Low complexity decoding schemes for MIMO systems

Abdellatif Salah

Thèse

présentée pour obtenir le grade de docteur

de TELECOM ParisTech

Spécialité : Électronique et Communications

Abdellatif SALAH

Schémas de décodage MIMO à Complexité Réduite

soutenu le 12, juillet 2010 devant le jury composé de

Pr. Dirk Slock	Rapporteurs
Dr. Cong Ling	
Pr. Mérouane Debbah	Examinateurs
Dr. Ahmed Saadani	
Pr. Hichem Besbes	
Dr. Ghaya Rekaya-Ben Othman	Directeurs de thèse
Pr. Jean-Claude Belfiore	

PhD thesis

TELECOM ParisTech

Communications and Electronics department

Digital communications group

Abdellatif SALAH

Low Complexity Decoding Schemes for MIMO Systems

Defense date: 07, 12 2010

Committee in charge:

Pr. Dirk Slock	Reporters
Dr. Cong Ling	
Pr. Mérouane Debbah	Examiners
Dr. Ahmed Saadani	
Pr. Hichem Besbes	
Dr. Ghaya Rekaya-Ben Othman	Advisors
Pr. Jean-Claude Belfiore	

TELECOM ParisTech

Dedicated to my father and mother for their encouragement during the progress of this work. I dedicate this thesis to my wife Manel for her suggestions and support and to our baby Malik. Thanks for all. My beloved brothers and sister: for your spirit, attention and pray.

Acknowledgment

This thesis bears my name on the cover, but it is the result not only of my work, but also that of many others who have helped or supported me along the way. Without them, the present work would not have been possible. This thesis would not have been possible without the kind support, the trenchant critiques, the probing questions, and the remarkable patience of my thesis advisor Dr. Ghaya Rekaya Ben Othman. I am heartily thankful to her for his encouragement, guidance and support from the initial to the final level enabled me to develop an understanding of the subject. I am also grateful to my advisor Pr. Jean-Claude Belfiore who provided me decisive and energetic support and for his assistance and willingness to discuss ideas.

I wish to thank TELECOM ParisTech for opening their doors to me, and for making all their resources available so that my work could be developed under the best conditions.

I would like to thank Pr. Dirk Slock, Dr Cong Ling, Pr. Mérouane Debbah, Dr. Ahmed Saadani, Pr. Hichem Besbes, for accepting to be the jury at my thesis defence. I'm grateful to my friends in France and in Tunisia for their support and encouragements. I am especially thankful to my son Malik for being the ultimate reason for finishing my thesis and to my wife Manel for his love and understanding. I am indebted to my parents: Sassi and Fattoum, my brothers: Mondher, Houcine, Omar, my sister Zohra for encouraging me and being patient. I would like to express ma gratitude to my family in-law for their care. Last, but not least, I'm thankful to my colleagues Rim Ouertani, Mireille Sarkiss, Sami Mekki, Maya Badr, Eric Bouton...I would like to thank also Rediet Getachew for his time spent reading my thesis and his thoughtful comments which led to great improvements.

Lastly, I offer my regards and blessings to all of those who supported me in any respect during the completion of the project.

Résumé

L'utilisation des antennes MIMO est une technique qui permet d'exploiter de façon très efficace la diversité spatiale et temporelle présente dans certains systèmes de communication, dont le canal sans fil. Le principal avantage de cette technique est une très grande efficacité spectrale. De nos jours, où le canal radio-mobile est de plus en plus utilisé pour transmettre tout type d'information, les méthodes permettant une utilisation plus efficace du spectre électromagnétique ont une importance fondamentale.

Les algorithmes de réception connus aujourd'hui sont très complexes, même en ce qui concerne les systèmes MIMO avec les codes espace-temps les plus simples. Cette complexité reste l'un des obstacles principaux à l'exploitation réelle.

Cette thèse présente une étude très détaillée de la complexité, la performance et les aspects les plus intéressants du comportement des algorithmes de la réception pour le décodage MIMO, étude qui présente un moyen rapide pour une éventuelle conception des architectures adaptées à ce problème.

Parmi les sujets présentés dans cette thèse, une étude approfondie de la performance et la complexité de ces algorithmes a été réalisée, ayant pour objectif d'avoir une connaissance suffisante pour pouvoir choisir, parmi le grand nombre d'algorithmes connus, le mieux adapté à chaque système particulier. Des améliorations aux algorithmes connus ont aussi été proposées et analysées.

Abstract

The use of MIMO antennas is a technique that allows to exploit in a very effective way the spatial and temporal diversity in certain systems of communication, of which the wireless communication systems. The main advantage of this technique is a good spectral efficiency. Nowadays, the mobile radio channel is increasingly used to transmit all type of information and methods allowing a more effective use of the spectrum have a fundamental importance.

Today, the well-known reception algorithms are very complex, even as regards the MIMO systems with the simplest space-time codes. This complexity remains one of the main obstacles in the real exploitation of this technique.

This thesis presents a detailed study of the complexity, the performance and the most interesting aspects of the behavior of the reception algorithms for MIMO decoding. This study presents a quick mean for a possible architectural conception adapted to this problem.

Among the subjects presented in this thesis, an in-depth study of the performance and the complexity of these algorithms was realized, having for objective to acquire enough knowledge to be able to choose, among the large number of known algorithms, the best adapted to every particular system. Improvements in the known algorithms were also proposed and analyzed.

Contents

List of Figures

List of abbreviations

AWGN	Additive White Gaussian Noise
BB	Branch and Bound
BeFS	Best First Search
BrFS	Breadth First Search
BER	Bit Error Rate
BICM	Bit Interleaved Coded Modulation
BLAST	Bell-labs LAyered Space-Time
BPSK	Binary Phase Shift Keying
BS	Base Station
CCDF	Cumulative Complementary Density Function
CLPS	Closest Lattice Point Search
CSIR	Channel State Information at the Receiver
CSITR	Channel State Information at the Transmitter and at the Receiver
DAST	Diagonal Algebraic Space-Time
DFE	Diagonal Algebraic Space-Time
DFS	Depth First Search
DL	Down Link
DSL	Digital Subscriber Line
ECC	Error Correcting Code
GC	Golden Code
IEEE	Institute of Electrical and Electronics Engineers
LDPC	Low-Density Parity Check
LDSTC	Linear Dispersion Space-Time Code
LLR	Log-Likelihood Ratio
LOS	Line-Of-Sight
LSD	List Sphere Decoder
LSTC	Layered Space-Time Code
MAP	Maximum A Posteriori
MIMO	Multiple-Input Multiple-Output
MISO	Multiple-Input Single-Output
ML	Maximum Likelihood
MMSE	Minimum Mean Squared Error
MMSE-GDFE	Minimum Mean Squared Error Generalized Decision Feedback Equalization
MSE	Mean Squared Error
NLOS	Non Line-Of-Sight

NVD	Non Vanishing Determinant
PER	Packet Error Rate
PSK	Phase-Shift Keying
OFDM	Orthogonal Frequency Division Multiplexing
OFDMA	Orthogonal Frequency-Division Multiple Access
OSIC	Ordered Successive Interferences Cancellation
OSTBC	Orthogonal Space-Time Block Code
QAM	Quadrature Amplitude Modulation
QPSK	QPSK Quadrature Phase Shift Keying
QRD	QR Decomposition
SB-Stack	Spherical Bound Stack decoder
SE	Schnorr-Euchner
SER	Symbol Error Rate
SD	Sphere Decoder
SIC	Successive Interferences Cancellation
SIMO	Single-Input Multiple-Output
SISO	Soft-Input Soft-Output
SSD	Shifted Sphere Decoder
SM	Spatial Multiplexing
SNR	Signal-to-Noise Ratio
SOVA	Soft-Output Viterbi Algorithm
ST	Space-Time
STBC	Space-Time Block Code
SVD	Singular Value Decomposition
UL	Up Link
UMTS	Universal Mobile Telecommunications System
WIMAX	Worldwide Interoperability for Microwave Access
WLAN	Wireless local Area Network
ZF	Zero-Forcing

List of Notations

Mathematical Notations

$E[a]$	Mean of the random variable a
σ_a^2	Variance of the random variable a
$\mathcal{N}(\mu, \sigma^2)$	Normal distribution with mean μ and variance σ^2
$\Re(z)$	Imaginary part of the complex z
$\Im(z)$	Real part of the complex z
$\underline{\boldsymbol{v}}$	Stacking the real part of $\boldsymbol{v}$ over its imaginary part as : $\underline{\boldsymbol{v}} = \begin{bmatrix} \mathfrak{Re}(\boldsymbol{v}) \\ \mathfrak{Im}(\boldsymbol{v}) \end{bmatrix}$
$\mathbb{Z}$	The set of all integers
$\mathbb{R}$	The set of all real numbers
$\mathbb{C}$	The set of all complex numbers
$\text{vec}(\boldsymbol{A})$	Vectorization of matrix $\boldsymbol{A}$
$\text{tr}(\boldsymbol{A})$	Trace of matrix $\boldsymbol{A}$
rank $(\boldsymbol{A})$	Rank of matrix $\boldsymbol{A}$
$\det(\boldsymbol{A})$	Determinant of matrix $\boldsymbol{A}$
$\boldsymbol{A}^T$	Transpose of matrix $\boldsymbol{A}$
$\boldsymbol{A}^H$	Hermitian conjugate of matrix $\boldsymbol{A}$
$\boldsymbol{A}^{-1}$	Inverse of $\boldsymbol{A}$
$\underline{\boldsymbol{A}}$	From complex valued matrix $\boldsymbol{A}$ to real valued matrix $\underline{\boldsymbol{A}}$ as follows : $\underline{\boldsymbol{A}} = \Psi(\boldsymbol{A}) = \begin{bmatrix} \mathfrak{Re}(\boldsymbol{A}) & -\mathfrak{Im}(\boldsymbol{A}) \\ \mathfrak{Im}(\boldsymbol{A}) & \mathfrak{Re}(\boldsymbol{A}) \end{bmatrix}.$
$\boldsymbol{I}_N$	Identity Matrix of dimension $N \times N$
$\boldsymbol{A}_{\setminus i}$	Matrix comprising all columns of $\boldsymbol{A}$ but the i^{th}
$\boldsymbol{A}_i^j$	Matrix comprising columns i through j of $\boldsymbol{A}$
$\boldsymbol{A}_{ik}^{jl}$	Matrix comprising rows i through j and columns k through l of $\boldsymbol{A}$
$\boldsymbol{A}_{\setminus ij}$	Matrix formed by removing the i^{th} row and j^{th} column

$$(x)^+ = \begin{cases} a & \text{if } x > 0 \\ 0 & \text{else} \end{cases}$$

Propagation Channel

$h(t,\tau)$	Impulse response of the linear filter modeling the propagation channel
$s(t)$	Transmitted signal
$n(t)$	Noise
$r(t)$	Received signal
B_c	Channel coherence band
B_s	Transmitted signal
$s(t)$	Bandwidth
T_s	Symbol duration

Résumé en Français

Dans un premier temps, on présente le fonctionnement général d'un système de communication numérique. On se focalise particulièrement sur les systèmes MIMO. D'abord, on présente le modèle du canal pour un système MIMO. Ainsi, différentes techniques de modélisation sont présentées et quelques modèles issus de la littérature sont classifiés. On introduit ensuite les techniques de diversité et les propriétés de lattice. La probabilité de coupure et le gain de multiplexage sont présentés. Le tradeoff diversité-multiplexage est aussi détaillé.

0.1 Canal MIMO et Modélisation du Systeme

0.1.1 Schemas de Transmission

Un diagramme pour un système de communication radio sans fils est présenté dans la figure (1.1). Dans cette figure, la source d'information pourrait être de la voix, de la vidéo ou des données. Le codeur source traite l'information et la formate en une séquence binaire bits $\in \{\pm 1\}$. L'objectif du codage source est de supprimer la redondance non structurée de la source. Le codage canal ajoute de la redondance structurée dans l'objectif de protéger l'information de la distorsion et du bruit du canal grâce á cette diversité. Le modulateur va mapper la séquence des bits codés en ondes radio convenantes pour une transmission á travers le canal. L'amplitude du signal décroit á cause de la distance entre l'émetteur et le récepteur. C'est la perte de propagation. A cause des obstacles, l'amplitude du signal est atténué, c'est ce qu'on appelle shadowing. Finalement, à cause de la propagation multi-trajets entre l'antenne émettrice et l'antenne réceptrice, le signal est déformé. L'évanouissement multi-trajets peut être constructif ou destructif. Le caractère constructif ou destructif de l'évanouissement du canal dépend de la fréquence porteuse du signal et ainsi appelé canal sélectif en fréquence. En plus des effets de propagation, vient s'ajouter un bruit au niveau du récepteur décorrelé du signal transmis : bruit thermique et interférences co-canal et issus des canaux adjacents.
L'objectif du récepteur est de reproduire à la sortie du décodeur source le signal émis aussi précisément que possible.

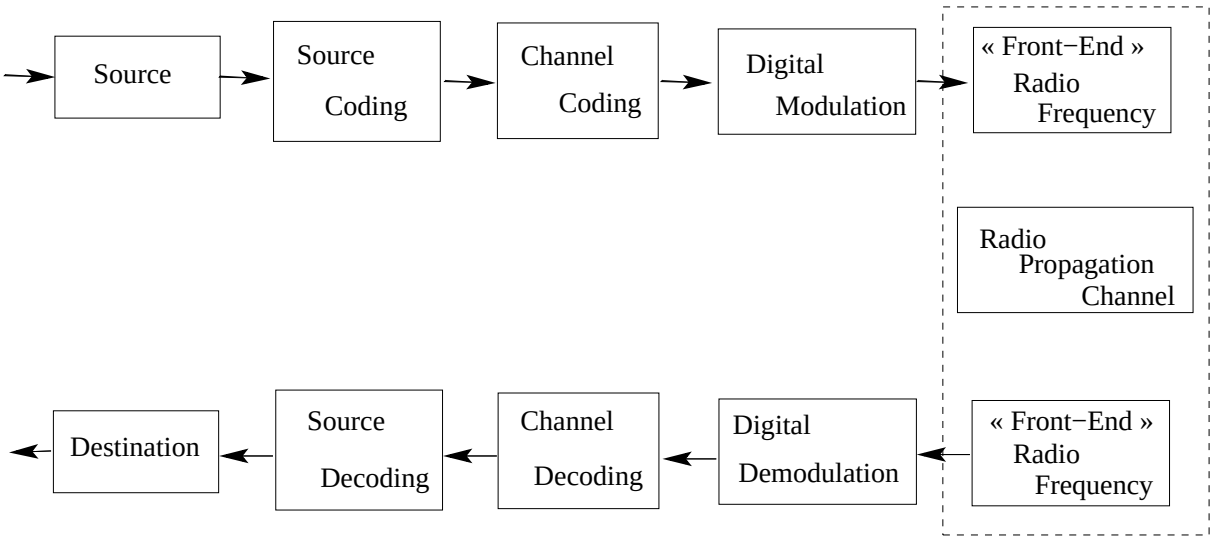


FIGURE 1 – Generic Architecture of a Digital Communication System

Ainsi, la modélisation du canal de transmission est essentielle pour concevoir un système de communication numérique performant, le modèle du canal dépend ainsi de l'environnement et du modèle de propagation. Comme montré dans la figure (1.2), le récepteur détecte les différentes versions du signal. L'écho de ce même signal n'est autre que l'interaction de l'onde avec l'environnement de propagation réel : diffraction, réflexion, dispersion (2).

