is given including comprehensive simulation to characterize the cell. In section four the steps to exploiting the MAC in an FIR filter is discussed. Finally conclusions are drawn in the last section.

II. CONVOLUTION APPROXIMATION

Suppose $h(t)$ is a time limited impulse response of a causal filter with duration D to be realized, and $x(t)$ is the input signal. The filter output is:

$$y(t) = \int_{t-D}^t x(\tau)h(t-\tau)d\tau \quad (1)$$

As an example, $h(t)$ can represent the impulse response of a root raised cosine filter where D is an integer multiple of symbol duration T : $D=NT$. In such cases, we are normally interested in the output at instants nT . The above equation can be written as follows:

$$y(nT) = \int_{(n-N)T}^{nT} x(\tau)h(nT-\tau)d\tau \quad (2)$$

Equation (1) is a continuos weighted integral of $x(t)$. The integral can be approximated by dividing the integration period into L subsections. Denoting the k -th integration gain $g_k = h((L-1-k)T)$, the filter output is approximated by:

$$y(nT) = \sum_{k=0}^{L-1} g_k \int_{nT-(L-k)\Delta t}^{nT-(L-k-1)\Delta t} x(\tau)d\tau \quad (3)$$

due to the resulting sinc function. As Δt becomes small, the effect of the sinc becomes negligible. In practical cases, one can recalculate the FIR filter taps to compensate for the frequency response deviation due to above approximation without need to choose Δt very small. This becomes important in high frequency filtering for which filter gain, g_k , should change every Δt that consequently implies a high clock rate.

III. MAC STRUCTURE

We first concentrate on realization of integration part of equation (3) using MAC structure. Among variety of different MAC, those having analog structures are more promising for high frequency applications. They are also more power efficient due to absence of ADCs. Two principal approaches are analog sampled-data processing using SI networks and analog data processing using analog integrator such as Gm-C approaches. As mentioned in the introduction section, the former method is very attractive in low or medium frequency. SI approach which is based on charging gate capacitor via a switch will be cumbersome in high frequency. In addition MOS switches dump an error charge into the gate capacitance of their respective memory transistors during their turn-off. Increasing memory-transistor size with respect to switch-transistor size can relax the problem in exchange of reducing effective frequency range of circuit. Other methods also exists (e.g. class AB)[6] but they result in a significant additional circuit complexity.

Figure 1 represents analog approach using a transconductor and a capacitor realizing an integrator (accumulation). The transconductor proposed in [7] presents an excellent frequency response which is used in this work. The input to the transconductor is applied via a cross-bar switch Figure 2 that provide 0 or 180 phase shift based on control line. When a cell has to be disconnected from ladder, M1 and M4 are off while M2 and M3 are on.

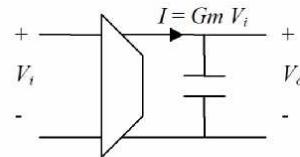


Figure 1- Transconductance-C integrator as accumulator.

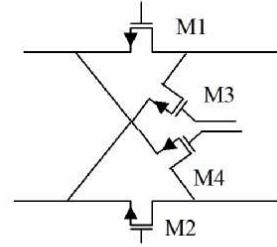


Figure 2- Cross switch to toggle between 0 and 180 as well as disconnecting a cell

As stated earlier the transconductor in this paper is based on CMOS inverter. In the following the principal performance of an inverter as a transconductor is given. Figure 3 shows a CMOS inverter as transconductor. With the same p-channel and n-channel transistors, the output current will be linear with input voltage. Using the square law formula for transistors and applying Kirchoff Current Law (KCL) at output node, the following equation is obtained.

$$i_o = \frac{1}{2} (k_n - k_p) (v_i - V_C)^2 + g_m (v_i - V_C) \quad (4)$$

where

Erreur ! Des objets ne peuvent pas être créés à partir des codes de champs de mise en forme. and

$$V_C = \frac{V_{DD} - V_{THn} - |V_{THp}|}{1 + \sqrt{k_n/k_p}} + V_{THn} \quad (5)$$

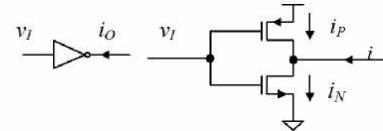


Figure 3- CMOS inverter and its circuit. It can be used as transconductor.

