

5/03

République Algérienne Démocratique et Populaire  
Ministère de l'Enseignement Supérieur et de La Recherche Scientifique

## Ecole Nationale Polytechnique



المدرسة الوطنية المتعددة التقنيات  
BIBLIOTHEQUE — المكتبة  
Ecole Nationale Polytechnique

### Département d'Electronique

Mémoire de Projet de Fin d'Etudes en vue de l'Obtention du Diplôme  
d'Ingénieur d'Etat en Electronique

#### *Thème*

# Evaluation de l'Architecture V-BLAST Appliquée aux Systèmes MIMO

Proposé et Dirigé par :  
Z.TERRA

Etudié par :  
MAKHLOUFIA Seddik

Soutenu le : 28 juin 2003 devant le jury composé de :

Dr A.BELOUHRANI  
Mr H.BOUSBIA-SALAH  
Mr Z.TERRA

Président  
Examineur  
Encadreur

Promotion : Juin 2003

République Algérienne Démocratique et Populaire  
Ministère de l'Enseignement Supérieur et de La Recherche Scientifique

## Ecole Nationale Polytechnique



المدرسة الوطنية المتعددة التقنيات  
المكتبة — BIBLIOTHEQUE  
Ecole Nationale Polytechnique

### Département d'Electronique

Mémoire de Projet de Fin d'Etudes en vue de l'Obtention du Diplôme  
d'Ingénieur d'Etat en Electronique

### *Thème*

# Evaluation de l'Architecture V-BLAST Appliquée aux Systèmes MIMO

Proposé et Dirigé par :  
Z.TERRA

Etudié par :  
MAKHLOUFLA Seddik

Soutenu le : 28 juin 2003 devant le jury composé de :

Dr A.BELOUHRANI  
Mr H.BOUSBIA-SALAH  
Mr Z.TERRA

Président  
Examineur  
Encadreur

Promotion : Juin 2003

# Remerciements

المدرسة الوطنية المتعددة التقنيات  
BIBLIOTHEQUE — المكتبة  
Ecole Nationale Polytechnique

Je tiens à remercier vivement :

- ☑ Dieu le tout puissant pour tout ce qu'il m'a donné.
- ☑ Mon père, ma belle-mère, mes frères et sœurs pour leurs encouragements et leur patience.
- ☑ Mon directeur de mémoire, le Docteur-Ingénieur TERRA de m'avoir proposé ce sujet et pour tous ses précieux conseils ainsi pour toute la documentation qu'il a mis à ma disposition.
- ☑ Docteur BELOUHRANI de l'honneur qu'il me fait en acceptant de présider le jury de ce mémoire.
- ☑ Monsieur BOUSBIA de l'honneur qu'il me fait en acceptant de faire partie de ce jury.
- ☑ Les enseignants, l'administration et le personnel de l'Ecole Nationale Polytechnique.



## *Dédicaces*

المدرسة الوطنية المتعددة التقنيات  
BIBLIOTHEQUE — المكتبة  
Ecole Nationale Polytechnique

A la mémoire de ma chère mère que dieu le tout puissant

l'accueille en son vaste paradis

A mon cher père et ma belle mère

A mes frères et sœurs

A ma grande famille

A mes amis et collègues

A tous ceux qui me sont chers

Je leur dédie ce modeste travail.

M.Seddik



مختص

في العشرة الأخيرة كان يبدو أن سعة وفعالية منظومات الاتصالات اللاسلكية تقترب شيئا فشيئا نحو السعة القاعدية الحديثة، غير أنه تبين أن استعمال العديد من الهوائيات في جهاز الإرسال و جهاز الاستقبال (تكنولوجيا MIMO : Multiple Input Multiple Output) يرفع نسبيا و بشكل ملحوظ الفعالية التطبيقية (النظرية) المحققة بشرط أن تكون بيئة الانتشار غنية بعوامل التبعثر (النشئك). منظومات MIMO تسمح بإرسال معلومات محددة و مستقلة عن بعضها البعض بنفس الشريط الطيفي المار و بدون إضافة طاقة جديدة. تقنية V-BLAST (Vertical Bell Labs Layered Space Time Architecture) هي واحدة من بين الوسائل الجديدة و المصروفة في المعالجة التطبيقية للإشارة التي تسمح بتحقيق السعة المرجوة، هذه التقنية طور من قبل Lucent Bell Labs.

هذه الدراسة هي محاكاة الخوارزمية V-BLAST في لغة البرمجة MATLAB وبحث و تقييم قدراتها و جداتها.

الكلمات المفتاحية

منظومات الاتصالات اللاسلكية، السعة القاعدية، منظومات MIMO، الفعالية التطبيقية، تقنية V-BLAST،  
Lucent Bell Labs، MATLAB.

### Abstract

In the past decade, it seemed that wireless communications systems would more and more approach fundamental capacity limits. However, it has been shown that using multiple antennas at the transmitter and the receiver (MIMO technology), the attainable spectral efficiency can be increased significantly. Provided that the propagation environment exhibits rich scattering, MIMO systems allow the transmission of multiple independent data streams within the same frequency band at no additional power expenditure.

One of the most prominent practical signal processing schemes realizing the promised capacity gains is V-BLAST (Vertical Bell Laboratories Layered Space-Time), which was developed by Lucent Bell Labs.

The goal of this project is to implement the V-BLAST algorithm in MATLAB, and to investigate its performance properties.

**Keywords :** wireless communications systems, fundamental capacity, MIMO, spectral efficiency, rich scattering, V-BLAST, Lucent Bell Labs, MATLAB.

### Résumé :

Dans la dernière décennie, il a semblé que les systèmes de communications sans fil s'approcheraient de plus en plus aux limites de la capacité fondamentale. Cependant, il est montré que l'utilisation des canaux multi-antennes à l'émission et à la réception (Canaux MIMO), augmente significativement l'efficacité spectrale de la transmission. A condition que l'environnement de propagation soit riche en dispersion, les systèmes MIMO permettent la transmission de séquences de données multiples et indépendantes dans la même bande de fréquence et sans augmenter la puissance d'émission.

L'un des plus connus des méthodes pratiques de traitement du signal réalisant la grande capacité des systèmes MIMO est l'architecture V-BLAST (Vertical Bell Laboratories Layered Space-Time) qui a été développée par Lucent Bell Labs.

Le but de ce projet est la simulation de l'algorithme V-BLAST sous MATLAB et l'évaluation des résultats de simulation.

**Mots clés :** communications sans fil, capacité fondamentale, Canal MIMO, efficacité spectrale, riche en dispersion, V-BLAST, Lucent Bell Labs, MATLAB.

# Sommaire



|                                                                                      |    |
|--------------------------------------------------------------------------------------|----|
| <b>Introduction générale</b> .....                                                   | 1  |
| <b>Chapitre 1 : Généralités sur la communication numérique</b> .....                 | 3  |
| 1.1 Introduction .....                                                               | 3  |
| 1.2 Historique sur la communication sans fil .....                                   | 3  |
| 1.3 Modulation .....                                                                 | 5  |
| 1.3.1 Signal passe bande .....                                                       | 6  |
| 1.3.2 Concept d'espace des signaux .....                                             | 8  |
| 1.3.3 Modulation d'amplitude sur deux porteuses en quadrature.....                   | 12 |
| 1.4 Récepteurs optimaux pour un canal AWGN.....                                      | 15 |
| 1.5 Probabilité d'erreur pour M-PSK.....                                             | 21 |
| 1.6 Probabilité d'erreur pour M-QAM.....                                             | 22 |
| <b>Chapitre 2 : Canal de communication sans fil MIMO</b> .....                       | 24 |
| 2.1 Modélisation du canal MIMO .....                                                 | 24 |
| 2.2 Modèle stochastique du canal MIMO .....                                          | 26 |
| 2.2.1 La distribution de fading de Nakagami-m .....                                  | 26 |
| 2.2.2 La distribution de fading de Rayleigh et de Rice .....                         | 27 |
| 2.3 Capacité du canal MIMO .....                                                     | 27 |
| 2.3.1 Capacité du canal SISO .....                                                   | 28 |
| 2.3.2 Capacité du canal MIMO.....                                                    | 29 |
| 2.3.3 Capacité en sortie (Outage capacity) .....                                     | 34 |
| 2.4 Evaluation de l'efficacité spectrale du canal MIMO .....                         | 35 |
| 2.5 Conclusion .....                                                                 | 38 |
| <b>Chapitre 3 : L'architecture <i>BLAST</i> (Bell-labs LAYered Space-Time)</b> ..... | 39 |
| 3.1 Introduction .....                                                               | 39 |
| 3.2 Description de l'architecture <i>BLAST</i> .....                                 | 40 |
| 3.3 Modélisation de l'architecture <i>BLAST</i> .....                                | 42 |
| 3.4 La détection V- <i>BLAST</i> .....                                               | 44 |
| 3.4.1 Notations .....                                                                | 44 |

|                                                       |                                                                                                           |           |
|-------------------------------------------------------|-----------------------------------------------------------------------------------------------------------|-----------|
| 3.4.2                                                 | Modèle de base du système V-BLAST .....                                                                   | 45        |
| 3.4.3                                                 | L'algorithme de détection ZF V-BLAST (Zero Forcing V-BLAST) .....                                         | 46        |
| 3.4.3.1                                               | Les étapes de l'algorithme ZF V-BLAST .....                                                               | 47        |
| 3.4.3.2                                               | Détermination de vecteur de mise à zéro ( <i>Nulling Vector</i> ).....                                    | 48        |
| 3.4.3.3                                               | Ordre optimal et sous optimal de la détection ZF V-BLAST.....                                             | 49        |
| 3.4.3.4                                               | Forme compacte de l'algorithme complet ZF V-BLAST.....                                                    | 50        |
| 3.4.3.5                                               | Complexité de Calcul et Stabilité Numérique.....                                                          | 50        |
| 3.4.4                                                 | L'algorithme MMSE V-BLAST (Minimum Mean Square Error).....                                                | 51        |
| 3.5                                                   | Conclusion .....                                                                                          | 52        |
| <br><b>Chapitre 4 : Simulation et résultats</b> ..... |                                                                                                           | <b>53</b> |
| 4.1                                                   | Introduction .....                                                                                        | 53        |
| 4.2                                                   | Résultats de simulation .....                                                                             | 54        |
| 4.2.1                                                 | L'efficacité de l'algorithme ZF V-BLAST .....                                                             | 54        |
| 4.2.2                                                 | Comparaison entre le codage binaire naturel et le codage de<br>Gray utilisés à la technique V-BLAST ..... | 56        |
| 4.2.3                                                 | L'effet de la diversité spatiale sur la performance de la transmission .....                              | 57        |
| 4.2.4                                                 | Inconvénient de l'algorithme ZF V-BLAST sans ordre optimal de détection.....                              | 58        |
| 4.2.5                                                 | Utilisation de la technique d'ordre optimal ( <i>Ordering</i> ).....                                      | 60        |
| 4.2.6                                                 | Utilisation de l'algorithme MMSE V-BLAST .....                                                            | 61        |
| 4.3                                                   | Conclusion .....                                                                                          | 63        |
| <br><b>Conclusion générale</b> .....                  |                                                                                                           | <b>64</b> |
| <b>Annexe : Programmes MATLAB</b>                     |                                                                                                           |           |
| <b>Bibliographie</b>                                  |                                                                                                           |           |

## Introduction générale

المدرسة الوطنية المتعددة التقنيات  
المكتبة — BIBLIOTHEQUE  
Ecole Nationale Polytechnique

Le domaine de la communication sans fil et des réseaux a fait une croissance explosive. La demande et l'achat de téléphones cellulaires sont prévus pour bientôt excéder l'achat et l'utilisation de téléphone traditionnel fixe. Le marché des dispositifs de communication sans fil connaît un développement continu. Avec cette croissance rapide vient la demande du consommateur pour plus et meilleures applications qui améliorent la performance et augmentent le débit de transmission. Toutes ces améliorations doivent être accomplies sous un nombre de contraintes. Le canal sans fil est par sa nature aléatoire et imprédictible et en général la performance d'un dispositif est plus moins sur un canal sans fil qu'un canal câblé. Le spectre ou la largeur de bande disponible au fournisseur de service sont souvent limités. Aussi, les exigences de puissance signifient que les dispositifs doivent utiliser de faible puissance possible pour les conserver et augmenter leurs fiabilités et pour tenir également les produits plus petits. Les concepteurs des systèmes de communication sans fil font face à un défi à deux parties, l'augmentation de débit des données et l'amélioration de la performance en maintenant la largeur de bande ou la puissance constante ou en faisant une légère augmentation.

Les systèmes sans fil se développent rapidement pour transmettre la voix, les données et les services multimédia de messagerie. Ces services exigent des canaux sans fil fiables qui ont des grandes capacités. Les systèmes qui communiquent sur un canal à une seule entrée et une seule sortie SISO (Single Input Single Output) ont des capacités limitées. De plus, dans quelques situations, la communication sur un canal SISO n'est pas fiable du fait au fading du multitrajets. Des recherches dans la théorie de l'information ces dernières années [Fos96], [Fos98] ont montré que de très hautes capacités peuvent être obtenues en utilisant plusieurs antennes à l'émetteur et au récepteur d'un système sans fil. Les résultats de ces recherches ont mené au développement d'une nouvelle architecture de transmission-réception appelée BLAST (Bell Labs Layered Space-Time Architecture) [Fos96] qui a atteint des débits et des performances non réalisables en utilisant des techniques traditionnelles.

BLAST est la première architecture proposée par Foschini [Fos96] pour objectif de réaliser les prédictions de capacité faites par l'analyse théorique de l'information des canaux MIMO. Plus tard dans [Fos98], les auteurs ont proposé une architecture modifiée qui était



plus pratiquement réalisable. Cette nouvelle architecture a été appelée V-BLAST (Vertical BLAST).

L'architecture V-BLAST a été inventée comme un moyen de constituer la diversité spatiale d'antennes en utilisant plusieurs antennes à l'émission et à la réception. C'est une technique pour améliorer la performance de taux d'erreur sans exiger une grande augmentation de la largeur de bande ou la conception des récepteurs excessivement complexes. Pour faciliter une compréhension de l'architecture V-BLAST il est utile de la simuler pour obtenir des résultats confirmant sa performance et validant les prédictions théoriques.

Le but de ce Projet est l'évaluation de la performance de V-BLAST. Au début, il sera supposé qu'on connaît exactement le canal au récepteur et l'architecture simulée de ce point de vue.

Ce mémoire est divisé en quatre chapitres :

Le premier chapitre présente des généralités sur la communication numérique, il contient un bref historique sur la communication sans fil, un rappel sur la modulation QAM et quelques concepts sur l'espace des signaux et les récepteurs optimaux pour un canal AWGN.

Le deuxième chapitre étudie les canaux MIMO et ses performances, il donne une description détaillée de la capacité d'un canal MIMO et de ses grands avantages.

Le troisième chapitre est consacré à l'architecture BLAST en détaillant plus précisément celle appelée V-BLAST. Nous présentons les deux variétés de l'architecture BLAST, D-BLAST (Diagonal BLAST) et V-BLAST (Vertical BLAST) et nous faisons ainsi une étude algorithmique des deux algorithmes de détection ZF (Zero Forcing) et MMSE (Minimum Mean Square Error) appliquées à l'architecture V-BLAST.

Le quatrième chapitre donne les résultats de simulation qui permettent l'évaluation de la performance de détection à partir de taux d'erreur par bit (BER Bit Error Rate) obtenu pour chaque simulation.

Enfin, nous donnons une conclusion générale sur ce travail.

## Chapitre 1

# Généralités sur la communication numérique

## 1.1 Introduction

Dans ce premier chapitre nous présentons un bref historique sur la communication sans fil. Nous citons également des généralités sur la communication numérique comme la représentation des signaux passe bande, le concept d'espace des signaux et la modulation numérique. Nous n'oublions pas la présentation et la description des récepteurs optimaux pour un canal AWGN, ainsi que l'évaluation de ses performances de détection à partir des formules théoriques de la probabilité d'erreur de détection.

## 1.2 Historique sur la communication sans fil

En 1876, le savant canadien Graham Bell invente le téléphone fixe, premier moyen de communication moderne. Onze ans plus tard, le physicien allemand Heinrich Hertz découvre les ondes radio. La première transmission radio fut réalisée en 1896, où le physicien italien Guglielmo Marconi a transmis le premier message sans fil. Aux Etats-Unis, les radiotéléphones mobiles sont en service depuis le début des années 20, époque où le Département de police de Détroit a commencé à s'en servir à bord de ses véhicules. L'invention britannique du radar (1935) a permis de franchir un pas important dans la compréhension de la partie des fréquences radioélectriques élevées du spectre électromagnétique. Au cours de la deuxième guerre mondiale, la radiotéléphonie a été perfectionnée à des fins militaires.

En 1946, le premier service radiotéléphonique mobile commercial a été testé par Bell Systems à St-Louis et a été introduit dans plusieurs villes des Etats-Unis à la fin des années 40. Ces services étaient destinés à relier au réseau terrestre public des usagers mobiles se trouvant à bord de véhicules. Le système reposait sur un seul émetteur puissant qui se servait de signaux fondés sur la modulation de fréquence susceptible de couvrir une zone dont le rayon à partir de l'émetteur pouvait aller jusqu'à 80 Km. Ces téléphones mobiles MF utilisaient la bande de fréquences des 120 kHz pour transmettre la voix sur une largeur de bande de 3 kHz seulement. Dans les années 60, un nouveau système téléphonique mobile a été lancé par Bell Systems; il s'agissait du service téléphonique mobile amélioré

(IMTS, Improved Mobile Telephone Service), qui apportait de nombreuses améliorations telles que la numérotation directe et la largeur de bande de 25-30 kHz du canal MF.

Les premiers systèmes cellulaires analogiques sont nés à la fin des années 60 et au début des années 70. Le terme « cellulaire » désigne la division de la zone de couverture en de nombreuses petites zones appelées cellules, chacune étant desservie par un émetteur et un récepteur de faible puissance. Les systèmes cellulaires analogiques n'étaient pas fondés sur un nouveau type de technologie mais sur la prise de conscience du fait que les techniques IMTS existantes pouvaient être organisées à une plus large échelle. Le même système MF analogique employé depuis la fin de la deuxième guerre mondiale pouvait maintenant être utilisé pour les communications vocales. Deux améliorations majeures ont permis de concrétiser le concept du cellulaire. La première a été l'invention du microprocesseur au début des années 70, qui a facilité la mise en œuvre des algorithmes de commande dans l'exploitation d'un système cellulaire et a permis d'accroître la complexité de ces algorithmes. Cette invention a rendu aussi possible la miniaturisation des combinés. La deuxième avancée technologique a été la liaison de commande numérique entre le mobile et la station de cellule.

Les travaux concernant les systèmes cellulaires numériques de deuxième génération ont débuté à la fin des années 80. Dans ces systèmes, non seulement les commandes mais aussi les signaux vocaux étaient numérisés. Les progrès technologiques tel qu'une meilleure qualité vocale ont été importants tant pour les techniciens que pour les clients. Le nouveau service numérique a accru la capacité et considérablement réduit le coût des communications hertziennes par rapport à celui des systèmes analogiques.

A la fin des années 80, l'UIT a commencé à élaborer des systèmes de troisième génération (3G) en portant son attention sur la nécessité d'harmoniser, sur le plan mondial, les normes relatives au spectre des fréquences radioélectriques et aux interfaces radioélectriques. Compte tenu de la croissance impressionnante du cellulaire mobile, les Membres de l'Union ont dû entreprendre d'élaborer des normes en ce qui concerne les systèmes 3G.

L'UIT a pour objectif d'établir une norme mondiale relative aux systèmes 3G dans le cadre d'une initiative appelée IMT-2000 (International Mobile Telecommunications—2000). Le système IMT-2000 est plus qu'un système cellulaire amélioré. Il prévoit l'intégration de différentes zones de couvertures de service – systèmes terrestres macrocellulaires, microcellulaires et picocellulaires, systèmes sans fil, systèmes à accès hertzien et systèmes à satellites—, ce qui permet d'assurer un service véritablement mondial, voir Figure 1.1. Les IMT –2000 devraient assurer une plateforme pour la répartition des services pour lesquels

il y a eu convergence (mobiles ou fixes, voix ou données, télécommunications, contenu ou calcul). [Dav00] [UMT99]

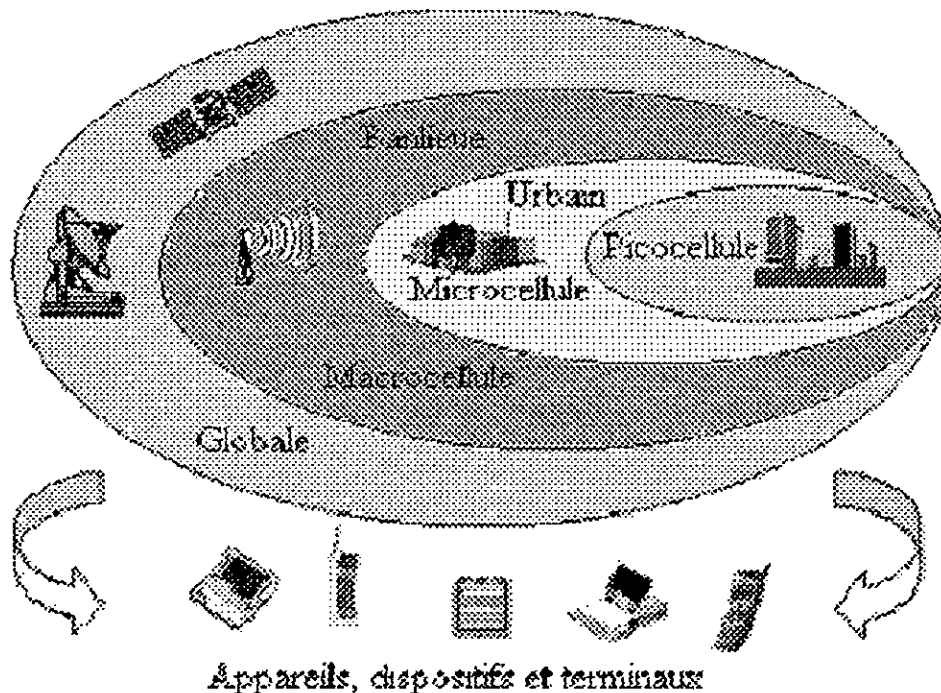


Figure 1.1 : Architecture du réseau IMT-2000

### 1.3 Modulation [Bak01]

La plupart des systèmes de communication actuels sont numériques. Ils fonctionnent directement sur des données numériques sortant de quelques dispositifs de calcul dont les entrées sont des signaux analogiques qui sont échantillonnés et quantifiés, la sortie de ces dispositifs sont des données numériques à transmettre sur un canal de communication. La plupart des canaux par lesquels ces données doivent être transmises, comme les fils téléphoniques, les fibres optiques, l'espace libre, ... etc. ont certaines caractéristiques et contraintes qui forcent les données transmises d'occuper une bande particulière de fréquences, ou une largeur de bande. Pour convertir les données numériques disponibles à la source en un signal qui peut être efficacement transmis par le canal, les données passent en un processus appelé modulation. La modulation a pour rôle de changer (ou bien décaler) le spectre de fréquences des signaux numériques, ou signaux bande de base, d'une telle façon pour créer un signal passe bande. Pour la majorité des systèmes, la fréquence porteuse est déterminée par une onde sinusoïdale qui est modulée par le signal bande de base pour produire le signal transmis (modulé). Pour se faire on prend un signal bande de base, le filtre et le mélanger ensuite avec une porteuse sinusoïdale. La porteuse sinusoïdale, produite par un oscillateur,

sert à changer la fréquence du signal bande de base en un spectre approprié pour la transmission sur le canal. La figure 1.2 représente un système de communication numérique simplifié sans codage de source ou du canal.

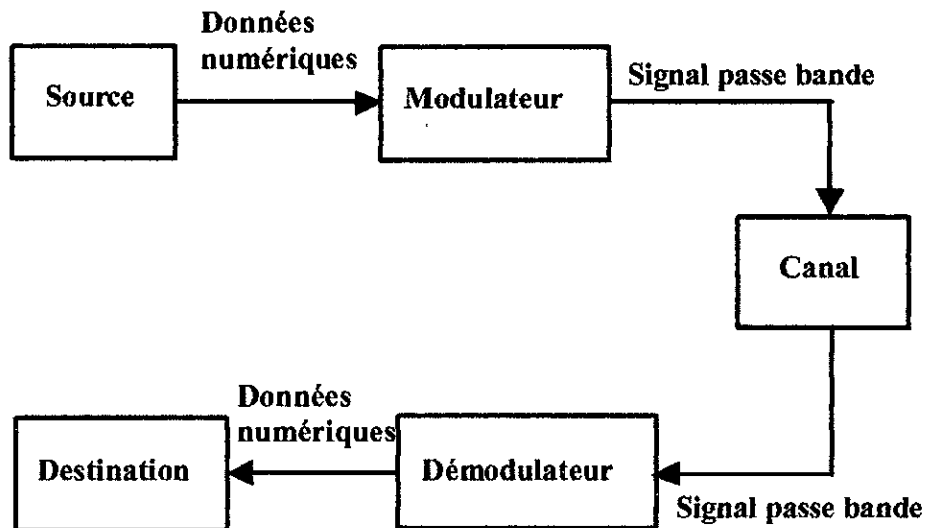


Figure 1.2 Structure générale d'un système de communication numérique simplifié

### 1.3.1 Signal passe bande

Il est extrêmement important d'être capable de comparer et de spécifier les avantages des différents systèmes et techniques de communication pour pouvoir choisir le meilleur système pour une application donnée. Il serait commode si nous pouvions comparer deux systèmes différents qui fonctionnent aux fréquences différentes sans devoir représenter les effets présentés par les différentes fréquences porteuses. Ce but peut être accompli si nous représentons des signaux passe bande comme des signaux équivalents passe bas qui conservent toutes les propriétés nécessaires pour évaluer la performance du système. Le spectre d'un signal passe bande sera contenu dans une bande relativement étroite de fréquences qui sont centrées autour de la fréquence porteuse  $\mp f_c$ . La figure 1.3 représente un exemple du spectre d'un signal passe bande arbitraire  $s(t)$ .

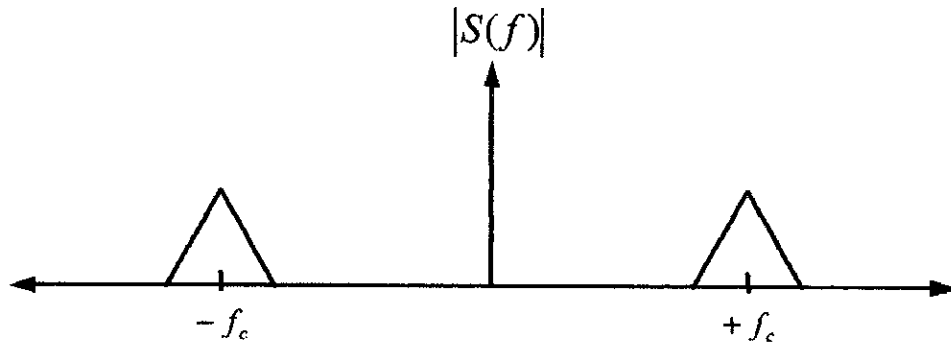


Figure 1.3 L'amplitude d'un signal passe bande dans le domaine fréquentiel

D'abord nous prenons la partie positive du spectre de fréquence du signal  $S(f)$  et construisons le signal "analytique" :

$$S_+(f) = 2u(f)S(f) \quad (1.1)$$

Où  $u(f)$  est l'échelon unité et  $S(f)$  est la transformée de Fourier de  $s(t)$ . Pour obtenir l'expression correspondante dans le domaine temporel nous pouvons prendre la transformée de Fourier inverse comme suit :

$$s_+(t) = F^{-1}\{S_+(f)\}$$

$$s_+(t) = F^{-1}\{2u(f)\} * F^{-1}\{S(f)\}$$

Où  $*$  représente le produit de convolution

$$s_+(t) = \left( \delta(t) + \frac{j}{\pi t} \right) * (s(t)) \quad (1.2)$$

$$s_+(t) = s(t) + \left( s(t) * \frac{j}{\pi t} \right)$$

Si on met  $\hat{s}(t) = \frac{1}{\pi t} * s(t)$ ,

Où  $\hat{s}(t)$  représente la transformée de Hilbert de  $s(t)$ ,

Alors,

$$s_+(t) = s(t) + j\hat{s}(t) \quad (1.3)$$

Comme  $s_+(t)$  est toujours un signal passe bande, nous pouvons le transformer en fréquence pour obtenir un signal équivalent passe bas  $s_l(t)$  qui peut être décrit par l'équation suivante :

$$s_l(t) = s_+(t)e^{-j2\pi f_c t} \quad (1.4)$$

Nous pouvons maintenant représenter  $s(t)$  par trois notations différentes sont :

- La notation en enveloppe Complexe
- La notation en Quadrature
- La notation en amplitude et phase

Dans la notation en enveloppe Complexe :

$$s(t) = \text{Re} \left[ s_l(t) e^{+j2\pi f_c t} \right] \quad (1.5)$$

où  $s_l(t)$  est un signal passe bas complexe et  $\text{Re}[\bullet]$  est la partie réelle du signal complexe.

Dans la notation en quadrature :

$$s(t) = x(t) \cos(2\pi f_c t) - y(t) \sin(2\pi f_c t) \quad (1.6)$$

Où  $x(t)$  et  $y(t)$  sont des signaux réels passe bas appelés les composants en-phase et en quadrature de phase de  $s(t)$ , respectivement.