La représentation mathématique simple pour un canal de communication est la représentation avec un filtre linéaire dont s'ajoute un bruit additif. Le signal transmis est ainsi corrompu par le bruit additif (3).

Figure (1.3) illustre les différentes configurations d'antennes utilisées pour pour définir des systèmes spatio-temporels. Single-input single-output (SISO) est le modele sans fil le plus connu, single-input multiple-output (SIMO) utilises une seule antenne de transmission et plusieurs en réception. Multiple-input single-output (MISO) utilises plusieurs antennes en transmission et une seule en réception. Considérons un système MIMO avec M antennes de transmission et N antennes en réception. Ainsi le signal en block reçu :

$$\boldsymbol{Y_{N\times T}} = \boldsymbol{H_{N\times M}} \cdot \boldsymbol{X_{M\times T}} + \boldsymbol{W_{N\times T}}, \tag{1}$$

Avec $\boldsymbol{H}$ est la matrice du canal avec des entrées complexes h_{ij} représentant les coefficients d'évanouissement entre le i^{me} antenne de réception et le j^{me} antenne de transmission, et sont modélisées par des variables gaussiens indépendants de moyenne zéro et de variance 0.5 par composante.

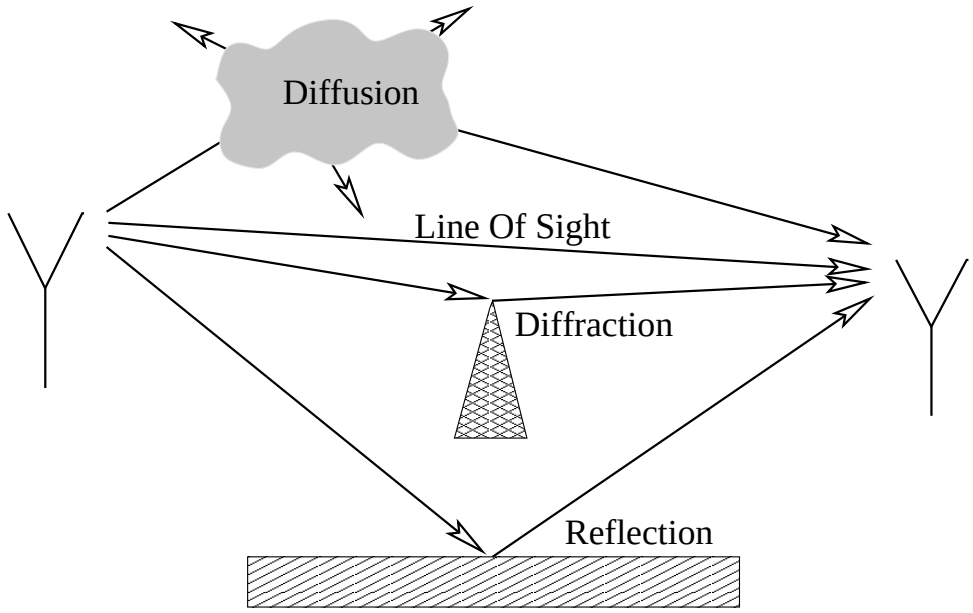


Figure 2 – Multipath Propagation of Radio Signals

Le canal MIMO $\boldsymbol{H}$ peut être représenté à un instant donné par une matrice de taille $N \times M$

$$\boldsymbol{H_{N\times M}} = \begin{bmatrix} h_{11} & h_{12} & \cdots & h_{1M} \\ h_{21} & h_{22} & \cdots & h_{2M} \\ \vdots & \vdots & \ddots & \vdots \\ h_{N1} & h_{N2} & \cdots & h_{NM} \end{bmatrix}, \tag{2}$$

La figure (1.4) représente ce système. $\boldsymbol{X_{M\times T}}$ est le bloc du signal transmis sur un T périodes symbole et une matrice de dimensions $M \times T$. T est la dimension temporelle, c'est ainsi le nombre d'utilisations canal (uc). $\boldsymbol{W_{N\times T}} \in \mathcal{N}(0,\sigma^2.\boldsymbol{I})$ est un bruit additif gaussien. Le canal est supposé constant durant la transmission d'un bloc (applé aussi des fois : trame) et change d'un bloc à un autre. Le canal est supposé aussi sans mémoire entre les blocs : ainsi les matrices associés aux différents blocs sont statistiquement indépendants. Ce canal est connu comme un canal plat en fréquence, un canal à évanouissement lent ou tout simplement un canal à évanouissement par bloc. Ces caractéristiques sont typiques dans des applications sans fils fixes (WiFi...), ou un changement lent du canal est prévu : un exemple sera un environnement bureautique ou les gens peuvent bouger en marchant. La matrice $\boldsymbol{H}$ est supposé de rang plein. Ceci étant justifié car la probablilité de generer une matrice aléatoirement présentant des lignes ou des colonnes non-indépendant est très proche de zero. En pratique, cela veut dire que les antennes de réception ou d'émission doivent être bien espacées. Cette exigence n'est pas considéré comme déraisonnable dans les applications modernes sans fil où la fréquence porteuse est dans la gamme de quelques

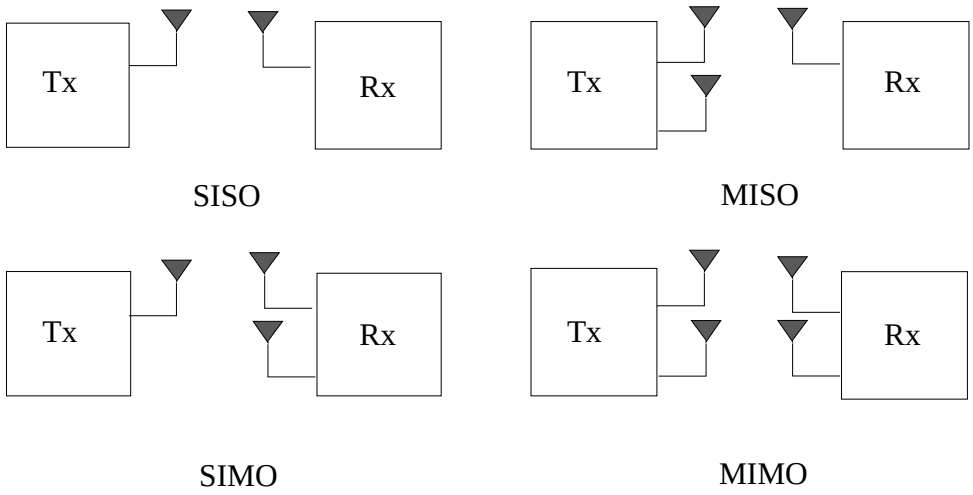


FIGURE 3 – Different antenna configurations in space-time systems.

gigahertz et donc la séparation requise serait de quelques centimètres. Chaque récepteur est supposé voir estimé $\boldsymbol{H}$ parfaitement à travers l'utilisation de certains procédés appropriés, telle qu'une séquence d'apprentissage transmis à chaque bloc. Cette situation est souvent décrite dans la littérature comme le récepteur ayant une connaissance parfaite de l'état du canal (CSI).
Les entrées $|h_{ij}|$ sont supposés avoir une distribution de Rayleigh. Les entrées de la matrice de canal $\boldsymbol{H}$ sont complexes et gaussiens. Chaque composant définit un gaussien réel et une partie imaginaire gaussienne de moyenne nulle et de variance égale à 0,5. Ainsi, le canal $\boldsymbol{H}$ est considéré comme un canal de Rayleigh.

0.1.2 Modulation and Demodulation

Modulation et la démodulation sont utilisées dans de nombreux types de transmission de données analogique et numérique. Le choix d'un type de modulation est basé sur la bande passante et le rapport signal bruit. Dans la modulation numérique, une porteuse analogique est modulée par un train de bits numériques. Les procédés de modulation numériques peuvent être considérés comme convertisseur numérique-analogique, et la démodulation comme convertisseur analogique-numérique. Les changements dans le signal de porteuse sont choisis parmi un nombre fini de symboles (l'alphabet de modulation). Si l'alphabet se compose de 2^{N_b} symboles, chaque symbole représente un message constitué de N_b bits. Habituellement, pour nos simulations, nous allons utiliser la Quadrature Amplitude Modulation (QAM). La modulation QAM est tout simplement une combinaison de modulation d'amplitude et de modulation par déplacement de phase. Ses points de la constellation sont généralement organisés dans une grille carrée avec un espacement vertical et horizontal égale. L'ensemble des points des constellations est un sous-ensemble final de $\mathbb{Z}[i]$. Depuis dans les télécommunications numériques, les données sont généralement

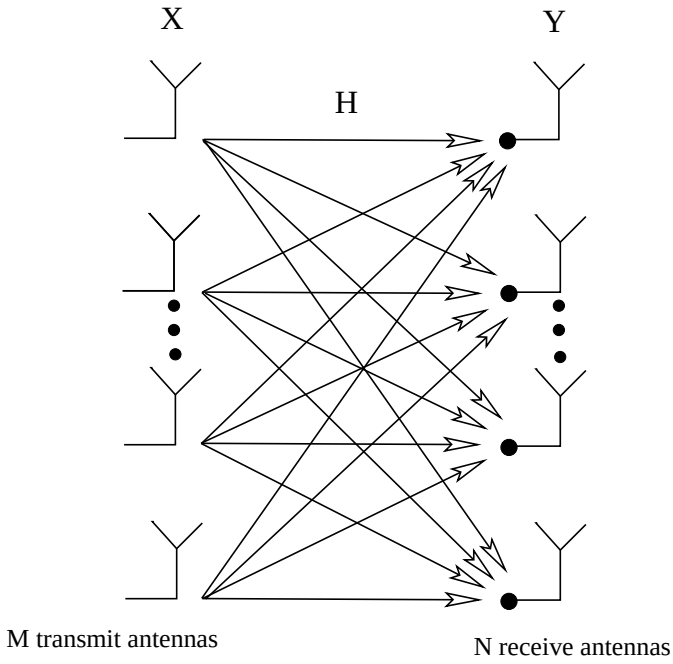


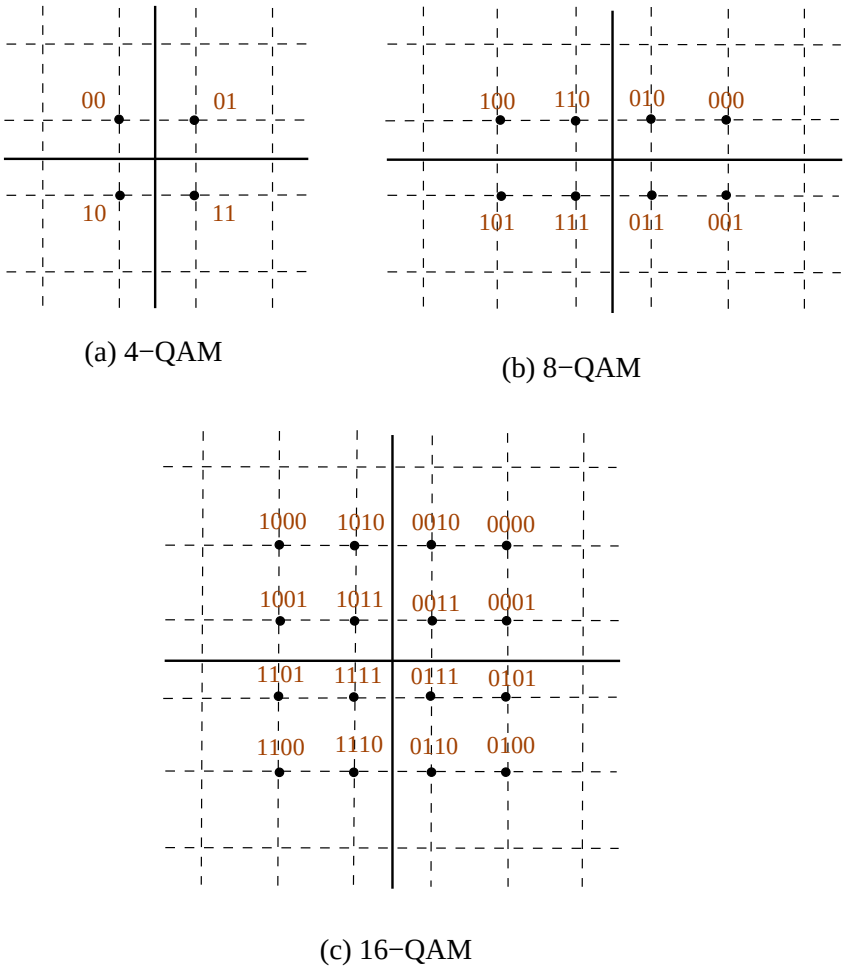
Figure 4 – MIMO System representation constituted of M transmit and N receive antennas

binaires, le nombre de points dans la grille est généralement une puissance de 2 (2, 4, 8 ...). Quelques exemples de constellations $q - QAM$ avec q = 4,8,16 sont présentées dans la figure (1.5).
En passant à une constellation d'ordre supérieur, il est possible de transmettre plus de bits par symbole. Toutefois, si l'énergie moyenne de la constellation reste la même, les points doivent être rapprochés et sont donc plus sensibles au bruit ; il en résulte un taux d'erreur binaire plus élevé et d'ordre supérieur QAM peut fournir des données moins fiable que d'ordre inférieur QAM, pour une énergie moyenne constante.