A balance structure can be developed as Figure 4 that effectively removes second order terms while we can take advantage of differential structure. It can be shown that following relationship hold for this circuit.

$$i_o = i_{o1} - i_{o2} = (k_n - k_p) (V_I - V_C) v_i + g_m v_i \quad (6)$$

In (6) V_I is common mode input voltage and V_C and g_m are defined in (5).

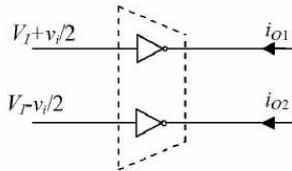


Figure 4- Balance transconductor using two inverters

Since there is no internal node in the circuit of Figure 3 or Figure 4, the transconductor can operate at very high frequency range. The principal problem with such simple circuit is its low output resistance due to channel length modulation in MOS transistors that reduces the gain. To modify this circuit, it is very important not to add any internal node(s) to preserve high frequency characteristic of the circuit. In Figure 5 gain enhancement is performed using two other transconductors. It is trivial to show that two added transconductor realize a resistance of $1/(g_{m3} - g_{m2})$ ohm at output nodes of g_{m1} . Proper setting of these values result a negative resistance that will cancel out the output resistance of all transconductors and yield a theoretical output resistance of infinity. In a practical circuit however, transistor mismatch may slightly degrade the gain. Based on equation (5) g_m is proportional to both supply voltage V_{DD} and aspect ratio of transistors (through k_n and k_p). Slight reduction of aspect ratio (W/L) for the transistors in g_{m3} can produce the required negative resistance. In order to still increase the gain, a fine tuning by V_{TN} may be needed.

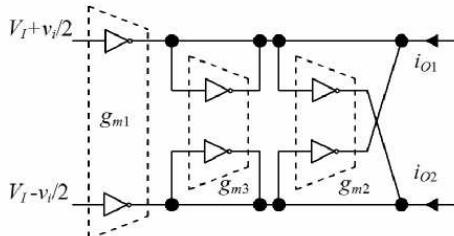


Figure 5- Balance transconductor and output resistance compensation

Figure 6 presents the overall circuit with cross switches that are used to change the polarity of input signal. It is used to take into account the possible negative filter taps. Transconductor structure is implemented using Cadence™ and AMS035 models. All n-channel transistor aspect ratio are $1.2\mu/1\mu$. For p-channel transistors in g_{m1} and g_{m2} the aspect ratio of $4.3\mu/1\mu$ is used. For g_{m3} it is designed to be $4\mu/1\mu$. Using a supply voltage of 3.3 volt for both V_{DD} and V'_{DD} , a gain of 40dB is obtained. We obtained 60 dB gain by setting V'_{DD} slightly under V_{DD} (3.1 volt).