Enfin, la notation en amplitude et phase :

$$s(t) = a(t) \cos(2\pi f_c t + \theta(t)) \quad (1.7)$$

Où  $a(t)$  et  $\theta(t)$  sont des signaux réels passe bas représentant l'amplitude et la phase du signal  $s(t)$ . Ces trois notations sont liées l'une à l'autre et on peut passer d'une représentation à l'autre en utilisant les équations suivantes :

$$x(t) = a(t) \cos(\theta(t)) \quad (1.8a)$$

$$y(t) = a(t) \sin(\theta(t)) \quad (1.8b)$$

$$a(t) = \sqrt{x^2(t) + y^2(t)} \quad (1.8c)$$

$$\theta(t) = \tan^{-1} \left[ \frac{y(t)}{x(t)} \right] \quad (1.8d)$$

$$s_l(t) = x(t) + jy(t) \quad (1.8e)$$

### 1.3.2 Concept d'espace des signaux

Pour développer une méthode générale pour la génération, l'analyse et la réception des signaux, il est nécessaire de construire une structure où les signaux de différents types peuvent être représentés d'une façon similaire. L'idée d'espace des signaux est introduite pour l'accomplir. Pour comprendre les espaces des signaux et leur utilité on doit d'abord définir

plusieurs termes et conditions. Un signal, que nous appellons  $x(t)$ , est une fonction de temps définie sur l'intervalle  $[a, b]$ . Le produit scalaire de deux signaux  $x_1(t)$  et  $x_2(t)$  est noté  $\langle x_1(t), x_2(t) \rangle$ , où :

$$\langle x_1(t), x_2(t) \rangle = \int_a^b x_1(t) x_2^*(t) dt \quad (1.9)$$

Deux signaux  $x_1(t)$  et  $x_2(t)$  sont orthogonaux si :

$$\langle x_1(t), x_2(t) \rangle = 0 \quad (1.10)$$

La norme d'un signal  $x(t)$  est notée  $\|x(t)\|$ , où

$$\|x(t)\| = [\langle x(t), x(t) \rangle]^{1/2}$$

$$\|x(t)\| = \sqrt{\int_a^b |x(t)|^2 dt} \quad (1.11)$$

L'énergie d'un signal  $x(t)$  est décrite par l'équation :

$$E_s = \|x(t)\|^2$$

$$E_s = \int_a^b |x(t)|^2 dt \quad (1.12)$$

Un signal  $x(t)$  est considéré normalisé s'il satisfait les équations suivantes :

$$\|x(t)\| = 1$$

$$E_s = \|x(t)\|^2 = (1)^2 = 1 \quad (1.13)$$

Un signal  $x(t)$  peut être exprimé comme suit :

$$x(t) = \sum_{i=1}^k w_i f_i(t) \quad (1.14)$$

Où  $w_i$  sont des poids ou des coefficients constants et  $f_i(t)$  sont des fonctions de base d'un repère orthonormal de fonctions de base. La dimension de la base des signaux,

$$S = \{s_1(t), s_2(t), \dots, s_{M-1}(t), s_M(t)\},$$

Composée des fonctions de base orthonormales,

$$F = \{f_1(t), f_2(t), \dots, f_{K-1}(t), f_K(t)\},$$



est égale à  $K$ , le nombre des fonctions de base.

L'ensemble des fonctions de base  $F$  peut être considéré comme une base complète orthonormale pour un ensemble de signaux  $S$  s'il satisfait trois conditions :

- 1) Condition complète : Tous les signaux de l'ensemble  $S$  peuvent être écrits comme une combinaison linéaire de fonctions de base de l'ensemble  $F$ .
- 2) Condition orthogonale : Toutes les paires de fonctions de base sont orthogonales tel que :

$$\langle f_i(t), f_j(t) \rangle = 0 \quad \forall_{i,j} \text{ Pour } i \neq j$$

- 3) Condition normale : Toutes les fonctions de base sont normalisées tel que :

$$\|f_i(t)\| = 1 \quad \forall_i$$

Le concept d'utiliser des fonctions de base pour représenter n'importe quel signal donné dans un ensemble de signaux mène à quelques simplifications qui le rendent commode pour l'analyse des systèmes.

Maintenant, n'importe quel signal donné peut être représenté comme suit :

$$s_i(t) = \sum_{k=1}^K w_k f_k(t)$$

Pour différents signaux, les coefficients  $w_i$  varieront mais les fonctions de base restent constantes. Cela implique que l'information appropriée pour distinguer un signal d'un autre contenue dans les coefficients  $w_i$ . Donc, nous pouvons représenter les différents signaux par un vecteur qui contient les coefficients des fonctions de base, pour que  $s_i(t)$  soit maintenant équivalent de :

$$S = [w_1 \quad w_2 \quad \dots \quad w_{K-1} \quad w_K]$$

Ayant établi l'équivalence entre un signal et sa représentation vectorielle, un espace de signaux peut maintenant être créé à partir des vecteurs d'un ensemble de signaux. Par exemple nous développons la représentation d'espace de signaux pour les signaux modulés en phase (PSK). Dans le PSK la forme générale pour n'importe quel signal est :

$$s_m(t) = \sqrt{\frac{2E_s}{T}} g(t) \cos\left(\frac{2\pi}{M}(m-1)\right) \cos(2\pi f_c t) - \sqrt{\frac{2E_s}{T}} g(t) \sin\left(\frac{2\pi}{M}(m-1)\right) \sin(2\pi f_c t) \quad (1.15)$$

où  $g(t)$  est l'impulsion de base et  $\frac{2\pi}{M}(m-1)$ ,  $m=1,2,\dots,M$ , est la phase.

Supposons que la base de ces signaux consiste en deux fonctions de base :

$$\begin{aligned}
 f_1(t) &= \sqrt{\frac{2}{T}} g(t) \cos(2\pi f_c t) \\
 f_2(t) &= -\sqrt{\frac{2}{T}} g(t) \sin(2\pi f_c t)
 \end{aligned}
 \tag{1.16}$$

Cela signifie que chaque signal  $s_m(t)$  sera représenté comme un vecteur de deux dimensions de la forme :

$$\begin{aligned}
 s_m &= [s_{m1} \quad s_{m2}] \\
 s_m &= \left[ \sqrt{E_s} \cos\left(\frac{2\pi}{M}(m-1)\right) \quad \sqrt{E_s} \sin\left(\frac{2\pi}{M}(m-1)\right) \right]
 \end{aligned}
 \tag{1.17}$$

où  $m = 1, 2, \dots, M$ .

Dans le cas où  $M = 2$  ce système a une modulation appelée "Binary Phase Shift Keying (BPSK)" et quand  $M = 4$  le système a une modulation appelée "Quadrature Phase Shift Keying" (QPSK), ou 4PSK. Le repère d'espace des signaux pour ces deux valeurs de  $M$  est représenté sur la figure 1.4.

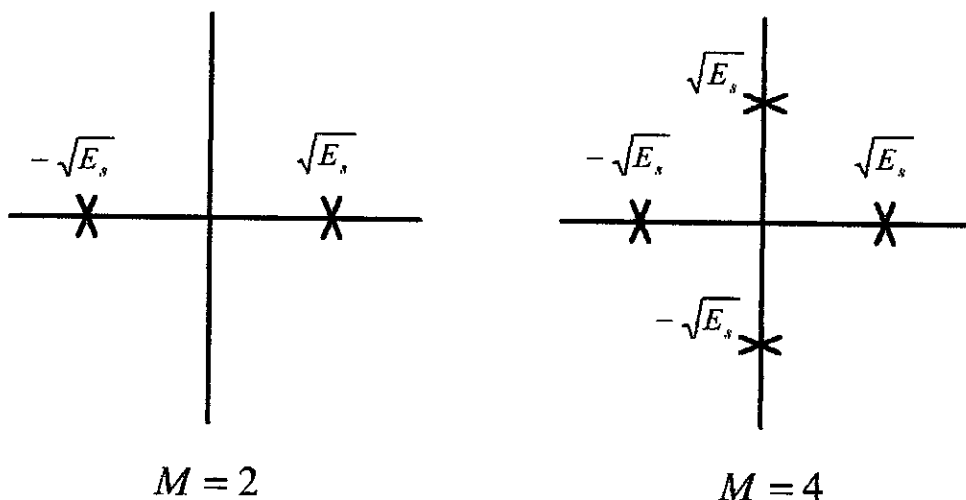


Figure 1.4

La distance Euclidienne entre deux signaux,  $x_i(t)$  et  $x_j(t)$  est définie par :

$$d_{ij} = \|x_i - x_j\|
 \tag{1.18}$$

où  $x_i$  et  $x_j$  sont les représentations vectorielle des deux signaux respectivement.

Donc, la distance Euclidienne entre deux signaux n'est rien que la distance entre les points tracés sur un diagramme d'espace des signaux.

Pour le cas de  $M$ -PSK cette distance peut être écrite :

$$d_{ij} = \sqrt{2E_s \left( 1 - \cos\left(\frac{2\pi}{M}(i-j)\right) \right)} \quad \forall_{i,j} \quad i \neq j \quad (1.19)$$

La distance Euclidienne minimale, notée par  $d_{\min}$ , est la distance la plus petite entre deux paires de points des signaux, ou dans le cas de PSK c'est la distance entre des points de signal adjacents.

Alors la distance minimale pour le PSK est :

$$d_{\min} = \sqrt{2E_s \left( 1 - \cos\left(\frac{2\pi}{M}\right) \right)} \quad (1.20)$$

### 1.3.3 Modulation d'amplitude sur deux porteuses en quadrature (QAM)

C'est un type de modulation particulier, qui consiste à prendre deux porteuses de même fréquence, mais déphasées l'une par rapport à l'autre de  $90^\circ$  (quadrature). Ces porteuses sont ensuite modulées en amplitude (PAM), puis additionnées l'une avec l'autre.

Le signal se présente sous la forme :

$$s(t) = x(t) \cdot \cos(2\pi f_c t) - y(t) \cdot \sin(2\pi f_c t)$$

$x$  et  $y$  sont les deux séquences de données :

$$\begin{aligned} x(t) &= \sum_k a_k g(t - kT), \\ y(t) &= \sum_k b_k g(t - kT) \end{aligned} \quad (1.21)$$

Où  $a_k$  et  $b_k$  sont les séquences de bits et  $g(t)$  l'impulsion de base qui vaut 1 pendant une période d'horloge, 0 ailleurs.

L'expression précédente peut se mettre aussi sous la forme :

$$s(t) = \sum_k A_k \cos(2\pi f_c t + \Phi_k) \cdot g(t - kT) \quad (1.22)$$

$$\text{avec} \quad \begin{cases} A_k = \sqrt{a_k^2 + b_k^2} \\ \Phi_k = \arctg \frac{b_k}{a_k} \end{cases} \quad (1.23)$$

Si  $a_k$  et  $b_k$  sont des bits notés +1 et -1 on obtient la configuration décrite dans le tableau ci-dessous.

Si  $a_k$  et  $b_k$  peuvent avoir chacun 4 états (notés  $\pm 1$  et  $\pm 3$ ), c'est à dire sont créés à partir d'un bloc de 4 bits, la configuration possède 16 états (16-QAM).

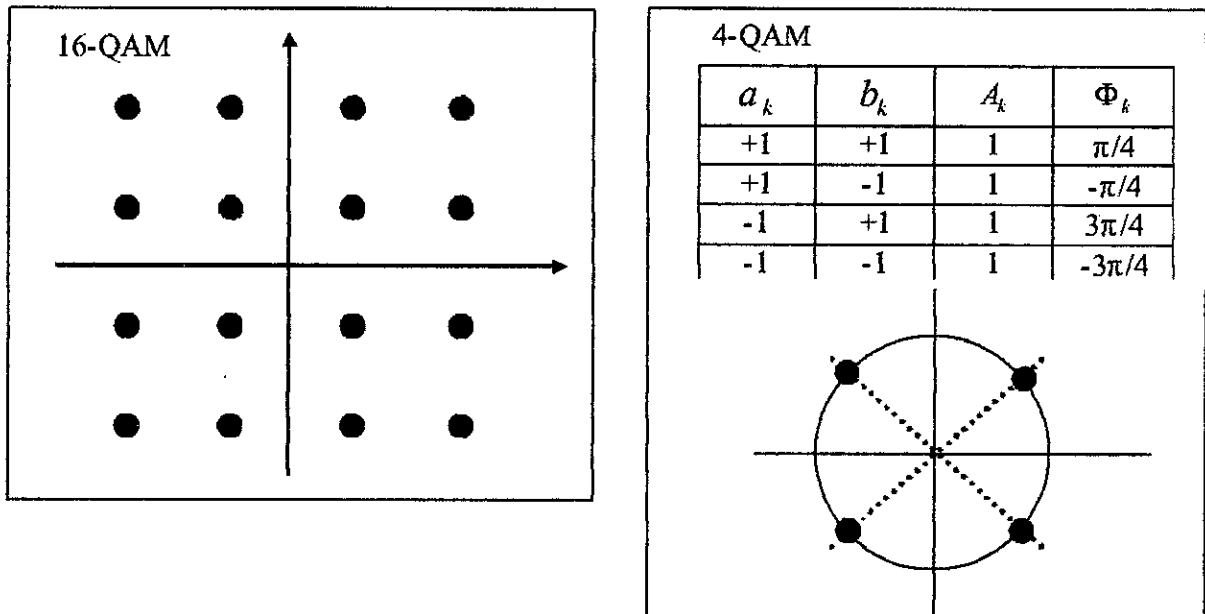


Figure 1.5

La modulation à plus de 4 états de phase est rarement utilisée seule, pour limiter le taux d'erreur on fait intervenir l'amplitude. Une modulation à 8 états peut par exemple associer 4 états de phase et 2 niveaux d'amplitude. Chaque symbole représente alors 3 bits, pour une bande passante donnée le débit est trois fois plus rapide qu'en PSK2. Dans le plan complexe les états peuvent avoir des positions très variées. La figure 1.6 représente deux exemples pour une modulation à 16 états (chaque symbole est codé sur 4 bits). On notera la différence entre une modulation 16-QAM et une modulation amplitude-phase.

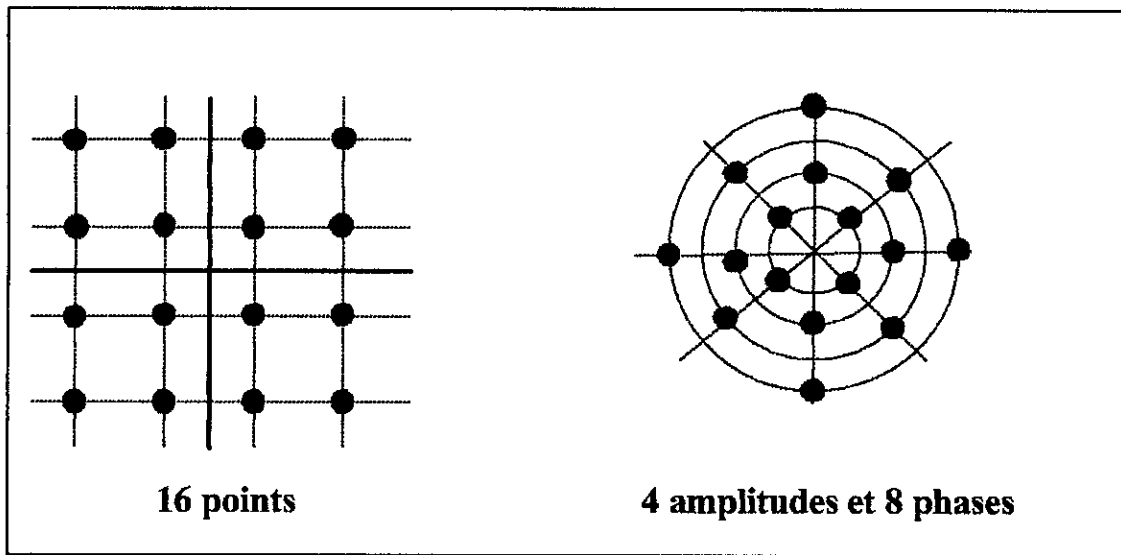


Figure 1.6 Modulation 16-QAM et Amplitude Phase à 16 états

Pour une QAM à  $2^n$  états le débit de bit est d'abord organisé en blocs de  $n$  bits lus à une cadence  $n$  fois plus lente que la fréquence bit d'entrée. Un circuit détermine alors les valeurs correspondantes de  $x$  et  $y$ , module en amplitude en conséquence les deux porteuses en quadrature et additionne les deux termes pour créer le symbole de sortie, voir figure 1.7.

Les deux voies en quadrature sont désignées le plus souvent par les lettres I et Q.

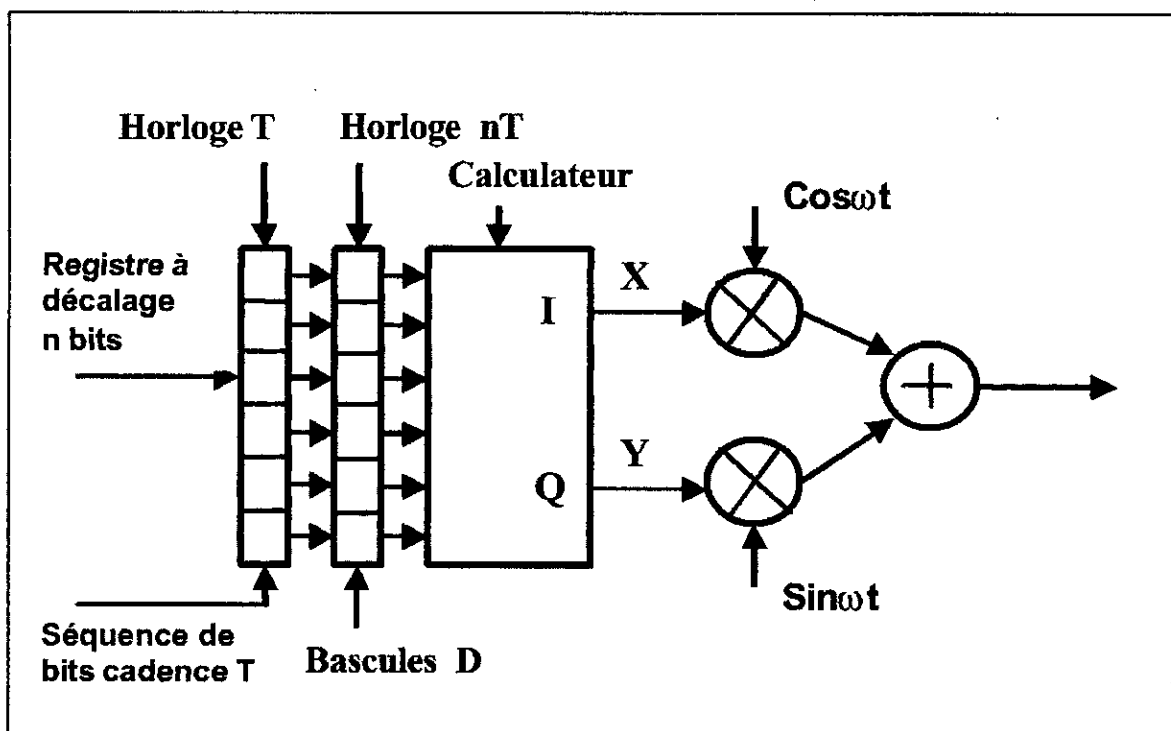


Figure 1.7 Modulation QAM à  $2^n$  états

### 1.4 Récepteurs optimaux pour un canal AWGN

Un récepteur peut être séparé en deux blocs, un démodulateur et un détecteur. Dans cette section nous examinerons deux types de démodulateurs, un basé sur l'utilisation des corrélateurs et l'autre sur les filtres adaptés. Nous discuterons également un détecteur optimal utilisant la règle de décision en sens de Maximum à posteriori (MAP) ou celle en sens de Maximum de Vraisemblance (ML).

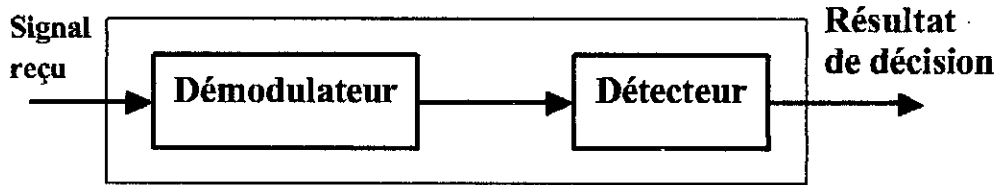


Figure 1.8 Diagramme d'un récepteur

Supposons que nous transmettons un signal  $s(t)$  de l'ensemble  $S$

$$S = \{s_1(t), s_2(t), \dots, s_{M-1}(t), s_M(t)\},$$

$s(t)$  s'écrit dans la base  $F = \{f_1(t), f_2(t), \dots, f_{K-1}(t), f_K(t)\}$  comme suit :

$$s_m(t) = \sum_{k=1}^K s_{mk} f_k(t) \quad (1.24)$$

Où

$$s_{mk} = \int_0^T s_m(t) f_k(t) dt$$

Le signal disponible au récepteur sera le signal original bruité par un bruit blanc additif Gaussien (AWGN). Le signal reçu sera donc :

$$r(t) = s(t) + n(t)$$

Où  $n(t)$  est un bruit blanc additif Gaussien (AWGN) de moyenne nulle et de densité spectrale de puissance de  $\frac{N_0}{2}$  et  $s(t)$  est le signal transmis.

Le bruit peut être représenté par :

$$n(t) = n'(t) + \sum_{k=1}^K n_k f_k(t) \quad (1.25)$$

Où

$$n_k = \int_0^T n(t) f_k(t) dt$$

Le signal  $n'(t)$  est non corrélé avec tous les signaux possibles transmis, donc il n'affecte pas la performance et peut être ignoré dans les équations qui suivent. Le signal reçu peut maintenant être représenté comme suit :

$$r(t) = \sum_{k=1}^K s_{mk} f_k(t) + \sum_{k=1}^K n_k f_k(t)$$

$$r(t) = \sum_{k=1}^K (s_{mk} + n_k) f_k(t) \quad (1.26)$$

$$r(t) = \sum_{k=1}^K r_k f_k(t)$$

Où  $r_k = s_{mk} + n_k$

On a deux types de démodulateurs optimaux :

1) Le démodulateur corrélateur qui projette le signal reçu  $r$  sur chaque fonction de base pour obtenir les composants de  $r$  ( $r = [r_1 r_2 \dots r_K]$ ), ce démodulateur est représenté sur la figure 1.9.

2) Le démodulateur à filtres adaptés (Figure 1.10) qui est équivalent au premier mais qui utilise des filtres adaptés, La réponse impulsionnelle de chaque filtre est tout simplement la fonction de base inversée dans le temps et décalée par une période  $T$ , c'est-à-dire :

$$h_k(t) = f_k(T - t)$$

Pour que la sortie du filtre soit identique à celle du corrélateur, elle doit être échantillonnée au temps  $t = T$ .

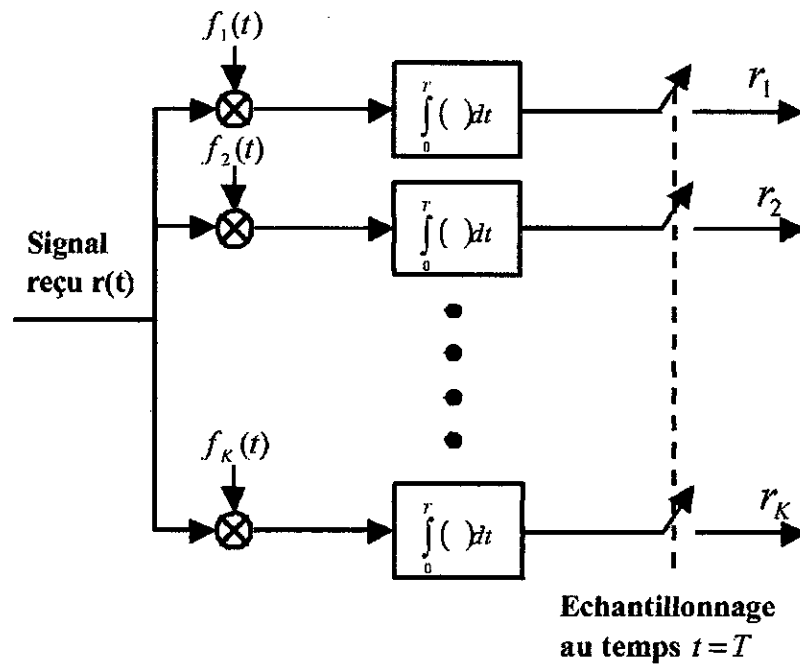


Figure 1.9 Démodulateur utilisant des corrélateurs

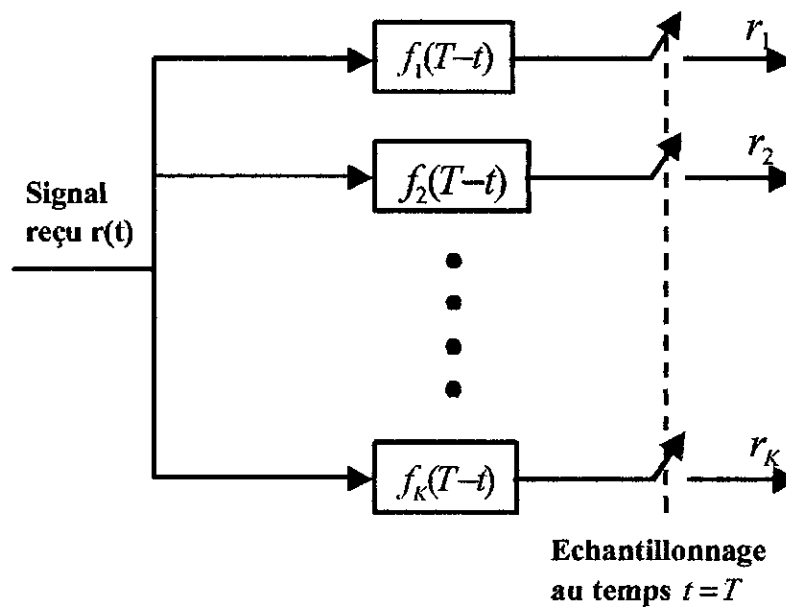


Figure 1.10 Démodulateur utilisant des filtres adaptés

Après que le signal a été reçu et démodulé, le vecteur  $r$  peut être utilisé pour calculer le signal estimé du signal transmis  $s$ . Si la probabilité d'une erreur d'estimation d'un symbole est :

$$P_s = \Pr[\hat{s} \neq s] \tag{1.27}$$



Et la probabilité qu'un signal donné  $s_i(t)$  transmis est  $p_i$ ,  $i=1,2,\dots,M$ , alors le but d'un récepteur optimal est de choisir une estimation qui minimise  $P_s$ . Cela peut être accompli en choisissant  $\hat{s} = s_m$  qui vérifie :

$$\Pr(s_m|r) > \Pr(s_i|r) \quad \forall i \neq m$$

En utilisant la règle de Bayes

$$\Pr(s_m|r) = \frac{\Pr(r|s_m)\Pr(s_m)}{\Pr(r)}$$

On obtient

$$\begin{aligned} \Pr(r|s_m)\Pr(s_m) &> \Pr(r|s_i)\Pr(s_i) \\ \Pr(r|s_m)p_m &> \Pr(r|s_i)p_i \quad \forall i \neq m \end{aligned} \quad (1.28)$$

L'équation 1.28 désigne la règle de décision en sens de maximum de probabilité à posteriori (MAP).

Si les signaux transmis sont équiprobables la règle de décision devient la décision en sens du maximum de vraisemblance ML [Pro00].

Le détecteur décide en faveur de  $\hat{s} = s_m$  si :

$$\Pr(r|s_m) > \Pr(r|s_i) \quad \forall i \neq m \quad (1.29)$$

Le vecteur  $S$  contient des coefficients scalaires qui sont conditionnellement déterministes, donc il n'influe pas sur la nature aléatoire du vecteur reçu. Le vecteur de bruit  $n$  est composé par des coefficients gaussiens de moyenne nulle et de variance

$$\sigma_n^2 = \frac{N_0}{2} \quad (1.30)$$

Tous les coefficients du vecteur  $n$  sont identiquement distribués et statistiquement indépendants.

$$\begin{aligned} E[r_k|s_k] &= E[(s_k + n_k)|s_k] \\ E[r_k|s_k] &= s_k + E[n_k] \\ E[r_k|s_k] &= s_k \end{aligned} \quad (1.31)$$

où la variance de  $(r|s)$  est  $\sigma_{r|s}^2 = \sigma_n^2 = \frac{N_0}{2}$

Comme chaque élément du vecteur  $\mathbf{n}$  est une variable aléatoire gaussienne statistiquement indépendante, alors chaque composant du vecteur reçu,  $r_k$  conditionné par  $s_k$  est aussi une variable aléatoire gaussienne statistiquement indépendante avec un moyen de  $s_k$  et une variance égale à  $\frac{N_0}{2}$ .

Donc

$$p(r|s_m) = \prod_{k=1}^K p(r_k|s_{mk}) \quad (1.32)$$

Pour calculer cette probabilité, il faut connaître la densité de probabilité de Gauss qui est égale à :

$$p(x) = \frac{1}{\sqrt{2\pi\sigma}} \exp\left\{-\frac{(x-m)^2}{2\sigma^2}\right\} \quad (1.33)$$

où  $\sigma^2$  est la variance de  $x$  et  $m$  sa moyenne.

Alors

$$p(r|s_m) = \prod_{k=1}^K \frac{1}{\sqrt{\pi N_0}} \exp\left\{-\frac{(r_k - s_{mk})^2}{N_0}\right\}$$

$$p(r|s_m) = \frac{1}{(\pi N_0)^{\frac{K}{2}}} \exp\left\{-\sum_{k=1}^K \frac{(r_k - s_{mk})^2}{N_0}\right\} \quad (1.34)$$

La règle de décision MAP décide pour  $\hat{s}$  qui maximise  $p(r|s_m)$ . Cela peut être exprimée mathématiquement par :

$$\hat{s} = \arg \max_{s_m \in S} \{p_m p(r|s_m)\} \quad (1.35)$$

Où  $\hat{s}$  est le signal estimé du signal transmis et  $S$  est l'ensemble des signaux de la constellation. L'opérateur "arg max" signifie que  $\hat{s}$  est égale au signal de  $S$  qui maximise la fonction entre accolades. En remplaçant (1.34) dans (1.35), on obtient :

$$\hat{s} = \arg \max_{s_m \in S} \left\{ p_m (\pi N_0)^{-\frac{K}{2}} \exp\left\{-\sum_{k=1}^K \frac{(r_k - s_{mk})^2}{N_0}\right\} \right\}$$

On prend le logarithme Népérien et après simplification on trouve la formule suivante :

$$\hat{s} = \underset{s_m \in S}{\operatorname{argmax}} \left\{ \ln(p_m) - \frac{K}{2} \ln(\pi N_0) - \frac{1}{N_0} \sum_{k=1}^K (r_k^2 - 2s_{mk}r_k + s_{mk}^2) \right\}$$

les termes  $\frac{K}{2} \ln(\pi N_0)$  et  $r_k^2$  sont communs pour tous les signaux possiblement reçus, donc

$$\hat{s} = \underset{s_m \in S}{\operatorname{argmax}} \left\{ \ln(p_m) + \frac{2}{N_0} \sum_{k=1}^K s_{mk}r_k - \frac{1}{N_0} \sum_{k=1}^K s_{mk}^2 \right\}$$

Sachant que  $\sum_{k=1}^K s_{mk}^2 = E_m$ , l'énergie de  $s_m$ ,

$$\hat{s} = \underset{s_m \in S}{\operatorname{argmax}} \left\{ \frac{N_0}{2} \left( \ln(p_m) + \frac{2}{N_0} \sum_{k=1}^K s_{mk}r_k - \frac{E_m}{N_0} \right) \right\}$$

$$\hat{s} = \underset{s_m \in S}{\operatorname{argmax}} \left\{ \frac{N_0}{2} \ln(p_m) + \sum_{k=1}^K s_{mk}r_k - \frac{E_m}{2} \right\} \quad (1.36)$$

Cette équation sera utilisée pour construire le récepteur MAP.