0.1.3 Les Techniques de diversité

Les techniques de diversité fonctionnent sur la dimension temps, fréquence ou espace, mais l'idée de base est la même. En envoyant des signaux qui transportent la même information par des voies différentes, multiples indépendamment atténués. Plusieurs répliques du signal sont obtenues au niveau du récepteur et une plus fiable détection peut être atteinte. Il ya trois types de schémas de diversité dans les communications sans fil

la diversité temporelle : Dans ce cas, les répliques du signal émis sont fournis dans le temps par une combinaison de codage de canal et des stratégies d'entrelacement temporelle. La principale exigence ici pour cette forme de diversité pour être efficace est que le



(a) 4−QAM

(b) 8−QAM

(c) 16−QAM

FIGURE 5 – Example of a $\mathbb{Z}[i]$ lattice of dimension 2

canal doit fournir suffisamment de variations dans le temps. Elle est applicable dans le cas où le temps de cohérence du canal est faible par rapport à la durée d'entrelacement désirée de symbole. Dans un tel cas, nous sommes assurés que le symbole entrelacé est indépendant du symbole précédent. Ceci le rend une réplique tout à fait nouvelle du symbole d'origine.

diversité fréquentielle : Ce type de diversité offre des répliques du signal original dans le domaine fréquentiel. Ceci est applicable dans les cas où la bande de cohérence du canal est faible par rapport à la bande passante du signal. Cela nous assure que les différentes parties du spectre verront des évanouissements indépendants.

La diversité spatiale : Elle est également appelé la diversité de l'antenne et est elle un moyen efficace de lutte contre l' évanouissement multi-trajets. Dans ce cas, les répliques du même signal transmis sont prévues dans différentes antennes du récepteur. Ceci est applicable dans les cas où l'espacement d'antenne est plus grande que la la distance de cohérence pour assurer des évanouissements indépendants à travers différentes antennes.

Fondamentalement, l'efficacité de tout schéma de diversité réside dans le fait que, au récepteur on doit fournir des versions indépendantes s du signal transmis. Dans un tel cas nous sommes assurés que la probabilité que deux signaux ou plus encourent un évanouissement profond sera très faible. Les contraintes qui pèsent sur le temps de cohérence, la bande de cohérence, et la distance de cohérence le confirme. Le schéma de diversité doit donc combiner de façon optimale les formes d'onde reçues diversifiées de manière à maximiser la qualité du signal qui en résulte.
Nous pouvons également classer la diversité en diversité d'émission et de réception.

diversité de réception : Maximum Ratio Combining est un schéma de diversité souvent appliqué dans les récepteurs pour améliorer la qualité du signal. Dans les téléphones cellulaires, il devient coûteuse et lourde à déployer. C'est une des raisons principales que la diversité d'émission est devenue populaire, puisque la diversité d'émission est plus facile à mettre en ÅŞuvre au niveau de la station de base.

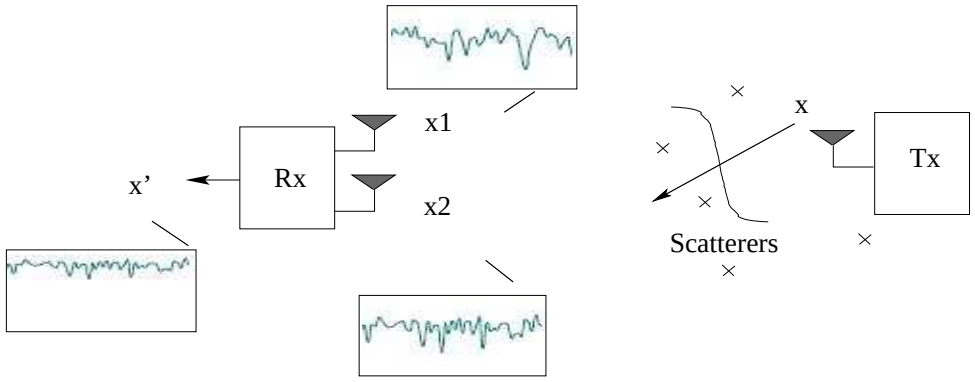


Figure 6 – Spatial Receive Diversity

Diversité de tranmission Dans ce cas, nous introduisons de la redondance contrôlée au niveau de l'émetteur, qui peut être ensuite exploité par des techniques de traitement de signal appropriées au niveau du récepteur. En général, cette technique nécessite une connaissance parfaite du canal au niveau de l'émetteur. Mais avec l'invention du codage espace-temps, comme le système Alamouti (4), il est devenu possible de mettre en oeuvre la diversité de transmission sans connaissance du canal. Ce fut l'une des raisons fondamentales pour lesquelles le MIMO industriel commence à prendre essort. Codes espace-temps pour la transmission MIMO exploite à la fois la diversité d'emission et de reception ce qui aboutit à une bonne qualité en reception.

Par conséquent, en MIMO, nous parlons beaucoup de la diversité de réception ou d'émission. En diversité de réception, le récepteur qui a plusieurs antennes reçoit plusieurs répliques du signal transmis même, en supposant que le la transmission est venu de la même source. Cela est vrai pour les canaux SIMO. Si le trajet du signal entre chaque paire d'antennes s'évanouit de manière indépendante, ainsi, quand un chemin s'évanouit, il est

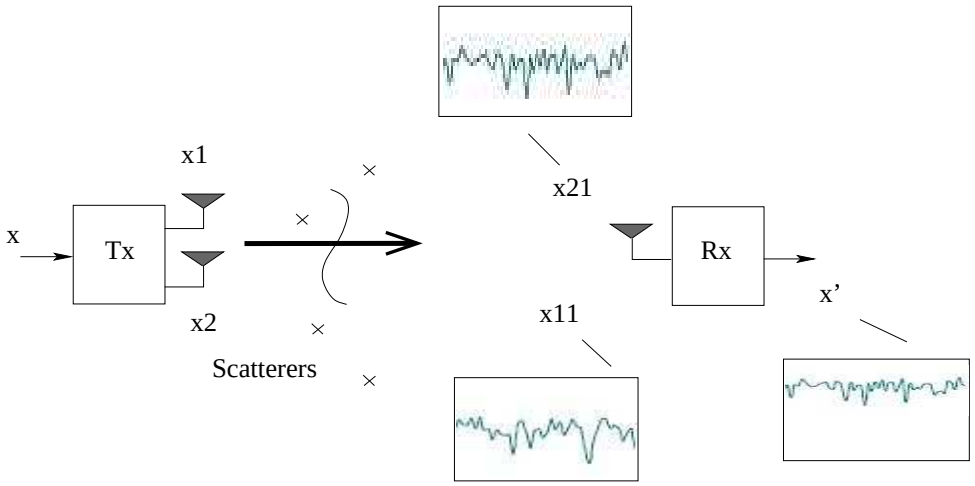


FIGURE 7 – Spatial Transmit Diversity

extrêmement peu probable que tous les autres chemins sont aussi en évanouissement. Si le nombre d'antennes de réception tend vers l'infini, la diversité tend vers l'infini et le canal tend vers un canal à bruit blanc gaussien additif (AWGN) (5).
Dans la catégorie de la diversité spatiale, il ya deux types de diversité en plus que nous devons examiner. Ce sont :

diversité de polarisation : Dans ce type de diversité de polarisation horizontale et verticale, les signaux sont transmis par deux antennes polarisées différemment et reçues respectivement par deux antennes polarisées différemment au niveau du récepteur. Différentes polarisations assurent qu'il n'y ait pas de corrélation entre les données, sans avoir à se soucier de la distance de cohérence entre les antennes.
Diversité d'Angle Cela s'applique à des fréquences porteuses de plus de 10 GHz. À ces fréquences, les signaux transmis sont hautement dispersés dans l'espace. Dans un tel cas, le récepteur peut avoir deux antennes très directives face dans des directions totalement différentes. Ceci permet au récepteur de détecter deux échantillons du même signal, qui sont totalement indépendantes les unes des autres.

Considérant un système MIMO $N \times M$, le gain de diversité maximal possible est égal à $N \times M$. À haut SNR, la probabilité d'erreur P_e diminue à d^{me} en puissance de SNR, correspondant à une pente de $-d$ dans la courbe de probabilité d'erreur (en échelle dB / dB).

$$P_e \propto \frac{1}{SNR^d} \tag{3}$$

Ainsi, la diversité est

$$d = -\lim_{SNR\to\infty} \frac{\log(P_e)}{\log(SNR)}. \tag{4}$$

Donc, pour SNR élevé, en moyenne la probabilité d'erreur diminue asymptotiquement,

donc la probabilité d'erreur diminue si nous envoyons des informations sur d chemins indépendants.

0.1.4 Définition de Lattice et ses propriétés

Un lattice Λ est un sous ensemble de rang p; pour $p < n$, de $\mathbb{R}^n$. Λ est ainsi un lattice de dimension p et il existe p vecteurs $\boldsymbol{v}_1,\boldsymbol{v}_2,\ldots,\boldsymbol{v}_p \in \mathbb{R}^n$ de dimension n tel que :

$$\Lambda = \Lambda(\boldsymbol{S}) = \{a_1\boldsymbol{v}_1 + a_2\boldsymbol{v}_2 + \ldots + a_p\boldsymbol{v}_p : a_i \in \mathbb{Z}\} \tag{5}$$

avec $\boldsymbol{S} = [\boldsymbol{v}_1,\boldsymbol{v}_2,\ldots,\boldsymbol{v}_p]$ est une matrice de dimensions $n \times p$. L'ensemlbe des vecteurs colonnes $\{\boldsymbol{v}_1,\boldsymbol{v}_2,\ldots,\boldsymbol{v}_p\}$ et la matrice $\boldsymbol{S}$ sont appelés respectivement la base et la matrice de base de Λ. Ainsi, un lattice est une combinaison linéaire entière des vecteurs de base. Dans le reste de document, un lattice ayant $\boldsymbol{S}$ comme matrice de base sera noté Λ_S.
Considérons ainsi quelques définitions utiles, (voir figure (1.8)) :

• La matrice de Gram d'un lattice Λ_S est $\boldsymbol{G} = \boldsymbol{S}^T\boldsymbol{S}$.

• Le lattice équivalent
Soit $\boldsymbol{Q}$ $dans M_n(\mathbb{R})$, de telle sorte que $\boldsymbol{Q}\boldsymbol{Q}^T = \boldsymbol{I}_n$.
Les deux lattices Λ_S et $\Lambda_{S.Q}$ sont équivalents (même lattice).

• volume fondamental d'un lattice
la volume fondamentale d'un lattice Λ_S ayant une base $\{\boldsymbol{v}_1,\boldsymbol{v}_2,\ldots,\boldsymbol{v}_p\}$ dans $\mathbb{R}^n$ est donné par :

$$\boldsymbol{V} = \{\boldsymbol{x} \in \mathbb{R}^n \backslash \boldsymbol{x} = a_1\boldsymbol{v}_1 + a_2\boldsymbol{v}_2 + \ldots + a_p\boldsymbol{v}_p, 0 \leq a_i < 1, i = 1\ldots p\} \tag{6}$$

Géométriquement, le determinant $det(\Lambda)$ d'un lattice Λ est défini comme étant le contenu du parallélépipède engendré par les bases du lattice. Généralement, un lattice peut avoir plusieurs bases possibles mais toujours le même déterminant.

• Cellule de Voronoï
La cellule de Voronoï d'un point $\boldsymbol{u}$ dans un lattice Λ est la région défini par

$$v(\boldsymbol{u}) = \{\boldsymbol{x} \in \mathbb{R}^n \backslash \|\boldsymbol{x} - \boldsymbol{u}\| \leq \|\boldsymbol{x} - \boldsymbol{y}\|, \boldsymbol{y} \in \Lambda\} \tag{7}$$

Ainsi, la cellule de Voronoï est une structure où chaque interieur d'une cellule consiste en tous les points proches d'un point particulier du lattice que tout autre point dans le lattice. Comme le lattice est uniforme, toutes les cellules de Voronoï sont identiques. Le volume fondamental d'un lattice est égale au volume d'une cellule de Voronoï. La représentation des systèmes MIMO comme lattice et les décoder avec un décodeur lattice a été initialement exploré by Damen et al. dans (6). On peut séparer la partie imaginaire et la partie réelle et vectorizer le système MIMO (codé et non codé) et obtenir ainsi une

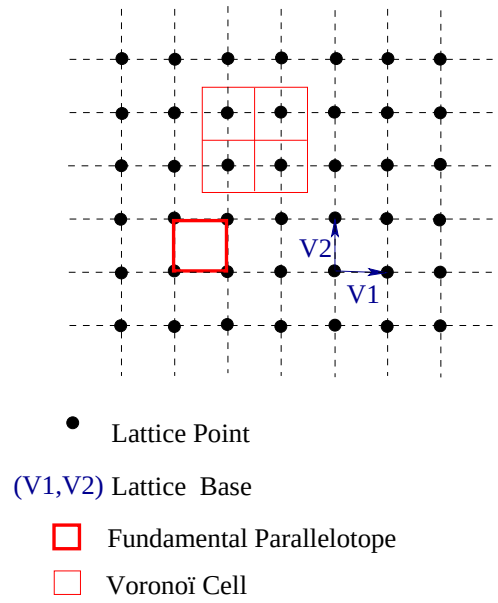


FIGURE 8 – Example of a $\mathbb{Z}[i]$ lattice of dimension 2

représentation en lattice du modèle du canal.