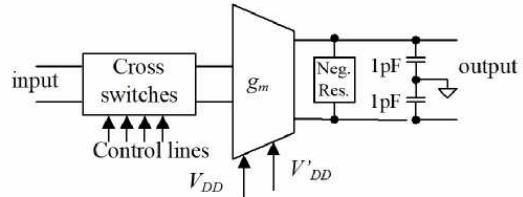


Figure 6- Overall cross switches and MAC

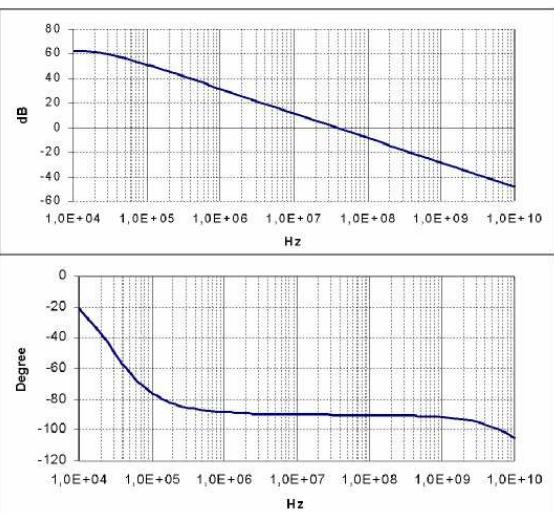


Figure 7- Frequency response of the MAC

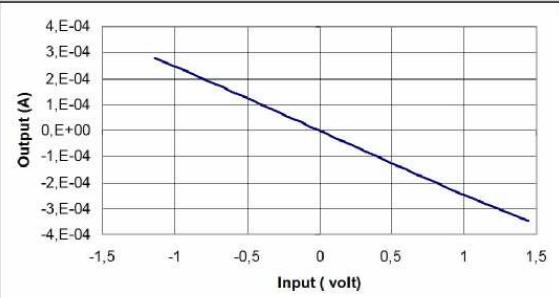


Figure 8 transconductor transfer function

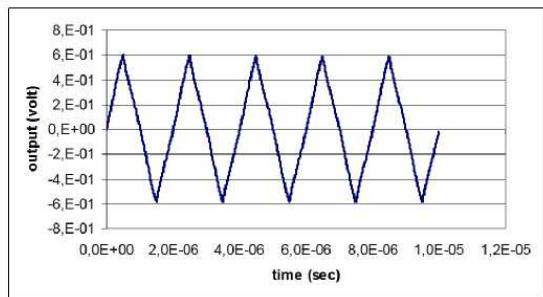


Figure 9- MAC response to a square wave input

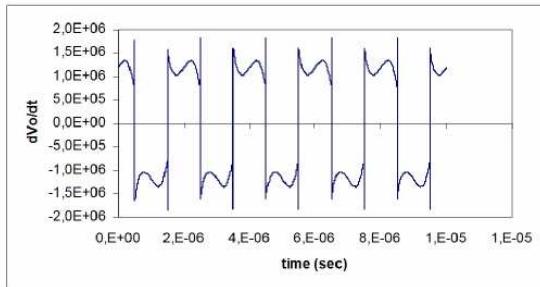


Figure 10- Derivation of triangular output of Figure 9

Figure 7 shows frequency response of the circuit in Figure 6. Low frequency pole is found to be located in 26 kHz owing to 1pF capacitance at the output and effective output resistance of transconductor. Unfortunately cascading cross switch with transconductor introduces internal node and thus a high frequency pole in 6.4GHz. The figure shows that the circuit can perform as an integrator for the range of 770kHz–240MHz with ± 2 degrees of phase error.

Figure 8 illustrates DC analysis of the transconductor and its linearity over the wide range.

A transient analysis was performed as follow. A low frequency pulse is given to integrator, including G_m and capacitor. Figure 9 shows the result. In ideal case output must be triangle. Owing to output resistance, there is a little deviation from ideal response. In Figure 10 derivation to the triangular wave is done to clarify the imperfection.

IV. FIR FILTER STRUCTURE

An FIR filter is a structure realizing. The integration can be done using a current source and a simple capacitor. The required current source is realized using the transconductance detailed in the last paragraph. In this section, we propose the overall structure to implement a reconfigurable mixed-signal FIR filter.