On pose 
$$z_m = \sum_{k=1}^K s_{mk}r_k$$

Il est possible d'utiliser  $M$  corrélateurs au récepteur pour projeter le signal reçu sur tous les composants de  $S$ . Cependant, pour une implémentation performante il est possible de projeter le signal reçu sur les fonctions de base puis faire une multiplication matrice-vecteur pour obtenir le produit scalaire entre les vecteurs signaux  $S_i$  et le vecteur reçu  $r$ .

La multiplication matrice-vecteur donne :  $z = Sr$

Où  $S$  est une matrice dont ses lignes sont les signaux vecteurs  $S_i$

$$S = \begin{bmatrix} s_1 \\ \vdots \\ s_M \end{bmatrix}$$

La sortie du nouveau démodulateur donne les  $z_m$ ,  $m=1,2,\dots,M$ . Le récepteur doit tenir compte des probabilités des signaux transmis et leurs énergies avant de choisir la métrique la plus grande et mettre  $\hat{s}$  égale au signal correspondant. La figure 1.11 représente un diagramme d'un récepteur MAP complet utilisant  $K$  corrélateurs.

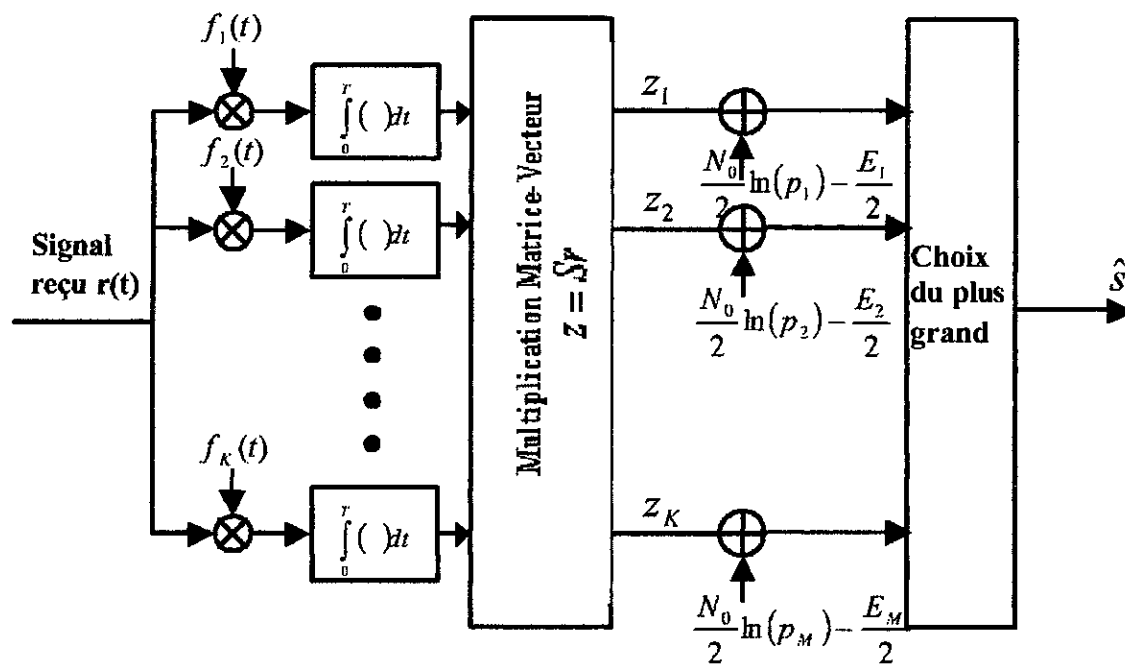


Figure 1.11 Récepteur complet réalisant la règle de décision MAP

Dans le récepteur de la figure 1.11, si les probabilités des signaux transmis sont égales, nous pouvons enlever les termes qui représentent les probabilités a priori des signaux,  $\frac{N_0}{2} \ln(p_m)$ , et créer un récepteur basé sur la règle de décision ML (Maximum de vraisemblance). Dans le cas où les énergies de chacun des signaux transmis sont égales, par exemple le M-PSK, alors nous pouvons enlever les termes  $E_m$  du récepteur.

Ainsi, pour le M-PSK avec des  $p_i$  égales, la décision est basée seulement sur les métriques  $Z_i$ .

### 1.5 Probabilité d'erreur pour M-PSK

La probabilité d'erreur symbole comme elle est indiquée précédemment (Equation 1.27) est

$$P_s = \Pr[\hat{s} \neq s]$$

La représentation vectorielle d'un signal M-PSK est donnée par la relation (1.17)

$$s_m = \left[ \sqrt{E_s} \cos\left(\frac{2\pi}{M}(m-1)\right) \quad \sqrt{E_s} \sin\left(\frac{2\pi}{M}(m-1)\right) \right]$$

Où  $E_s$  est l'énergie du signal  $s_m$ .

La probabilité d'erreur par symbole approximée pour un bruit AWGN pour des grandes valeurs de SNR calculée en [Pro00] est donnée par la formule suivante :

$$\begin{aligned} P_M &\approx 2Q\left(\sqrt{2\gamma_s} \sin \frac{\pi}{M}\right) \\ &= 2Q\left(\sqrt{2k\gamma_b} \sin \frac{\pi}{M}\right) \end{aligned} \quad (1.37)$$

Où  $\gamma_s$  est le rapport signal sur bruit par symbole

et  $\gamma_b$  est le rapport signal sur bruit par bit tel que  $\gamma_s = k\gamma_b$  et  $k = \log_2 M$

et  $Q$  est la fonction d'erreur donnée en [Pro00] par :

$$\begin{aligned} Q(x) &= \frac{1}{\sqrt{2\pi}} \int_x^{\infty} e^{-\frac{t^2}{2}} dt \quad x \geq 0 \\ Q(x) &= \frac{1}{2} \operatorname{erfc}\left(\frac{x}{\sqrt{2}}\right) \end{aligned}$$

La figure 1.12 représente la probabilité d'erreur par symbole pour M-PSK (M=16, 32 et 64).

La probabilité d'erreur bit équivalente pour M-PSK est approximée dans [Pro00] par :

$$P_b \approx \frac{1}{k} P_M \quad (1.38)$$

## 1.6 Probabilité d'erreur pour M-QAM

La probabilité d'erreur par symbole pour un bruit AWGN et pour M-QAM calculée en [Pro00] est donnée par la formule suivante :

$$P_M = 1 - \left( 1 - 2 \left( 1 - \frac{1}{\sqrt{M}} \right) Q \left( \sqrt{\frac{3}{M-1} \frac{E_{moy}}{N_0}} \right) \right)^2 \quad (1.39)$$

Cette probabilité est représentée sur la figure 1.13 pour M-QAM (M=2, 4, 8, 16, 32 et 64).

Où  $\frac{E_{moy}}{N_0}$  est le rapport signal sur bruit moyen par symbole.

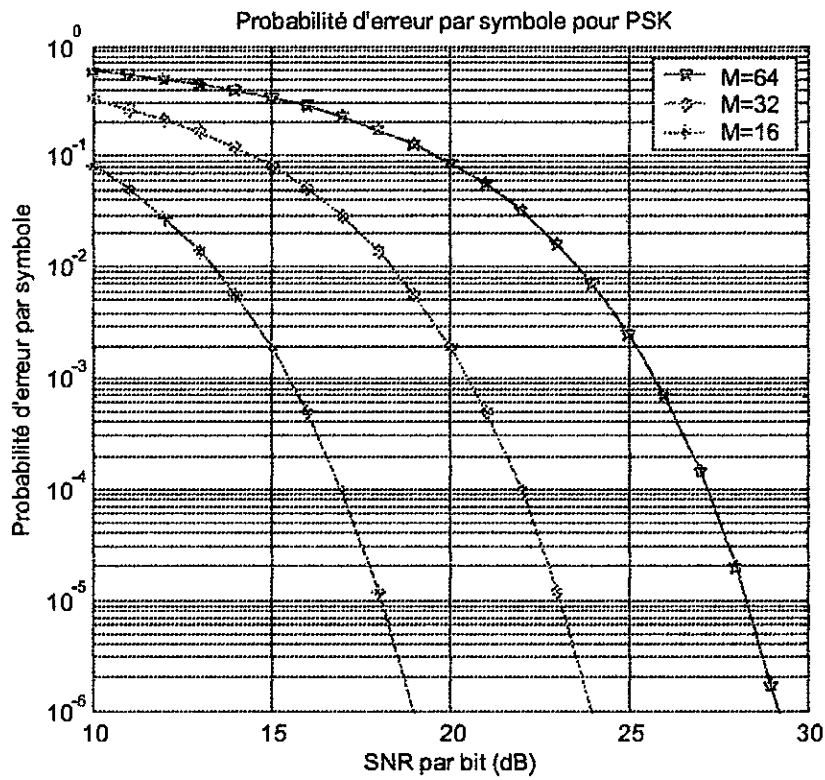


Figure 1.12 Probabilité d'erreur par symbole pour M-PSK

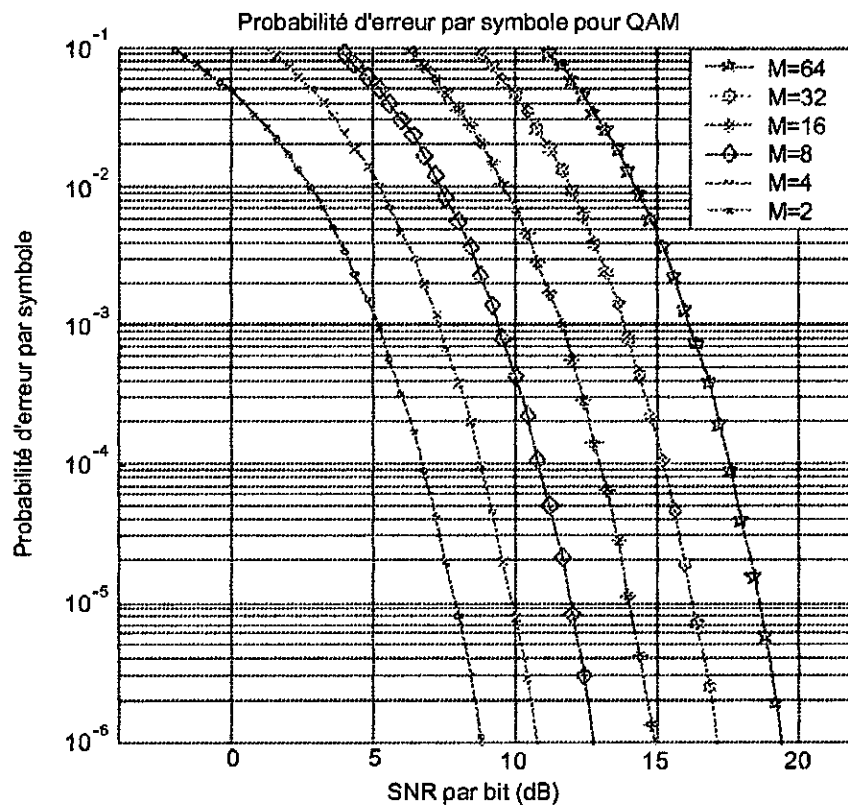


Figure 1.13 Probabilité d'erreur par symbole pour M-QAM

## Chapitre 2

# Canal de communication sans fil MIMO

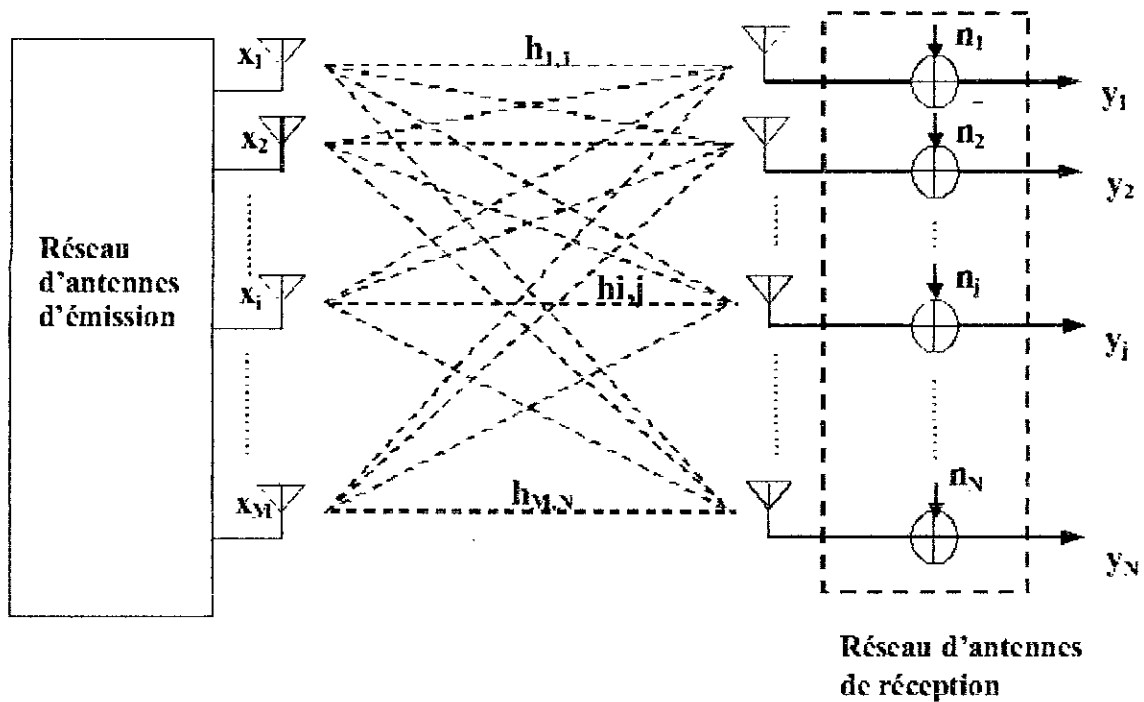
## 2.1 Modélisation du canal MIMO

Pour évaluer efficacement la performance du canal de transmission MIMO, il faut proposer des modèles qui représentent tous les effets principaux sur les différents signaux d'émission. Le modèle le plus utilisé généralement pour représenter les systèmes MIMO est le fading de Rayleigh plat quasi-statique indépendant "*independent quasi-static flat Rayleigh fading*" à tous les éléments d'antennes. Cela a été utilisé dans [Fos96], [Ala98], [Tar98] et [Tar99] où des nouveaux algorithmes de traitement du signal ont été présentés. La simplicité de ce modèle du canal a rendu l'analyse des performances de ces algorithmes moins compliqués, permettant aux auteurs de placer plus d'accent en présentation des algorithmes d'émission et de réception. Ce modèle est aussi utilisé par les concepts théoriques fondamentaux de l'information sur la capacité des canaux MIMO [Fos98] et les techniques de transmission de diversité [Ara99].

Les suppositions de base du modèle de canal MIMO "*independent quasi-static flat Rayleigh fading*" d'après [Rav02] sont :

- Un grand nombre de diffuseurs est présent dans le canal pour que le signal reçu à chaque antenne de réception du système MIMO soit la somme de plusieurs composants de multi trajets. Dans ce cas la distribution du signal reçu à chaque antenne sera Gaussienne complexe, son amplitude a une distribution de Rayleigh.
- Le retard du canal dû au multi trajets est inférieur à une période d'un symbole. Cette supposition garantit le fading plat "*flat fading*".
- Les caractéristiques du canal restent constantes au moins pendant la période de transmission d'un encadrement entier (ensemble des symboles). Cette supposition représente le fading quasi-statique "*quasi-static fading*".
- Les antennes à l'émetteur et le récepteur du système MIMO sont placés assez loin (dans l'espace) tel que l'effet du canal à un signal émis par une antenne particulière diffère pour ceux émis par les autres antennes. Cela confirme la supposition d'indépendance ou le fading non corrélé spatialement.

On peut représenter le système MIMO par le schéma suivant :



**M** : nombre d'antennes d'émission

**N** : nombre d'antennes de réception

Figure 2.1 Modèle du canal MIMO

La modélisation mathématique du schéma ci-dessus est donnée par l'équation vectorielle suivante :

$$y = Hx + n \tag{2.1}$$

Tel que :

$x = [x_1 \ x_2 \ \dots \ x_{M-1} \ x_M]^T$ , le signal vecteur d'émission de dimension  $M \times 1$ .

$y = [y_1 \ y_2 \ \dots \ y_{N-1} \ y_N]^T$ , le signal vecteur de réception de dimension  $N \times 1$ .

$n = [n_1 \ n_2 \ \dots \ n_{N-1} \ n_N]^T$ , le vecteur de bruit blanc additif gaussien de dimension  $N \times 1$ .

$$H = \begin{bmatrix} h_{11} & \dots & h_{1M} \\ h_{21} & \dots & h_{2M} \\ \vdots & \dots & \vdots \\ h_{N1} & \dots & h_{NM} \end{bmatrix} \quad , \text{ la matrice du canal de dimension } N \times M .$$

Le coefficient  $h_{ij}$  ( $1 \leq i \leq N$  et  $1 \leq j \leq M$ ) de la matrice  $H$  représente le gain complexe du canal entre l'émetteur  $j$  et le récepteur  $i$ .

$$h_{ij} = \alpha + j\beta$$



$h_{ij} = |h_{ij}| e^{j\Phi_{ij}}$  où  $|h_{ij}| = \sqrt{\alpha^2 + \beta^2}$  et  $\Phi_{ij} = \tan^{-1} \left( \frac{\beta}{\alpha} \right)$   
 $|h_{ij}|$  et  $\Phi_{ij}$  sont respectivement l'atténuation et le déphasage à la réception dû à la propagation à travers le canal sans fil qui relie l'antenne d'émission  $j$  avec l'antenne de réception  $i$ .

## 2.2 Modèle stochastique du canal MIMO [Wen02]

Le scénario de propagation dans les systèmes de communication sans fil est très complexe et le signal transmis atteint le récepteur après plusieurs réflexions sur le chemin de propagation (propagation multi trajets). Si le milieu de propagation est assez riche en dispersion, alors la méthode stochastique est la plus appropriée pour modéliser le canal. Les éléments de la matrice  $H$  sont supposés des variables aléatoires prises d'une fonction de densité de probabilité.

La distribution des phases des coefficients de la matrice  $H$  est supposée uniforme dans l'intervalle  $[-\pi, \pi[$  et celle des amplitudes est supposée une distribution de Nakagami- $m$ , de Rayleigh ou de Rice.

Soit  $H_{pq}$  ( $p = 1, \dots, N$  et  $q = 1, \dots, M$ ) les coefficients de la matrice  $H$  exprimés :

$H_{pq} = X_{pq} + jY_{pq} = A_{pq} e^{j\Phi_{pq}}$  en coordonnées cartésiennes et polaires respectivement.

### 2.2.1 La distribution de fading de Nakagami- $m$

L'amplitude  $A_{pq}$  est distribuée selon la loi de Nakagami- $m$  si la fonction de densité de probabilité de  $A_{pq}$  est :

$$p_{A_{pq}}(A_{pq}) = \frac{2}{\Gamma(m_f)} \left( \frac{m_f}{\Omega} \right)^{m_f} A_{pq}^{2m_f-1} e^{-m_f A_{pq}^2 / \Omega}, \quad A_{pq} \geq 0 \quad (2.2)$$

Où

- $\Omega = E\{A_{pq}^2\}$  est le moment d'ordre 2 (la puissance moyenne). La lettre  $m$  dans le nom « Nakagami- $m$  » est le paramètre de la distribution appelé facteur d'évanouissement (*Fading figure*).

- $m_f = \frac{\Omega^2}{E[(A_{pq}^2 - \Omega)^2]}$ ,  $m_f \geq \frac{1}{2}$ . Si  $m_f$  est entier, pour  $m_f$  et  $m_f/2$ , la variable aléatoire de Nakagami est la racine carrée de la somme des carrés de  $2m_f$  variables aléatoires gaussiennes indépendantes. La distribution de Nakagami couvre un grand ensemble de distributions :
  - pour  $m_f = 1/2$  elle devient une distribution gaussienne,
  - pour  $m_f = 1$  elle est égale à la distribution de Rayleigh,
  - et pour  $m_f$  tend vers l'infini, le canal devient statique et la densité de probabilité devient une impulsion égale à  $\sqrt{\Omega}$ .

### 2.2.2 La distribution de fading de Rayleigh et de Rice :

Pour le canal de Rayleigh, l'amplitude  $A_{pq}$  a une distribution de Rayleigh

$$P_{A_{pq}}(A_{pq}) = \frac{A_{pq}}{\Omega} e^{-A_{pq}^2/2\Omega}, \quad A_{pq} \geq 0 \quad (2.3)$$

Et dans le cas où l'amplitude  $A_{pq}$  a une moyenne différente de zéro, alors elle a une distribution de Rice

$$P_{A_{pq}}(A_{pq}) = \frac{A_{pq}}{\Omega} e^{-(A_{pq}^2 + A_0^2)/2\Omega} I_0\left(\frac{A_{pq}A_0}{\Omega}\right), \quad A_{pq} \geq 0 \quad (2.4)$$

Où  $A_0$  est la moyenne et  $I_0(\cdot)$  est la fonction de Bessel modifiée de première espèce d'ordre zéro. La distribution de Rice est Souvent caractérisée par le rapport :

$$K = \frac{A_0^2}{2\Omega}$$

Pour  $K=0$ , la distribution de Rice devient Rayleigh. Noter aussi que la distribution de Nakagami- $m$  peut être approximée par la distribution de Rice, surtout pour les grandes valeurs du paramètre de fading  $m_f$ .

### 2.3 Capacité du canal MIMO [Hol01]

Les systèmes multi-antennes à l'émission et à la réception (MIMO) sont aujourd'hui considérés comme étant un des secteurs de recherche les plus prometteurs de communications

sans fil. C'est dû au fait qu'un canal MIMO peut donner un gain de capacité significatif très important par rapport au canal SISO traditionnel. L'augmentation de l'efficacité spectrale offerte par les systèmes MIMO est basée sur l'utilisation de la diversité spatiale (ou les antennes) à l'émission et à la réception.

La capacité d'un canal est le maximum d'information moyenne transmise à travers le canal [Sob]. Il est désirable de concevoir les systèmes de transmission qui exploitent la capacité du canal autant que possible. Représentons l'entrée et la sortie d'un canal sans fil sans mémoire par les variables aléatoires  $X$  et  $Y$  respectivement, la capacité du canal est définie comme suit :

$$C = \max_{p(x)} I(X; Y) \quad (2.5)$$

Où  $I(X; Y)$  représente l'information mutuelle entre  $X$  et  $Y$ . L'équation (2.5) montre que l'information mutuelle est maximisée selon toutes les distributions statistiques  $p(x)$  possibles de l'émetteur. L'information mutuelle est une mesure de la quantité d'information d'une variable aléatoire contenant une autre variable. L'information mutuelle entre  $X$  et  $Y$  peut être aussi écrite :

$$I(X; Y) = H(Y) - H(Y|X) \quad (2.6)$$

Où  $H(Y|X)$  représente l'entropie conditionnelle entre les variables aléatoires  $X$  et  $Y$ .

L'entropie d'une variable aléatoire peut être décrite comme une mesure de la quantité d'information moyenne exigée pour décrire la variable aléatoire. Elle peut être aussi décrite comme une mesure de l'incertitude de la variable aléatoire. L'information mutuelle peut être décrite comme étant la réduction de l'incertitude d'une variable aléatoire en raison de la connaissance de l'autre. Noter que l'information mutuelle entre  $X$  et  $Y$  dépend des propriétés du canal (matrice de canal  $H$ ) et les propriétés de  $X$  (la distribution de probabilité de  $X$ ).

La capacité est mesurée en bits par seconde (bits/s). Il est parfois préférable de la représenter par unité de la bande passante du canal. La capacité du canal est alors mesurée en bits/s/Hz.

### 2.3.1 Capacité du canal SISO

La capacité ergodique (moyenne) d'un canal aléatoire avec  $M=N=1$  et une puissance moyenne transmise  $P_T$  peut être exprimée comme suit :

$$C = E_H \left\{ \max_{p(x): P^* \leq P_T} I(X; Y) \right\} \quad (2.7)$$

Où  $P$  est la puissance moyenne d'un seul mot transmis à travers le canal et  $E_{\{ \cdot \}}$  est l'espérance de toutes les réalisations du canal. En comparant cette équation avec la définition dans (2.5), la capacité du canal est maintenant définie comme le maximum de l'information mutuelle entre l'émetteur et le récepteur sur toutes les distributions statistiques de l'entrée qui satisfont la contrainte de puissance. Si chaque symbole de canal à l'émetteur est noté par  $s$ , la contrainte de puissance moyenne peut être exprimée comme suit :

$$P = E[|s|^2] \leq P_T \quad (2.8)$$

En utilisant (2.7), la capacité ergodique (moyenne) d'un canal SISO ( $M = N = 1$ ) avec un gain de canal aléatoire complexe  $h_{11}$  est donnée par :

$$C = E_H \left\{ \log_2(1 + \rho |h_{11}|^2) \right\} \quad (2.9)$$

Où  $\rho$  est le rapport signal sur bruit (SNR) au récepteur. Si  $|h_{11}|$  a une distribution de Rayleigh,  $|h_{11}|^2$  suit une distribution de Khi-deux de deux degrés de liberté. L'équation (2.9) peut être alors écrite par :

$$C = E_H \left\{ \log_2(1 + \rho \chi_2^2) \right\} \quad (2.10)$$

Où  $\chi_2^2$  est la variable aléatoire de Khi-deux de deux degrés de liberté. La figure 2.2 représente la capacité de Shannon pour un canal gaussien (courbe pointillée) et la capacité d'un canal à fading de Rayleigh (courbe continue) selon l'équation (2.10).

### 2.3.2 Capacité du canal MIMO

La capacité ergodique (moyenne) d'un canal aléatoire MIMO sachant que la puissance maximale transmise est  $P_T$  peut être exprimée comme suit :

$$C = E_H \left\{ \max_{p(\mathbf{x}) : E\{\Phi\} \leq P_T} I(\mathbf{x}; \mathbf{y}) \right\} \quad (2.11)$$

Où  $\Phi = E\{\mathbf{x}\mathbf{x}^*\}$  est la matrice de covariance du signal vecteur transmis  $\mathbf{X}$ . La puissance totale transmise est limitée à  $P_T$ , indépendamment du nombre d'antennes d'émission.

---

<sup>†</sup> L'indice \* désigne le transposé Hermitien

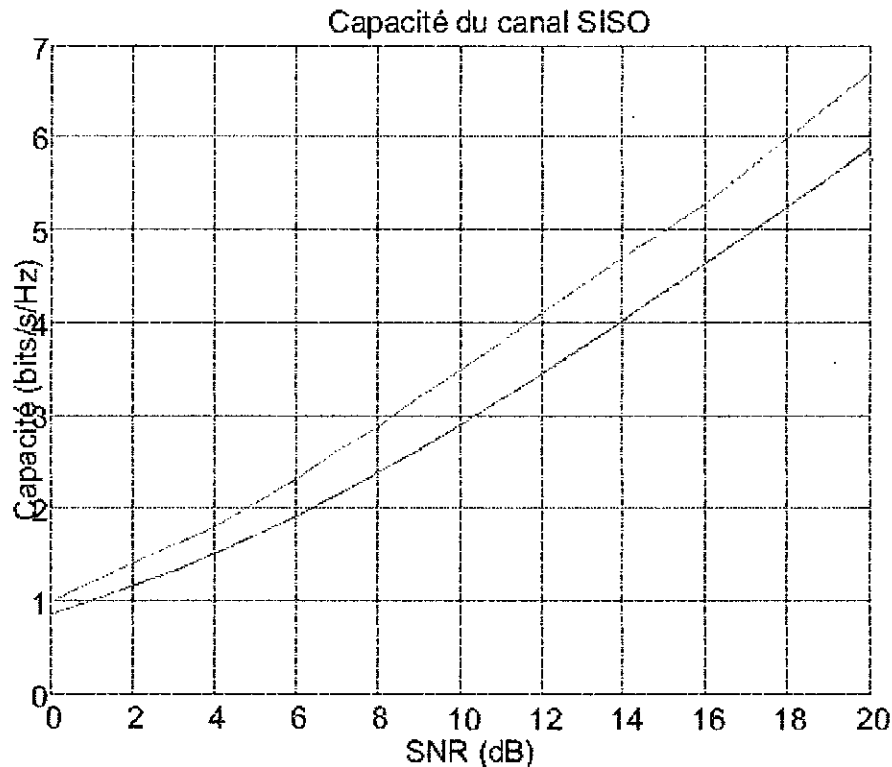


Figure 2.2 : Capacité d'un canal SISO à fadings de Rayleigh (courbe continue) comparée par rapport à la capacité de Shannon du canal SISO (courbe pointillée).

En utilisant (2.1) et la relation entre l'information mutuelle et l'entropie, (2.11) peut être analysée pour un  $H$  donné comme suit :

$$I(x; y) = h(y) - h(y | x) \quad (2.12)$$

$$= h(y) - h(Hx + n | x) \quad (2.13)$$

$$= h(y) - h(n | x) \quad (2.14)$$

$$= h(y) - h(n) \quad (2.15)$$

Où  $h(\cdot)$  représente l'entropie différentielle d'une variable aléatoire. Le vecteur transmis  $x$  et le vecteur de bruit  $n$  sont supposés indépendants. L'équation (2.15) est maximisée quand  $y$  est gaussien, puisque la distribution normale maximise l'entropie pour une variance donnée [Cov91]. L'entropie différentielle d'un vecteur réel gaussien  $y \in R^n$  avec une moyenne de zéro et une matrice de covariance  $K$  est égale à  $\frac{1}{2} \log_2 ((2\pi e)^n \det K)$ . Pour un vecteur complexe gaussien  $y \in C^n$ , l'entropie différentielle est inférieure ou égale à

$\log_2 \det(\pi e K)$ , avec l'égalité si et seulement si  $y$  est un complexe circulaire symétrique Gaussien avec  $E\{yy^*\} = K$  [Tel95]. En supposant la distribution optimale gaussienne pour le vecteur de transmission  $x$ , la matrice de covariance du vecteur complexe reçu  $y$  est donnée par :

$$E\{yy^*\} = E\{(Hx + n).(Hx + n)^*\} \quad (2.16)$$

$$= E\{Hx x^* H^*\} + E\{nn^*\} \quad (2.17)$$

$$= H\Phi H^* + K^n \quad (2.18)$$

$$= K^d + K^n \quad (2.19)$$

Les indices  $d$  et  $n$  désignent respectivement la partie désirable et la partie de bruit de (2.19).