0.1.4.1 Pour le cas du système non codé

Ici, on suppose que T, la dimension temporelle, est égale à 1. Soit le mapping inversible $\Psi : \mathbb{C}^M \rightarrow \mathbb{R}^{\underline{M}}$ du vecteur complexe $\boldsymbol{v}$ en empilant la partie réelle du vecteur $\boldsymbol{v}$ sur la partie imaginaire, défini par

$$\underline{\boldsymbol{v}} = \Psi(\boldsymbol{v}) = \begin{bmatrix} \mathfrak{Re}(\boldsymbol{v}) \\ \mathfrak{Im}(\boldsymbol{v}) \end{bmatrix}, \tag{8}$$

Où $\underline{M} = 2M$.
Où $N \neq 1$, le mapping $\Psi : \mathbb{C}^{N \times M} \rightarrow \mathbb{R}^{\underline{N} \times \underline{M}}$ and $\underline{N} = 2N$ d'une matrice complexe $\boldsymbol{A}$ à une matrice reelle $\underline{\boldsymbol{A}}$ est definit comme suit :

$$\underline{\boldsymbol{A}} = \Psi(\boldsymbol{A}) = \begin{bmatrix} \mathfrak{Re}(\boldsymbol{A}) & -\mathfrak{Im}(\boldsymbol{A}) \\ \mathfrak{Im}(\boldsymbol{A}) & \mathfrak{Re}(\boldsymbol{A}) \end{bmatrix}. \tag{9}$$

Ainsi, on peut réécrire l'équation (1.5) en séparant la partie réelle et la partie imaginaire comme suit :

$$\underline{\boldsymbol{y}}_{\underline{N}} = \underline{\boldsymbol{H}}_{\underline{N} \times \underline{M}} \underline{\boldsymbol{x}}_{\underline{M}} + \underline{\boldsymbol{z}}_{\underline{N}}. \tag{10}$$

Comme la matrice $\boldsymbol{H}_{\boldsymbol{N \times M}}$ est de rang plein, la matrice $\underline{\boldsymbol{H}}_{\underline{N} \times \underline{M}}$ est ainsi de rang plein. On obtient donc une représentation en lattice du système MIMO non codé. Les dimensions du lattice sont $\underline{N} \times \underline{M}$ et la matrice génératrice est $\underline{\boldsymbol{H}}_{\underline{N} \times \underline{M}}$.

0.1.4.2 Pour le cas du système codé

Dans ce cas $\boldsymbol{X}$ n'est plus un vecteur symbole comme dans le cas d'un système non codé. Mais, il représente plutôt la matrice mot de code à envoyer. Le mot de code reçu reste comme décrit dans l'équation (1.5). Deux étapes doivent être suivies pour avoir une représentation en lattice du système MIMO codé :
1) représenter le système code comme un système non code,
2) séparer les parties imaginaires et réelles. Premier pas consiste dans la vectorisation. Ainsi, l'équation (1.5) devient

$$\boldsymbol{y_{N\cdot T}} = \begin{pmatrix} \boldsymbol{H_{N\times M}} & & 0 \\ & \ddots & \\ 0 & & \boldsymbol{H_{N\times M}} \end{pmatrix} \cdot \underbrace{\begin{pmatrix} \phi_{11} & \dots & \phi_{1,M\cdot T} \\ \vdots & \ddots & \vdots \\ \phi_{M\cdot T,1} & \dots & \phi_{M\cdot T,M\cdot T} \end{pmatrix}}_{\boldsymbol{\phi_{M\cdot T\times M\cdot T}}} \cdot \underbrace{\begin{pmatrix} x_1 \\ \vdots \\ x_{M\cdot T} \end{pmatrix}}_{\boldsymbol{x_{M\cdot T}}} + \begin{pmatrix} z_1 \\ \vdots \\ z_{N\cdot T} \end{pmatrix},$$

Avec Comme rĂŹesultat, on obtient un système ĂŹequivalent à (1.5)

$$\begin{aligned} \boldsymbol{y_{N\cdot T}} &= \boldsymbol{H_{1,N\cdot T\times M\cdot T}} \cdot \boldsymbol{\phi_{M\cdot T\times M\cdot T}} \cdot \boldsymbol{x_{M\cdot T}} + \boldsymbol{z_{N\cdot T}} \\ &= \boldsymbol{\mathcal{H}_{N\cdot T\times M\cdot T}} \cdot \boldsymbol{x_{M\cdot T}} + \boldsymbol{z_{N\cdot T}} \end{aligned} \tag{11}$$

La séparation de la partie réelle et imaginaire est appliquée à l'équation précédente comme pour l'équation (1.15), et le système codé devient

$$\underline{\boldsymbol{y}}_{\underline{\boldsymbol{N}}\cdot\underline{\boldsymbol{T}}} = \underline{\boldsymbol{\mathcal{H}}}_{\underline{\boldsymbol{N}}\cdot\underline{\boldsymbol{T}}\times\underline{\boldsymbol{M}}\cdot\underline{\boldsymbol{T}}} \cdot \underline{\boldsymbol{x}}_{\underline{\boldsymbol{M}}\cdot\underline{\boldsymbol{T}}} + \underline{\boldsymbol{z}}_{\underline{\boldsymbol{N}}\cdot\underline{\boldsymbol{T}}} \tag{12}$$

Où on definit par $\underline{\boldsymbol{\mathcal{H}}}_{\underline{\boldsymbol{N}}\cdot\underline{\boldsymbol{T}}\times\underline{\boldsymbol{M}}\cdot\underline{\boldsymbol{T}}}$ la matrice équivalente génératrice du lattice.

0.2 Décodage MIMO

0.2.1 Introduction

Dans cette partie, on présente l'etat de l'art des algorithmes de décodage MIMO existant en littérature. On discutera dans un premier temps, les decodeurs sous-optimaux, comme le decodeur de Forcage à Zero (Zero Forcing : ZF), l'algorithme de me minimisation de l'erreur quadratique (Minimum Mean Square Error : MMSE), et les algorithmes à retour de décision, etc.
On présente ensuite, les décodeurs MIMO optimaux, et on se focalisera en particulier sur ceux basés sur une représentation en lattice et sur les algorithmes séquentiels. On distingue ainsi deux categories, les decodeurs utilisant la stratégie de Pohst comme le Sphère Décodeur et le Schnorr-Euchner, algorithmes utilisants la stratégie de Dijkstra (ex : best-first-search) comme le decodage par stack et le decodage de Fano. On montrera que cette dernière catégorie offrira un compromis complexité-performance.

0.2.2 Du modèle canal vers le design de Lattice

La théorie de Lattice est appliquée efficacement pour encoder et décoder l'information dans la transmission numérique avec antennes multiples. La théorie de lattice est un outil mathématique puissant pour représenter le canal géométriquement et comprendre son comportement dans l'objectif de designer un bon modulateur et un bon démodulateur.

0.2.3 Décodeurs MIMO : Principes de base et structures

Dans cette thèse, les schémas MIMO étudiés sont les schémas spatio-temporels. Ainsi, il est possible de représenter le système comme un système non-codé équivalent. Pour guise de simplicité, on considérera le schémas de transmission non codé. Donc, les algorithmes de décodage étudiés et proposés pour la suite reste valable pour un système codé, seulement les dimensions du système vont changer. Rappelons, l'equation qui represente le système MIMO avec M antennes de transmission et N antennes de réception. Soit la matrice mot de code $\boldsymbol{X}$ de dimension $M \times T$et le signa reçu $\boldsymbol{Y}$de dimensions $N \times T$ vérifiant :

$$\boldsymbol{Y} = \boldsymbol{H}\boldsymbol{X} + \boldsymbol{W} \tag{13}$$

Apres vectorisation, le système peut être écrit sous cette forme :

$$\boldsymbol{y_{eq}} = \boldsymbol{H_{eq}}\boldsymbol{s} + \boldsymbol{w_{eq}} \tag{14}$$

Où $\boldsymbol{y_{eq}}$ et $\boldsymbol{w_{eq}}$ sont les vecteurs colonnes composés de NT éléments obtenus de $\boldsymbol{Y}$ et $\boldsymbol{W}$. $\boldsymbol{s}$ est le vecteur composé de p symboles encodés avec la matrice mot de code $\boldsymbol{X}$. La matrice du canal équivalente $\boldsymbol{H_{eq}}$ de dimensions $NT \times p$ inclut la réponse du canal et l'opération du codage spatio-temporel. Dans la suite, pour simplifier les notations, on va plus mentionner l'indice $_{\boldsymbol{eq}}$. Soit aussi $n = NT$. Comme résultat, les nouveaux dimensions du système sont $n \times p$.
On suppose aussi une transmission cohérente (matrice canal connu à l'émission). Le system devient ainsi :

$$\boldsymbol{y} = \boldsymbol{H} \cdot \boldsymbol{s} + \boldsymbol{w}, \tag{15}$$

L'objectif derrière la transmission MIMO est de trouver l'estimation du vecteur transmis. Le decodage optimal est le maximum de vrai ressemblance (maximum likelihood : ML). Il s'agit de trouver le vecteur le plus proche $\boldsymbol{s}$ qui minimise la métrique

$$\hat{\boldsymbol{s}} = \arg\min_{\boldsymbol{s} \in \mathcal{C}_s} \|\boldsymbol{y} - \boldsymbol{H} \cdot \boldsymbol{s}\| \tag{16}$$

Où $\mathcal{C}_s$ est l'ensemble formé par les vecteurs de la constellation. Le récepteur ML cherche dans tous les vecteurs de la constellation, le vecteur signal le plus probablement transmis. En analysant la structure des décodeurs, on peut déduire que trois phases peuvent être distinguées dans la construction. Chaque décodeur peut inclure quelques phases ou toutes les phases. Ceci dépendra du compromis compexité-performance recherché. Les décodeurs incluant toutes les phases offriront une estimation plus optimale mais souffriront néanmoins d'une énorme complexité.
Phase Zéro : Prétraitement

Le prétraitement est une phase optionnelle. Etant donné un problème de recherche, la phase de prétraitement est utile pour améliorer l'efficacité du décodage. Le prétraitement peut être séparé en deux étapes indépendantes : le prétraitement à gauche et le prétraitement à droite. Ces deux étapes seront détaillés plus tard.

Phase deux : Un premier point

Cette phase permet d'obtenir une première estimation. L'avantage est d'obtenir rapidement un premier résultat qui n'est pas forcement optimal. Quelques fois, le récepteur a besoin d'obtenir une estimation rapide. Dans ce cas, cette phase peut etre suffisante puisque elle offre une complexité très réduite même si le résultat n'est pas optimal. Ce premier point peut être amélioré en utilisant d'autres phases, mais ca rendra le cout de la complexité plus important.

Phase deux : un meilleur Point

Le premier point obtenu précédemment sera amélioré pour avoir une estimation plus fiable. Ce premier point est généralement utilisé comme initialisation pour la deuxième phase pour trouver un point plus fiable. Cette phase ne peut pas être indépendante de la première phase.

0.2.4 Les classes des décodeurs MIMO

0.2.4.1 Les décodeurs MIMO sous-optimaux

Le décodeur ZF Le récepteur ZF est un récepteur linéaire. Il se comporte comme un filtre linéaire $\boldsymbol{F}$ et il sépare les flux des datas pour décoder ainsi indépendamment chaque flux. On suppose que le matrice du canal $\boldsymbol{H}$ est inversible et on estime le vecteur transmis comme suit :

$$\hat{\boldsymbol{s}} = \left(\boldsymbol{H}^H \boldsymbol{H}\right)^{-1} \boldsymbol{H}\boldsymbol{s} = \boldsymbol{H}^{\dagger}\boldsymbol{s} \tag{17}$$

où † représente le pseudo-inverse. Puisque l'inverse de $\boldsymbol{H}$ ne peut exister que si les colonnes de $\boldsymbol{H}$ sont independents, il est supposé que les elements de $\boldsymbol{H}$ sont i.i.d. Ainsi :

$$\boldsymbol{F_{ZF}} = \left(\boldsymbol{H}^H \boldsymbol{H}\right)^{-1} \boldsymbol{H} \tag{18}$$

et

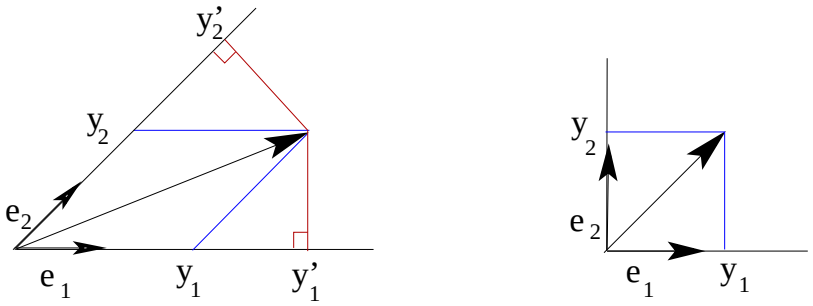
$$\boldsymbol{F_{ZF}} \cdot \boldsymbol{y} = \boldsymbol{s} + \boldsymbol{F_{ZF}} \cdot \boldsymbol{w}. \tag{19}$$

Ainsi, une simple détection permettra d'estimer $\hat{\boldsymbol{s}}$ en utilisant une quantification dans la constellation QAM grâce à la fonction $\boldsymbol{Q}_{QAM}$:

$$\hat{\boldsymbol{s}} = \boldsymbol{Q}_{QAM}\left\{\boldsymbol{F_{ZF}} \cdot \boldsymbol{y}\right\} \tag{20}$$

Le décodage ZF peut être vu comme une projection orthogonale du vecteur reçu sur la base constitué des vecteurs lignes de la matrice $\boldsymbol{H}$. Figure (2.1) montre un exemple de projection en dimension 2. Si la base n'est pas orthogonal, la projection de $\boldsymbol{y}$ génère une erreur de décodage. Si la base est orthogonale, la projection n'induit pas une erreur de décodage et la solution obtenue est bien la solution ML.