To realize equation (3), the input signal should be multiplied by the filter taps during Δt seconds. In the next time interval, the transconductance gain is changed to the next filter coefficient. This operation is continued until the end of the filter response. The capacitor voltage is equal to the convolution result and represents filter output at instant nT . We suppose that the filter taps have been already stocked in a RAM with M bits per taps. In order to modify the transconductance gain according to filter tap, we repeat it M times. In order to introduce the weighting a R-4R ladder is used as shown in Figure 11. Control signals are used to invert transconductance gains (for negative taps) or to disconnect a specified cell if the corresponding bit of the tap is zero.

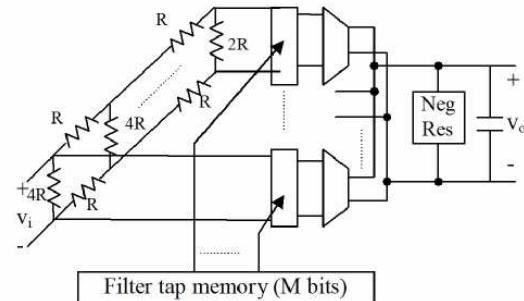


Figure 11- Filter architecture

Using this structure the filter output is calculated at instant nT . The integrating capacitor should be discharged using a switching MOS at the beginning of a new integration period D . This circuit should be repeated N times where N is D/T . In practice and for a root raised cosine filter the impulse response duration is about 6 times as symbol duration. So, the circuit in Figure 11 should be repeated 6 times in order to produce the all required samples. A digital control circuit should be placed to synchronize the filter tap memory addressing and discharging of output capacitors.

In comparison to a digital implementation, the number of transistor used is drastically reduced using this structure. This leads to a very less power consumption and much lower chip area. It is a promising structure for high speed applications with respect to sampled-data approaches. With this structure, there is no need to use sample and hold and analog to digital converters, that reduces the complexity of overall system.

V. CONCLUSION

In this paper we proposed a new structure for mixed-signal realization of FIR filter with high-frequency characteristics. It is based on CMOS technology that can be integrated along with digital circuitry. A high bandwidth MAC structure using CMOS inverter is proposed and employed as a building block of our FIR filter. Furthermore, the filter coefficients can be modified in real time. This feature is very suitable in adaptive filtering where the filter taps are modified continuously.

REFERENCES

- [1] J. Hagenauer and M. Winkelhofer, "The analog decoder," in Proc. Int. Symp. Information Theory, Cambridge, MA, 1988, p. 145.
- [2] H.-A. Loeliger, F. Lustenberger, M. Helfenstein, and F. Tarköy, "Probability propagation and decoding in analog VLSI," in Proc. Int. Symp. Information Theory, Cambridge, MA, 1988, p. 146
- [3] Srinivasan, V.; Rosen, G.; Hasler, P.; "Low-power realization of FIR filters using current-mode analog design techniques" ,Conference on Signals, Systems and Computers, Volume 2, 7-10 Nov. 2004 Page(s):2223 – 2227;
- [4] Rocha-Perez, J.M.; Silva-Martinez, J.; "SC implementation of FIR filters for digital communication systems", Third International Workshop on Design of Mixed-Mode Integrated Circuits and Applications, 1999 Page(s):179 - 182
- [5] D.G. Haigh, C. Toumazou, S.J. Harrold, K. Steptoe, J.I. Sewell and R. Bayrums, "Design optimization and testing of a GaAs switched capacitor filter," *IEEE Trans. Circuits Syst.*, vol. 38, pp. 825-837, Aug. 1991.

- [6] N.C. Battersby and C. Toumazou, "Class AB switched-current memory for analog sampled data systems, " *Electronics Letters*, vol. 27, pp. 873-875, May 1991.
- [7] Bram Nauta, "A CMOS Transconductance-C Filter Technique for Very High Frequencies ", *IEEE JOURNAL OF SOLID-STATE CIRCUITS*, VOL. 27, NO. 2. FEBRUARY 1992.