On donne alors l'information maximale mutuelle d'un canal aléatoire MIMO par :

$$\begin{aligned} I &= h(y) - h(n) \\ &= \log_2 [\det(\pi e (K^d + K^n))] - \log_2 [\det(\pi e K^n)] \\ &= \log_2 [\det(K^d + K^n)] - \log_2 [\det(K^n)] \\ &= \log_2 [\det((K^d + K^n).(K^n)^{-1})] \\ &= \log_2 [\det(K^d (K^n)^{-1} + I_N)] \\ &= \log_2 [\det(H\Phi H^* (K^n)^{-1} + I_N)]. \end{aligned}$$

Quand l'émetteur n'a aucune connaissance sur le canal, il est optimal d'utiliser une distribution uniforme de puissance à l'émission [Tel95]. Dans ce cas la matrice de covariance d'émission est donnée par :  $\Phi = \frac{P_T}{M} I_M$ . on suppose que le bruit dans chaque branche du récepteur est

non corrélé, dans ce cas la matrice de covariance de bruit  $K^n = \sigma^2 I_N$ . La capacité ergodique (moyenne) pour un canal MIMO AWGN complexe peut être exprimée comme suit [Tel95][Fos98] :

$$C = E_H \left\{ \log_2 \left[ \det \left( I_N + \frac{P_T}{\sigma^2 M} H H^* \right) \right] \right\} \quad (2.20)$$

$$C = E_H \left\{ \log_2 \left[ \det \left( I_N + \frac{\rho}{M} H H^* \right) \right] \right\} \quad (2.21)$$

Où  $\rho = \frac{P_T}{\sigma^2}$  est le rapport signal sur bruit à chaque antenne de réception.

Si on analyse plus loin la capacité du canal MIMO présentée en (2.21), il est possible de diagonaliser la matrice produite  $HH^*$  par la décomposition en valeurs propres ou bien par la décomposition en valeurs singulières. En utilisant la décomposition en valeurs propres, le produit matriciel  $HH^*$  est écrit comme suit :

$$HH^* = EAE^* \quad (2.22)$$

Où  $E$  est la matrice des vecteurs propres avec des colonnes orthonormales et  $A$  est une matrice diagonale qui contient les valeurs propres sur sa diagonale principale. En utilisant cette notation, (2.21) peut être écrite :

$$C = E_H \left\{ \log_2 \left[ \det \left( I_N + \frac{\rho}{M} EAE^* \right) \right] \right\} \quad (2.23)$$

Le produit matriciel peut être exprimé en utilisant la décomposition en valeurs singulières sur la matrice  $H$  du canal qui peut être écrite :

$$H = U \Sigma V^* \quad (2.24)$$

Où  $U$  et  $V$  sont des matrices unitaires de vecteurs singuliers gauche et droite respectivement,  $\Sigma$  est une matrice diagonale qui contient les valeurs singulières sur sa diagonale principale.

Tous les éléments sur la diagonale sont nuls sauf les premiers  $K$  éléments ( $K$  est le rang de la matrice  $H$ ). En remplaçant (2.24) dans (2.21), la capacité du canal MIMO peut être écrite :

$$C = E_H \left\{ \log_2 \left[ \det \left( I_N + \frac{\rho}{M} U \Sigma \Sigma^* U^* \right) \right] \right\} \quad (2.25)$$

Maintenant, il est plus facile de remarquer que la capacité totale d'un canal MIMO est égale à la somme des capacités des sous-canaux SISO AWGN parallèles.

Le nombre des sous-canaux parallèles est déterminé par le rang de la matrice du canal. En général, le Rang de la matrice du canal est conditionné par :

$$\text{Rang}(H) = k \leq \min\{M, N\} \quad (2.26)$$

En utilisant (2.26) et sachant que :

$$\det \left( I_N + \frac{\rho}{M} EAE^* \right) = \det \left[ E \left( I_N + \frac{\rho}{M} A \right) E^* \right]$$

$$\begin{aligned}
&= \det \left( I_N + \frac{\rho}{M} A \right) \\
&= \prod_{i=1}^k \left( 1 + \frac{\rho}{M} \lambda_i \right)
\end{aligned}$$

L'expression (2.23) peut être écrite :

$$C = E_H \left\{ \sum_{i=1}^k \log_2 \left( 1 + \frac{\rho}{M} \lambda_i \right) \right\} \quad (2.27)$$

Où les  $\lambda_i$  sont les valeurs propres de la matrice diagonale  $A$ .

De la même façon, on peut trouver :

$$C = E_H \left\{ \sum_{i=1}^k \log_2 \left( 1 + \frac{\rho}{M} \sigma_i^2 \right) \right\} \quad (2.28)$$

Où  $\sigma_i^2$  sont les carrés des valeurs singulières de la matrice diagonale  $\Sigma$ .

Avec une combinaison optimale au récepteur et en utilisant la diversité au récepteur seulement ( $M=1$ ), la capacité du canal peut être exprimée comme suit [Fos98a] :

$$C = E_H \left\{ \log_2 \left( 1 + \rho \cdot \chi_{2N}^2 \right) \right\} \quad (2.29)$$

Où  $\chi_{2N}^2$  est la variable aléatoire de Khi-deux de  $2N$  degrés de liberté. S'il y a  $M$  antennes d'émission et une combinaison optimale entre les  $N$  antennes de récepteur, la capacité peut être écrite<sup>2</sup> [Fos98a] :

$$C = E_H \left\{ M \cdot \log_2 \left( 1 + \frac{\rho}{M} \cdot \chi_{2N}^2 \right) \right\} \quad (2.30)$$

L'équation (2.30) représente la limite supérieure du canal MIMO à fading de Rayleigh. Dans la Figure 2.3, la capacité de Shannon d'un canal SISO est comparée avec la limite supérieure de (2.30) avec  $M=N=6$ . Bien que cette limite du canal MIMO représente un cas spécial, la figure 2.3 montre clairement le potentiel de la technologie MIMO.

---

<sup>2</sup> En supposant qu'il n'y a pas d'interférences entre les  $N$  signaux reçus.



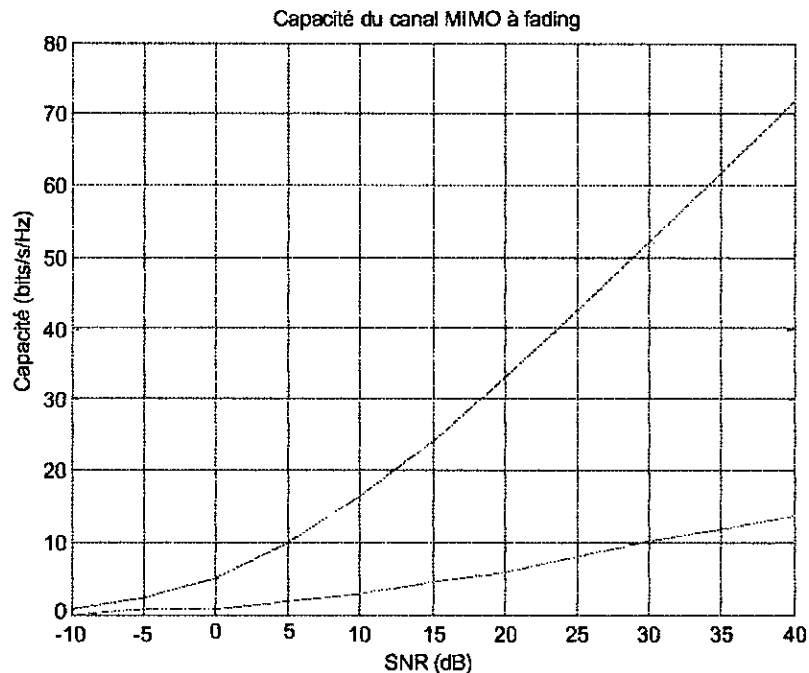


Figure 2.3 : Capacité de Shannon d'un canal SISO (courbe pointillée) comparée par rapport à la capacité d'un canal MIMO à fading de Rayleigh (courbe continue) avec  $M = N = 6$ .

### 2.3.3 Capacité en sortie (Outage capacity)

Dans ce qui précède la capacité ergodique (moyenne) a été exprimée comme une mesure de l'efficacité spectrale du canal MIMO. La capacité ergodique dans (2.7) et (2.11) a été définie comme la moyenne de la valeur maximale de l'information mutuelle entre le signal transmis et le signal reçu où la maximisation a été effectuée selon toutes les distributions statistiques possibles de l'émission. Une autre mesure de la capacité du canal fréquemment utilisée est la capacité en sortie (Outage capacity). Avec cette dernière, la capacité du canal est associée à une probabilité de sortie (Outage). La capacité est traitée comme une variable aléatoire qui dépend de la réponse instantanée du canal et reste constante pendant la transmission d'un bloc fini d'information. Si la capacité du canal diminue au-dessous de la capacité en sortie (Outage capacity), alors il n'y a aucune possibilité de décoder le bloc d'information transmis sans erreurs (Si le codage est utilisé à l'émission). La probabilité pour laquelle la capacité est inférieure au capacité en sortie ( $C_{outage}$ ) est  $q$ . Cela peut être exprimé en termes mathématiques par :

$$\text{Prob}\{C \leq C_{\text{outage}}\} = q \quad (2.31)$$

Dans ce cas, (2.31) représente une limite supérieure en raison du fait qu'il y'a une probabilité finie  $q$  pour laquelle la capacité du canal est inférieure à la capacité en sortie (Outage capacity). Elle peut être écrite comme une limite inférieure, représentant le cas où il y'a une probabilité finie  $(1 - q)$  pour laquelle la capacité du canal est supérieure à  $C_{\text{outage}}$ , c-à-d :

$$\text{Prob}\{C > C_{\text{outage}}\} = 1 - q \quad (2.32)$$

## 2.4 Evaluation de l'efficacité spectrale du canal MIMO

Comme nous l'avons vu précédemment, La capacité d'un canal est le maximum d'information moyenne transmise à travers le canal [Sob]. Cette capacité dépend du milieu de propagation et du signal transmis. A partir de la formule (2.21), la capacité du canal MIMO dépend de la matrice  $H$  du canal et du rapport signal sur bruit à la réception  $\rho$  (ou bien de la puissance émise et du bruit au niveau de la réception).

La capacité maximale d'un canal MIMO est atteinte virtuellement quand chacun des  $M$  signaux transmis est reçu par le même réseau d'antennes sans interférence. Elle peut être également décrite comme si chaque signal transmis sera reçu par une antenne de réception séparée, donc par  $M \times N$  antennes de réception. Avec une combinaison optimale entre les  $N$  antennes du récepteur, la capacité peut être écrite (formule 2.30) :

$$\begin{aligned} C &= E_H \left\{ M \cdot \log_2 \left( 1 + \frac{\rho}{M} \cdot \chi_{2N}^2 \right) \right\} \\ &= M \cdot E_H \left\{ \log_2 \left( 1 + \frac{\rho}{M} \cdot \chi_{2N}^2 \right) \right\} \end{aligned} \quad (2.33)$$

Nous remarquons que la capacité du canal MIMO augmente presque linéairement avec le nombre d'antennes d'émission.

A partir d'une simulation sous MATLAB, nous calculons la capacité moyenne de 1000 échantillons aléatoires (réalisations de la matrice  $H$  du canal) pour des nombres d'antennes de 1 à 20 antennes ( $M=1:20$ ). Les résultats pour un  $\text{SNR}=0$ ,  $\text{SNR}=10$  et  $\text{SNR}=20$  sont représentés sur la figure 2.4 :

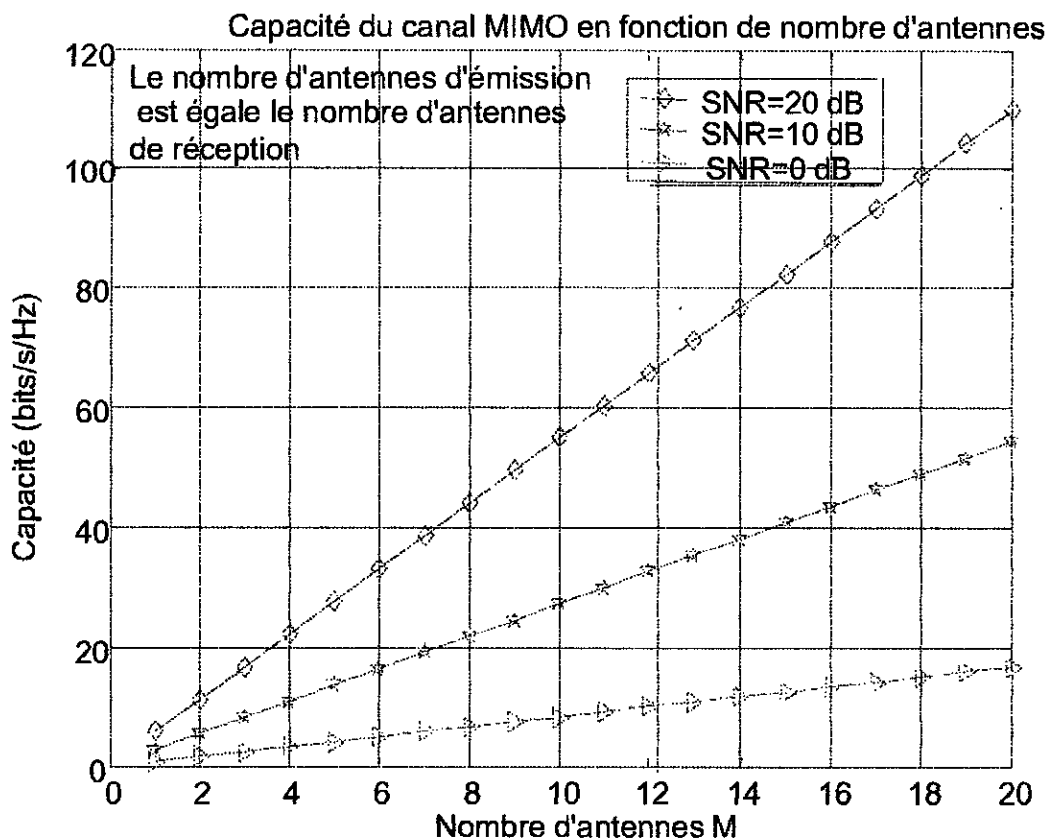


Figure 2.4 Variation de la capacité du canal MIMO en fonction de nombre d'antennes d'émission et de la réception ( $M=N$ ,  $SNR=0$ ,  $SNR=10$  et  $SNR=20$ ) [Programme1]

Nous voyons clairement que la capacité du canal MIMO augmente linéairement en fonction du nombre d'antennes d'émission et de réception. Nous avons utilisé ici le même nombre d'antennes à l'émission et à la réception. Ça représente un grand avantage pour les systèmes multi-antennes à l'émission et à la réception (MIMO).

Le graphe donné par la Figure 2.5 représente l'augmentation de l'efficacité spectrale du canal MIMO avec l'augmentation du rapport signal sur bruit (SNR).

Nous remarquons que la capacité du canal MIMO augmente rapidement avec l'augmentation du rapport signal sur bruit (SNR). Cela encourage les opérateurs et les investisseurs dans ce domaine d'utiliser les systèmes MIMO surtout dans le cas d'émission de grande puissance.

Maintenant nous allons voir la variation de la capacité du canal MIMO en fonction de la dimension de la matrice  $H$  du canal (Le nombre d'antennes d'émission et de réception) et en fonction du rapport signal sur bruit (SNR) en même temps, voir Figure 2.6.

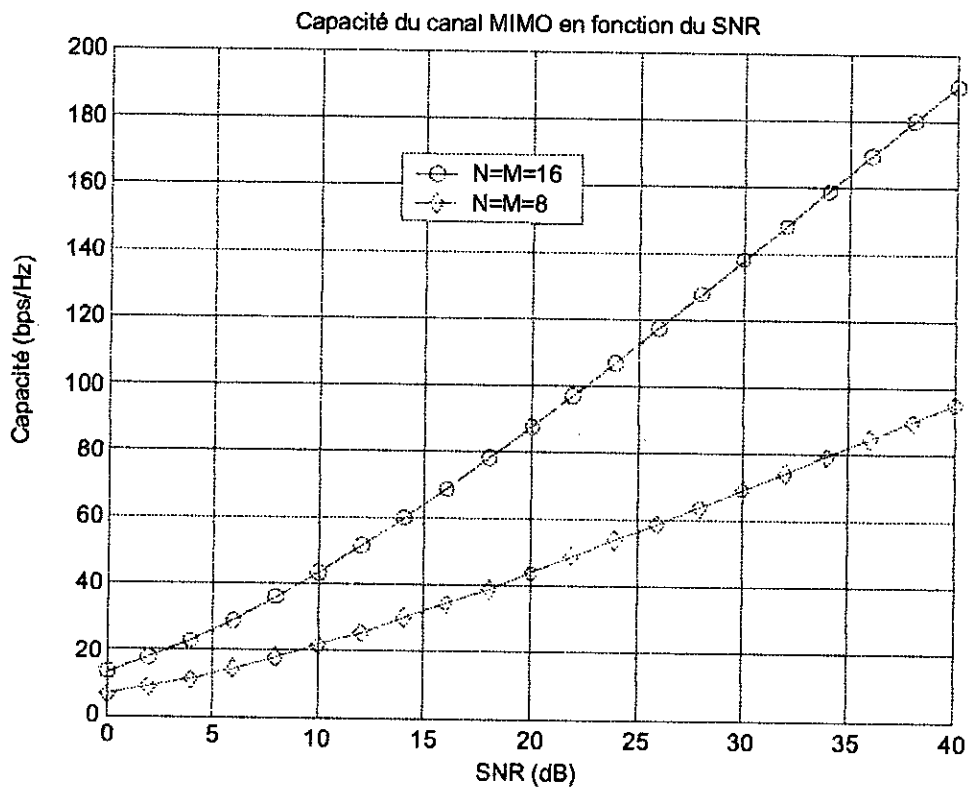


Figure 2.5 Variation de la capacité d'un canal MIMO en fonction du rapport signal sur bruit (SNR) [Programme2].

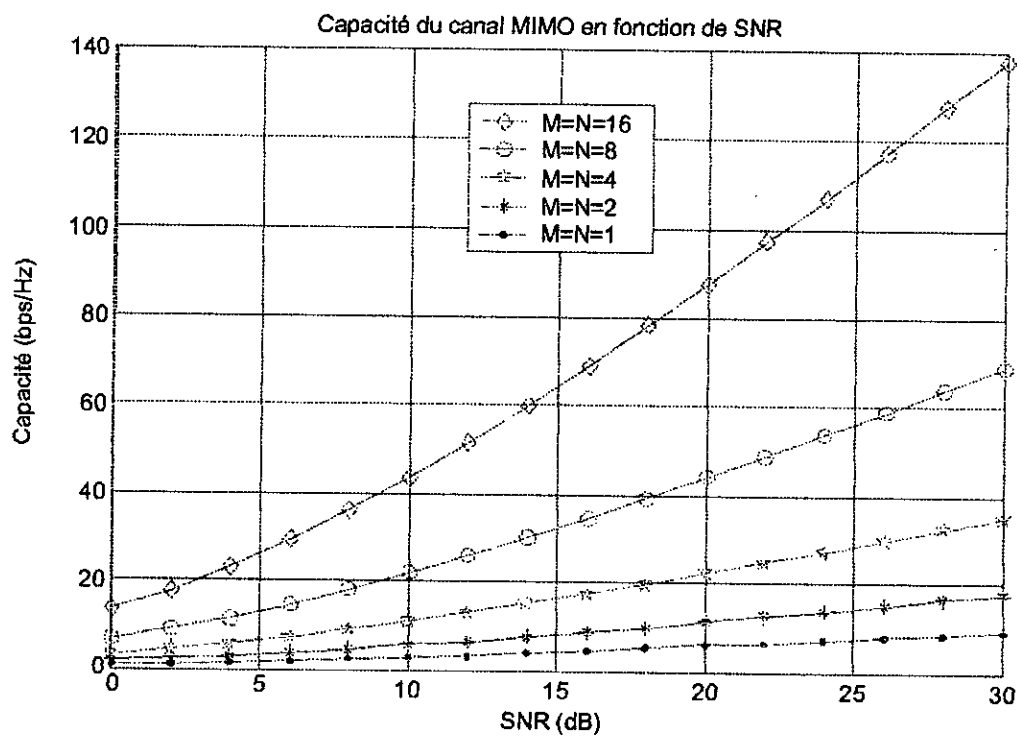


Figure 2.6 Variation de la capacité du canal MIMO en fonction du rapport signal sur bruit et des nombres d'antennes d'émission et de réception [Programme3].

## 2.5 Conclusion

A partir des résultats présentés précédemment, nous pouvons dire que le canal MIMO peut atteindre une grande efficacité spectrale si en le comparant avec les systèmes conventionnels qui n'utilisent pas la diversité spatiale.

La haute efficacité spectrale atteinte par un système MIMO est faite dans le cas où l'environnement est riche en dispersion, les signaux de chaque émetteur apparaissent fortement non corrélés dans chacune des antennes de réception. Quand les signaux sont transmis par des canaux non corrélés entre l'émetteur et le récepteur, chacun des signaux transmis suit un chemin spatial différent des autres. Le récepteur peut utiliser les différences de la signature spatiale simultanément et à la même fréquence pour séparer les signaux qui sont provenus de différentes antennes d'émission et qui ont parcouru différents chemins du milieu de propagation.

Le problème qui se pose maintenant c'est comment exploiter ces avantages et ces grandes capacités du canal MIMO. La grande difficulté est au niveau de la réception, c-à-d comment on doit séparer et détecter les signaux transmis au niveau des antennes de réception.

Le premier algorithme proposé pour résoudre ce problème ou bien pour exploiter les systèmes MIMO est l'algorithme D-BLAST (Diagonal Bell Labs Layered Space-Time) qui a été proposé par Foschini en 1996, cet algorithme est capable de réaliser une partie substantielle de la capacité MIMO. Cependant, une haute complexité de l'implémentation de l'algorithme est son inconvénient substantiel. Une version simplifiée de l'algorithme BLAST est connue sous le nom V-BLAST (vertical BLAST). Elle est capable de réaliser une grande efficacité spectrale tandis qu'elle est relativement simple à implémenter.

Nous verrons dans les chapitres suivants la description et l'analyse de l'architecture V-BLAST, une simulation sous *MATLAB* est prévue afin d'évaluer ses différents résultats.

## Chapitre 3

# L'architecture *BLAST* (Bell-labs

# Layared Space-Time)

## 3.1 Introduction

Dans la prochaine génération des réseaux de communications sans fil outre les services de téléphonie et de transmission de données déjà disponibles aujourd'hui, le nouveau système IMT 2000/UMTS (Universal Mobile Telecommunications System) permettra en effet le surf sans fil sur internet, le commerce électronique et même la transmission vidéo avec une qualité télévisuelle – tout cela grâce à une grande vitesse de transmission des données jusqu'à 200 fois plus rapide qu'aujourd'hui.

Alors, les services de haut débit et de meilleure qualité de transmission de données sont exigés. Cependant, la distorsion du signal transmis devient plus importante quand le débit augmente dans un canal multi-trajets et cela cause des erreurs de détection au récepteur. En effet, Les systèmes qui communiquent sur un canal sans fil SISO (Single Input Single Output) ont une capacité limitée. Dans quelques situations, la communication sur un canal SISO n'est pas fiable à cause du fading dû aux multi-trajets, pour cela, il faut introduire des canaux sans fil fiables avec grandes capacités.

Des recherches théoriques de l'information en ces dernières années [Fos96], [Fos98], [Fos98a] ont montré que des très hautes capacités peuvent être obtenues en utilisant des systèmes sans fil Multi-antennes à l'émission et à la Réception MIMO (Multiple Input Multiple Output). L'augmentation de l'efficacité spectrale offerte par les systèmes MIMO est basée sur l'utilisation de la diversité spatiale à l'émission et à la réception. Les résultats de ces recherches ont mené au développement d'une nouvelle architecture d'émission-réception appelée *Bell Labs Layered Space-Time Architecture (BLAST)* [Fos96] qui fournit des débits de transmission non atteints en utilisant des techniques traditionnelles. [Rav02]

Jusqu'au milieu des années 90, il était généralement admis que l'unique façon d'augmenter la capacité de la communication sans fil était d'augmenter la largeur de la bande et la puissance émise.

Jerry Foschini, de Bell Laboratories de Lucent Technology, au New Jersey, a ouvert en 1996 de nouveaux horizons à la recherche dans le domaine de la technologie sans fil en faisant la démonstration expérimentale d'une nouvelle architecture qu'il l'a baptisé *BLAST* (Bell Labs Layered Space-Time) et qui accroît la capacité de la communication sans fil.

Plutôt que d'augmenter la largeur de bande, les concepteurs du système de communication sans fil *BLAST* se sont servis de plusieurs antennes pour émettre et recevoir les signaux multiples transmis en parallèle (en couches espace/temps comme l'indique le nom du système), sur les mêmes fréquences, grâce à une répartition de la puissance de transmission normale. [Se02]

Le codage espace-temps et/ou des algorithmes spéciaux de traitement de signal doivent être implémentés au récepteur pour réaliser au moins la partie de la capacité du canal MIMO. L'algorithme *D-BLAST* (**D**iagonal **B**ell Labs **L**Ayered **S**pace-**T**ime) a été proposé par Foschini pour cette fin, qui est capable de réaliser une partie substantielle de la capacité MIMO. Cependant, une haute complexité de l'implémentation de l'algorithme est son inconvénient substantiel. Une version simplifiée de l'algorithme *BLAST* est connue sous le nom *V-BLAST* (vertical *BLAST*). Elle est capable de réaliser une grande efficacité spectrale tandis qu'elle est relativement simple à implémenter. [Loy02]

### 3.2 Description de l'architecture *BLAST* [BLA00]

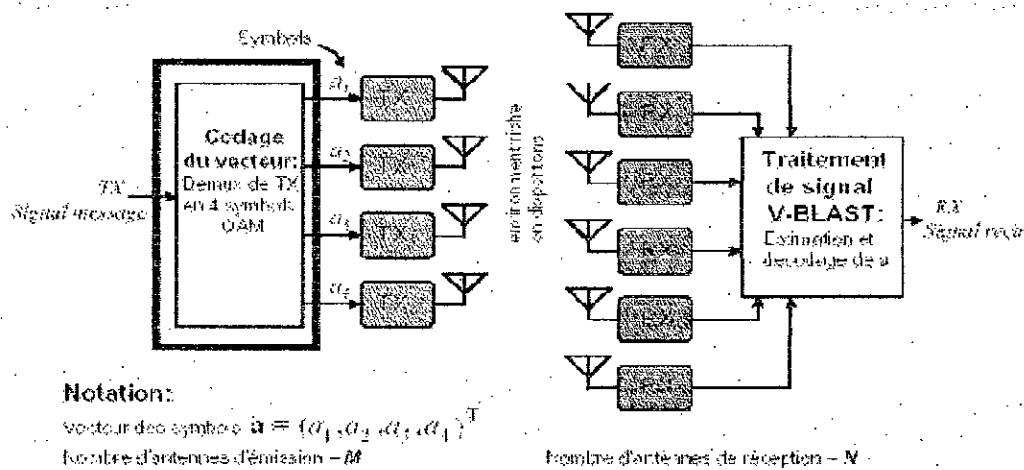
L'architecture *BLAST* est une technique de communication sans fil qui utilise des réseaux d'antennes à l'émission et à la réception pour fournir des vitesses de transmission (débits) très grandes par rapport aux approches conventionnelles.

Dans les systèmes sans fil, les ondes hertziennes ne se propagent pas simplement de l'antenne d'émission à l'antenne de réception, mais rebondissent et se dispersent aléatoirement sur les objets et les obstacles dans l'environnement de propagation. Cette dispersion s'appelle multi-trajets, elle donne des copies multiples ("des images") du signal transmis parvenant au récepteur via des différents chemins dispersés. Dans les systèmes conventionnels sans fil, le multi-trajet représente un obstacle significatif à la transmission précise, parce que les images parviennent au récepteur à des temps légèrement différents et peuvent ainsi se superposer de façon destructive et s'annulent enfin. Pour cette raison, le multi-trajet est traditionnellement vu comme une grande perte. Cependant, en employant l'architecture *BLAST*, il est possible d'exploiter les multi-trajets, c'est-à-dire exploiter les caractéristiques de dispersion du milieu

de propagation pour augmenter, plutôt que dégrader, l'efficacité de la transmission en traitant la multiplicité de chemins dispersés comme des sous-canaux séparés parallèles.

L'architecture BLAST accomplit cela en divisant le signal message aux plusieurs séquences, puis on émit simultanément et parallèlement ces séquences à travers un réseau d'antennes d'émission, voir Figure 3.1. Toutes les séquences sont transmises dans la même bande de fréquence, donc le spectre est exploité très efficacement. Puisque les séquences du signal message sont envoyées en parallèles sur plusieurs antennes, la vitesse de transmission est augmentée presque proportionnellement avec l'augmentation de nombre d'antennes d'émission utilisées.

### V-BLAST: Le grand schema



Un système V-BLAST avec  $M=4$  et  $N=6$

Figure 3.1 Un système V-BLAST avec  $M=4$  et  $N=6$

Au récepteur, on utilise encore un réseau d'antennes pour recevoir les séquences multiples transmises et leurs images dispersées. Chaque antenne du réseau reçoit toutes les séquences transmises en superposition, non séparément. Cependant, si la dispersion due au multi-trajet est suffisante, alors toutes les multiples séquences sont dispersées différemment, puisqu'elles proviennent de différentes antennes d'émission qui sont placées aux points différents dans l'espace. En utilisant un traitement de signal sophistiqué, ces légères différences de la dispersion permettent aux séquences d'être identifiées et récupérées. En effet, le multi-trajet est exploité pour fournir un parallélisme spatial très utile qui est employé pour améliorer le débit de transmission de données. Ainsi, en utilisant la technique BLAST, plus qu'il y'a la propagation multi-trajet, plus la transmission soit rapide et fiable.



Les algorithmes de traitement du signal *BLAST* utilisés au récepteur sont le coeur de la technique. Au niveau des antennes de réception, des processeurs très rapide traitent les signaux captés par l'ensemble des antennes de réception simultanément, ensuite ils extraient le signal le plus fort des signaux reçus, et après les signaux faibles restant, qui sont plus faciles à récupérer une fois que les signaux plus forts ont été enlevés comme une source d'interférence. La capacité de séparer les sous-séquences dépend aussi des différences entre les chemins de propagation.