En pratique, la matrice du canal n'est pas orthogonale. Plusieurs travaux dans la littérature permettent d'obtenir des bases équivalentes composées des vecteurs les plus courtes et les plus orthogonales possibles, ces techniques sont appelées les techniques de réduction (43). Ainsi, apliquer ces tes techniques comme le prétraitement suivi un décodage ZF permettent d'obtenir un



(a) random H (b) Orthogonal H

FIGURE 9 – Orthogonal Projection of the received vector

décodage pré-ML en terme de performance.
A la sorite du filter ZF, le bruit résultant est $\tilde{\boldsymbol{w}} = \boldsymbol{F_{ZF}} \cdot \boldsymbol{w}$. La matrice de covariance est défini par

$$\boldsymbol{R}_{\tilde{\boldsymbol{w}}\tilde{\boldsymbol{w}}} = E\left[(\tilde{\boldsymbol{w}}) \cdot (\tilde{\boldsymbol{w}})^H\right] = \sigma^2 \left(\boldsymbol{H}^H \cdot \boldsymbol{H}\right)^{-1} = \sigma^2 \boldsymbol{G}^{-1}. \tag{21}$$

Par conséquence, le bruit n'est pas blanc, $\boldsymbol{R}_{\tilde{\boldsymbol{w}}\tilde{\boldsymbol{w}}} \neq \boldsymbol{R}_{\boldsymbol{ww}} = \sigma^2 \boldsymbol{I}$. En plus, si on applique la decomposition SVD de la matrice de Gram $\boldsymbol{G}$, on obtient $\boldsymbol{G} = \boldsymbol{U} \cdot \boldsymbol{D} \cdot \boldsymbol{V}^H$, où $\boldsymbol{U}$ et $\boldsymbol{V}$ sont des matrices unitaires et $\boldsymbol{D}$ est une matrice diagonale contenant les valeurs singulières de $\boldsymbol{G}$. En utilisant la proprieté que les valeurs singulières de la matrice de Gram sont égales au carré des valeurs propres de $\boldsymbol{H}$, noté λ_1,λ_2,...,λ_n, la matrice de covariance $\tilde{\boldsymbol{w}}$ est donné par

$$\boldsymbol{R}_{\tilde{\boldsymbol{w}}\tilde{\boldsymbol{w}}} = \sigma^2 \boldsymbol{V} \cdot \begin{bmatrix} \sqrt{\lambda_1}^{-1} & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & \sqrt{\lambda_n}^{-1} \end{bmatrix} \cdot \boldsymbol{U}^H \tag{22}$$

Ainsi, le problème connu de forçage à zero est l'amplification du bruit provoqué par l'inversion des valeurs propres de $\boldsymbol{H}$. Ces valeurs propres sont grandes pour une matrice mal-conditionnée. Le décodeur à retour de decision : ZF-DFE L'idée générale du décodeur ZF-DFE est de traiter le vecteur reçu $\boldsymbol{y}$ pour estimer le vecteur transmis $\boldsymbol{s}$ en estimant chaque composante s_k, une par une, en annulant les effets de ces symboles déj à décodés, et annulant ceux déj à inconnus. En pratique, si un symbole $\hat{s}_k$ est estimé, le décodeur exploite cette décision pour estimer $\hat{s}_{k-1}$, $\hat{s}_{k-2}$,..., $\hat{s}_1$. Ainsi, ce décodeur non linéaire est appelé un décodeur à retour de décision (DFE : Decision Feedback Equalization). Le décodeur The ZF-DFE utilise le critère ZF pour décoder le symbole $\hat{s}_k$. Le DFE inclut un filtre feedforward qui opère sur le signal reçu pour supprimer l'interférence inter-symboles ISI, un filtre de feedback qui opère sur les symboles déj à détectés pour supprimer l'ISI. Le DFE est généralement plus performant que le l'égaliseur linéaire traditionnel. Et puisqu'il s'agit d'une détection successive, la décomposition QR est très utile.

$$\begin{aligned} \boldsymbol{y} &= \boldsymbol{H} \cdot \boldsymbol{s} + \boldsymbol{w} \\ &= \boldsymbol{QR} \cdot \boldsymbol{s} + \boldsymbol{w} \end{aligned} \tag{23}$$

Dans l'objectif d'exploiter la structure triangulaire supérieure de la matrice $\boldsymbol{R}$, on multiplie les deux parties de l'équation (2.11) du coté gauche par le transposé de $\boldsymbol{Q}$.

$$\begin{aligned} \boldsymbol{y}_1 &= \boldsymbol{Q}^T \boldsymbol{y} \\ &= \boldsymbol{R} \cdot \boldsymbol{s} + \boldsymbol{Q}^T \boldsymbol{w} \end{aligned} \tag{24}$$

Comme $\boldsymbol{R}$ est triangulaire supérieure, pour la première itération, le décodeur estime le symbole s_n en utilisant cette équation

$$\hat{s}_n = \boldsymbol{Q}_{QAM}\left\{\frac{y_{1,n}}{r_{nn}}\right\} \tag{25}$$

Pour decoder le symbole d'information s_k, le décodeur utilise les symboles $\hat{s}_j$, $j = k+1,\ldots,n$ précédemment estimés, en utilisant cette équation

$$\hat{s}_k = \boldsymbol{Q}_{QAM}\left\{\frac{1}{r_{kk}}\left(y_{1,n} - \sum_{j=k+1}^{n} r_{kj}\cdot\hat{s}_j\right)\right\}, 1 \leq k \leq n \tag{26}$$

Malheureusement, ZF-DFE performance est entravée par la propagation des erreurs. La degradation dans les performances du décodeur DFE survient quand une détection erronée est injectée dans le filtre feedback. Ainsi, au lieu de supprimer l'ISI, le DFE peut amplifier l'ISI. La propagation d'erreur peut induire des erreurs de décision et augmenter ainsi la probabilité d'erreur binaire et symbole. Le décodeur MMSE Le récepteur ZF élimine l'interférence mais amplifie le bruit. Ceci, peut être pas trop signifiant pour des hauts SNR, mais pour des SNR faibles, il sera pratique de designer un filtre qui maximise le rapport global signal sur bruit et interférence (SINR). Une possibilité sera de minimiser le bruit total résultant, i.e. trouver le filtre optimal $\boldsymbol{F_{MMSE}}$ qui minimise l'erreur quadratique moyenne :

$$\begin{aligned}\boldsymbol{F_{MMSE}} &= \arg\min_{F}\left(E\left\{\|\hat{\boldsymbol{s}} - \boldsymbol{s}\|\right\}\right)\\ &= \arg\min_{F}\left(E\left\{\|\boldsymbol{F}\cdot\boldsymbol{y} - \boldsymbol{s}\|\right\}\right)\end{aligned} \tag{27}$$

Ainsi, le filtre MMSE peut être écrit comme suit :

$$\boldsymbol{F_{MMSE}} = \boldsymbol{H}^H\cdot\left(\boldsymbol{H}^H\boldsymbol{H} + \frac{\sigma^2}{\sigma_s^2}\boldsymbol{I}\right)^{-1}, \tag{28}$$

où σ_s^2 représente la puissance moyenne des composantes du vecteur $\boldsymbol{s}$, i.e $E\left[\boldsymbol{s}\boldsymbol{s}^H\right] = \sigma_s^2\boldsymbol{I}_p$.
Le critère MMSE a des performances meilleur que le Forçage à Zero pour des SNR faibles, mais avec un désavantage : le récepteur doit connaitre la variance du bruit. Aussi, pour des SNR élevés, le MMSE et le ZF sont équivalents.
Le récepteur MMSE offre un bon compromis entre la suppression d'interférence et la réduction du bruit. Pour un SNR élevé, le récepteur MMSE devient un récepteur ZF. Pour les SNR faibles, le récepteur MMSE devient similaire à un filtre adapté :

$$\boldsymbol{F}_{MMSE} = \begin{cases}\boldsymbol{F}_{ZF} & \text{if SNR is high}\\ \frac{\sigma^2}{\sigma_s^2}\boldsymbol{H}^H & \text{if SNR is low}\end{cases}$$

Comparaison des décodeurs sous-optimaux Dans la figure (3.13), on compare les performances et les complexités des différents décodeurs sous-optimaux présentés ultérieurement. Ainsi, on considère un système MIMO 2×2 avec un multiplexage spatial. On utilise simplement une constellation 4-QAM. Le canal est Rayleigh quasi-statique. L'efficacité spectrale est de 4b/s/Hz. Les performances sont calculés en terme de BER en fonction du SNR. Le SNR est calculé avec cette équation :

$$SNR = 10log_{10}\left(\frac{n\sum_{i=1}^{p}E_{s_i}}{2\sum_{i=1}^{p}log_2\left(q\right)N_0}\right) \qquad \text{dB}$$

où E_{s_i} est l'énergie moyenne par dimension de l'information complexe appartenant à un constellation $q - QAM$ et $\sigma^2 = 2N_0$.
Dans la figure (3.13), on montre aussi la complexité des décodeurs sous-optimaux en terme de nombre de multiplications par mot de code. Pour tous ces décodeurs, les opérations sont des opérations matricielles appliquées au signal reçu et complètement indépendantes de la variance du bruit. Ce qui peut explique la complexité constante avec le SNR.
Même si tous les décodeurs sous-optimaux offrent une complexité faible et constante âĂŞ qui est très utile dans les implémentations pratiques - ils ne permettent pas des bonnes performances et ne profitent pas de la diversité offerte par les systèmes MIMO.
D'un autre coté, l'utilisation des décodeurs sous-optimaux peut être très intéressante si le nombre des antennes de réception est grand comparé au nombre d'antennes de transmission car on profite de la diversité de réception élevé. Mais, afin de récupérer la diversité totale offerte par les systèmes MIMO et les codes espace-temps, nous devrions nous concentrer sur les décodeurs optimales.

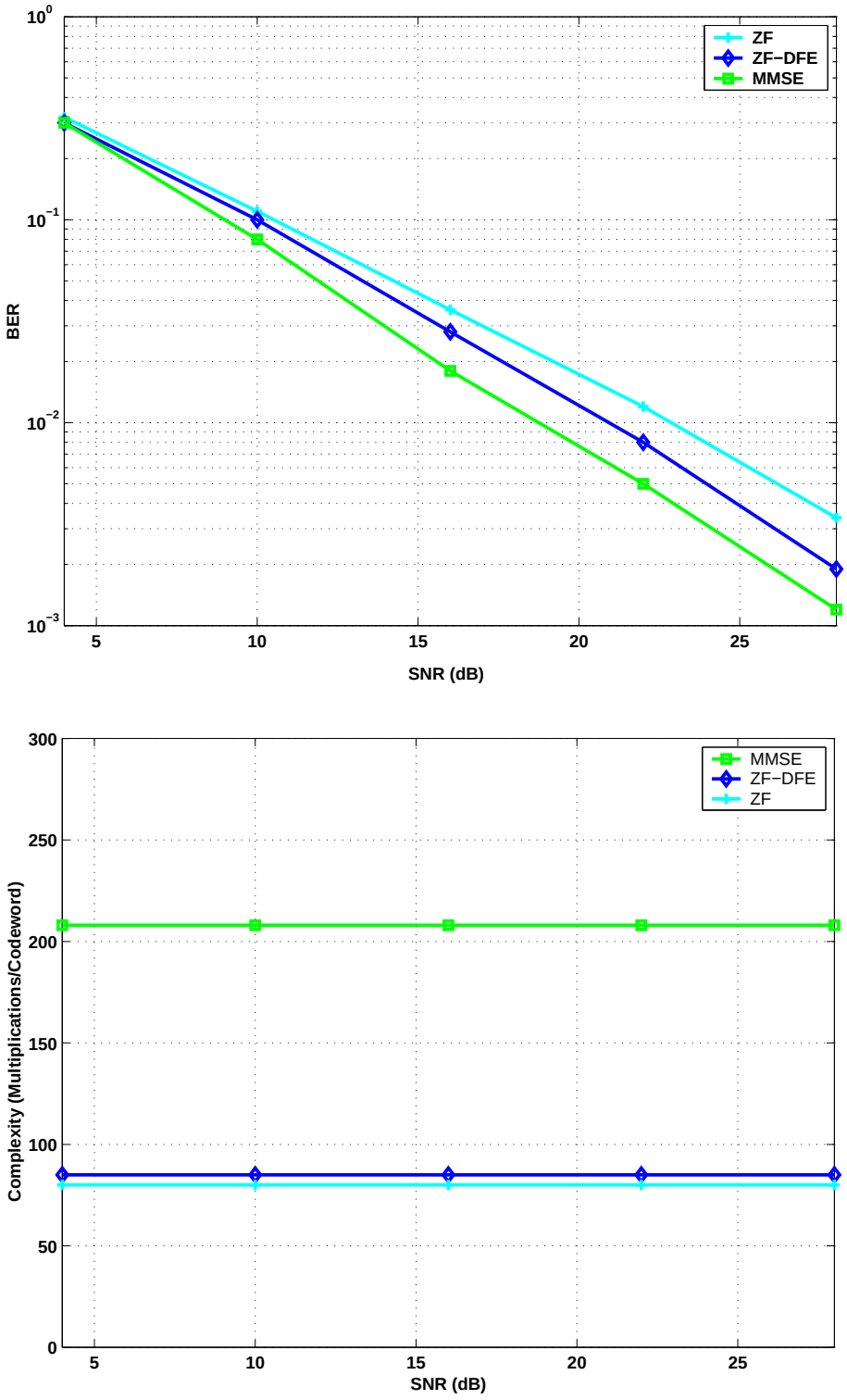


Figure 10 – Linear and Non-Linear Sub-Optimal Decoders Performance and Complexity Comparison for 2×2 MIMO System using Spatial Multiplexing with a 4-QAM Constellation

0.2.4.2 Les decodeurs MIMO pour lattice

Le maximum likelihood (ML) conduit à la meilleure performance en terme de taux d'erreur, mais il est extrêmement exigeantes en termes de complexité. Pour les constellations, de taille q, le décodage ML consiste à chercher parmi q^p candidats possibles. Ce ci est abordable quand q et p sont petits, mais pas pour les systèmes à grande efficacité spectrale. La complexité croissante est causée par la recherche dans toutes les combinaisons possibles , bien que beaucoup d'entre eux sont probablement pas le bon candidat : en raison de la distribution gaussienne du bruit, des mots de code qui sont loin du vecteur reçu sont beaucoup moins probable que des mots de code proche du vecteur reçu. Le décodage par Lattice permet une réduction significative de la complexité comparé au ML exhaustif, d'abord 1) ca évite le besoin d'un contrôle compliqué des bornes (44) et 2) permet une utilisation plus efficace des algorithmes des prétraitements (ex., l'algorithme LLL (43)) qui sont connus pour avoir offrir une réduction significative de complexité. La recherche du point le plus proche d'un point donné a été très largement étudié dans la théorie de lattice. En général, l'algorithme de recherche optimal doit exploiter dans la structure de lattice. Pour les lattices en général, qui n'ont pas une structure particulière, le problème est NP-difficile.

Une approche commune au problème du point le plus proche est d'identifier une région dans lequel le point optimal de lattice doit exister, et après étudier tous les points de lattice dans cette région, et et éventuellement de réduire sa taille dynamiquement. En général, le développement des algorithmes du point le plus proche suivent deux branches inspirées par deux articles fondateurs : Phost (45) en 1981 a examiné les points de lattice qui appartiennent à un hyper sphère, Kannan (49) en 1983 a utilisé un parallélépipède rectangulaire. Les deux papiers apparaissent plus tard dans des versions étendus : Pohst (48) et Kannan (en suivant les travaux de Helfrich (47)) et (46). En (45), cependant, Pohst a proposé une stratégie efficace pour énumérer tous les points du réseau intérieur d'une sphère avec un certain rayon. Bien au pire des cas, la complexité est exponentielle en q, cette stratégie a été largement utilisé dans plusieurs problĂŹematiques de recherche de points en raison de son efficacité dans de nombreux scénarios (voir (52) pour un examen exhaustif des ouvrages connexes).