Vahid Meghdadi (M'03) was born in Mashad, Iran. He received B.S. and M.S. degrees in communication engineering from Sharif University of Technology, Tehran, Iran in 1989 and 1992 respectively, and PhD degree from University of Limoges, Limoges, France in 1998. Since 2000, he has been assistant professor in electrical engineering department of Ecole Nationale Supérieure Ingénieurs de Limoges (ENSIL), France. His main research interest consists of digital communication systems, channel coding, MIMO systems, MIMO-OFDM time and frequency synchronization and channel estimation, and analog decoding.

Jean Pierre Cances (M'04) was born in Brive, France, on April 17, 1966. He received the Eng. degree from the Ecole Nationale Supérieure des Télécommunications de Bretagne (ENSTBr) in 1990 and the Ph.D. degree from the Ecole Nationale Supérieure des Télécommunications (ENST-Telecom Paris) in 1993. He also received the teaching degree Agrégation in physics in 1993. He is currently a full professor at ENSIL Limoges University where he teaches signal processing. His professional interests include channel coding, iterative processing for MIMO systems and optical communications.

Référence

- [1] M. Choi and A. Abidi, "A 6b 1.3 GSample/s A/D converter in 0.35- μ m CMOS," in IEEE Int. Solide-State Circuits Conf. Dig. Tech. Papers, Feb. 2001, pp. 126-127
- [2] M. Frey, H.A. Loeliger, P. Merkli, P. Strelbel, "Two experimental analog decoders", Proc. 2005 IEEJ Int. Analog VLSI Workshop, Bordeaux, France, Oct. 19-21, 2005.
- [3] B. Vucetic, J. Yuan, Turbo Codes: Principles and Applications, Dordrecht : Kluwer, 2000
- [4] J. Hagenauer, M. Mörz, E. Offer, "Analog Turbo Networks in VLSI: The Next Step in Turbo Decoding and Equalization", International Symposium on Turbo Codes, ENST, Bretagne, France, September 2000, p. 209–218
- [5] J. Hagenauer, "Soft Decision MAP Decoding of Binary Linear Block Codes Via global Optimization", IEEE International Symposium on Information Theory (ISIT '97), Ulm, June 1997
- [6] J. Hagenauer, M. Moerz, A. Schaefer, "Analog decoders and receivers for high speed applications", International Zurich Seminar on Broadband Communications, 19-21 Feb. 2002, pages: 3-1 - 3-8
- [7] J. Lazzaro, S. Reyckebusch, M. A. Mahowald and C.A. Mead, "Winner-take-all network of O(N) complexity," in Touretzky, D. (Ed): "Advances in neural information processing systems 1", (Morgan Kaufmann, San Mateo, CA, 1989), pp. 703-711.
- [8] I. Baturone, J. L. Huertas, A. Barriga, S. Sánchez-Solano, "Current-Mode Multiple-Input Max Circuit" Electronics Letters, Vol. 30, no. 9, pp. 678-679, April 1994.
- [9] M.R. Zahabi, V. Meghdadi, J.P. Cances, A. Saemi, J.M. Dumas, B. Barelau, "Analog CMOS Kernel for ML Decoding of Convolutional Codes", 14th Iranian Conference on Electrical Engineering ICEE, May 16-18, 2006