Sous la supposition théorique largement utilisée de dispersion indépendante de Rayleigh, la capacité théorique de l'architecture *BLAST* augmente presque linéairement avec le nombre d'antennes, quand la puissance totale transmise est tenue constante. Dans le cas réel bien sûr, la dispersion sera moins favorable que la supposition indépendante de Rayleigh et il reste à voir combien de capacité est en réalité disponible dans les divers environnements de propagation. Néanmoins, même dans les environnements relativement pauvres de dispersion, la technique *BLAST* doit être capable de fournir des capacités plus grandes par rapport aux architectures conventionnelles. Un prototype réalisé dans les laboratoires Bell a déjà montré une efficacité spectrale de 20 - 40 bits par seconde par Hertz de largeur de bande, des chiffres non atteints en employant les techniques conventionnelles.

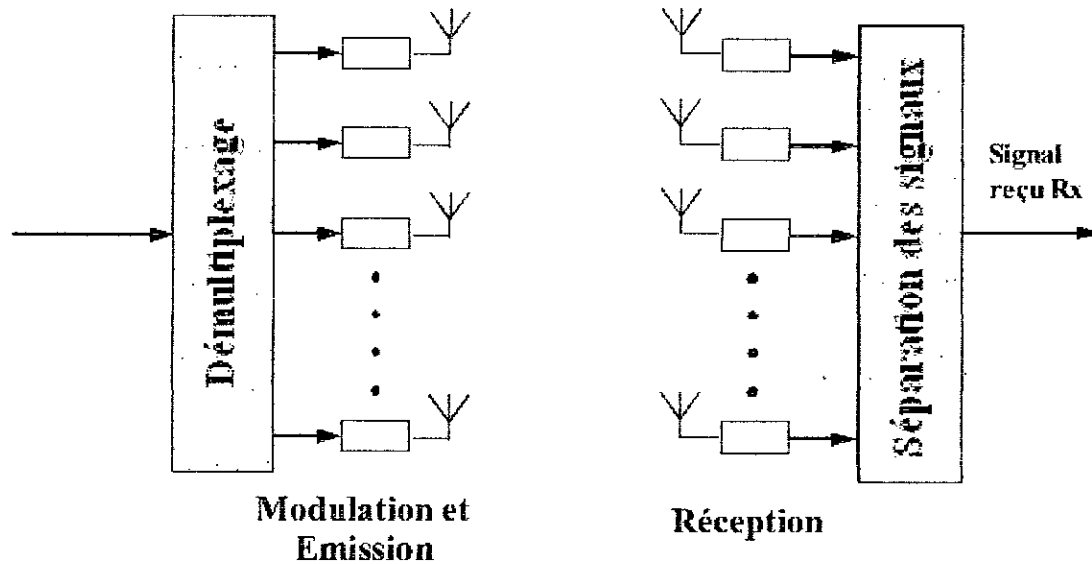
### 3.3 Modélisation de l'architecture *BLAST* [Bea01]

Il existe deux variétés de l'architecture *BLAST* :

- Architecture diagonale D-*BLAST* [Fos96] ;
- Architecture verticale V-*BLAST* [Fos98].

Les préfixes D et V désignent la direction dans laquelle les séquences du signal message sont codées et le signal reçu est traité en couches Espace-temps.

L'architecture générale du système est la même pour les deux algorithmes de *BLAST* (Figure 3.2), bien que les opérations de démultiplexage et de séparation des signaux sont différentes. Pour les deux cas, le démultiplexeur divise la séquence d'information incidente (*incoming data*) en  $n_s$  séquences de bit (*bit streams*) notés  $a$  à  $d$  pour 4 antennes d'émission de l'exemple de la Figure 3.3.

Figure 3.2 : Architecture du système *BLAST*

Pour V-BLAST, chaque séquence d'information (*data stream*) après démultiplexage est toujours dirigée à la même antenne d'émission comme elle est montrée dans la Figure 3.3(a) (bien que l'architecture originale de V-BLAST elle-même n'utilise pas de codage, elle peut utiliser un certain codage pour améliorer la performance). Cependant, pour D-BLAST chaque séquence (*data stream*) après démultiplexage est codée Individuellement et dirigée cycliquement aux différentes antennes dans chaque période des symboles, voir Figure 3.3(b). La transmission de chaque séquence (*stream*) codée est distribuée en espace-temps.

Pour l'algorithme V-BLAST la séparation et la détection des signaux à la réception se fait symbole par symbole en utilisant la Combinaison de mise à zéro (*Nulling*) et d'annulation (*Cancellation*) des symboles [Fos98].

Au contraire, D-BLAST traite les signaux reçus en couches diagonales dans l'espace-temps correspondants à chaque séquence codée (*coded stream*) individuellement [Fos96]. Par exemple, soit la diagonale contenant la séquence  $a$ , dans la Figure 3.3(b), est détectée, ça serait annulée et le traitement continuerait en diagonales contenant les séquences  $d, c, b$  et  $a$  à nouveau, etc.

Pour que les deux architectures fonctionnent efficacement, il faut avoir plus d'antennes de réception que d'antennes d'émission, bien que n'importe quelle augmentation du nombre des éléments d'antennes de réception peut améliorer mieux la performance de la détection. Deuxièmement, les processus de détection exigent aussi une connaissance précise de la

matrice de réponse du canal MIMO et donc n'importe quelle implémentation doit aussi inclure l'estimation du canal.

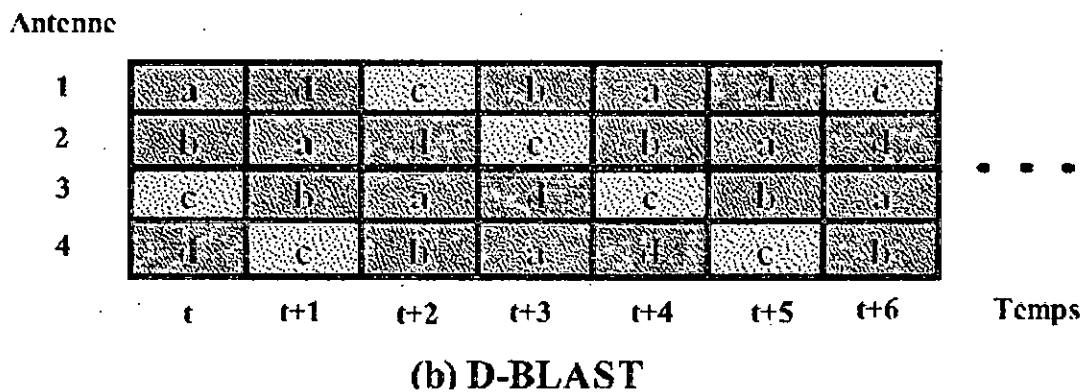
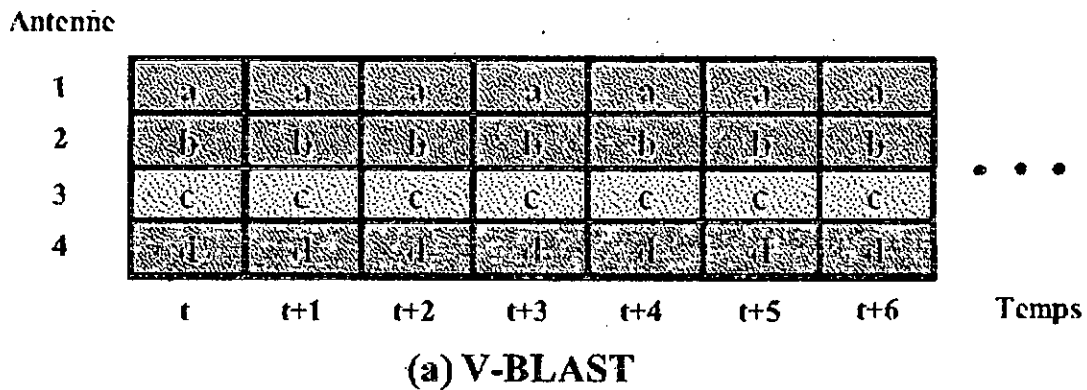


Figure 3.3 : Direction (*Assignment*) des séquences (*data streams*) aux antennes d'émission

Comme exemple de la performance atteinte par ces architectures, une implémentation expérimentale de V-BLAST est faite aux Bell Labs a atteint des efficacités spectrales entre 20 et 40 bps/Hz dans un environnement interne.[Fos98]

### 3.4 La détection V-BLAST

#### 3.4.1 Notations

Les vecteurs sont notés par des lettres minuscules et les matrices sont notées par des lettres majuscules. D'autres notations employées sont comme suit :

- $\overline{(\bullet)}$  Complément de  $(\bullet)$
- $(\bullet)^T$  Transposé de  $(\bullet)$
- $(\bullet)^*$  Transposé hermitien de  $(\bullet)$
- $(\bullet)^+$  Pseudo-inverse de  $(\bullet)$

- $(\bullet)^{-1}$  Inverse de  $(\bullet)$
- $\|(\bullet)\|$  Norme euclidienne de  $(\bullet)$
- $E\{(\bullet)\}$  Espérance de  $(\bullet)$
- $H^{N \times M}$  Matrice de dimension  $N \times M$
- $(H)_j$  Colonne  $j$  de la matrice  $H$
- $(H)_i$  Ligne  $i$  de la matrice  $H$

### 3.4.2 Modèle de base du système V-BLAST

Considérons un système V-BLAST avec  $M$  antennes d'émission et  $N \geq M$  antennes de réception, comme indiqué sur la Figure 3.4.

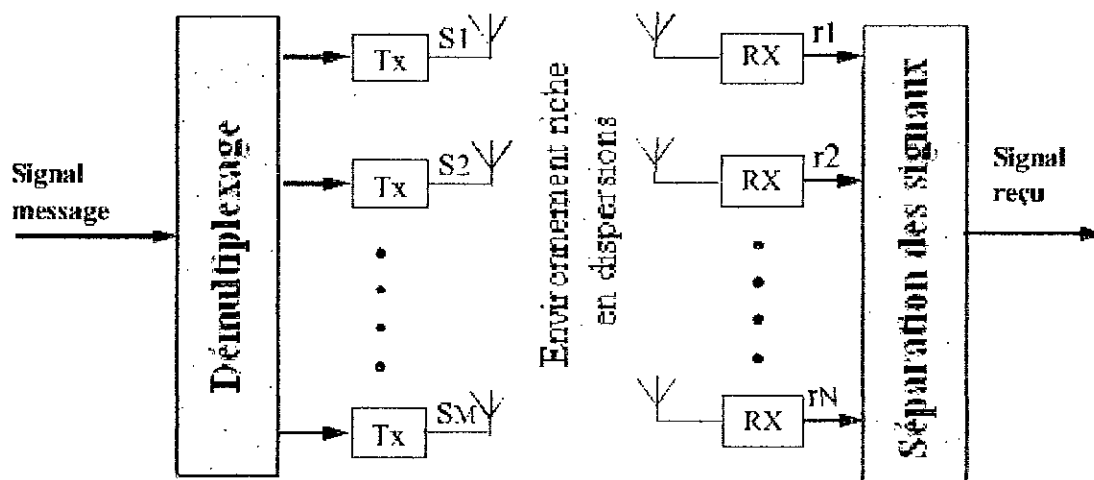


Figure 3.4 : Le modèle de base de V-BLAST

À l'émetteur la séquence d'entrée est démultiplexée (divisée) en  $M$  séquences, chaque séquence est modulée par la même constellation  $\Omega$  et envoyée à l'antenne d'émission correspondante. Ainsi, à chaque période d'un symbole un signal vecteur de taille  $M$ ,  $s = [s_1, s_2, \dots, s_M]^T$ , est envoyé au récepteur sur un canal sans fil à fading plat quasi-statique et riche en dispersion.

Au récepteur, le signal reçu  $r_i$  par l'antenne de réception  $i$  est la superposition de  $M$  signaux transmis bruités par le canal MIMO plus un bruit blanc gaussien additif (AWGN)  $v_i$ ,

$$r_i = \sum_{j=1}^M h_{ij} s_j + v_i, \quad (i = 1, 2, \dots, N) \quad (3.1)$$

Où  $h_{ij}$  sont les fonctions de transfert entre l'émetteur  $j$  et le récepteur  $i$ , qui sont des variables aléatoires complexes gaussiennes indépendantes identiquement distribuées de moyenne 0 et de variance unité, et  $v_i$  est aussi une variable aléatoire complexe gaussienne indépendante identiquement distribuée de moyenne zéro et de variance  $\sigma^2$ .

La supposition du fading plat quasi-statique signifie que le coefficient  $h_{ij}$  reste constant pendant un intervalle de temps contient plusieurs périodes d'émission, et change d'un intervalle à l'autre de façon indépendante. La supposition que le canal est riche en dispersion est effectuée quand les antennes sont suffisamment séparées (plusieurs longueurs d'onde) et dans le cas où le scénario contient plusieurs diffuseurs autour de l'émetteur ou du récepteur, cela implique aussi que le gain du canal soit complexe gaussien et indépendant d'un coefficient à un autre. Par conséquent, le système V-BLAST convient particulièrement aux environnements intérieurs. Quelques applications sont les réseaux sans fil fixes et les réseaux locaux sans fil (LAN).

Il est commode de décrire le modèle du système en notations matricielles :

$$r = Hs + v \quad (3.2)$$

où

$$H = \begin{bmatrix} h_{11} & \dots & h_{1M} \\ h_{21} & \dots & h_{2M} \\ \vdots & \dots & \vdots \\ h_{N1} & \dots & h_{NM} \end{bmatrix} = \begin{bmatrix} (H)_1 \\ \vdots \\ (H)_N \end{bmatrix} = [(H)_1, \dots, (H)_M] \quad (3.3)$$

$$r = \begin{bmatrix} r_1 \\ \vdots \\ r_N \end{bmatrix}, \quad s = \begin{bmatrix} s_1 \\ \vdots \\ s_M \end{bmatrix}, \quad v = \begin{bmatrix} v_1 \\ \vdots \\ v_N \end{bmatrix}$$

### 3.4.3 L'algorithme de détection ZF V-BLAST (Zero Forcing V-BLAST)

Théoriquement, la détection basée sur le maximum de vraisemblance serait optimale pour la détection V-BLAST. Cependant, elle est trop complexe pour l'implémenter. Par exemple,

dans le cas de 6 antennes d'émission avec une modulation QPSK, un total de  $4^6 = 4096$  comparaisons doit être fait pour chaque symbole transmis [Guo01].

Il est présenté dans [Fos98] qu'une basse complexité peut être obtenue en utilisant l'algorithme sous-optimal de détection ZF V-BLAST (Forçage à Zéro). À chaque période d'un symbole, il détecte d'abord le symbole transmis le plus fort (dans le sens du rapport signal sur bruit SNR à la réception), ensuite annule l'effet de ce symbole détecté de chacun des signaux reçus et continue ensuite à détecter le symbole le plus fort parmi les restant, et ainsi de suite. Supposons que le récepteur sait exactement la matrice du canal  $H$ , qui peut être accomplie par des moyens classiques d'estimation du canal, par exemple l'insertion des bits à l'émission pour connaître le canal à la réception.

### 3.4.3.1 Les étapes de l'algorithme ZF V-BLAST

L'algorithme ZF V-BLAST [Fos98] [Guo01] contient quatre étapes récursives :

➤ **Etape 1 : l'ordre optimal de détection (*Ordering*)**

Elle consiste à déterminer l'ordre optimal de détection pour les  $M$  symboles transmis, cela correspond à choisir l'ordre de la ligne de  $H^T$  qui a la norme euclidienne minimale parmi les  $M$  lignes de  $H^T$ .

$$G = H^T = (H^T H)^{-1} H^T \quad (3.4)$$

$$k = \arg \min_i \|(G)_i\|^2 \quad (3.5)$$

Où  $G$  est mentionnée comme la matrice de mise à zéro (*Nulling Matrix*).

Le choix de cet ordre sera expliqué après.

➤ **Etape 2: la mise à zéro (*Nulling*)**

Choisir la ligne  $(G)_k$  comme le vecteur de mise à zéro (*Nulling Vector*)  $w_k$ , utiliser ce vecteur pour obtenir le signal transmis le plus fort au sens du rapport signal sur bruit à la réception  $y_k$  et annuler tous les signaux restant.

$$w_k = (G)_k \quad (3.6)$$

$$y_k = w_k^T \cdot r \quad (3.7)$$

➤ **Etape3 : Quantification (*Slicing*)**

La valeur estimée du plus fort signal  $y_k$  est déterminée par le choix de la valeur la plus proche au  $y_k$  dans la constellation des signaux  $\Omega$ .

$$\hat{s}_k = \arg \min_{\hat{s} \in \Omega} \|\hat{s} - y_k\|^2 \quad (3.8)$$

➤ **Etape4 : l'annulation (*Cancellation*)**

Dès que le symbole le plus fort à la réception est détecté, son effet doit être éliminé du signal vecteur reçu pour réduire la complexité de détection pour les signaux restant.

$$r \leftarrow r - \hat{s}_k (H)_k \quad (3.9)$$

Egalement la  $k^{\text{ème}}$  colonne  $(H)_k$  de la matrice  $H$  doit être annulée,

$$H \leftarrow (H)_{\bar{k}} \quad (3.10)$$

$(H)_{\bar{k}}$  est la matrice  $H$  avec la  $k^{\text{ème}}$  colonne annulée.

Le processus ci-dessus correspond à la détection d'une couche, après la détection de cette couche nous retournons à la première étape pour détecter la couche suivante, et ainsi de suite jusqu'à la détection de toutes les  $M$  couches.

### 3.4.3.2 Détermination de vecteur de mise à zéro (*Nulling Vector*) [Fos98]

Pour le ZF V-BLAST, le  $k_i^{\text{ème}}$  vecteur de mise à zéro (*Nulling Vector*) est défini comme le seul vecteur ayant la norme minimale satisfaisant :

$$w_{k_i}^T (H)_{k_i} = \begin{cases} 0 & j > i \\ 1 & j = i \end{cases} \quad (3.11)$$

Il n'est pas difficile de démontrer que le vecteur (3.11) est tout simplement la  $k_i^{\text{ème}}$  ligne de  $H_{k_i}^+$ . Où la notation  $H_{k_i}^-$  signifie la matrice obtenue après l'annulation des colonnes  $k_1, k_2, \dots, k_i$  de  $H$  et  $^+$  désigne le pseudo-inverse de Moore-Penrose.

Le SNR post-déecté du  $k_i^{\text{ème}}$  symbole détecté est obtenue par remplacement de  $r_1 = Ha + v$  et (3.11) dans  $y_{k_i} = w_{k_i}^T r_1$  et après, on prend l'espérance

$$\rho_{k_i} = \frac{\langle |a_{k_i}|^2 \rangle}{\sigma^2 \|w_{k_i}\|^2} \quad (3.12)$$

Où l'espérance dans le numérateur est prise de la constellation  $\Omega$ .

Dans le cas de l'utilisation de l'annulation (*Cancellation*) des symboles, l'ordre de détection des symboles du vecteur transmis devient important du point de vue de la performance du système.

### 3.4.3.3 Ordre optimal et sous optimal de la détection ZF V-BLAST

Le meilleur ordre de détection peut être trouvé en essayant toutes les  $M!$  permutations possibles, mais c'est prohibitivement complexe. Dans [Fos98], un ordre optimal qui effectue une optimisation globale est présenté. Soit l'ordre :

$$P \equiv \{k_1, k_2, \dots, k_M\} \quad (3.13)$$

une permutation des entiers  $1, 2, 3, \dots, M$  qui détermine l'ordre selon lequel les composants (les couches) du signal vecteur transmis  $s$  sont détectés.

Le SNR [Fos98] de la  $k_i^{\text{ème}}$  couche détectée de  $s$  est :

$$\rho_{k_i} = \frac{E \left[ |s_{k_i}|^2 \right]}{\sigma^2 \|w_{k_i}\|^2} \quad (3.14)$$

Où l'espérance dans le numérateur est prise de la constellation  $\Omega$ . En raison de la même constellation,  $\rho_{k_i}$  est seulement déterminé par le vecteur  $w_{k_i}$  (*Nulling Vector*). Il est prouvé dans [Fos98] que simplement le choix du meilleur  $P_{k_i}$ , c'est-à-dire le minimum des  $\|w_{k_i}\|^2$  à chaque couche dans le processus de détection mène à l'ordre optimal de détection. Une nouvelle réduction du nombre de calculs des pseudo-inverses peut être réalisée en utilisant un algorithme d'ordre sous-optimal proposé par [Won00]. Au lieu de l'ordre des couches restantes après chaque couche détectée, l'ordre sous-optimal est obtenu en triant dans un ordre croissant, pour  $i = 1, 2, \dots, M$ , seulement une fois en détectant la première couche. Ainsi,



la complexité d'ordre est réduite sans causer une dégradation significative dans la performance générale [Guo01].

### 3.4.3.4 Forme compacte de L'algorithme complet ZF V-BLAST

L'algorithme complet de détection ZF VBLAST peut être décrit d'une façon compacte comme une procédure récursive incluant la détermination de l'ordre optimal, comme suit :

**Initialisation :**

$$i \leftarrow 1 \quad (3.15a)$$

$$G_1 = H^T = (H^* H)^{-1} H^* \quad (3.15b)$$

$$k_1 = \arg \min_j \|(G_1)_j\|^2 \quad (3.15c)$$

**Récursion :**

$$w_{k_i} = \underline{(G_i)_{k_i}} \quad (3.15d)$$

$$y_{k_i} = w_{k_i}^T r_i \quad (3.15e)$$

$$\hat{s}_{k_i} = \arg \min_{\hat{s} \in \Omega} \|\hat{s} - y_{k_i}\|^2 \quad (3.15f)$$

$$r_{i+1} = r_i - \hat{s}_{k_i} (H)_{k_i} \quad (3.15g)$$

$$G_{i+1} = H_{k_i}^{-1} \quad (3.15h)$$

$$k_{i+1} = \arg \min_{j \in \{k_1, \dots, k_i\}} \|(G_{i+1})_j\|^2 \quad (3.15i)$$

$$i \leftarrow i+1 \quad (3.15j)$$

### 3.4.3.5 Complexité de calcul et stabilité numérique

La principale complexité de calcul de l'algorithme ZF V-BLAST est au niveau de l'ordre optimal de détection (*Ordering*), qui consacre plus de 80 % du coût de calcul total (incluant l'estimation du canal). C'est dû aux calculs répétés des matrices pseudo-inverses impliqués dans chaque étape d'ordre. Aussi, quand le nombre d'antennes d'émission et de réception est grand, les calculs répétés des matrices pseudo-inverses pourraient mener à une instabilité

numérique [Dam01]. Cela signifie qu'un petit changement dans  $H$  pourrait causer un grand changement dans  $H^{-1}$ . Donc, on suppose un algorithme qui a une petite complexité de calcul et une stabilité numérique [Guo01].

Les erreurs à la détection sont générées à l'étape de mise à zéro (*Nulling*) et de quantification (*Slicing*) et sont accumulées à l'étape d'annulation (*Cancellation*).

Une solution pour améliorer la performance de mise à zéro est de remplacer la mise à zéro ZF "ZF *Nulling*" par l'erreur quadratique minimale moyenne "MMSE *Nulling*", qui améliore la performance du système sous ZF V-BLAST par 8 dB à un taux d'erreur de  $10^{-3}$  pour un système (4, 4) [Bar01].

### 3.4.4 L'algorithme MMSE V-BLAST (Minimum Mean Square Error)

Une solution pour améliorer la performance de mise à zéro est de remplacer la mise à zéro ZF "ZF *Nulling*" par l'erreur quadratique minimale moyenne "MMSE *Nulling*". La différence entre MMSE et ZF se situe au niveau de calcul des vecteurs d'annulation (*Nulling Vectors*)  $w_{k_i}$ .

Pour MMSE  $w_{k_i} = \underline{(G_i)_{k_i}}$

Tel que,

$$G_{i+1} = \left( H_i^* H_i + \frac{N_0}{E_s} I_{(M-i) \times (M-i)} \right)^{-1} H_i^* \quad , \text{ [Bar01]} \quad (3.16)$$

Donc l'algorithme MMSE V-BLAST est tout simplement le suivant :

**Initialisation :**

$$i \leftarrow 1 \quad (3.17a)$$

$$G_i = H^{\dagger} = \left( H^* H + \frac{N_0}{E_s} I_{M \times M} \right)^{-1} H^* \quad (3.17b)$$

$$k_i = \arg \min_j \left\| \underline{(G_i)_j} \right\|^2 \quad (3.17c)$$

**Récursion :**

$$w_{k_i} = \underline{(G_i)_{k_i}} \quad (3.17d)$$

$$y_{k_i} = w_{k_i}^T r_i \quad (3.17e)$$

$$\hat{s}_{k_i} = \underset{\hat{s} \in \Omega}{\operatorname{argmin}} \|\hat{s} - y_{k_i}\|^2 \quad (3.17f)$$

$$r_{i+1} = r_i - \hat{s}_{k_i} (H)_{k_i} \quad (3.17g)$$

$$G_{i+1} = H_{k_i}^{-1} \quad (3.17h)$$

$$k_{i+1} = \underset{j \notin \{k_1, \dots, k_i\}}{\operatorname{argmin}} \left\| \underline{(G_{i+1})_j} \right\|^2 \quad (3.17i)$$

$$i \leftarrow i+1 \quad (3.17j)$$

### 3.5 Conclusion

Dans ce chapitre, on a présenté l'architecture *BLAST* avec ses deux variétés *D-BLAST* et *V-BLAST* et on l'a modélisé aussi pour faciliter son étude. On a présenté également l'algorithme original de détection *ZF V-BLAST* avec ses différents détails ainsi que sa version améliorée *MMSE V-BLAST* qui améliore considérablement la performance de la détection (diminuer le taux d'erreur à la détection).

On essaiera dans le chapitre suivant de simuler sous *MATLAB* ces deux algorithmes : *ZF V-BLAST* et *MMSE V-BLAST*, d'analyser et d'évaluer ses résultats.

## Chapitre 4

# Simulation et résultats

### 4.1 Introduction

Nous avons étudié dans le chapitre précédent les algorithmes ZF V-BLAST et MMSE V-BLAST, nous évaluons maintenant les performances de ces algorithmes au niveau de la détection. Pour cela, nous avons simulé sous MATLAB une chaîne de transmission pour évaluer ces deux algorithmes.

La simulation est conçue comme suit :

1. Génération d'un signal aléatoire à partir d'une constellation M-QAM.
2. Génération d'une matrice aléatoire complexe du canal, ses coefficients sont des variables aléatoires complexes gaussiennes de moyennes nulles et de variances unités.
3. Génération d'un vecteur complexe de bruit AWGN, ses coefficients sont des variables aléatoires complexes gaussiennes de moyennes nulles et de variances unités.
4. Constitution du vecteur reçu, qui est un mélange des vecteurs émis subissant des distorsions dues au canal plus l'influence du bruit AWGN.
5. Ensuite, l'application des algorithmes de détection (ZF V-BLAST ou MMSE V-BLAST) sur le signal vecteur reçu.
6. Enfin, la comparaison des signaux émis avec ceux détectés et le calcul du taux d'erreur par bit (*BER*).

Pour évaluer efficacement la performance de détection de ces deux algorithmes, des réalisations ont été faites selon le tableau 4.1 :

| Tests<br>Modulation | Nombre de<br>Simulations | Nombre de<br>Symboles testés | Nombre de<br>Bits testés |
|---------------------|--------------------------|------------------------------|--------------------------|
| 4-QAM               | 12 500                   | 50 000                       | 100 000                  |
| 8-QAM               | 4 500                    | 36 000                       | 108 000                  |
| 16-QAM              | 1 600                    | 25 600                       | 102 400                  |

Tableau 4.1 : Statistiques des réalisations

Les résultats obtenus seront représentés dans ce chapitre comme suit :

- L'efficacité de l'algorithme ZF V-BLAST
- Comparaison entre le codage binaire naturel et le codage de Gray appliqués au technique V-BLAST
- L'effet de la diversité spatiale sur la performance de la transmission
- Inconvénient de l'algorithme ZF V-BLAST sans ordre optimal de détection
- Utilisation de la technique d'ordre optimal
- Utilisation de l'algorithme MMSE V-BLAST

Les programmes réalisés sous MATLAB pour l'évaluation de la capacité du canal MIMO et les algorithmes ZF V-BLAST et MMSE V-BLAST sont donnés en Annexe.

## 4.2 Résultats de la simulation

Dans cette section nous présentons les résultats de la simulation pour évaluer la performance des algorithmes de détection ZF V-BLAST et MMSE V-BLAST selon l'ordre donné ci-dessus.

### 4.2.1 L'efficacité de l'algorithme ZF V-BLAST

La figure 4.1 montre l'efficacité de l'algorithme ZF V-BLAST sans ordre optimal de détection pour une constellation 4-QAM ou QPSK. Elle compare également notre résultat de la simulation et le résultat de Baro et Bauch de " Institute for Communications Engineering, Munich University of Technology" [Bar01].

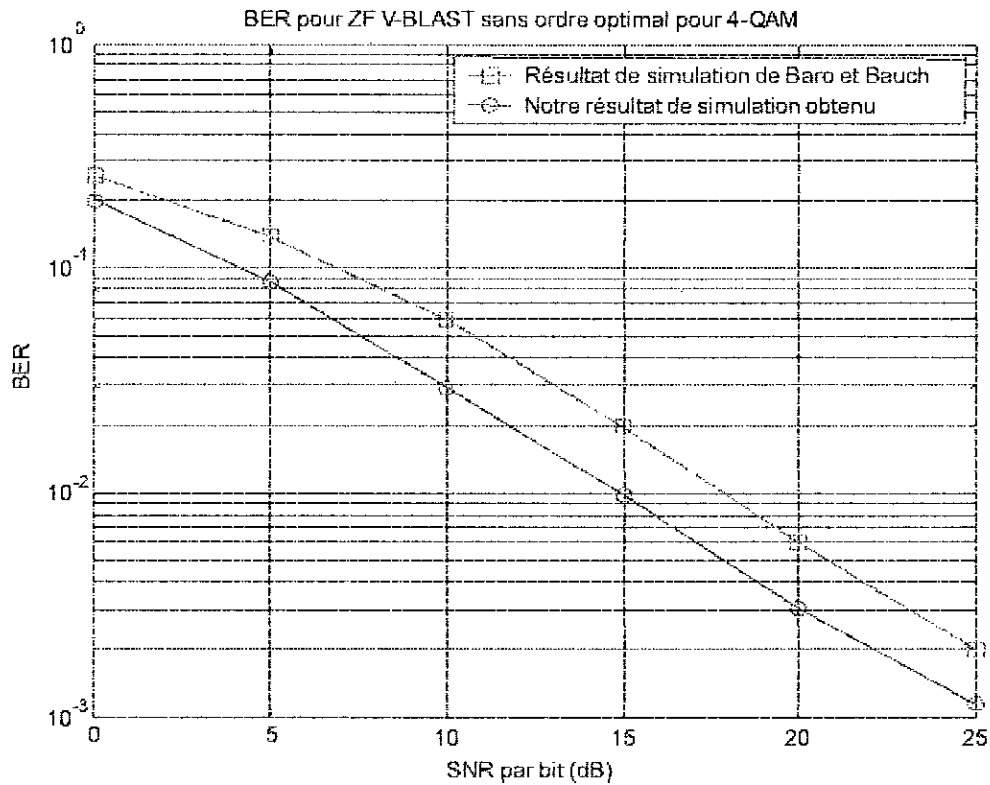


Figure 4.1 : Comparaison des résultats de notre simulation et celle réalisée par [Bar01] pour une ZF V-BLAST 4-QAM sans ordre optimal de détection [Programme 4].