La stratégie d'énumération de Pohst a été initialement introduite en communication numérique par Viterbo et Biglieri (50). Dans (51), Viterbo et Boutros l'ont appliqué pour la détection ML pour les constellations multi-dimensionnelles transmis sur un canal évanouissant à une antenne et donnent le diagramme d'une éventuelle implémentation. Agrell et al. (52) ont propose l'utilisation d'un raffinement Schnorr-Euchner (53) de l'énumération de Pohst dans la recherche du point le plus proche. Algorithme du décodeur par Sphère L'algorithme de décodeur par sphère a été initialement développé dans les années 1980, mais a récemment attiré beaucoup d'attention dans la communauté MIMO grâce à sa performance similaire au décodeur ML exhaustive à une complexité raisonnable. L'idée principale est de limiter la recherche parmi les candidats possibles localisés dans un sphère de rayon $\sqrt{C}$ centré sur le vecteur reçu (voir figure (2.3)). Dans cette partie on suppose un système MIMO symétrique, $M = N$. En appliquant un mapping du système complexe vers un système réelle de l'équation (2.3) comme décrit dans les équations (1.13) et (1.14), on obtient

$$\underline{\boldsymbol{y}} = \underline{\boldsymbol{H}} \cdot \underline{\boldsymbol{s}} + \underline{\boldsymbol{w}}. \tag{29}$$

Après, on considère la décomposition QR de la matrice $\underline{\boldsymbol{H}} = \boldsymbol{QR}$. Après la multiplication de deux cotés de l'équation (2.17) par $\boldsymbol{Q}^T$, le système devient

$$\begin{aligned} \boldsymbol{y_1} &= \boldsymbol{Q}^T \cdot \underline{\boldsymbol{y}} \\ &= \boldsymbol{R} \cdot \underline{\boldsymbol{s}} + \boldsymbol{w_1}. \end{aligned} \tag{30}$$

$\boldsymbol{Q}$ est orthogonale et la multiplication par $\boldsymbol{Q}^T$ ne change pas le système précédent. Le système est de dimension $2n$ puisque $M = N$ et puisqu'on est passé à la représentation dans le domaine réel. Maintenant, trouver le point le plus proche dans le sphère est équivalent à résoudre l'inéquation suivante :

$$\min_{\underline{\boldsymbol{s}} \in C_{\underline{\boldsymbol{s}}}} \|\boldsymbol{y_1} - \boldsymbol{R} \cdot \underline{\boldsymbol{s}}\|^2 \leq C \tag{31}$$

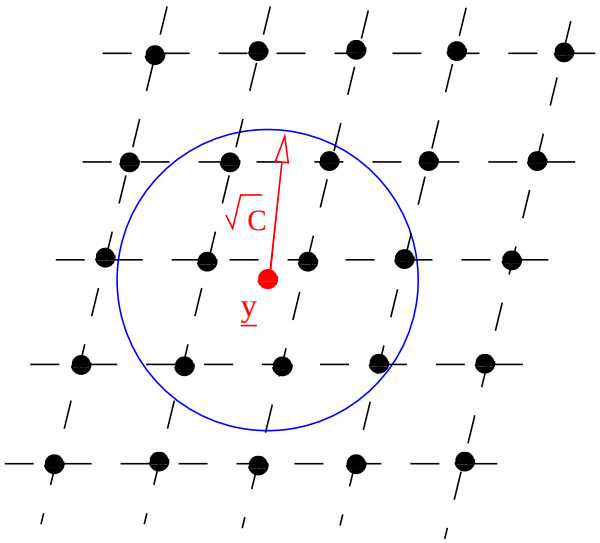


Figure 11 – Sphere Decoding

L'algorithme de décodage Schnorr-Euchner L'algorithme de Schnorr-Euchner nous étudions ici a été présenté dans (52). Il a été utilisé dans des applications de cryptographie. Cet algorithme a le même principe que le SD : la recherche du point le plus proche. Cet algorithme est basé sur deux étapes. La première étape consiste à rechercher le point de Babai (BP), ce qui représente une première estimation, mais n'est pas nécessairement le point le plus proche. Trouver le BP nous donne une borne sur l'erreur. Dans la deuxième étape, nous modifions le BP jusqu' à ce que le point le plus proche est atteint. Nous zigzagons autour de chaque composant BP en vue de construire le point le plus proche(contrairement au sphère décodeur, il n'ya pas de minimum et maximum pour chaque composante du BP). Le temps nécessaire pour trouver le point le plus proche est étroitement liée à BP, ce qui signifie étroitement liée au SNR. En fait, si le BP est très loin du point le plus proche, c'est à dire pour les rapports signal sur bruit faible, l'algorithme prend beaucoup plus de temps à converger. Toutefois, si le BP est proche de point le plus proche, c'est à dire pour les rapports signal sur bruit élevés, l'algorithme converge rapidement.

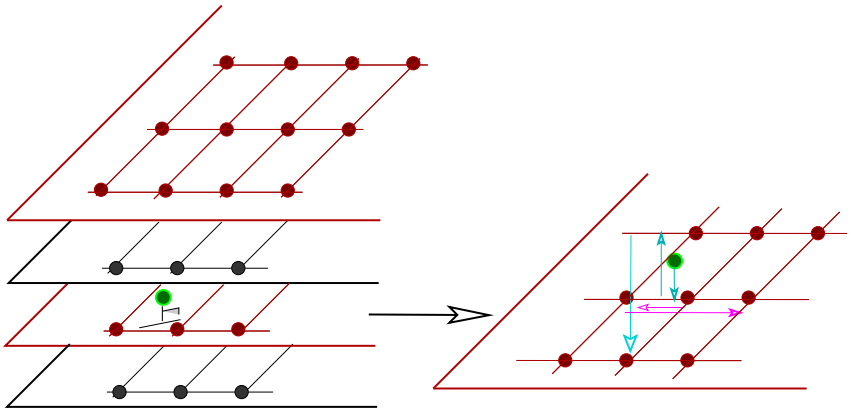


FIGURE 12 – SE Search Strategy

L'idée clef est de voir le lattice comme une superposition des hyperplans et puis commencer la recherche pour le point le plus proche dans l'hyperplan (voir figure 2.4).
Rappelons l'équation (2.18). La forme triangulaire supérieure de $\boldsymbol{R}$ permet de voir le lattice comme plusieurs couches. Ainsi, la matrice $\boldsymbol{R}$ peut être écrit comme suit

$$\boldsymbol{R} = \begin{bmatrix} \boldsymbol{R_1} \\ \boldsymbol{r_{2n}} \end{bmatrix}, \tag{2.30}$$

Où $\boldsymbol{R_1}$ est une matrice $(2n-1) \times 2n$ Composée du top $2n-1$ lignes de la matrice $\boldsymbol{R}$. La matrice $\boldsymbol{R}$ est triangulaire, le vecteur $\boldsymbol{r_{2n}} = (0,,0,r_{2n,2n})$ est orthogonal à l'espace généré par la matrice $\boldsymbol{R_1}$. Maintenant, l'algorithme de recherche dans le lattice de dimension $2n$ va être détaillé récursivement comme un nombre fini de $2n-1$ opérations dimensionnelles. Le lattice $\Lambda_{\boldsymbol{R}}$ peut etre vu comme une superposition infini d'hyperplans de dimension $2n-1$ générés par la matrice $\boldsymbol{R_1}$:

$$\Lambda_{\boldsymbol{R}} = \cup \{\boldsymbol{c} + t_{2n}\boldsymbol{r_{2n}} / \boldsymbol{c} \in \Lambda_{\boldsymbol{R_1}}, t_{2n} \in \mathbb{Z}\}. \tag{2.31}$$

Une projection successive sur les différents hyperplans du lattice permet de trouver une première estimation du point le plus proche. C'est le 'Babai point' et il correspond au point ZF-DFE (55). Une fois ce point est trouvé, il constitue le point de départ pour visiter les autres points. L'objectif est de trouver le point le plus proche, il n'est donc pas nécessaire de considérer les points ayant une distance supérieure au 'Babai point'. Ainsi, le Schnorr-Euchner (SE) est un algorithme dans une sphère centre sur le point reçu avec comme rayon initial la distance entre le point reçu et le BP.

0.3 Décodeur par Stack à bornes sphériques dur et souple

0.3.1 Introduction

Le décodage ML consiste à chercher le point le plus proche au point reçu appartenant au lattice. La recherche exhaustive consiste à visiter tous les points de lattice qui est impossible à réaliser en pratique. Ainsi, pour décoder le vecteur reçu, il est nécessaire de définir une région de recherche finie.
Le décodage par stack était originalement designé pour décoder les codes binaires à treillis, ou le mot de code est choisi dans un alphabet fini. Néanmoins, considérant le lattice, le mot de code est pris dans le corps fini $\mathbb{Z}^{2n}$ ce qui aboutira à une structure en arabre. Appliquer le decodage par stack semble impossible dans ce cas. Notre objectif est de proposer une version modifié de l'algorithme stack dans le but de decoder le lattice et de réduire la complexité. Donc, on propose ici un nouvel algorithme qui combine la région de recherche du sphère décodeur et la stratégie de recherche du décodage par stack.

0.3.2 Décodeur par Stack à bornes sphériques

0.3.2.1 Décodage SB-Stack pour les lattices

A - 1^{ere} approche

Appliquant le décodage par stack, on cherche le point le plus proche dans une région finie $\Lambda \subset \mathbb{Z}^{2n}$. Malheureusement, la troncature de l'arbre aura une incidence sur les performances du décodeur. En fait, si le mot de code transmis appartient à Λ, le décodeur saura systématiquement le décoder, mais une erreur se produira si le mot de code est en dehors de la zone de recherche. Ainsi, le principal défi est de savoir comment choisir la zone de recherche optimale Λ. Déj à, la forme triangulaire de la base de lattice rappelle la strategie d'enumeration Schnorr Euchner (SE) (52). Notre algorithme de recherche est similaire au SE et base sur l'utilisation du BP, néanmoins la stratégie de recherche et la construction de l'arbre sont totalement différents. En effet, le SE consiste à énumérer tous les noeuds possibles dans une région borné en zigzagant autour du BP en utilisant la stratégie de recherche en profondeur (Depth First Search : DeFS). Dans cette première approche, on s'inspire de l'algorithme SE et on propose un arbre centré sur le BP $\boldsymbol{u}$.
A chaque niveau, il énumère les points voisins dans le lattice $\boldsymbol{u} \pm \boldsymbol{t} = (u_1 \pm t_1, u_2 \pm t_2, ..., u_{2n} \pm t_{2n})$ où $\boldsymbol{t}$ est un vecteur dans $\mathbb{Z}^{2n}$, la stratégie BeFS est encore appliquée sur cet arbre. En appliquant cet algorithme, on peut délimiter la taille de l'arbre en choisissant le nombre des noeuds voisins de lattice du BP à prendre. Néanmoins, le point ML n'est pas garanti d'être dans l'arbre. Pour l'atteindre, on doit élargir la région de recherche ce qui implique d'avoir un arbre dense ce qui implique plus de complexité. Dans la figure (3.1), on montre les performances du décodage par stack contraint à quelques régions de recherche. Ainsi, on trace le taux d'erreur symbole comme fonction du SNR, pour un système MIMO 4×4 en utilisant le multiplexage spatial et un canal quasi statique de Rayleigh. Au début, on procède en considérant une région de recherche défini

par $\Lambda_a = \{u_i - 1, u_i, u_i + 1, i = 1, \ldots, 2n\}$. Ceci veut dire que les points de lattice concernés directement par l'algorithme de recherche sont les points voisins directs du BP. Néanmoins, dans des mauvaises conditions canal, le point ML peut être loin et inaccessible. Ceci est visible dans le graphe (a), ou les performances sont sous-optimales et provoque une perte de 2dB du ML. Pour le même système, on a progressivement élargi la région de recherche et on a observé le comportement de l'algorithme. Les graphes (a)-(d) montrent les performances obtenues en considérant les régions de recherche $\Lambda_a = \{u_i - 1, u_i, u_i + 1\}$, $\Lambda_b = \Lambda_a \bigcup \{u_i - 2, u_i + 2\}$, $\Lambda_c = \Lambda_b \bigcup \{u_i - 3, u_i + 3\}$ et $\Lambda_d = \Lambda_c \bigcup \{u_i - 4, u_i + 4\}$. Comme montré dans la figure (3.1), le décodeur fournit des performances sous-optimales, mais on s'approche des performances ML quand on elargit la région de recherche. Mais la complexité augmente avec les performances. Ainsi, un compromis peut être établi et cet algorithme de décodage peut être d'un grand intérêt. Ainsi, au début de l'algorithme, le compromis performance-complexité est fixé ce qui définit la région approprié. Dans les simulations de la figure (3.1), on a considéré un vecteur uniforme $\boldsymbol{t}$. On peut également utilisé un vecteur $\boldsymbol{t}$ avec des valeurs larges de t_i pour les premiers composants et des petites valeurs de t_i pour les derniers. Le choix peut être judicieux à cause du problème de propagation d'erreur dans l'arbre de recherche.

B - 2^{nd}approach (Décodeur SB-Stack)

Dans la première approche, l'arbre de recherche est centré autour du BP. Toutefois, cette dernière est généralement une estimation approximative de la transmise mot de code, puis le centrage de la région de recherche sur ce point n'est pas optimale puisque la solution ML peut ne pas être à l'intérieur, comme le montre la figure (3.2). Par conséquent, nous proposons ici une seconde approche pour le décodage du lattice inspiré de l'algorithme de décodeur par sphère. Le principe du décodeur sphère consiste à énumérer tous les points du lattice trouvés dans une sphère d'un rayon $\sqrt{C}$ centrée sur le point reçu. Chaque fois un point est trouvé , le rayon est mis à jour, ce qui limite la nombre des points énumérés, mais garantit également le critère du point le plus proche. Le décodeur par sphère utilise la stratégie de DeFS. Nous appelons cette deuxième approche : le decodeur par stack à bornes sphĂŹeriques. (SB-Stack). L'algorithme SB-Stack explore seulement les points du réseau l'intérieur de la sphère de rayon $\sqrt{C}$ en utilisant la stratégie BeFS, qui conduit à la définition d'une limite supérieure et une inférieure pour chaque point du lattice.