- [10] F.R. Kschischang, B.J. Frey, H.A. Loeliger, "Factor graphs and the sum-product algorithm", IEEE Transactions on Information Theory, Vol. 47, Issue 2, Feb 2001 Pages: 498 - 519
- [11] H.A. Loeliger, F. Lustenberger, M. Helfenstein, F Tarkoy, "Probability propagation and decoding in analog VLSI", IEEE Transactions on Information Theory, vol 47, No 2, Fev. 2001, pages: 837-843
- [12] J. Hagenauer, M. Mörz, E Offer, "A Circuit Based Interpretation of Analog MAP Decoding with Binary Trellises", 3rd ITG Conference Source and Channel Coding (SCC'00), München, January 2000, ITG-Fachbericht 159, VDE-Verlag, Berlin, p. 175–180
- [13] J.B. Anderson, S.M. Hladik, "Tailbiting MAP decoders" IEEE Journal on Selected Areas in Communications, Volume 16, Issue 2, Feb. 1998 Pages: 297 - 302
- [14] F.R. Kschischang, V. Sorokine, "On the trellis structure of block codes", IEEE Transactions on Information Theory, Vol. 41, Issue 6, Part 2, Nov. 1995, Page: 1924 - 1937
- [15] S.Yu,C.Winstead, C.Myers, R.Harrison, and C.Schlegel, "An analog decoder for (8,4) Hamming code with Serial Input Interface", Proceedings of IEEE International Symposium on Circuits and Systems (ISCAS02), May 2002, Arizona, USA.
- [16] C. Winstead, J. Dai; S. Yu; C. Myers, R.R. Harrison, C. Schlegel, "CMOS analog MAP decoder for (8,4) Hamming code" IEEE Journal of Solid-State Circuits, Volume 39, Issue 1, Jan. 2004 Pages: 122 - 131.
- [17] A.R. Calderbank, G.D. Forney, A. Vardy, "Minimal tail-biting trellises: the Golay code and more", IEEE Transactions on Information Theory, Volume 45, Issue 5, July 1999 Pages: 1435 - 1455
- [18] K. Andrews, D. Divsalar, S. Dolinar, J. Hamkins, F. Pollara, "Very High Speed Analog Decoders", Internal JPL report, D-22054 August 24, 2001
- [19] S. Hemati, A.H. Banihashemi, "Dynamics and performance analysis of analog iterative decoding for low-density parity-check (LDPC) codes", IEEE Transactions on Communications, Vol 54, Issue 1, Jan. 2006 Pages: 61 - 70.
- [20] V.C. Gaudet, P.G. Gulak, "A 13.3-Mb/s 0.35-/spl mu/m CMOS analog turbo decoder IC with a configurable interleaver", IEEE Journal of Solid-State Circuits, Vol. 38, Issue 11, Nov. 2003 Pages: 2010 - 2015

- [21] D. Vogrig, A. Gerosa, A. Neviani, A.Gi. Amat, G. Montorsi, S Benedetto, "A 0.35- μ m CMOS analog turbo decoder for the 40-bit rate 1/3 UMTS channel code", IEEE Journal of Solid-State Circuits, Vol. 40, Issue 3, Mar 2005 Pages: 753 - 762
- [22] C. Berrou, A. Glavieux, P. Thitimajshima, "Near Shannon limit error-correcting coding and decoding: Turbo-codes", in prpc. Int. Conf. Communications 1993 (ICC 93), Geneva, Switzerland, May 1993, Pages :1064 - 1070
- [23] M. Perenzoni, A. Gerosa, A. Neviani, "Analog CMOS implementation of Gallager's iterative decoding algorithm applied to a block turbo code", ISCAS 2003, Vol. 5, 25-28 May 2003, Pages: V-813 - V-816
- [24] M. Arzel, C. Lahuec, F. Seguin, D. Gnaedigtt, M. Jezequel, "Analog slice turbo decoding" IEEE International Symposium on Circuits and Systems (ISCAS 2005), 23-26 May 2005, Pages: 332 - 335 Vol. 1
- [25] D. Gnaedig, E. Boutillon, M. Jéezéquel, V. Gaudet, and P. Gulak, "On multiple slice turbo codes," in Proc. 3rd Int. Symp. on Turbo Codes and Related Topics, Brest, France, September 2003, pp. 343–346.
- [26] N. Sadeghi, H.M. Nik, C. Schlegel, V.C. Gaudezt, "Analog FFT interface for Ultra-Low Power Analog Receiver Architecture", Analog Decoding Workshop, 5-6 June 2006, Torino, Italy
- [27] J.S. Garrido, R.J. Piechocki, K. Maharatna, D. McNamara, "Analog decoding in MIMO systems", Analog Decoding Workshop, 5-6 June 2006, Torino, Italy
- [28] B. Nauta, "A CMOS transconductance-C filter technique for very high frequencies", IEEE Journal of Solid-State circuits, Vol. 27 Issue 2, Feb 1992, Pages: 142-153
- [29] Bahl, Cocke, Jelinek and Raviv, "Optimal Decoding of Linear Codes for Minimizing Symbol Error Rate", IEEE Trans. Info. Theory, Vol IT-20, pp 284-287, March 1974.
- [30] Felix Lustenberger, et al. "All-Analog Decoder for a Binary (18,9,5) Tail-Biting Trellis Code", Proceedings of ESSCIRC'99, Duisburg, September 21-23, 1999, pp. 362-365.