Nous voyons clairement que le taux d'erreur par bit diminue quand le rapport signal sur bruit (SNR) augmente, cela est conforme avec les résultats théoriques du fait que la probabilité d'erreur dépend de la distance euclidienne entre les points adjacents de la constellation et que cette probabilité diminue quand cette distance entre les points adjacents augmente, ceci reste valable quand le niveau du signal augmente ou la puissance du signal augmente, l'influence du bruit sur la transmission diminue. Afin de valider nos résultats une deuxième constatation concernant la comparaison entre nos résultats et ceux données par Baro et Bauch [Bar01], nous remarquons que la forme des deux courbes est la même avec une petite différence, par exemple si nous prenons la valeur du SNR=10, nous avons trouvé un taux d'erreur de 3 bits erronés par 100 bits, Baro et Bauch ont trouvé un taux d'erreur de 6 bits erronés par 100 bits. Pour la valeur 25 dB du SNR, nous avons trouvé un taux de 0.001 mais Baro et Bauch ont trouvé 0.002, nous pouvons considérer que ces différences sont acceptables et peuvent être dues aux différences des données ou des paramètres prises à la simulation.

### 4.2.2 Comparaison entre le codage binaire naturel et le codage de Gray utilisés a la technique V-BLAST

Dans la deuxième étape, nous comparons les performances de codage binaire naturel et le codage de Gray appliqués au niveau de cette technique (algorithme V-BLAST). La figure 4.2 montre l'amélioration effectuée par l'utilisation du codage de Gray au lieu du codage binaire naturel.

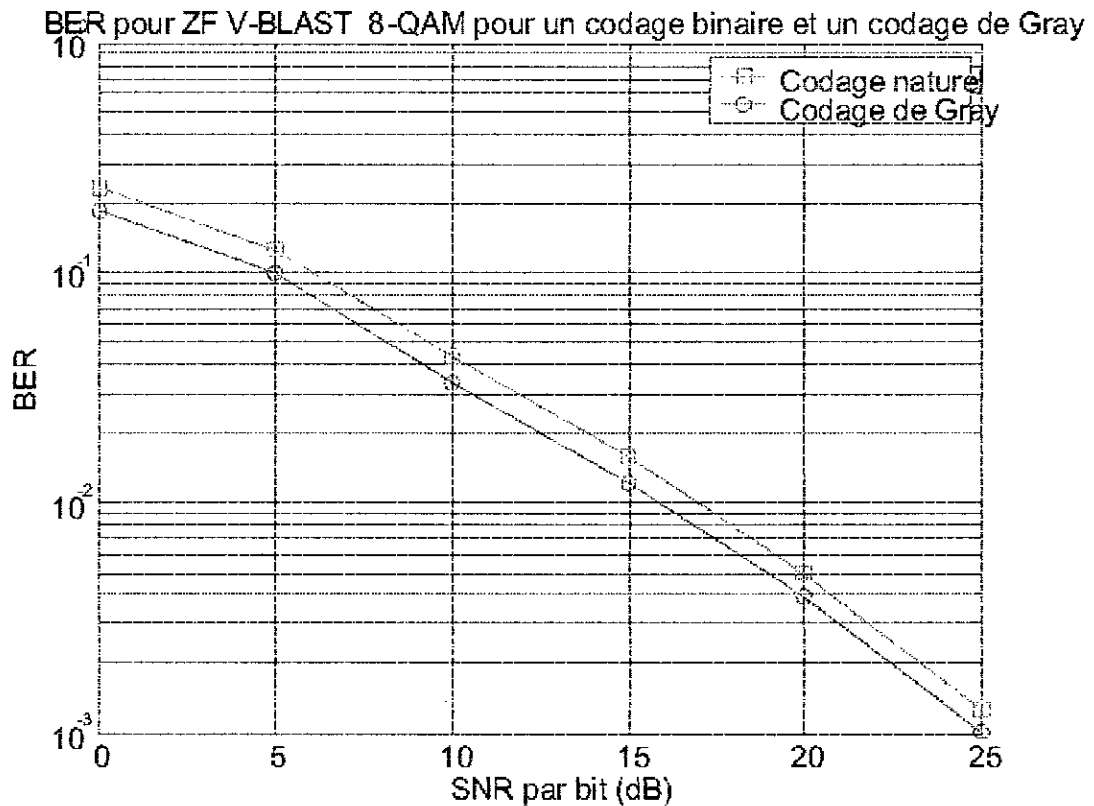


Figure 4.2 : Comparaison des performances des codes Gray et Binaire Naturel appliqués à l'algorithme ZF V-BLAST pour une Constellation 8-QAM [Programme5].

On remarque également une différence pour les constellations 4QAM et 16-QAM comme la montre la figure 4.3.

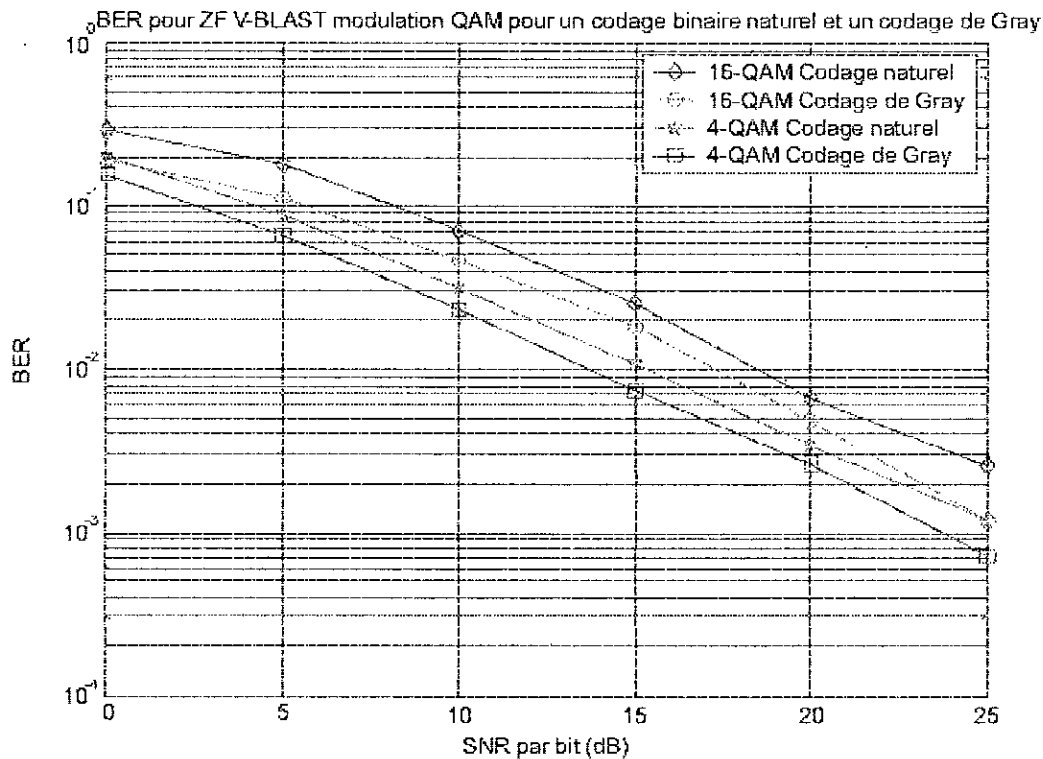


Figure 4.3 : Comparaison des performances des codes Gray et Binaire naturel appliqués à l'algorithme ZF V-BLAST pour les constellation 4-QAM et 16-QAM.

On peut dire que cette amélioration est logique et facile à comprendre l'avantage du code Gray. En effet, pour le code Gray chaque deux points adjacents de la constellation sont codés par deux nombres binaires qui se différencient d'un bit seulement, ce qui minimise automatiquement le Taux Erreur Bit à la détection, ce qui n'est pas le cas si on utilise le code binaire naturel. C'est pour cette raison nous utiliserons uniquement le code Gray dans ce qui nous reste de la simulation.

### 4.2.3 L'effet de la diversité spatiale sur la performance de la transmission

Nous verrons maintenant l'effet de la diversité spatiale au niveau d'émission et de la réception sur la performance de la transmission. Nous avons vu dans le deuxième chapitre que l'efficacité spectrale augmente linéairement avec le nombre d'antennes d'émission et de réception (Figure 2.4). La question qui se pose maintenant, avec quelle performance (au sens du BER) cette efficacité spectrale augmente?

La figure 4.4 montre la performance de détection en fonction de la diversité spatiale.



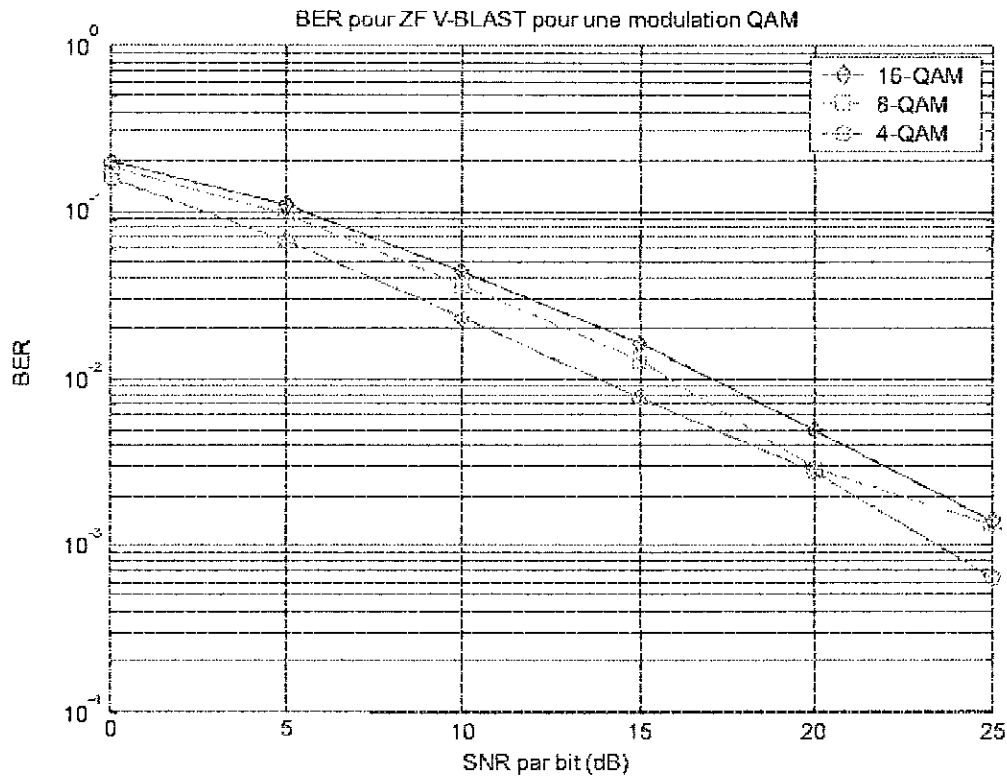


Figure 4.4 : La performance de la transmission en fonction de la diversité spatiale [Programme6].

Nous remarquons bien qu'il n'y a pas une grande dégradation au niveau de BER si on augmente la diversité spatiale, par exemple pour la valeur du SNR= 0, on trouve presque les mêmes BER pour les trois cas. Pour la valeur du SNR=10, BER=0.024 pour 4-QAM, BER=0.037 pour 8-QAM et BER=0.048 pour 16-QAM. Mais si on combine l'avantage de l'efficacité spectrale et la performance de la détection, on peut dire que la diversité spatiale dès qu'elle augmente réalise un grand avantage pour le système et la transmission en générale.

#### 4.2.4 Inconvénient de l'algorithme ZF V-BLAST sans ordre optimal de détection

La figure 4.5 représente le taux d'erreur par Bit (BER) calculé pour chaque antenne d'émission en fonction du rapport signal sur bruit, c-à-d le taux de nombre de bits erronés à la détection d'une couche sur le total des bits émis par l'antenne qui correspond à cette couche.

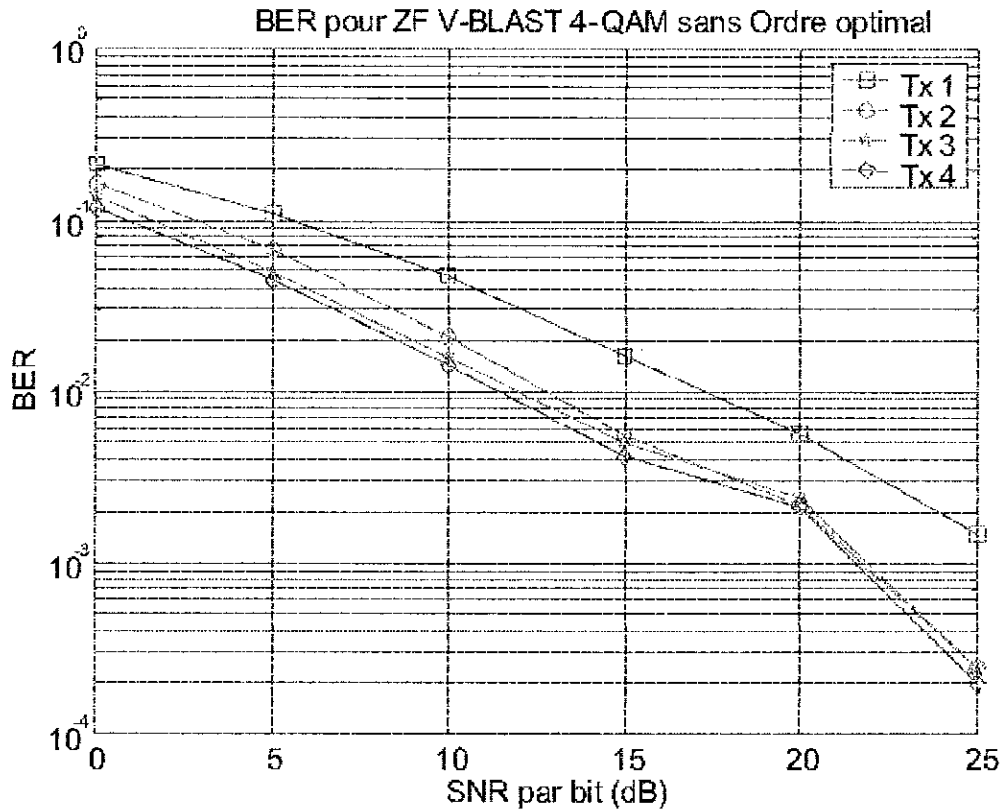


Figure 4.5 : Taux Erreur Bit pour chaque antenne d'émission pour une constellation 4-QAM sans utilisation de la technique d'ordre optimal de la détection [Programme 7].

Quand la première couche est détectée et soustraite du système, la détection des couches suivantes s'améliore. Cela signifie que nous faisons la suppression d'interférence due au signal de la première couche. Donc l'élimination des couches déjà détectées signifie l'élimination des interférences par rapport aux couches qui ne sont pas encore détectées. Et cela minimise automatiquement le BER pour les couches à détecter par rapport à celles déjà détectées.

Dans cette simulation pour la figure 4.5, nous n'avons pas utilisé la technique d'ordre optimal de détection. Cela nous permet de séparer la performance pour le signal provenant de l'antenne 1 qui est détecté le premier jusqu'au signal provenant de l'antenne 4 détecté le dernier.

Comme nous voyons dans la figure 4.5, le plus grand BER est enregistré pour l'antenne 1, et diminue chaque fois qu'un symbole d'une couche est détecté et éliminé du signal reçu pour les antennes restantes qui profitent de la soustraction des symboles précédemment détectés.

### 4.2.5 Utilisation de la technique d'ordre optimal de détection (Ordering)

Plusieurs méthodes sont possibles si on veut améliorer la performance de V-BLAST. Une première technique, proposée dans [Fos98], encourage l'ordre optimal de détection des signaux à détecter. Le rapport signal sur bruit (SNR) post-détection est employé comme un critère d'ordre optimal des signaux à détecter, voir le paragraphe 3.4.3.3).

L'avantage d'ordre optimal de détection (*Ordering*) est représenté sur la figure 4.6 qui montre la performance de V-BLAST pour chaque couche du système et sur la figure 4.7 qui représente l'amélioration de la performance de détection du système entier dans le cas d'utilisation de la technique d'ordre optimal de détection.

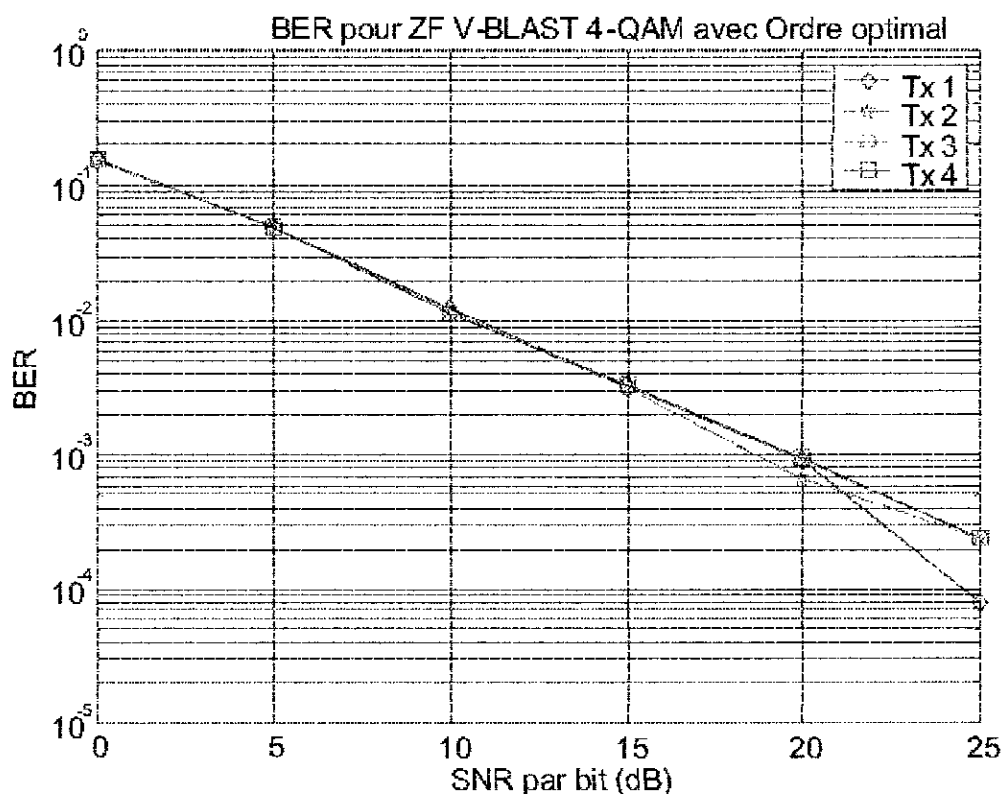


Figure 4.6 : Taux Erreur Bit pour chaque antenne d'émission pour une constellation 4-QAM avec utilisation de la technique d'ordre optimal de la détection [Programme8].

Nous remarquons que les quatre courbes qui représentent les performances des quatre couches sont presque confondues ce qui signifie la même probabilité de BER pour la détection des signaux provenant de n'importe quelle antenne d'émission. On peut remarquer aussi la diminution de BER par rapport à la simulation précédente concernant ZF V-BLAST sans ordre optimal de détection, cela peut être confirmé sur la figure 4.7:

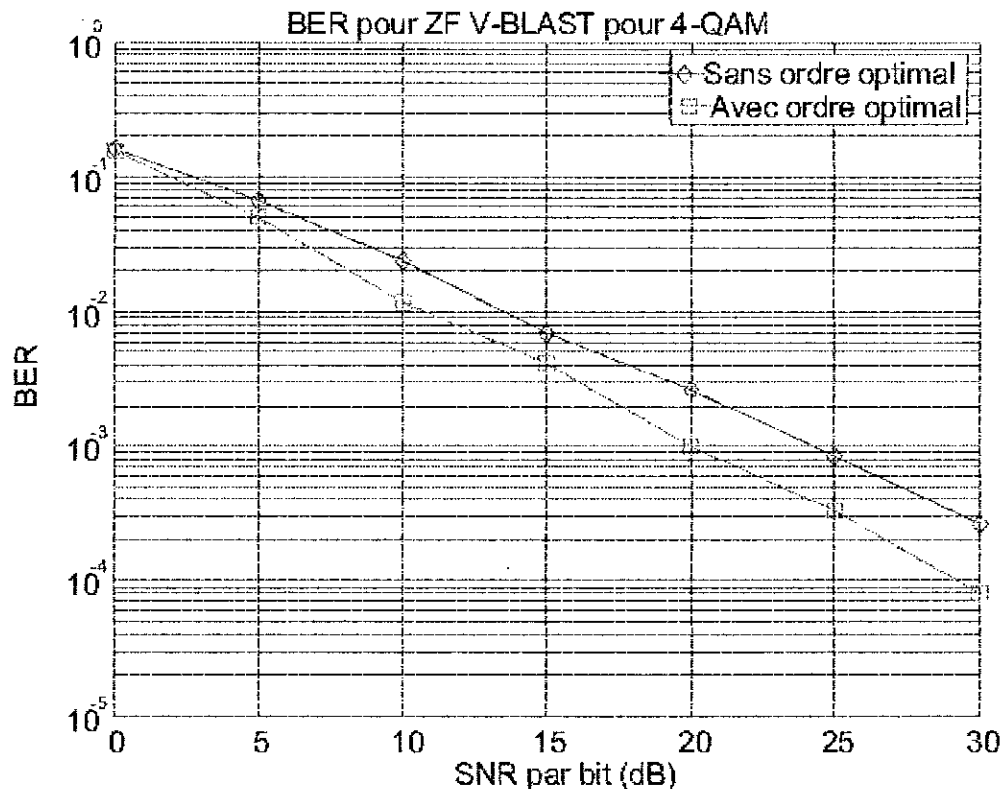


Figure 4.7 : Comparaison des résultats de ZF V-BLAST sans et avec ordre optimal de détection [Programme9].

Nous observons bien la diminution de BER dans le cas d'utilisation d'ordre optimal de détection (Ordering) surtout dans les grandes valeurs du rapport signal sur bruit (SNR).

Mais l'opération d'ordre optimal de détection (Ordering) deviendra très complexe si on utilise un nombre d'antennes supérieures ou égales à 8, elle exige beaucoup de test et beaucoup d'études de cas. C'est pour ça, dans ce mémoire nous nous limiteront l'application d'ordre optimal de détection au système  $4 \times 4$ .

#### 4.2.6 Utilisation de l'algorithme MMSE V-BLAST

Une autre technique pour améliorer la performance de détection est de remplacer Zero-Forcing Nulling proposée dans [Fos98] par l'algorithme puissant MMSE (Minimum Mean Square Error) (voir section 3.2.4). En plus d'annuler les interférences, le niveau du bruit sur le canal est tenu en compte. Un inconvénient est, cependant, que le SNR doit être connu et estimé au récepteur.

La figure 4.8 représente la comparaison entre l'algorithme ZF V-BLAST et MMSE V-BLAST sans et avec ordre optimal de détection.

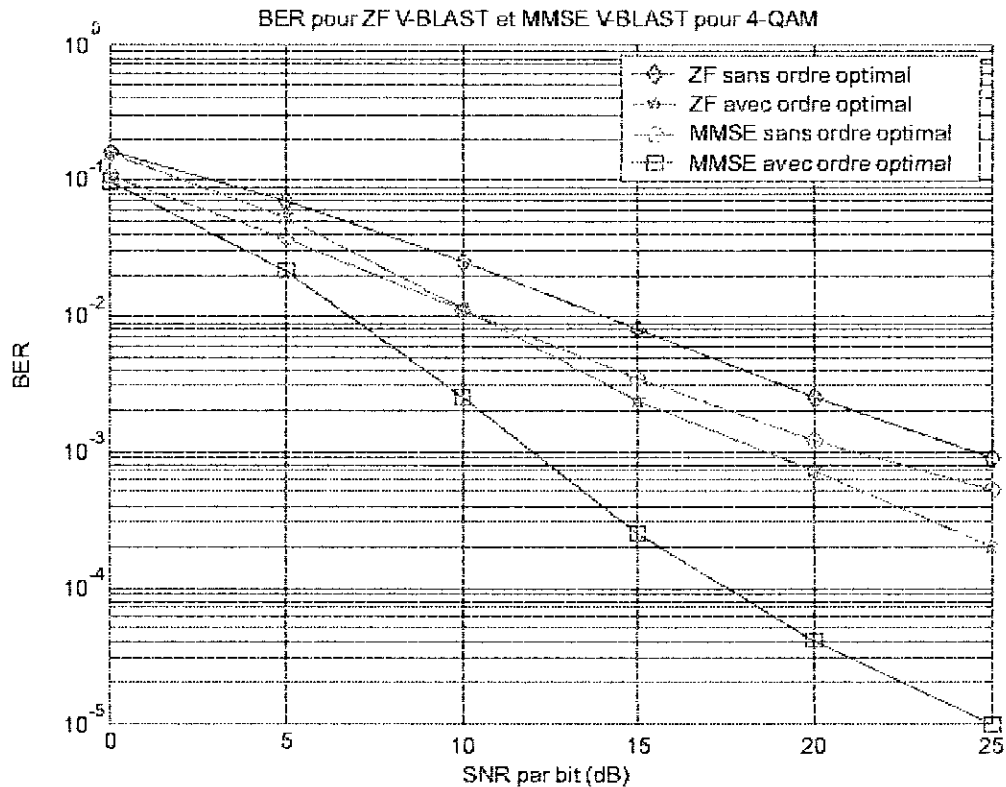


Figure 4.8 : Comparaison entre ZF V-BLAST et MMSE V-BLAST sans et avec ordre optimal de détection [Programme 10].

Nous remarquons que l'ordre optimal des signaux améliore la performance de la détection et minimise remarquablement le taux d'erreur par bit.

Pour ZF V-BLAST avec ordre optimal de détection, l'amélioration est approximativement de 4 dB à un taux d'erreur de  $10^{-3}$  par rapport à ZF sans ordre optimal de détection. Si nous appliquons de plus MMSE V-BLAST, l'amélioration comparée avec ZF est seulement 2 dB en négligeant l'ordre optimal. La combinaison de MMSE et l'ordre optimal de détection effectue une amélioration de 8 dB comparée avec ZF avec ordre optimal et 12 dB comparée avec ZF originale sans ordre optimal.

On peut remarquer aussi la grande performance réalisée par MMSE V-BLAST avec ordre optimal de détection par rapport aux autres techniques surtout au niveau des grandes valeurs du rapport signal sur bruit. Donc l'algorithme MMSE V-BLAST est le plus performant dans ce niveau.

### 4.3 Conclusion

Nous avons montré la performance de l'algorithme ZF appliquée au système V-BLAST et évalué plusieurs possibilités d'amélioration de cette performance. Nous avons d'abord analysé l'amélioration obtenue selon une solution MMSE comparée avec Zero-Forcing. En outre, nous avons montré les avantages et les améliorations obtenues par l'utilisation d'ordre optimal de détection des signaux.

Nous résumons les plus importants résultats obtenus de la simulation par :

1. les algorithmes ZF V-BLAST et MMSE V-BLAST peuvent réaliser une grande partie de la capacité théorique du canal MIMO à partir des détections performantes à la réception.
2. le code Gray améliore la performance de la détection par rapport au code binaire naturel.
3. la technique d'ordre optimal de détection effectue une amélioration importante de la performance de détection.
4. l'algorithme MMSE V-BLAST constitue une technique plus efficace pour diminuer le taux d'erreur par bit et pour améliorer la performance de détection.

## Conclusion générale

L'objectif de notre travail est l'évaluation de la performance de l'architecture V-BLAST au niveau de la détection en étudiant les algorithmes liés à cette architecture (ZF V-BLAST et MMSE V-BLAST) afin d'exploiter les grandes capacités offertes par les canaux MIMO.

Le premier chapitre a présenté des généralités sur la communication numérique. Le deuxième chapitre a été consacré à l'introduction et la description du canal sans fil MIMO, il a présenté également les modèles stochastiques et la grande capacité offerte par les canaux MIMO. Le troisième chapitre était une introduction au concept d'architecture BLAST. Son but était de présenter le modèle du système V-BLAST et d'expliquer bien les algorithmes appliqués au niveau de la détection; ZF et MMSE V-BLAST. Dans le dernier chapitre nous avons présenté les résultats de simulation de l'algorithme ZF V-BLAST et ses versions améliorées. Le premier résultat a montré bien l'efficacité de l'algorithme ZF V-BLAST qui nous permet de dire que l'architecture V-BLAST exploite bien les grandes capacités offertes par les systèmes MIMO avec moins de taux d'erreur. Nous avons montré aussi l'avantage du codage de Gray par rapport le binaire naturel. Nous avons évalué également des possibilités d'amélioration de ZF V-BLAST comme l'utilisation de la technique d'ordre optimal de détection et l'algorithme MMSE V-BLAST qui ont présenté des améliorations considérables de la performance de détection.

A partir des résultats obtenus, nous pouvons dire que cette architecture peut exploiter les grandes capacités offertes par les canaux MIMO, donc elle peut trouver son application dans les services de transmission à haut débit comme l'internet sans fil à haut débit.

Cette architecture peut être combinée avec d'autres techniques pour améliorer en plus les performances de détection. L'utilisation du codage par block et décodage Turbo ou encore le codage en Trellis en combinaison avec V-BLAST améliore considérablement la performance de détection et diminue le taux d'erreur. Il existe d'autres techniques permettant l'amélioration de la quantification durant le processus de détection comme l'utilisation de la décision douce au lieu de la décision ferme, ou l'utilisation du Turbo BLAST qui permet de diminuer considérablement le BER.