Le but de la SB-Stack est de trouver le noeud feuille ayant le moindre coût et se situant dans la zone de recherche sphérique. A partir du noeud racine, l'algorithme calcule les limites supérieure et inférieure de la première composante $\underline{s}_{2n}$, notés respectivement $b_{inf,2n}$ et $b_{sup,2n}$ et génère tous les noeuds à l'intérieur de ces limites.

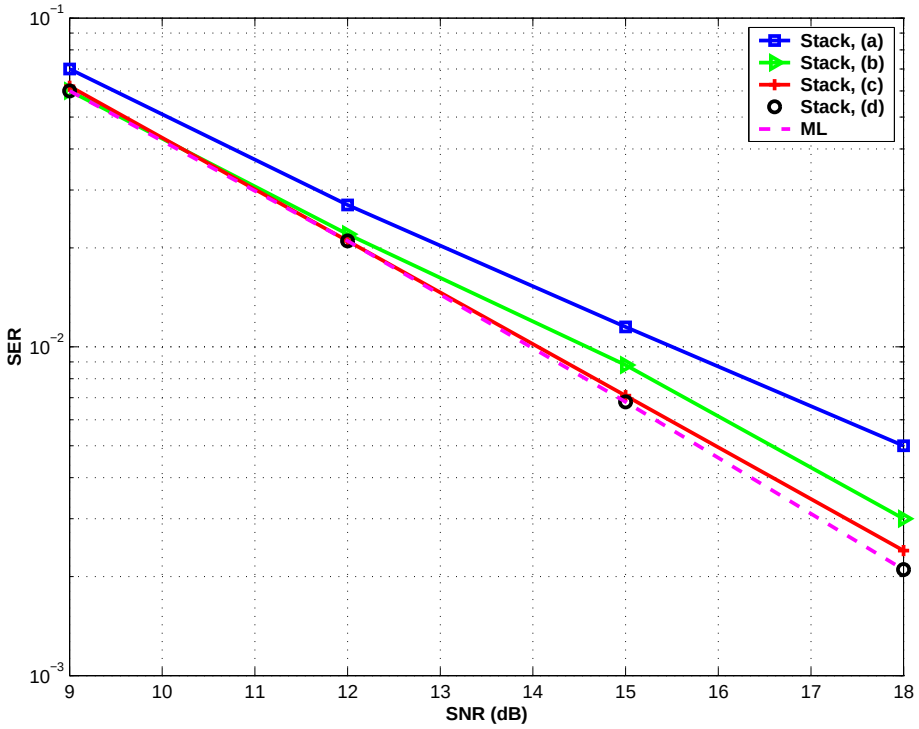


Figure 13 – Performance of a MIMO System using a SM with $M = N = 4$, obtained for different search regions

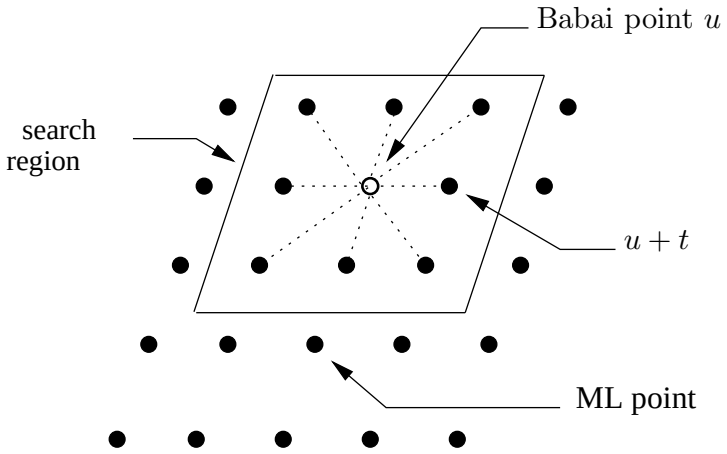


Figure 14 – Example of a Lattice defined in $\mathbb{Z}^2$; the Search Region does not contain the ML point

Noeuds générés sont stockés avec leurs coûts respectifs dans la pile (le stack). Après cela, l'algorithme réordonne les noeuds dans la mémoire dans un ordre croissant en fonction de leurs coûts, sélectionne le haut noeud, puis calcule les limites du niveau suivant. Puis, il génère tous les enfants possibles du noeud supérieur et les stocke dans la mémoire.Après cela, le noeud supérieur est retiré de la pile. Cette procĂŹedure est répétée jusqu' à ce qu'un noeud de feuille atteint le

haut de la mémoire. Notez que, bien que la similitude apparente entre le traditionnel décodeur par sphère et l'algorithme SB-Stack, ces deux algorithmes de recherche soulèvent de grandes différences. De cette façon, contrairement au décodeur par sphère, le rayon dans le décodeur SB-stack reste inchangée au cours de tout le processus de décodage, alors que pour le décodeur par sphère, le rayon est mis à jour chaque fois qu'un point est trouvée. Cependant, la stratégie de recherche pour l'algorithme de stack est tout à fait différente. En fait, à chaque étape, l'algorithme peut revenir en arrière à un niveau supérieur ayant un moindre coût avant d'atteindre un noeud feuille qui correspond à une solution candidate. Ainsi, il n'est pas possible de surveiller la distance ML lorsque l'algorithme est en cours. Par conséquent, le rayon doit être fixé.

0.4 Réduction de la complexité de l'algorithme de décodage par stack

0.4.1 Introduction

Cette partie vise à améliorer le décodage par stack en termes de complexité. Comme montré avant, l'algorithme de décodage par stack est un bon candidat pour résoudre le compromis complexité-performance tout en donnant une excellente structure qui fournit des sorties souples. Notre but est alors de proposer une version modifiée de l'algorithme afin de réduire la consommation de la complexité et le temps.

0.4.2 Décodage par Stack parallèle

Le décodage par stack montre des bonnes caractéristiques pour être mis en oeuvre et améliore la complexité par rapport aux décodeurs ML SE et SD. Mais il souffre encore de grande complexité. Le traitement parallèle peut réduire le temps en exécutant des instructions simultanément. Nous proposons ici une nouvelle version du décodeur par stack basée sur un traitement parallèle. SD a été traitée en parallèle in (91) mais avec une perte de performance.
Le parallélisme est une forme de calcul dans lequel de nombreuses instructions sont effectuées simultanément. Toutefois, les programmes parallèles sont plus difficiles à écrire que ceux séquentiels.
Pour résoudre un problème, un algorithme est réalisé, qui produit un flux série d'instructions. ces instructions sont exécutées sur une unité centrale de traitement. Une seule instruction peut etre exécuté à un moment donné âĂŞ après que l'instruction est terminée, la suivante est exécutée. Le traitement en parallèle utilise plusieurs ressources de calcul simultanément pour résoudre un problème. Le problème est divisé en parties qui sont indépendantes de sorte que chaque ressource de calcul peut exécuter sa partie de l'algorithme simultanément avec les autres. La structure parallèle permet le décodage des parties réelle et imaginaire de chaque symbole de manière indépendante et en même temps. La représentation en lattice donné avant par la fonction Ψ impose une restriction majeure de l'algorithme de recherche arborescente. Plus précisément, la recherche doit être exécutée en série d'un niveau à un autre sur l'arbre. Traitement de chaque niveau pour estimer les symboles doit avoir l'estimation de symboles précédents qui sont nécessaires pour cal-

culer les coûts pour chaque enfant. La pile standard de décodage en utilisant l'arbre de recherche commence au plus haut niveau et de traverser l'arbre avec un niveau à la fois, et calcule pour chaque étape des coûts du noeud enfant.

0.4.3 Représentation en nouveau lattice

Selon la représentation en lattice Ψ, il est impossible, par exemple de calculer le coût pour un noeud dans le niveau k sans assigner une estimation pour les niveaux d'avant. Cette approche signifie que le décodage de toute $\underline{s}_k$ nécessite une valeur estimée pour tous les $\underline{s}_k$ précédents, où $j = k-1,...,2n$.
L'idée derrière ce travail est de détendre la structure d'arbre de recherche qui le rend plus souple pour le parallélisme. Ainsi, on peut décoder chaque paire de niveaux adjacents dans l'arbre, et chaque niveau de cette paire est indépendant de l'autre. Pour cela, on devrait commencer par donner une seconde forme de la représentation matrice de canal. Au lieu de la fonction Ψ défini par :

$$\begin{aligned} \underline{\mathbf{H}} &= \Psi(\mathbf{H}) \\ &= \begin{bmatrix} \mathfrak{Re}(\mathbf{H}) & -\mathfrak{Im}(\mathbf{H}) \\ \mathfrak{Im}(\mathbf{H}) & \mathfrak{Re}(\mathbf{H}) \end{bmatrix} \end{aligned} \tag{32}$$

On utilise une autre fonction Ω et on donne une autre représentation en lattice défini dans (91) par :

$$\begin{aligned} \underline{\tilde{\mathbf{H}}} &= \Omega(\mathbf{H}) \\ &= \begin{bmatrix} \Psi(H_{1,1}) & \cdots & \Psi(H_{1,n}) \\ \vdots & \ddots & \vdots \\ \Psi(H_{n,1}) & \cdots & \Psi(H_{n,n}) \end{bmatrix}, \end{aligned} \tag{33}$$

On suppose que Ψ peut être appliqué aux composantes complexes comme pour le cas matriciel. En utilisant cette représentation du canal, on change l'ordre de détection des symboles reçus à la forme suivante :

$$\begin{aligned} \underline{\tilde{\mathbf{y}}} &= \Omega(\mathbf{y}) \\ &= \begin{bmatrix} \mathfrak{Re}(\mathbf{y}_1) \\ \mathfrak{Im}(\mathbf{y}_1) \\ \vdots \\ \mathfrak{Re}(\mathbf{y}_n) \\ \mathfrak{Im}(\mathbf{y}_n) \end{bmatrix}, \end{aligned} \tag{34}$$

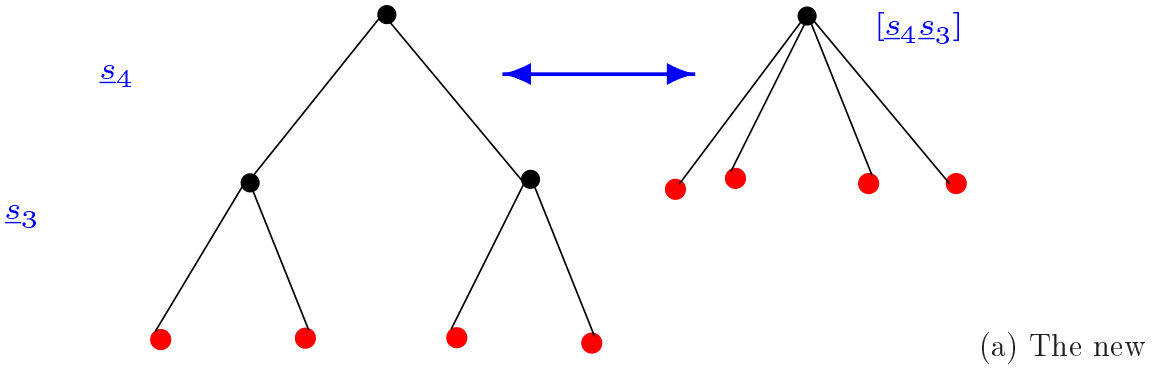
Ce qui veut dire que le premier et le second niveau de l'arbre correspondent aux parties réelle et imaginaire de $\mathbf{s}_n$.

0.4.3.1 Apercu des techniques de decodage parallele

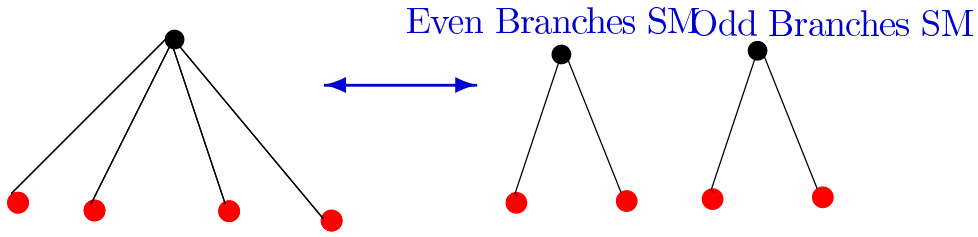
Apres l'application de la décomposition QR, cette structure devient très intéressante. En effet, grâce à l'orthogonalité entre les colonnes de chaque ensemble, tous les elements $\underline{r}_{k,k+1}$ pour $k = 1,3,\cdots,2n$ dans la matrice triangulaire supérieure sont nulles. La localisation de ces zéros est très importante puisque ils introduisent une orthogonalité entre la partie réelle et la partie imaginaire pour chaque symbole détecté. Par exemple, pour la matrice triangulaire supérieure 4×4 $\underline{\tilde{\mathbf{R}}}$, on obtient la forme suivante

$$\underline{\tilde{\mathbf{R}}} = \begin{bmatrix} \underline{r}_{1,1} & 0 & \underline{r}_{1,3} & \underline{r}_{1,4} \\ 0 & \underline{r}_{2,2} & \underline{r}_{2,3} & \underline{r}_{2,4} \\ 0 & 0 & \underline{r}_{3,3} & 0 \\ 0 & 0 & 0 & \underline{r}_{4,4} \end{bmatrix}. \tag{35}$$

En utilisant cet exemple, la figure (4.2) définit le décodage par stack parallèle qui va traiter deux niveaux dans chaque ĂŹetape en dupliquant le traitement et en gardant une seule mémoire pour sauvegarder les fils.



(a) The new lattice permits to calculate the four nodes simultaneously



(b) The four nodes calculation is distributed on two state machines

FIGURE 15 – Parallel processing principle

A travers la figure (4.1), on peut comprendre l'importance du traitement parallèle dans l'amélioration du vitesse de décodage. La nouvelle représentation du lattice permet de calculer deux dimensions dans une seule ĂŹetape grace à l'indépendance entre les deux dernières couches de la matrice de génération du lattice. Ainsi, le temps de l'exécution est deux fois plus faible puisque deux dimensions sont visitées pour chaque noeud.