Chapitre 4 Conclusions et Perspectives

Avec les nouvelles applications multimédia d'aujourd'hui, il est clair qu'une capacité importante en terme de débit de transmission est nécessaire. Pour augmenter la quantité d'informations émises il faudrait augmenter le nombre de bits transmis à chaque utilisation de canal. Ceci peut se faire en augmentant la taille de la constellation utilisée et aussi en utilisant la diversité d'antenne. Nous avons donc travaillé sur les codes temps espace (en bloc et en treillis) et nous avons pu exploiter le gain de diversité et de codage offerts par ces systèmes. Un des problèmes sévères liés à ces systèmes est le problème de la synchronisation (synchronisation de paquet, de temps et de fréquence). Nous avons étudié les méthodes existantes et nous avons pu améliorer leur performance. Comme nous l'avons détaillé dans le chapitre 2, une estimation conjointe de temps et de fréquence accompagnée de l'estimation de canal nous a fait approcher de la borne inférieure de Cramer-Rao.

Cette approche est basée sur l'envoi d'une séquence connue dite "séquence d'entraînement". Quand tous les paramètres ont été estimés, il va falloir suivre leur variation. Nous pensons qu'il est judicieux de proposer des systèmes itératifs à retour de décision pour maintenir la synchronisation. C'est une voie sur laquelle nous travaillons. De l'autre côté, pour la liaison montante de la quatrième génération des portables, nous envisageons une modulation OFDMA utilisant des antennes multiples. Le problème de synchronisation d'un tel système n'est pas encore complètement résolu. Des méthodes itératives utilisant les algorithmes de EM ou de SAGE sont des moyens à mettre en œuvre pour résoudre le problème de synchronisation dans ce contexte complexe.

Pour réaliser ces algorithmes de plus en plus compliqués et avec des débits de plus en plus importants, il faut trouver des algorithmes sous optimaux pour diminuer la complexité tout en gardant une performance acceptable. Comme nous l'avons présenté dans le chapitre 3, il est possible de traiter le signal directement en analogique. Les méthodes itératives étant les parties très gourmandes en terme de calcul peuvent être réalisées en analogique moyennant un traitement en parallèle. Ceci est faisable car les blocs de traitement de signal ne comportent

que peu de transistors en comparaison avec leur équivalent numérique; un traitement massivement parallèle est donc possible. Les décodeurs analogiques ont été déjà fabriqués dans les laboratoires de recherche, nous avons généralisé ce principe sur d'autres types de blocs comme par exemple les filtres FIR. Il est possible aussi d'effectuer des opérations matricielles directement dans le domaine analogique. C'est-à-dire que nous cherchons à fabriquer un récepteur tout analogique ou au moins d'identifier et de fabriquer les parties où ce type d'implantation est justifié. Ceci pourrait non seulement améliorer la consommation, mais aussi augmenter la vitesse de traitement.

C'est un axe de recherche qui fait appel à des experts en système de communications aussi bien qu'aux experts de circuits VLSI. Sachant que dans notre laboratoire de recherche, Xlim, les deux expertises existent, cette ambition se justifie très naturellement.