## Annexe

### Programmes MATLAB

#### 1- Capa\_nombre.m :

```

%
%-----Capa_nombre.m-----
% Variation de la capacité du canal MIMO en fonction des nombres
% d'antennes d'émission et de la réception (or les prend égaux)
%
%
snr = '0 10 20'; % Rapport signal sur bruit (SNR)
Pt = 10.^(snr/10); % la puissance totale selon snr
Ns = 1000; % Nombre de simulations
N=1:20; % N est le nombre des antennes
Cp1=zeros(1,20);
Cp2=zeros(1,20);
Cp3=zeros(1,20);
r=1;
while r<=3,
    for M=1:20,
        % Génération de Ns matrices aléatoires du canal
        E = 1/sqrt(2)*randn(M,M,Ns)+i*1/sqrt(2)*randn(M,M,Ns);

        C_t=C;
        for r=1:Ns,
            [U S V] = svd(E(:,:,r)*E(:,:,r)');
            Vp=diag(S);
            C=C;
            for s=1:M,
                C= C+log2(1+Pt(r)/M*Vp(s));
            end
            C_t=C_t+C;
        end
        C_moy=C_t/Ns;
        if r==1
            Cp1(M)=C_moy;
        elseif r==2
            Cp2(M)=C_moy;
        elseif r==3
            Cp3(M)=C_moy;
        end
    end
    r=r+1;
end
plot(N,Cp3,'o-d',N,Cp2,'x-p',N,Cp1,'w->');grid
xlabel('Nombre d'antennes M');ylabel('Capacité (bits/s/Hz)');
title('Capacité du canal MIMO en fonction de nombre d'antennes');
legend('SNR=20 dB','SNR=10 dB','SNR=0 dB');

```



## 2- Capa\_snr.m :

```

%
%-----Capa_snr.m-----
%      Simulation de la capacité d'un canal MIMO en fonction      %
%      du rapport signal sur bruit (SNR)                          %
%-----
%
snr = '0 : 2 : 40';          % Rapport signal sur bruit (SNR)
Pt = 10.^(snr/10);          % la puissance totale selon snr
Ns = 1000;                  % Nombre de simulations
Np = length(snr);

C_moy8=zeros(1,Np);
C_moy16=zeros(1,Np);
r=1;
while r<=Np,
    if r==1,
        M=8,N=8; % Nombre des antennes d'émission et de réception
    else
        M=16,N=16; % Nombre des antennes d'émission et de réception
    end

    % Génération de Ns matrices aléatoires du canal
    E = 1/sqrt(2)*randn(N,M,Ns)+i*1/sqrt(2)*randn(N,M,Ns);

    % Calcul de la capacité du canal
    k=1;
    while k<=Np
        C_t=0;
        for r=1:Ns,
            'U S V' = svd(E(:, :,r)*E(:, :,r)');
            Vp=diag(S);
            C=C;
            for s=1:min(M,N),
                C=C+log2(1+Pt(k)/M*Vp(s));
            end
            C_t=C_t+C;
        end
        if r==1
            C_moy8(k)=C_t/Ns;
        else
            C_moy16(k)=C_t/Ns;
        end
        k=k+1;
    end
    r=r+1;
end
plot(snr,C_moy16,'b-o',snr,C_moy8,'r-d');
xlabel('SNR (dB)'); ylabel('Capacité (bps/Hz)');grid;
title('Capacité du canal MIMO en fonction du SNR');
legend('N-M=16', 'N-M=8');

```

## 3- Efficacite.m :

```

%
%-----Efficacite.m-----
% Simulation de la capacité d'un canal MIMO en fonction du rapport signal &
% sur bruit (SNR) pour différentes dimensions de sa matrice aléatoire E %
%-----
%
snr = '0 : 2 : 30'; % Rapport signal sur bruit (SNR)
Pt = 10.^(snr/10); % la puissance totale selon snr
Ns = 1000; % Nombre de simulations
Np = length(snr);

C_moy1=zeros(1,Np);
C_moy2=zeros(1,Np);
C_moy4=zeros(1,Np);
C_moy8=zeros(1,Np);
C_moy16=zeros(1,Np);
r=1;
while r<=Np,
    if r==1,
        M=1;N=1; % Nombre des antennes d'émission et de réception
    elseif r==2
        M=2;N=2; % Nombre des antennes d'émission et de réception
    elseif r==3,
        M=4;N=4; % Nombre des antennes d'émission et de réception
    elseif r==4
        M=8;N=8; % Nombre des antennes d'émission et de réception
    elseif r==5,
        M=16;N=16; % Nombre des antennes d'émission et de réception
    end
    % Génération de Ns matrices aléatoires du canal
    E = 1/sqrt(2)*randn(N,M,Ns)+i*1/sqrt(2)*randn(N,M,Ns);

    % Calcul de la capacité du canal
    k=1;
    while k<=Np
        C_t=0;
        for r=1:Ns,
            [U S V] = svd(E(:, :, r)*E(:, :, r)');
            Vp=diag(S);
            C=C;
            for s=1:min(M,N),
                C=C+log2(1+Pt(k)/M*Vp(s));
            end
            C_t=C_t+C;
        end
        if r==1,
            C_moy1(k)=C_t/Ns;
        elseif r==2
            C_moy2(k)=C_t/Ns;
        elseif r==3,
            C_moy4(k)=C_t/Ns;
        elseif r==4
            C_moy8(k)=C_t/Ns;
        elseif r==5,
            C_moy16(k)=C_t/Ns;
        end
        k=k+1;
    end
end

```

```

    r=r+1;
end
plot(snr,C_moy16,'o-d',snr,C_moy8,'r-o',snr,C_moy4,'w-g',snr,C_moy2,
'o-+',snr,C_moy1,'r.-');
xlabel('SNR (dB)'), ylabel('Capacité (bps/Hz)'); grid;
title('Capacité du canal RING en fonction de SNR');
legend('M-N-16','M-N-8','M-N-4','M-N-2','M-N-1')

```

#### 4- ZF\_so4.m :

```

%
%-----ZF_so4.m-----
% Simulation de l'algorithme ZF V-BLAST sans ordre optimal pour 4-QAM %
% et comparaison entre le résultat de cette simulation et celle %
% de Balz et Bauch 'Barcl' %
%-----
%
M=4; N=4; % Nombre d'antennes d'émission et de réception
snr = 'C : 5 : 25'; % Rapport signal sur bruit par bit (SNR) en dB
Pt = 8*10.^(snr/10); % La puissance totale selon snr
Np = length(snr);
Ns = 12500; % Nombre de simulations

BER=0;
for np=1:Np,
    Pt=Pt(np);
    SS='';SeSe='';
    for ns=1:Ns,
        M=4;i=sqrt(-1);
        % Génération des matrices aléatoires du canal
        E = 1/sqrt(2)*randn(N,M)+i*1/sqrt(2)*randn(N,M);
        % Génération d'un vecteur de bruit aléatoire gaussien (RACN)
        B=wgn(N,1,C)+i*wgn(N,1,C);
        % La constellation 4-QAM
        C=sqrt(Pt/M)/sqrt(2)*[1+i -1+i -1-i 1-i]';
        % Génération du signal vecteur transmis de la constellation 4-QAM
        S=sqrt(Pt/M)/sqrt(2)*(randarc(M,1)+i*randarc(M,1));
        R=E*S+B; % Le signal vecteur reçu R=[R1 R2 R3 R4]'
        %-----* La détection du premier symbole *-----
        Se=zeros(M,1);
        it=1;
        G=inv(E'*E)*E';
        Wk=G(1,:);
        Yk=dot(Wk',R);
        Sc='';
        for s=1:4,
            Ss=norm(C(s)-Yk)^2;
            Sc=[Sc;Ss];
        end
        s=1;
        while s<=4,
            if Sc(s)==min(Sc)
                Sk=C(s);
            end
            s=s+1;
        end
        Se(it)=Sk;
        M=M-1;
        %-----* La détection des symboles restant *-----
        for it=2:4,
            R=R-Sk*E(:,1);

```

```

E=E(:,2:M+1);
G=inv(E'*E)*E';
Wk=G(1,:);
Yk=dot(Wk',R);
Sc='';
for s=1:4,
    Se=norm(C(s)-Yk)^2;
    Sc=[Sc;Se];
end
s=1;
while s<=4,
    if Sc(s)==min(Sc)
        Sk=C(s);
    end
    s=s+1;
end
Se(it)=Sk;
M=M-1;
end
SS=[SS;S];
SeSe=[SeSe;Se];
end % Fin d'exécution de Ns simulations
-----
SSs=zeros(1,4*Ns);SeSes=zeros(1,4*Ns);
for d=1:4*Ns,
    if SS(d)==C(1)
        SSs(d)=0;
    elseif SS(d)==C(2)
        SSs(d)=1;
    elseif SS(d)==C(3)
        SSs(d)=2;
    elseif SS(d)==C(4)
        SSs(d)=3;
    end
end
for d=1:4*Ns,
    if SeSe(d)==C(1)
        SeSes(d)=0;
    elseif SeSe(d)==C(2)
        SeSes(d)=1;
    elseif SeSe(d)==C(3)
        SeSes(d)=2;
    elseif SeSe(d)==C(4)
        SeSes(d)=3;
    end
end
SNR=srr(np)
[Ne,ser]=siterr(SSs,SeSes);
BER=[BER ser]
end % Fin de calcul de Taux d'Erreur par Bit pour tous les Np aux
ZF_so=BER % Notre résultat
% les résultats obtenus par Baro et Bauch
Baro=[0.26 0.14 0.06 0.02 0.006 0.002]
semilogy(snr,Baro,'r-o',srr,ZF_so,'b-o');grid
xlabel('SNR par bit (dB)');ylabel('BER');
legend('Résultat de simulation de Baro et Bauch','Notre résultat de
Simulation obtenu');
title('BER pour ZF V-BLAST sans ordre optimal pour 4-QAM ');

```

## 5- ZF\_so\_Nat\_gray8.m :

```

%
%-----ZF_so_Nat_gray8.m-----
%      Simulation de l'algorithme EP V-BLAST pour une modulation 8-QAM      %
%      pour un codage binaire naturel et un codage de Gray                %
%-----
%
M=8; N=8;                                % Nombre d'antennes d'émission et de réception
snr = 'C : 5 : 25';                       % Rapport signal sur bruit par bit (SNR) en dB
PT = 24*10.^(snr/10);                      % La puissance totale selon snr
Np = length(snr);
Ns = 4500;                                % Nombre de simulations

BERn=''; BERg='';
for np=1:Np,
    Pt=PT(np);
    SS=''; SeSe='';
    for ns=1:Ns,
        M=8; i=sqrt(-1);
        % Génération des matrices aléatoires du canal
        E = 1/sqrt(2)*randr(N,M)+i*1/sqrt(2)*randr(N,M);
        % Génération d'un vecteur de bruit aléatoire gaussien (AWGN)
        B=wgn(N,1,0)+i*wgn(N,1,0);
        % La constellation 8-QAM rectangulaire
        C=sqrt(Pt/48)*[1+i -1+i -1-i 1-i 3+i -3+i -3-i 3-i]';
        % Génération du signal vecteur transmis de la constellation 8-QAM
        S=sqrt(Pt/48)*randsrc(8,1,'1+i -1+i -1-i 1-i 3+i -3+i -3-i 3-i');
        R=E*S+B; % Le signal vecteur reçu R=E*S+B= [R1 R2 R3 R4 R5 R6 R7 R8]
        %-----* La détection du premier symbole *-----
        Se=zeros(M,1);
        it=1;
        G=inv(E'*E)*E';
        Wk=G(1,:);
        Yk=dot(Wk',R);
        Sc='';
        for s=1:8,
            Ss=norm(C(s)-Yk)^2;
            Sc=[Sc;Ss];
        end
        s=1;
        while s<=8,
            if Sc(s)==min(Sc)
                Sk=C(s);
            end
            s=s+1;
        end
        Se(it)=Sk;
        M=M-1;
        %-----* La détection des symboles restant *-----
        for it=2:8,
            R=R-Sk*E(:,1);
            E=E(:,2:M+1);
            G=inv(E'*E)*E';
            Wk=G(1,:);
            Yk=dot(Wk',R);
            Sc='';
            for s=1:8,
                Ss=norm(C(s)-Yk)^2;
                Sc=[Sc;Ss];
            end
        end
    end
end

```

```

s-1;
while s<-8,
    if Sc(s)--min(Sc)
        Sk=C(s);
    end
    s=s+1;
end
Se(it)-Sk;
M=M-1;
end
SS='SS;S';
SeSe='SeSe;Se';
end % Fin d'exécution de Ns simulations
%-----
SSs=zeros(1,8*Ns);SeSes=zeros(1,8*Ns);
for c=1:2,
    if c==1
        for d=1:8*Ns,
            if SS(d)--C(1)
                SSs(d)=0;
            elseif SS(d)--C(2)
                SSs(d)=1;
            elseif SS(d)--C(3)
                SSs(d)=2;
            elseif SS(d)--C(4)
                SSs(d)=3;
            elseif SS(d)--C(5)
                SSs(d)=4;
            elseif SS(d)--C(6)
                SSs(d)=5;
            elseif SS(d)--C(7)
                SSs(d)=6;
            elseif SS(d)--C(8)
                SSs(d)=7;
            end
        end
    end
    for d=1:8*Ns,
        if SeSe(d)--C(1)
            SeSes(d)=0;
        elseif SeSe(d)--C(2)
            SeSes(d)=1;
        elseif SeSe(d)--C(3)
            SeSes(d)=2;
        elseif SeSe(d)--C(4)
            SeSes(d)=3;
        elseif SeSe(d)--C(5)
            SeSes(d)=4;
        elseif SeSe(d)--C(6)
            SeSes(d)=5;
        elseif SeSe(d)--C(7)
            SeSes(d)=6;
        elseif SeSe(d)--C(8)
            SeSes(d)=7;
        end
    end
    SNR=snr(rp)
    'Nr,corr' =siterr(SSs,SeSes);
    BERR='BERR corr'
else
    for d=1:8*Ns,
        if SS(d)--C(1)

```

```

        SSs(d)-0;
    elseif SS(d)--C(2)
        SSs(d)-1;
    elseif SS(d)--C(3)
        SSs(d)-3;
    elseif SS(d)--C(4)
        SSs(d)-2;
    elseif SS(d)--C(5)
        SSs(d)-4;
    elseif SS(d)--C(6)
        SSs(d)-5;
    elseif SS(d)--C(7)
        SSs(d)-7;
    elseif SS(d)--C(8)
        SSs(d)-6;
    end
end
for d=1:8*Ns,
    if SeSe(d)--C(1)
        SeSeS(d)-0;
    elseif SeSe(d)--C(2)
        SeSeS(d)-1;
    elseif SeSe(d)--C(3)
        SeSeS(d)-3;
    elseif SeSe(d)--C(4)
        SeSeS(d)-2;
    elseif SeSe(d)--C(5)
        SeSeS(d)-4;
    elseif SeSe(d)--C(6)
        SeSeS(d)-5;
    elseif SeSe(d)--C(7)
        SeSeS(d)-7;
    elseif SeSe(d)--C(8)
        SeSeS(d)-6;
    end
end
`Ng,berg`-citerr(SSs,SeSeS);
BERg-`BERg berg`
end
end
end
end % Fin de calcul de Taux Erreur par Bit pour tous les Np sur
semilogy(snr,BERn, 'r-s',snr,BERg, 'b-o');grid
xlabel('SNR par bit (dB)');ylabel('BER');
title('BER pour ZF V-BLAST 2-QAM pour un codage binaire et un codage de
Gray');
legend('Codage naturel','Codage de Gray');

```

## 6- ZF\_so4\_8\_16\_QAM.m :

```

%
%-----ZF_so4_8_16_QAM.m-----
% Simulation de l'algorithme ZF V-BLAST pour une modulation M-QAM %
% M=4, 8 et 16 %
%-----
%
snr = `0 : 5 : 25` ; % Rapport signal sur bruit par bit (SNR) en dB
Np = length(snr);

BER4-'';BER8-'';BER16-'';
for r=1:3 % r=1: simulation pour 4-QAM, r=2 pour 8-QAM et r=3 pour 16-QAM
    if r==1

```

```

M=4;N=4;
PT = 8*10.^(snr/10); % La puissance totale selon snr
Ns = 12500; % Nombre de simulations
elseif r==2
M=8;N=8;
PT = 24*10.^(snr/10); % La puissance totale selon snr
Ns = 4500; % Nombre de simulations
elseif r==3
M=16;N=16;
PT = 64*10.^(snr/10); % La puissance totale selon snr
Ns = 1500; % Nombre de simulations
end
for np=1:Np,
Pt=PT(np);
Ss='';SeSe='';
for ns=1:Ns,
if r==1
M=4;N=4;
elseif r==2
M=8;N=8;
elseif r==3
M=16;N=16;
end
% Génération des matrices aléatoires du canal
E = 1/sqrt(2)*randn(N,M)+i*1/sqrt(2)*randn(N,M);
% Génération d'un vecteur de bruit aléatoire gaussien (AWGN)
B=wgn(N,1,C)+i*wgn(N,1,C);
if r==1
% La constellation 4-QAM
C=sqrt(Pt/M)/sqrt(2)*[1+i -1+i -1-i 1-i];
% Génération du signal vecteur transmis de la constellation
% 4-QAM
S=sqrt(Pt/M)/sqrt(2)*(randsrc(N,1)+i*randsrc(N,1));
elseif r==2
% La constellation 8-QAM rectangulaire
C=sqrt(Pt/48)*[1+i -1+i -1-i 1-i 3+i -3+i -3-i 3-i];
% Génération du signal vecteur transmis de la constellation
% 8-QAM
S=sqrt(Pt/48)*randsrc(M,1,[1+i -1+i -1-i 1-i 3+i -3+i -3-i
3-i]);
elseif r==3
% La constellation 16-QAM rectangulaire
C=sqrt(Pt/160)*[1+i -1+i -1-i 1-i 3+i 1+3*i -1+3*i -3+i -3-i
-1-3*i 1-3*i 3-i 3+3*i -3+3*i -3-3*i 3-3*i];
% Génération du signal vecteur transmis de la constellation
% 16-QAM
S=sqrt(Pt/160)*randsrc(M,1,[1+i -1+i -1-i 1-i 3+i 1+3*i
-1+3*i -3+i -3-i -1-3*i 1-3*i 3-i 3+3*i -3+3*i -3-3*i 3-3*i]);
end
R=E*S+B; % Le signal vecteur reçu
%-----* La détection du premier symbole *-----
Se=zeros(M,1);
it=1;
G=inv(E'*E)*E';
Wk=G(1,:);
Yk=dot(Wk',R);
Sc='';
for s=1:N,
Ss=norm(C(s)-Yk)^2;
Sc=[Sc;Ss];
end
end

```



```

s=1;
while s<-N,
    if Sc(s)--min(Sc)
        Sk=C(s);
    end
    s=s+1;
end
Se(it)=Sk;
M=M-1;
*-----* La détecteur des symboles restant *-----
for it=2:N,
    R=R-Sk*E(:,1);
    E=E(:,2:M+1);
    G=inv(E'*E)*E';
    Wk=G(1,:);
    Yk=dot(Wk',R);
    Sc='';
    for s=1:N,
        Ss=norm(C(s)-Yk)^2;
        Sc=[Sc;Ss];
    end
    s=1;
    while s<-N,
        if Sc(s)--min(Sc)
            Sk=C(s);
        end
        s=s+1;
    end
    Se(it)=Sk;
    M=M-1;
end
SS=[SS;S'];
SeSe=[SeSe;Se'];
end % Fin d'exécution de Ms simulations
SSs=zeros(1,N*Ns);SeSeS=zeros(1,N*Ns);
if r==1
    for d=1:N*Ns,
        if SS(d)--C(1)
            SSs(d)=0;
        elseif SS(d)--C(2)
            SSs(d)=1;
        elseif SS(d)--C(3)
            SSs(d)=3;
        elseif SS(d)--C(4)
            SSs(d)=2;
        end
    end
    for d=1:N*Ns,
        if SeSe(d)--C(1)
            SeSeS(d)=0;
        elseif SeSe(d)--C(2)
            SeSeS(d)=1;
        elseif SeSe(d)--C(3)
            SeSeS(d)=3;
        elseif SeSe(d)--C(4)
            SeSeS(d)=2;
        end
    end
end
SNR=snr(rp)
[Ne4,ser4]=siterr(SSs,SeSeS);
BER4='BER4 ser4'

```

```
elseif r==2
    for d=1:N*Ns,
        if SS(d)==C(1)
            SSs(d)=0;
        elseif SS(d)==C(2)
            SSs(d)=1;
        elseif SS(d)==C(3)
            SSs(d)=3;
        elseif SS(d)==C(4)
            SSs(d)=2;
        elseif SS(d)==C(5)
            SSs(d)=4;
        elseif SS(d)==C(6)
            SSs(d)=5;
        elseif SS(d)==C(7)
            SSs(d)=7;
        elseif SS(d)==C(8)
            SSs(d)=6;
        end
    end
end
for d=1:N*Ns,
    if SeSe(d)==C(1)
        SeSees(d)=0;
    elseif SeSe(d)==C(2)
        SeSees(d)=1;
    elseif SeSe(d)==C(3)
        SeSees(d)=3;
    elseif SeSe(d)==C(4)
        SeSees(d)=2;
    elseif SeSe(d)==C(5)
        SeSees(d)=4;
    elseif SeSe(d)==C(6)
        SeSees(d)=5;
    elseif SeSe(d)==C(7)
        SeSees(d)=7;
    elseif SeSe(d)==C(8)
        SeSees(d)=6;
    end
end
end
SNR=snr(np)
`Nes,ber8`=biterr(SSs,SeSees);
BER8=`BER8 ber8`
elseif r==3
    for d=1:N*Ns,
        if SS(d)==C(1)
            SSs(d)=0;
        elseif SS(d)==C(2)
            SSs(d)=1;
        elseif SS(d)==C(3)
            SSs(d)=3;
        elseif SS(d)==C(4)
            SSs(d)=2;
        elseif SS(d)==C(5)
            SSs(d)=4;
        elseif SS(d)==C(6)
            SSs(d)=8;
        elseif SS(d)==C(7)
            SSs(d)=9;
        elseif SS(d)==C(8)
            SSs(d)=5;
        elseif SS(d)==C(9)
            SSs(d)=6;
        end
    end
end
```

```

        SSo(d)-7;
elseif SS(d)--C(10)
        SSo(d)-11;
elseif SS(d)--C(11)
        SSo(d)-10;
elseif SS(d)--C(12)
        SSo(d)-6;
elseif SS(d)--C(13)
        SSo(d)-12;
elseif SS(d)--C(14)
        SSo(d)-13;
elseif SS(d)--C(15)
        SSo(d)-15;
elseif SS(d)--C(16)
        SSo(d)-14;
end
end
for d=1:N*Ns,
    if SeSe(d)--C(1)
        SeSec(d)-0;
    elseif SeSe(d)--C(2)
        SeSec(d)-1;
    elseif SeSe(d)--C(3)
        SeSec(d)-3;
    elseif SeSe(d)--C(4)
        SeSec(d)-2;
    elseif SeSe(d)--C(5)
        SeSec(d)-4;
    elseif SeSe(d)--C(6)
        SeSec(d)-8;
    elseif SeSe(d)--C(7)
        SeSec(d)-9;
    elseif SeSe(d)--C(8)
        SeSec(d)-5;
    elseif SeSe(d)--C(9)
        SeSec(d)-7;
    elseif SeSe(d)--C(10)
        SeSec(d)-11;
    elseif SeSe(d)--C(11)
        SeSec(d)-10;
    elseif SeSe(d)--C(12)
        SeSec(d)-6;
    elseif SeSe(d)--C(13)
        SeSec(d)-12;
    elseif SeSe(d)--C(14)
        SeSec(d)-13;
    elseif SeSe(d)--C(15)
        SeSec(d)-15;
    elseif SeSe(d)--C(16)
        SeSec(d)-14;
    end
end
SNR=snr(rp)
[Ne16,ber16]=biterr(SSo,SeSec);
BER16='BER16 ber16'
end
end % Fin de calcul de Taux Erreur par Bit pour tous les Np snr
end % Fin de calcul de Taux Erreur par Bit pour les trois constellations
semilogy(snr,BER16,'b-d'),snr,BER8,'m-s',snr,BER4,'r-o');grid
xlabel('SNR par bit (dB)');ylabel('BER');
title('BER pour 2P V-BLAST pour une modulation M-QAM');

```

```
legend('16-QAM','8-QAM','4-QAM');
```

### 7- ZF\_sans\_Ordre\_O.m :

```
%
%-----*ZF_sans_Ordre_O.m*-----
%      Simulation de l'algorithme ZF V-BLAST sans ordre      %
%      optimal de détection pour une modulation 4-QAM      %
%-----
%
M=4; N=4;          % Nombre d'antennes d'émission et de réception
srr = '0 : 5 : 25'; % Rapport signal sur bruit par bit (SNR) en dB
Pt = 8*10.^(srr/10); % La puissance totale selon srr
Np = length(srr);
Ns = 12500;       % Nombre de simulations

BER1='';BER2='';BER3='';BER4='';
for rp=1:Np,
    Pt=Pt(rp);
    SS1='';SeSe1='';SS2='';SeSe2='';SS3='';SeSe3='';SS4='';SeSe4='';
    for rs=1:Ns,
        M=4;i=sqrt(-1);
        % Génération des matrices aléatoires du canal
        E = 1/sqrt(2)*randn(N,M)+i*1/sqrt(2)*randn(N,M);
        % Génération d'un vecteur de bruit aléatoire gaussien (AWGN)
        B=wgn(N,1,C)+i*wgn(N,1,C);
        % La constellation 4-QAM
        C=sqrt(Pt/M)/sqrt(2)*[1+i -1+i -1-i 1-i];
        % Génération du signal vecteur transmis de la constellation 4-QAM
        S=sqrt(Pt/M)/sqrt(2)*(randsrc(M,1)+i*randsrc(M,1));
        R=E*S+B;          % Le signal vecteur reçu R=[R1 R2 R3 R4]
        %-----* La détection du premier symbole *-----
        Se=zeros(M,1);
        it=1;
        G=inv(E'*E)*E';
        Wk=G(1,:);
        Yk=dot(Wk',R);
        Sc='';
        for s=1:4,
            Ss=norm(C(s)-Yk)^2;
            Sc=[Sc;Ss];
        end
        s=1;
        while s<=4,
            if Sc(s)==min(Sc)
                Sk=C(s);
            end
            s=s+1;
        end
        Se(it)=Sk;
        M=M-1;
        %-----* La détection des symboles restant *-----
        for it=2:4,
            R=R-Sk*E(:,1);
            E=E(:,2:M+1);
            G=inv(E'*E)*E';
            Wk=G(1,:);
            Yk=dot(Wk',R);
            Sc='';
            for s=1:4,
                Ss=norm(C(s)-Yk)^2;
```

```

        Sc='Sc;Ss';
    end
    s-1;
    while s<-4,
        if Sc(s)--min(Sc)
            Sk=C(s);
        end
        s-s+1;
    end
    Se(it)-Sk;
    M-M-1;
end
SS1='SS1;S(1)';
SeSe1='SeSe1;Se(1)';
SS2='SS2;S(2)';
SeSe2='SeSe2;Se(2)';
SS3='SS3;S(3)';
SeSe3='SeSe3;Se(3)';
SS4='SS4;S(4)';
SeSe4='SeSe4;Se(4)';
end % Fin d'exécution de Ns simulations
%-----
SSs1=zeros(1,Ns);SeSe1s1=zeros(1,Ns);
SSs2=zeros(1,Ns);SeSe2s2=zeros(1,Ns);
SSs3=zeros(1,Ns);SeSe3s3=zeros(1,Ns);
SSs4=zeros(1,Ns);SeSe4s4=zeros(1,Ns);
for d=1:Ns,
    if SS1(d)--C(1)
        SSs1(d)=0;
    elseif SS1(d)--C(2)
        SSs1(d)=1;
    elseif SS1(d)--C(3)
        SSs1(d)=3;
    elseif SS1(d)--C(4)
        SSs1(d)=2;
    end
end
for d=1:Ns,
    if SeSe1(d)--C(1)
        SeSe1s1(d)=0;
    elseif SeSe1(d)--C(2)
        SeSe1s1(d)=1;
    elseif SeSe1(d)--C(3)
        SeSe1s1(d)=3;
    elseif SeSe1(d)--C(4)
        SeSe1s1(d)=2;
    end
end
for d=1:Ns,
    if SS2(d)--C(1)
        SSs2(d)=0;
    elseif SS2(d)--C(2)
        SSs2(d)=1;
    elseif SS2(d)--C(3)
        SSs2(d)=3;
    elseif SS2(d)--C(4)
        SSs2(d)=2;
    end
end
for d=1:Ns,
    if SeSe2(d)--C(1)

```

```
        SeSep2(d)-0;
    elseif SeSep2(d)--C(2)
        SeSep2(d)-1;
    elseif SeSep2(d)--C(3)
        SeSep2(d)-3;
    elseif SeSep2(d)--C(4)
        SeSep2(d)-2;
    end
end
for d=1:Ns,
    if SS3(d)--C(1)
        SS3(d)-0;
    elseif SS3(d)--C(2)
        SS3(d)-1;
    elseif SS3(d)--C(3)
        SS3(d)-3;
    elseif SS3(d)--C(4)
        SS3(d)-2;
    end
end
for d=1:Ns,
    if SeSep3(d)--C(1)
        SeSep3(d)-0;
    elseif SeSep3(d)--C(2)
        SeSep3(d)-1;
    elseif SeSep3(d)--C(3)
        SeSep3(d)-3;
    elseif SeSep3(d)--C(4)
        SeSep3(d)-2;
    end
end
for d=1:Ns,
    if SS4(d)--C(1)
        SS4(d)-0;
    elseif SS4(d)--C(2)
        SS4(d)-1;
    elseif SS4(d)--C(3)
        SS4(d)-3;
    elseif SS4(d)--C(4)
        SS4(d)-2;
    end
end
for d=1:Ns,
    if SeSep4(d)--C(1)
        SeSep4(d)-0;
    elseif SeSep4(d)--C(2)
        SeSep4(d)-1;
    elseif SeSep4(d)--C(3)
        SeSep4(d)-3;
    elseif SeSep4(d)--C(4)
        SeSep4(d)-2;
    end
end
SNR=snr(rp)
`N1,ser1`-siterr(SS31,SeSep1);
BER1=`BER1 ser1`
`N2,ser2`-siterr(SS32,SeSep2);
BER2=`BER2 ser2`
`N3,ser3`-siterr(SS33,SeSep3);
BER3=`BER3 ser3`
`N4,ser4`-siterr(SS34,SeSep4);
```

```

BER4='BER4_ser4'
end
% Fix de calcul de Taux Error par Bit pour tous les Np sur
semilogy(snr, BER1, '-s', snr, BER2, '-o', snr, BER3, '-p', snr, BER4, '-d'); grid
xlabel('SNR par bit (dB)'); ylabel('BER');
title('BER pour ZF V-BLAST 4-QAM sans Orde optimal');
legend('Tx 1', 'Tx 2', 'Tx 3', 'Tx 4');

```

## 8- ZF avec Ordre O.m :

```

%
%-----*ZF_avec_Ordre_O.m*-----
%      Simulation de l'algorithme ZF V-BLAST avec ordre optimal      %
%      de détection pour une modulation 4-QAM                      %
%-----
%
M=4; N=4;           % Nombre d'antennes d'émission et de réception
snr = 10 : 5 : 25;  % Rapport signal sur bruit par bit (SNR) en dB
Pt = 8*10.^(snr/10); % la puissance totale selon snr
Np = length(snr);
Ns = 12500;        % Nombre de simulations

BER1=''; BER2=''; BER3=''; BER4='';
for np=1:Np,
    Pt=Pt(np);
    SS1=''; SeSe1=''; SS2=''; SeSe2=''; SS3=''; SeSe3=''; SS4=''; SeSe4='';
    for ns=1:Ns,
        M=4; i=sqrt(-1);
        % Génération des matrices aléatoires du canal
        E = 1/sqrt(2)*randn(N,M)+i*1/sqrt(2)*randn(N,M);
        % Génération d'un vecteur de Bruit aléatoire gaussien (AWGN)
        B=wgn(M,1,C)+i*wgn(M,1,C);
        % La constellation 4-QAM
        C=sqrt(Pt/M)/sqrt(2)*[1+i -1+i -1-i 1-i]';
        % Génération du signal vecteur transmis de la constellation 4-QAM
        S=sqrt(Pt/M)/sqrt(2)*(randarc(M,1)+i*randarc(M,1));
        R=E*S+B;           % Le signal vecteur reçu R=E*S+B= R1 R2 R3 R4'
        %-----*Utilisation de la technique d'ordre optimal*-----
        M=4;
        K='';
        Se=zeros(M,1); E1=E;
        for it=1:4,
            G=inv(E'*E)*E';
            Dc='';
            for m=1:M,
                D=norm(G(m,:))^2;
                Dc=[Dc,D];
            end
            m=1;
            while m<=M
                if Dc(m)--min(Dc)
                    k=m;
                end
                m=m+1;
            end
            Wk=G(k,:);
            Yk=dot(Wk',R);
            Sc='';
            for s=1:N,
                Ss=norm(C(s)-Yk)^2;
                Sc=[Sc,Ss];
            end

```

```
s=1;
while s<=N,
    if Sc(s)==min(Sc)
        Sk=C(s);
    end
    s=s+1;
end
R=R-Sk*H(:,k);
if M==4
    Sk1=Sk;
elseif M==3
    Sk2=Sk;
elseif M==2
    Sk3=Sk;
elseif M==1
    Sk4=Sk;
end
if M==4
    if k==1
        H=H1(:,2:4);H2=H;
    elseif k==2
        H=[H1(:,1),H1(:,3),H1(:,4)];H2=H;
    elseif k==3
        H=[H1(:,1),H1(:,2),H1(:,4)];H2=H;
    elseif k==4
        H=[H1(:,1),H1(:,2),H1(:,3)];H2=H;
    end
elseif M==3
    if k==1
        H=H2(:,2:3);H3=H;
    elseif k==2
        H=[H2(:,1),H2(:,3)];H3=H;
    elseif k==3
        H=[H2(:,1),H2(:,2)];H3=H;
    end
elseif M==2
    if k==1
        H=H3(:,2);H4=H;
    elseif k==2
        H=H3(:,1);H4=H;
    end
end
M=M-1;
K=[K;k];
end
Ka=K;
if K(1)==1
    K(2)=K(2)+1;
    if K(2)==2
        K(3)=K(3)+2;
        if K(3)==3
            K(4)=4;
        else K(4)=3;
        end
    elseif K(2)==3
        K(3)=2*K(3);
        if K(3)==2
            K(4)=4;
        else K(4)=2;
        end
    elseif K(2)==4
```



```
        K(3)-K(3)+1;
        if K(3)--2
            K(4)-3;
        else K(4)-2;
        end
    end
elseif K(1)--2
    if K(2)--1
        K(3)-K(3)+2;
        if K(3)--3
            K(4)-4;
        else K(4)-3;
        end
    else K(2)-K(2)+1;
        if K(2)--3
            if K(3)--1
                K(4)-4;
            else K(3)-4;K(4)-1;
            end
        elseif K(2)--4
            if K(3)--1
                K(4)-3;
            else K(3)-3;K(4)-1;
            end
        end
    end
elseif K(1)--3
    if K(2)--1
        K(3)-2*K(3);
        if K(3)--2
            K(4)-4;
        else K(4)-2;
        end
    elseif K(2)--2
        if K(3)--1
            K(4)-4;
        else K(3)-4;K(4)-1;
        end
    else K(2)-4;
        if K(3)--1
            K(4)-2;
        else K(4)-1;
        end
    end
elseif K(1)--4
    if K(2)--1
        K(3)-K(3)+1;
        if K(3)--2
            K(4)-3;
        else K(4)-2;
        end
    elseif K(2)--2
        if K(3)--1
            K(4)-3;
        else K(3)-3;K(4)-1;
        end
    elseif K(2)--3
        if K(3)--1
            K(4)-2;
        else K(4)-1;
        end
    end
end
```

```

        end
    end
    %----- Le vecteur signal estimé -----
    Se(K(1))=Sk1;
    Se(K(2))=Sk2;
    Se(K(3))=Sk3;
    Se(K(4))=Sk4;
    %-----
    SS1='SS1;S(1)';
    SeSe1='SeSe1;Se(1)';
    SS2='SS2;S(2)';
    SeSe2='SeSe2;Se(2)';
    SS3='SS3;S(3)';
    SeSe3='SeSe3;Se(3)';
    SS4='SS4;S(4)';
    SeSe4='SeSe4;Se(4)';
end % Fin d'exécution de Ns simulations
%-----
SSs1=zeros(1,Ns);SeSeS1=zeros(1,Ns);
SSs2=zeros(1,Ns);SeSeS2=zeros(1,Ns);
SSs3=zeros(1,Ns);SeSeS3=zeros(1,Ns);
SSs4=zeros(1,Ns);SeSeS4=zeros(1,Ns);
for d=1:Ns,
    if SS1(d)==C(1)
        SSs1(d)=0;
    elseif SS1(d)==C(2)
        SSs1(d)=1;
    elseif SS1(d)==C(3)
        SSs1(d)=3;
    elseif SS1(d)==C(4)
        SSs1(d)=2;
    end
end
for d=1:Ns,
    if SeSe1(d)==C(1)
        SeSeS1(d)=0;
    elseif SeSe1(d)==C(2)
        SeSeS1(d)=1;
    elseif SeSe1(d)==C(3)
        SeSeS1(d)=3;
    elseif SeSe1(d)==C(4)
        SeSeS1(d)=2;
    end
end
for d=1:Ns,
    if SS2(d)==C(1)
        SSs2(d)=0;
    elseif SS2(d)==C(2)
        SSs2(d)=1;
    elseif SS2(d)==C(3)
        SSs2(d)=3;
    elseif SS2(d)==C(4)
        SSs2(d)=2;
    end
end
for d=1:Ns,
    if SeSe2(d)==C(1)
        SeSeS2(d)=0;
    elseif SeSe2(d)==C(2)
        SeSeS2(d)=1;
    elseif SeSe2(d)==C(3)

```

```

        SeSe2(d)-3;
    elseif SeSe2(d)--C(4)
        SeSe2(d)-2;
    end
end
for d=1:Ns,
    if SS3(d)--C(1)
        SS3(d)-0;
    elseif SS3(d)--C(2)
        SS3(d)-1;
    elseif SS3(d)--C(3)
        SS3(d)-3;
    elseif SS3(d)--C(4)
        SS3(d)-2;
    end
end
for d=1:Ns,
    if SeSe3(d)--C(1)
        SeSe3(d)-0;
    elseif SeSe3(d)--C(2)
        SeSe3(d)-1;
    elseif SeSe3(d)--C(3)
        SeSe3(d)-3;
    elseif SeSe3(d)--C(4)
        SeSe3(d)-2;
    end
end
for d=1:Ns,
    if SS4(d)--C(1)
        SS4(d)-0;
    elseif SS4(d)--C(2)
        SS4(d)-1;
    elseif SS4(d)--C(3)
        SS4(d)-3;
    elseif SS4(d)--C(4)
        SS4(d)-2;
    end
end
for d=1:Ns,
    if SeSe4(d)--C(1)
        SeSe4(d)-0;
    elseif SeSe4(d)--C(2)
        SeSe4(d)-1;
    elseif SeSe4(d)--C(3)
        SeSe4(d)-3;
    elseif SeSe4(d)--C(4)
        SeSe4(d)-2;
    end
end
SNR=snr(np)
`N1,ser1`-biterr(SSo1,SeSeo1);
BER1=`BER1 ser1`
`N2,ser2`-biterr(SSo2,SeSeo2);
BER2=`BER2 ser2`
`N3,ser3`-biterr(SSo3,SeSeo3);
BER3=`BER3 ser3`
`N4,ser4`-biterr(SSo4,SeSeo4);
BER4=`BER4 ser4`
end
% Fin de calcul de Taux Erreur par Bit pour tous les Np snr
semilogy(snr,BER1,'-d',snr,BER2,'-g',snr,BER3,'-o',snr,BER4,'-s');grid
xlabel('SNR par bit (dB)');ylabel('BER');

```

```
title('BER pour ZF V-BLAST 4-QAM avec ordre optimal');
legend('Tx 1', 'Tx 2', 'Tx 3', 'Tx 4');
```

### 9- ZFso\_a4.m :

```
%
%-----ZFso_a4.m-----
%      Simulation de l'algorithme ZF V-BLAST pour 4-QAM      %
%      sans et avec ordre optimal de détection              %
%-----
%
M=4; N=4;          % Nombre des antennes d'émission et de réception
snr = '0 : 5 : 30'; % Rapport signal sur bruit par bit (SNR) en dB
Pt = 8*10.^(snr/10); % La puissance totale selon snr
Np = length(snr);
Ns = 12500;        % Nombre de simulations

BER1=''; BER2='';
for n=1:2,
    for np=1:Np,
        Pt=Pt(np);
        SS=''; SeSe='';
        for ns=1:Ns,
            M=4; i=sqrt(-1);
            % Génération des matrices aléatoires du canal
            E = 1/sqrt(2)*randn(N,M)+i*1/sqrt(2)*randn(N,M);
            % Génération d'un vecteur de bruit aléatoire gaussien (AWGN)
            B=wgn(N,1,C)+i*wgn(N,1,C);
            % La constellation 4-QAM
            C=sqrt(Pt/M)/sqrt(2)*[1+i -1+i -1-i 1-i]';
            % Génération du signal vecteur transmis de la constellation
            % 4-QAM
            S=sqrt(Pt/M)/sqrt(2)*(randsrc(N,1)+i*randsrc(N,1));
            R=E*S+B; % Le signal vecteur reçu R=E*S+B= [R1 R2 R3 R4]
            if n==1
                %-----* La détection du premier symbole *-----
                Se=zeros(M,1);
                it=1;
                G=inv(E'*E)*E';
                Wk=G(1,:);
                Yk=dot(Wk',R);
                Sc='';
                for s=1:4,
                    Ss=norm(C(s)-Yk)^2;
                    Sc=[Sc;Ss];
                end
                s=1;
                while s<=4,
                    if Sc(s)==min(Sc)
                        Sk=C(s);
                        end
                        s=s+1;
                    end
                Se(it)=Sk;
                M=M-1;
                %-----* La détection des symboles restant *-----
                for it=2:4,
                    R=R-Sk*E(:,1);
                    E=E(:,2:M+1);
                    G=inv(E'*E)*E';
                    Wk=G(1,:);
```

```

Yk-dot(Wk',R);
Sc='';
for s=1:4,
    Ss=norm(C(s)-Yk)^2;
    Sc=[Sc;Ss];
end
s=1;
while s<=4,
    if Sc(s)--min(Sc)
        Sk=C(s);
    end
    s=s+1;
end
Se(it)=Sk;
M=M-1;
end
SS=[SS;S'];
SeSe=[SeSe;Se'];
else
    M=4;
    K='';
    Se=zeros(M,1);E1=E;
    for it=1:4,
        G=inv(E'*E)*E';
        Dc='';
        for m=1:M,
            D=norm(G(m,:))^2;
            Dc=[Dc;D];
        end
        m=1;
        while m<=M
            if Dc(m)--min(Dc)
                k=m;
            end
            m=m+1;
        end
        WK=G(k,:);
        Yk-dot(Wk',R);
        Sc='';
        for s=1:N,
            Ss=norm(C(s)-Yk)^2;
            Sc=[Sc;Ss];
        end
        s=1;
        while s<=N,
            if Sc(s)--min(Sc)
                Sk=C(s);
            end
            s=s+1;
        end
        R=R-Sk*E(:,k);
        if M--4
            Sk1=Sk;
        elseif M--3
            Sk2=Sk;
        elseif M--2
            Sk3=Sk;
        elseif M--1
            Sk4=Sk;
        end
        end
        if M--4

```

```

        if k--1
            E=E1(:,2:4);E2=E;
        elseif k--2
            E=[E1(:,1),E1(:,3),E1(:,4)];E2=E;
        elseif k--3
            E=[E1(:,1),E1(:,2),E1(:,4)];E2=E;
        elseif k--4
            E=[E1(:,1),E1(:,2),E1(:,3)];E2=E;
        end
    elseif M--3
        if k--1
            E=E2(:,2:3);E3=E;
        elseif k--2
            E=[E2(:,1),E2(:,3)];E3=E;
        elseif k--3
            E=[E2(:,1),E2(:,2)];E3=E;
        end
    elseif M--2
        if k--1
            E=E3(:,2);E4=E;
        elseif k--2
            E=E3(:,1);E4=E;
        end
    end
    end
    M=M-1;
    K=[K;k];
end
Ka=K;
if K(1)--1
    K(2)=K(2)+1;
    if K(2)--2
        K(3)=K(3)+2;
        if K(3)--3
            K(4)=4;
        else K(4)=3;
        end
    elseif K(2)--3
        K(3)=2*K(3);
        if K(3)--2
            K(4)=4;
        else K(4)=2;
        end
    elseif K(2)--4
        K(3)=K(3)+1;
        if K(3)--2
            K(4)=3;
        else K(4)=2;
        end
    end
end
elseif K(1)--2
    if K(2)--1
        K(3)=K(3)+2;
        if K(3)--3
            K(4)=4;
        else K(4)=3;
        end
    end
    else K(2)=K(2)+1;
    if K(2)--3
        if K(3)--1
            K(4)=4;
        else K(3)=4;K(4)=1;
        end
    end
end

```

```

        end
        elseif K(2)--4
            if K(3)--1
                K(4)-3;
            else K(3)-3;K(4)-1;
            end
        end
    end
end
elseif K(1)--3
    if K(2)--1
        K(3)-2*K(3);
        if K(3)--2
            K(4)-4;
        else K(4)-2;
        end
    elseif K(2)--2
        if K(3)--1
            K(4)-4;
        else K(3)-4;K(4)-1;
        end
    else K(2)-4;
        if K(3)--1
            K(4)-2;
        else K(4)-1;
        end
    end
elseif K(1)--4
    if K(2)--1
        K(3)-K(3)+1;
        if K(3)--2
            K(4)-3;
        else K(4)-2;
        end
    elseif K(2)--2
        if K(3)--1
            K(4)-3;
        else K(3)-3;K(4)-1;
        end
    elseif K(2)--3
        if K(3)--1
            K(4)-2;
        else K(4)-1;
        end
    end
end
end
%-----* Le vecteur signal estimé *-----
Se(K(1))-Sk1;
Se(K(2))-Sk2;
Se(K(3))-Sk3;
Se(K(4))-Sk4;
%-----
SS='SS;S';
SeSe='SeSe;Se';
end
end % Fin d'exécution de Ns simulations
%-----
SSs=zeros(1,4*Ns);SeSeS=zeros(1,4*Ns);
for d=1:4*Ns,
    if SS(d)--C(1)
        SSs(d)-C;
    elseif SS(d)--C(2)

```

```

        SSs(d)-1;
    elseif SS(d)--C(3)
        SSs(d)-3;
    elseif SS(d)--C(4)
        SSs(d)-2;
    end
end
for d=1:4*Ns,
    if SeSe(d)--C(1)
        SeSeS(d)-0;
    elseif SeSe(d)--C(2)
        SeSeS(d)-1;
    elseif SeSe(d)--C(3)
        SeSeS(d)-3;
    elseif SeSe(d)--C(4)
        SeSeS(d)-2;
    end
end
SNR=snr(rp)
if r==1
    [N1,ser1]=siterr(SSs,SeSeS);
    BER1='BER1_ser1'
else
    [N2,ser2]=siterr(SSs,SeSeS);
    BER2='BER2_ser2'
end
end % Fin de calcul de Taux Erreur par Bit pour tous les Np srx
end
semilogy(snr,BER1,'-d',snr,BER2,'-s');grid
xlabel('SNR par bit (dB)');ylabel('BER');
title('BER pour ZF V-BLAST pour 4-QAM');
legend('Sans ordre optimal','Avec ordre optimal');

```

### 10- ZF\_MMSEso\_a4.m :

```

%
%-----ZF_MMSEso_a4.m-----
%      Simulation des algorithmes ZF V-BLAST et MMSE V-BLAST      %
%      pour 4-QAM sans et avec ordre optimal de détection.      %
%-----
%
M=4; N=4; % Nombre des antennes d'émission et de réception
snr = 0 : 5 : 25; % Rapport signal sur bruit par bit (SNR) en dB
PT = 2*10.^(snr/10); % la puissance totale selon snr
Np = length(snr);
Ns=12500; % Nombre de simulations

ZFBERSo=''; ZFBERao=''; MMSEBERSo=''; MMSEBERao='';
for r=1:4,
    for rp=1:Np,
        Pt=PT(rp);
        SS=''; SeSe='';
        for ns=1:Ns,
            M=4;i=sqrt(-1);
            % Génération des matrices aléatoires du canal
            E = 1/sqrt(2)*randr(N,M)+i*1/sqrt(2)*randr(N,M);
            % Génération d'un vecteur de bruit aléatoire gaussien (AWGN)
            B=wgn(N,1,C)+i*wgn(N,1,C);
            % la constellation 4-QAM
            C=sqrt(Pt/M)/sqrt(2)*[1+i -1+i -1-i 1-i]';
            % Génération du signal vecteur transmis de la constellation

```



```

% 4-QAM
S=sqrt(Pt/M)/sqrt(2)*(randsrc(N,1)+i*randsrc(N,1));
R=E*S+B; % Le signal vecteur reçu R=E*S+B= [R1 R2 R3 R4]
if (r--1|r--3)
    %-----* La détection de premier symbole *-----
    Se=zeros(M,1);
    it=1;
    if r==1
        G=inv(E'*E)*E';
    else
        G=inv(E'*E+4/Pt*eye(M))*E';
    end
    Wk=G(1,:);
    Yk=dot(Wk',R);
    Sc='';
    for s=1:4,
        Ss=norm(C(s)-Yk)^2;
        Sc=[Sc;Ss];
    end
    s=1;
    while s<=4,
        if Sc(s)==min(Sc)
            Sk=C(s);
        end
        s=s+1;
    end
    Se(it)=Sk;
    M=M-1;
    %-----* La détection des symboles restant *-----
    for it=2:4,
        R=R-Sk*E(:,1);
        E=E(:,2:M+1);
        if r==1
            G=inv(E'*E)*E';
        else
            G=inv(E'*E+4/Pt*eye(M))*E';
        end
        Wk=G(1,:);
        Yk=dot(Wk',R);
        Sc='';
        for s=1:4,
            Ss=norm(C(s)-Yk)^2;
            Sc=[Sc;Ss];
        end
        s=1;
        while s<=4,
            if Sc(s)==min(Sc)
                Sk=C(s);
            end
            s=s+1;
        end
        Se(it)=Sk;
        M=M-1;
    end
    SS=[SS;S'];
    SeSe=[SeSe;Se'];
elseif (r==2|r==4)
    M=4;
    K='';
    Se=zeros(M,1);E1=E;
    for it=1:4,

```

```

if r==2
    G=inv(E'*E)*E';
else
    G=inv(E'*E+Pt*eye(M))*E';
end
Dc='';
for m=1:M,
    D=norm(G(m,:))^2;
    Dc=[Dc;D];
end
m=1;
while m<=M
    if Dc(m)==min(Dc)
        k=m;
        end
        m=m+1;
end
Wk=G(k,:);
Yk=dot(Wk',R);
Sc='';
for s=1:N,
    Ss=norm(C(s)-Yk)^2;
    Sc=[Sc;Ss];
end
s=1;
while s<=N,
    if Sc(s)==min(Sc)
        Sk=C(s);
        end
        s=s+1;
end
R=R-Sk*E(:,k);
if M==4
    Sk1=Sk;
elseif M==3
    Sk2=Sk;
elseif M==2
    Sk3=Sk;
elseif M==1
    Sk4=Sk;
end
if M==4
    if k==1
        E=[E1(:,2:4);E2-E;
elseif k==2
        E=[E1(:,1),E1(:,3),E1(:,4)]';E2-E;
elseif k==3
        E=[E1(:,1),E1(:,2),E1(:,4)]';E2-E;
elseif k==4
        E=[E1(:,1),E1(:,2),E1(:,3)]';E2-E;
end
elseif M==3
    if k==1
        E=[E2(:,2:3);E3-E;
elseif k==2
        E=[E2(:,1),E2(:,3)]';E3-E;
elseif k==3
        E=[E2(:,1),E2(:,2)]';E3-E;
end
elseif M==2
    if k==1

```

```
        E=E3(:,2);E4=E;
        elseif k==2
            E=E3(:,1);E4=E;
        end
    end
    end
    M=M-1;
    K=[K;k];
end
Ka=K;
if K(1)==1
    K(2)=K(2)+1;
    if K(2)==2
        K(3)=K(3)+2;
        if K(3)==3
            K(4)=4;
        else K(4)=3;
        end
    elseif K(2)==3
        K(3)=2*K(3);
        if K(3)==2
            K(4)=4;
        else K(4)=2;
        end
    elseif K(2)==4
        K(3)=K(3)+1;
        if K(3)==2
            K(4)=3;
        else K(4)=2;
        end
    end
elseif K(1)==2
    if K(2)==1
        K(3)=K(3)+2;
        if K(3)==3
            K(4)=4;
        else K(4)=3;
        end
    else K(2)=K(2)+1;
        if K(2)==3
            if K(3)==1
                K(4)=4;
            else K(3)=4;K(4)=1;
            end
        elseif K(2)==4
            if K(3)==1
                K(4)=3;
            else K(3)=3;K(4)=1;
            end
        end
    end
elseif K(1)==3
    if K(2)==1
        K(3)=2*K(3);
        if K(3)==2
            K(4)=4;
        else K(4)=2;
        end
    elseif K(2)==2
        if K(3)==1
            K(4)=4;
        else K(3)=4;K(4)=1;
        end
    end
end
```

```

        end
        else K(2)-4;
            if K(3)--1
                K(4)-2;
            else K(4)-1;
            end
        end
    elseif K(1)--4
        if K(2)--1
            K(3)-K(3)+1;
            if K(3)--2
                K(4)-3;
            else K(4)-2;
            end
        elseif K(2)--2
            if K(3)--1
                K(4)-3;
            else K(3)-3;K(4)-1;
            end
        elseif K(2)--3
            if K(3)--1
                K(4)-2;
            else K(4)-1;
            end
        end
    end
    %-----* Le vecteur signal estimé *-----
    Se(K(1))-Sk1;
    Se(K(2))-Sk2;
    Se(K(3))-Sk3;
    Se(K(4))-Sk4;
    %-----
    SS='SS;S';
    SeSe='SeSe;Se';
end
end % Fin d'exécution de Ns simulations
%-----
SSs=zeros(1,4*Ns);SeSec=zeros(1,4*Ns);
for d=1:4*Ns,
    if SS(d)--C(1)
        SSs(d)-0;
    elseif SS(d)--C(2)
        SSs(d)-1;
    elseif SS(d)--C(3)
        SSs(d)-3;
    elseif SS(d)--C(4)
        SSs(d)-2;
    end
end
for d=1:4*Ns,
    if SeSe(d)--C(1)
        SeSec(d)-0;
    elseif SeSe(d)--C(2)
        SeSec(d)-1;

    elseif SeSe(d)--C(3)
        SeSec(d)-3;
    elseif SeSe(d)--C(4)
        SeSec(d)-2;
    end
end
end
end

```

## Annexe

---

```
SNR=srr(rp)
if r==1
    'N1,ser1'-biterr(SSs,SeSeo);
    ZFBERso='ZFBERso ser1'
elseif r==2
    'N2,ser2'-biterr(SSs,SeSeo);
    ZFBERao='ZFBERao ser2'
elseif r==3
    'N2,ser2'-biterr(SSs,SeSeo);
    MMSEBERso='MMSEBERso ser2'
elseif r==4
    'N2,ser2'-biterr(SSs,SeSeo);
    MMSEBERao='MMSEBERao ser2'
end
end % Fin de calcul de Taux Erreur par Bit pour tous les Np srr
end
semilogy(srr,ZFBERso,'-d',srr,ZFBERao,'-p',srr,MMSEBERso,
'-o',srr,MMSEBERao,'o-s');grid
xlabel('SNR par bit (dB)');ylabel('BER');
title('BER pour ZF V-BLAST et MMSE V-BLAST pour 4-QAM');
legend('ZF sans ordre optimal','ZF avec ordre optimal','MMSE sans ordre
optimal','MMSE avec ordre optimal');
```

## Bibliographie

- [Ala98] S. M. Alamouti, "A simple transmit diversity technique for wireless communications", *IEEE J. Selected Areas Commun.*, vol. 16, no. 8, pp. 1451-1458, Oct. 1998.
- [Ara99] N. Aradhana, M. D. Trott, G. W. Wornell, "Performance limits of coded diversity methods for transmitter antenna arrays", *IEEE Transactions on Information theory*, vol. 45, no. 7, pp. 2418-2433, November 1999.
- [Bak01] D. A. Baker, "Space-Time block coding with imperfect channel estimates ". Thesis of Master of Science in Electrical Engineering at West Virginia University, pp. 3-20, 2001.
- [Bar01] S. Baro, G. Bauch, "Improving Performance of Layered Space-Time Architectures" ITG-Diskussionsitzung "Intelligente Antennen" TU Ilmenau, 16. März 2001.
- [Bar01a] S. Baro, G. Bauch, A. Pavlic, A. Semmler, "Improving BLAST Performance using Space-Time Block Codes and Turbo Decoding". Institute for Communications Engineering, Munich University of Technology, 2001.
- [Bea01] M. A. Beach, D. P. McNamara, P. N. Fletcher & P. Karisson, "MIMO a solution for advanced wireless access ?", University of Bristol, p. 233, 2001.
- [BLA00] "BLAST High-Level Overview".  
[www1.bell-labs.com/project/blast/high-level-overview.html](http://www1.bell-labs.com/project/blast/high-level-overview.html)
- [Cov91] T. M. Cover, J. A. Thomas, "Elements of Information theory", John Wiley & Sons, Inc, 1991.
- [Dam01] M.O. Damen, "Joint Coding/Decoding in a Multiple Access System, application to Mobile Communications", Thèse de Doctorat Soutenue le 22 Octobre 1999.
- [Dav00] R. David, "Historique", Juillet 2000.
- [Fos96] G. J. Foschini, "Layered space-time architecture for wireless communication in fading environments when using multiple antennas", *Bell Labs. Tech. J.*, vol. 2, Autumn 1996.
- [Fos98] P.W. Wonianski, G.J. Foschini, G. D. Golden, and R. A. Valenzuela, "V-BLAST: An architecture for realizing very high data rates over rich scattering wireless channels", in *ISSSE*, 1998, pp. 295-300.
- [Fos98a] G. J. Foschini, M. J. Gans, "On limits of wireless communications in a fading environment when using multiple antennas", *Wireless Personal Communications*, 6: pp. 311-335, aug. 1998.
- [Guo01] Z. Guo, P. Nilsson, "On detection Algorithm and Hardware Implementation for V-BLAST". Dept. of Electrosience, Lund University, Sweden, 2001.

[Hol01] B.Holter, "On the capacity of the MIMO channel -A tutorial introduction-", Oct. 2001.

[Loy02] S.Loyka, F.Gagnon, "Performance Analysis of the V-BLAST Algorithm : An Analytical Approach", 2002 International Zurich Seminar on broadband communications, Access-Transmission-Networking, February 19-21 Zurich Switzerland.

[Pro00] J. G. Proakis, "Digital Communications", McGraw-Hill, Inc., 2000.

[Rav02] N.Ravikiran, "Performance Analysis of Space-Time Coded Modulation Techniques using GBSB-MIMO Channel Models", Thesis Master of Science in Electrical Engineering Virginia Polytechnic Institute and State University, June 2002.

[Sci02] M.Sellathurai, "Fiche d'information",  
[http://www.nserc.ca/news/2002/p020307\\_b4.htm](http://www.nserc.ca/news/2002/p020307_b4.htm), mars 2002.

[Sob] P.Sobieski L.Vandendorpe, "Signaux de communications",  
[http://www.tele.ucl.ac.be/ELEC2880/tout\\_2880.pdf](http://www.tele.ucl.ac.be/ELEC2880/tout_2880.pdf)

[Tar98] V.Tarokh, N.Seshadri and A. R. Calderbank, "Space-time codes for high data rate wireless communication : Performance criterion and code construction", IEEE Trans. Inform. Theory, vol.44, pp. 744-765, March 1998.

[Tar99] V.Tarokh, A.Naguib, N.Seshadri and A. R. Calderbank, "Combined array processing and space-time coding", IEEE Trans. Inform. Theory, vol. 45, no. 4, pp. 1121-1128, May 1999.

[Tel95] I.Telatar, "Capacity of multi-antenna Gaussian channels", AT&T Technical Memorandum, June 1995.

[UMT99] Forum UMTS. Rapport no. 8 "The Future Mobile Market: Global trends and developments with a focus on Western Europe", Mars 1999.

[Wen02] M.Wennstrom, "On MIMO systems and adaptive arrays for wireless communication, Analysis and Practical Issues", Thèse de Doctorat de Uppsala University, 2002.

[Won00] K.W.Wong, C.Y.Tsui, and R.S.Cheng, "A low complexity architecture of the v-blast system," in *IEEE Wireless Communications and Networking Conference*, 2000.