Introduction

Except for the written word, human communication has always been transmitted over the atmosphere. Voice, sounds, smoke signals, etc. use mechanical or visual cues to transmit messages. With the advent of telephony and telegraphy, long-distance communications became possible while being inexpensive and reliable. These techniques used electrical wire as the communication medium with great success : thousands of homes and cities all over the world deployed hundreds of thousands of kilometers of cable to allow people to communicate. However, it was soon determined that the exclusive use of wires severely limited the use and flexibility of the new communication systems. From an economic perspective, providing each home with a wired telephone cable is very expensive. Replacing the wired infrastructure when it fails is even more so. From a practical point of view, the end point of the wire is, by necessity, fixed ; this means that, in order to profit from it, a person has to be present in a specific place. Also, wired communications are essentially one-to-one ; sharing the same connection between several people is, at best, cumbersome. Over the last century, enormous advances have been made in telecommunications. Ever greater amounts of information have been made available to the public from a variety of sources : radio, television, internet, multimedia mobile phones. The old wire seems to be ill-suited to the new possibilities offered by technology. In fact, more and more information is transmitted wirelessly, returning to the ancient use of the atmosphere as communication medium. The reasons are simple ; in short, wireless communications overcome the basic limitations described above. There is no need to lay out huge amounts of material to homes, businesses and offices. The transmitted signal can cover a large area (even the whole planet), allowing the use of the medium from any location, and making broadcasting trivially easy.
That's why, wireless communications is among the most active fields of technology development of our time and is becoming a key element of modern society. This huge development is especially driven by the transformation of what has been previously considered as a medium for supporting voice telephony into a medium for supporting other services, such as the transmission of video, images, text, and data. Thus, from satellite transmission, radio and television broadcasting to mobile telephony, wireless communications has revolutionized the way societies function.

Similar to gains in wireline capacity over the last two decades, the demand for new wireless capacity is growing at a very rapid speed and there is an increasing demand for higher data rates, better quality of service, and higher network capacity compared with that obtainable from DSL (Digital Subscriber Line) and cable.

Even if there are still many technical problems to be solved in wireline communications, demands for additional wireline capacity can be easily realized with the addition of new infra-

structure, such as additional optical fibers, routers, switches, and so on. On the other hand, the traditional resources that have been used to add capacity to wireless systems are radio bandwidth and transmitter power. Thus, wireless system designers are facing a number of challenges. These include the limited availability of the radio frequency spectrum and a complex space-time varying wireless environment.

Unfortunately, the two needed resources are among the most severely limited in the deployment of modern wireless networks : radio bandwidth because of the very tight situation with regard to useful radio spectrum, and transmitter power because mobile and other portable services require the use of battery power, which is limited. So, the appearance of systems using the spatial dimension to increase spectral efficiency and quality of transmission constituted a real revolution at the end of 1990s.

As a result, Multiple-Input Multiple-Output (MIMO) systems have emerged as a most promising technology in these measures and offering a powerful paradigm for meeting these challenges. MIMO communication systems can be defined intuitively by considering that multiple antennas are used at the transmitting end as well as at the receiving end. The core idea behind MIMO is that signals sampled in the spatial domain at both ends are combined in such a way that they either create effective multiple parallel spatial data pipes (therefore increasing the data rate), and/or add diversity to improve the quality (Bit-Error Rate or BER) of the communication.
Clearly, the benefits from multiple antennas arise from the use of a new dimension-space. Hence, because the spatial dimension comes as a complement to time (the natural dimension of digital communication data), MIMO technology is also known as 'space-time' wireless or 'smart' antennas. In 1995, Telatar shows that the capacity of a MIMO system grows linearly with the minimum of the number of antennas at the transmitter and at the receiver (7). Thus, with the emergence of MIMO systems, multipaths were effectively converted into a benefit for communication systems. MIMO indeed takes advantage of random fading, and possibly delay spread, to multiply transfer rates. The prospect of dramatic improvements in wireless communication performance at no cost of extra spectrum was further illustrated in the now famous paper by Telatar (7).

Simultaneously, Bell Labs developed the so-called BLAST architecture (19) that achieved spectral efficiencies up to 10-20 bits/s/Hz, while the first space-time coding architectures appeared (25).

Space-time coding is a powerful approach to exploit features of MIMO systems. The design of high-performance high-rate space-time codes has been a problem of great interest and several recent approaches have been developed to obtain such codes. These codes permit to further exploit degrees of freedom of the MIMO system by introducing dependency between temporal and spatial domains in order to bring spatial diversity and coding gain.

The MIMO success story had begun. Today, MIMO appears as an ideal technology for large-scale commercial wireless products such as wireless local area and third generation networks.

In the commercial area, Iospan Wireless Inc. developed the first commercial system in 2001 that used MIMO-OFDMA technology. Iospan technology supported both diversity coding and

spatial multiplexing. In 2005, Airgo Networks had developed a pre-11n version based on their patents on MIMO. Following that in 2006, several companies (including at least Broadcom, Intel, and Marvell) have fielded a MIMO-OFDM solution based on a pre-standard for IEEE 802.11n WiFi standard. Also in 2006, several companies (Beceem Communications, Samsung, Runcom Technologies, etc.) have developed MIMO-OFDMA based solutions for IEEE 802.16e WIMAX broadband mobile standard. All upcoming 4G systems will also employ MIMO technology. Several research groups have demonstrated over 1 Gbit/s prototypes.

Facing this very promising technique, the main difficulty is to define optimal systems for an aimed application. Furthermore, the notion of complexity is essential in the conception of an optimal system. Today, only schemes with low number of antennas or using orthogonal space-time codes with low coding gain are implemented in commercialized wireless communication systems thanks to their low complexities.

On the other hand, the complexity of reception of high spectral efficiency systems, using a large number of antennas, considerably increases with spectral efficiency. In 2007 and for 4G communication, NTT DoCoMo completed a trial in which they reached a maximum packet transmission rate of approximately 5 Gbit/s in downlink with 12×12 MIMO technique using a 100 MHz frequency bandwidth while moving at 10 km/h, and is planning on releasing the first commercial network (10). Samsung also proposed a demonstrator with an 8×8 MIMO system capable of transmitting up to 3.5 Gbit/s (11) (12). Throughputs offered by these systems are revolutionary for the wireless communications world. However their complexity of implementation is still too important to consider their integration in the commercialized public systems of communications who must be ergonomic, cheap and reliable.

Thus, this thesis confronts the problem of receiver design and the main goal of this research is to propose powerful MIMO decoding algorithms with low complexity and offering a complexity-performance tradeoff.

Several decoding algorithms offering optimal performances were proposed in the literature. We distinguish in particular decoders having strategies based on searching inside a tree such as the Sphere Decoder, the algorithm of Schnorr-Euchner and sequential decoders such as the Fano decoder and the Stack decoder. Although these latter ones present low complexity compared to exhaustive decoders, their complexity increases in function of the number of antennas and the size of the used constellation. Thus, the purpose behind this work is to bring bring as many enhancements as possible for the MIMO decoder especially in terms of complexity. This thesis consists of four chapters. **Chapter 1** provides an overview of the recent developments in space-time coding and signal processing techniques appropriate for MIMO communication systems. First, we will introduce the MIMO channel and system modeling. Then, we will review the information theory results on the capacities of wireless systems employing multiple transmit and receive antennas. Finally, some space time codes are described and their construction techniques and criteria are also briefly touched upon. **Chapter 2** begins by examining the basic principles and structures of MIMO decoders. Then, it details the different decoder classes and gives an overview of their way of working. After that, some well-known pre-processing techniques are described. At the end of this chapter, we focus on the diversity Multiplexing Tradeoff of MIMO decoders. In **Chapter 3**, we propose at first a new sequential decoding algorithm that we call SB-Stack (Spherical Bound-Stack decoder). This one is based, at the same time, on the Sphere

Decoder and the Sequential Stack Decoder. The Sphere Decoder uses the DeFS (Depth-First-Search) strategy while the stack decoder uses the BeFS (Best-First-Search) strategy. We show that the SB-Stack decoder combines the advantages of both decoders. In fact, it uses the search strategy of stack decoder and the search region defined by the sphere centered on the received point. Consequently, the SB-Stack decoder offers lower complexity than the original decoders without sacrificing optimal performances. In the second half of this chapter, a new soft output MIMO decoder is proposed. We exploit the suitable structure of the stack algorithm to provide a soft output, required when STBC are concatenated with error correcting codes. The proposed algorithm is an extension of the hard SB-Stack decoder. A straightforward idea was to exploit internal nodes still stored in the stack at the end of hard decoding process to calculate the LLR (Log-Likelihood Ratio). We show that the potential gain of such a method is rather large relative to classical soft decoders. In **Chapter 4**, we focus on reducing the complexity of the stack decoding algorithm using different techniques. First, we propose the use of parallel processing for stack decoding, this is in order to decode signals transmitted on linear MIMO channels while reducing time consumption of hardware architecture. Then, we detail some strategies reducing the stack decoding complexity like the child-sibling strategy and the complex strategy (decoding over complex symbol alphabets) . We show also the improvement introduced by these strategies compared to the original stack decoding. At the end of this chapter, we suggest a solution to the problem of the increase in decoder complexity in the case of severe channels. Indeed, these decoders seem to be converted to exhaustive search decoders visiting all constellation points. Thus, we introduce time and complexity constraints and we propose to finish the stack decoding by a ZF-DFE decoding for hard decoding and a K-Best decoding for soft outputs.
In summary, a range of signal processing tools appropriate for use in MIMO communication systems have been developed in the work presented in this thesis and simulation results are provided to demonstrate the effectiveness of the proposed techniques.

Chapitre 1

MIMO Channel Description and Information Theory notions

In this first chapter, we will first present the general functioning of a digital communication system. We will focus particularly on MIMO systems. First, we will present the channel modeling problem for MIMO systems. Thus, different techniques of modeling are presented and some MIMO models found in literature are classified. After that, we will introduce diversity techniques and lattice properties. In the second half of this chapter, we will recall some information theory notions applied to MIMO systems. Different formulations of capacity for MIMO channels are presented for different cases such as the knowledge of Channel State Information at the Transmitter and the Receiver and the knowledge of Channel State Information at the Receiver only. Then, the outage probability and the multiplexing gain are presented. Also, we will detail the Diversity Multiplexing Tradeoff which is an essential notion for the comprehension of MIMO systems' functioning.
By the end of this chapter, we will present some known Space Time Codes.

1.1 MIMO Channel and System Modeling

1.1.1 Transmission Scheme

A block diagram of a wireless communications system is shown in Figure (1.1). In this figure the source of information could be a voice signal, a video signal or a data. The source encoder processes the information and formats the information into a sequence of information bits $\in \{\pm 1\}$. The goal of the source encoder is to remove the unstructured redundancy from the source so that the rate of information bits at the output of the source encoder is as small as possible within a constraint on complexity. The channel encoder adds structured redundancy to the information bits for the purpose of protecting the data from distortion and noise in the channel. The modulator maps the sequence of coded bits into waveforms that are suitable for

transmission over the channel. First, the signal amplitude decreases due to the distance between

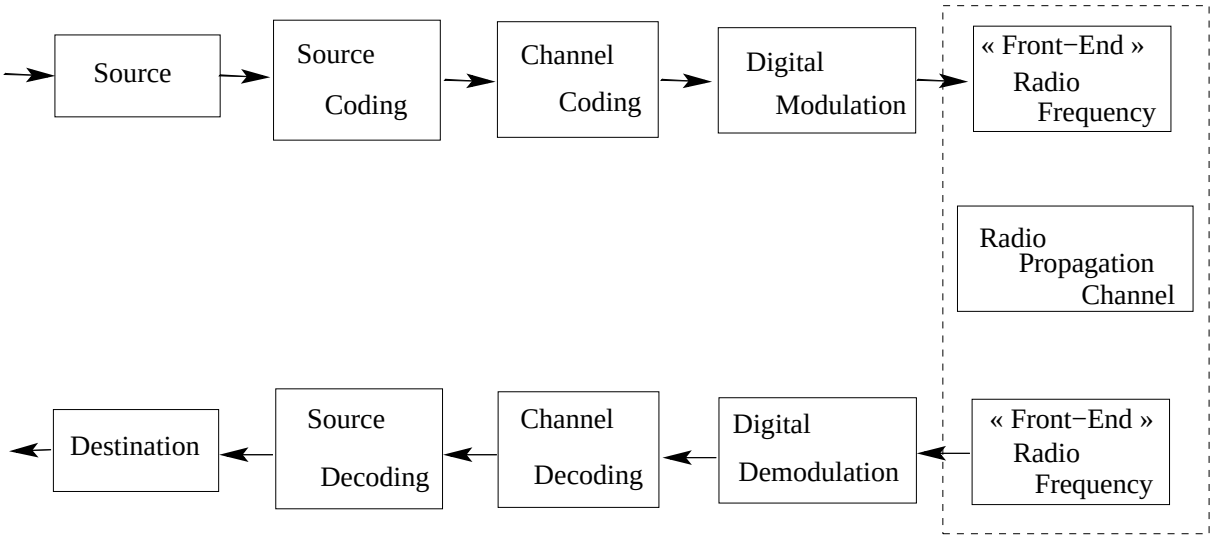


Figure 1.1 – Generic Architecture of a Digital Communication System

the transmitter and receiver. This is generally referred to as propagation loss. Second, due to obstacles the signal amplitude is attenuated. This is called shadowing. Finally, because of multiple propagation paths between the transmitter antenna and the receiver antenna, the signal waveform is distorted. Multipath fading can be either constructive, if the phases of different paths are the same, or destructive, if the phases of the different paths cause cancellation. The destructive or constructive nature of the fading depends on the carrier frequency of the signal and is thus called frequency selective fading. In addition to propagation effects, typically there is noise at the receiver that is uncorrelated with the transmitted signal. Thermal noise due to motion of the electrons in the receiver is one form of this noise. Other users occupying the same frequency band or in adjacent bands with interfering sidelobes is another source of this noise.
The receiver's goal is to reproduce at the output of the source decoder the information-bearing signal, be it a voice signal or a data, as accurately as possible. The structure of the receiver is that of a demodulator, channel decoder, and source decoder. The demodulator maps a received waveform into a sequence of decision variables for the coded data. The channel decoder attempts to determine the information bits using the knowledge of the codebook (set of possible encoded sequences) of the encoder. The source decoder then attempts to reproduce the information. In the following, we will not focus on source coding.

Thus, the modeling of transmission channel is essential to conceive a performing digital communications system and we can conclude that channel model depends on system environment and the propagation model. As it is shown in figure (1.2) the receiver detects different versions of signal. This echo of the same signal is the interaction of wave with the real propagation environment : diffraction, reflection, dispersion (2).

The simplest mathematical model for a communication channel is its representation with a linear filter with an additive noise where transmitted signal is corrupted by an additive random noise process (3). Such linear filters are characterized by time-variant channel impulse response

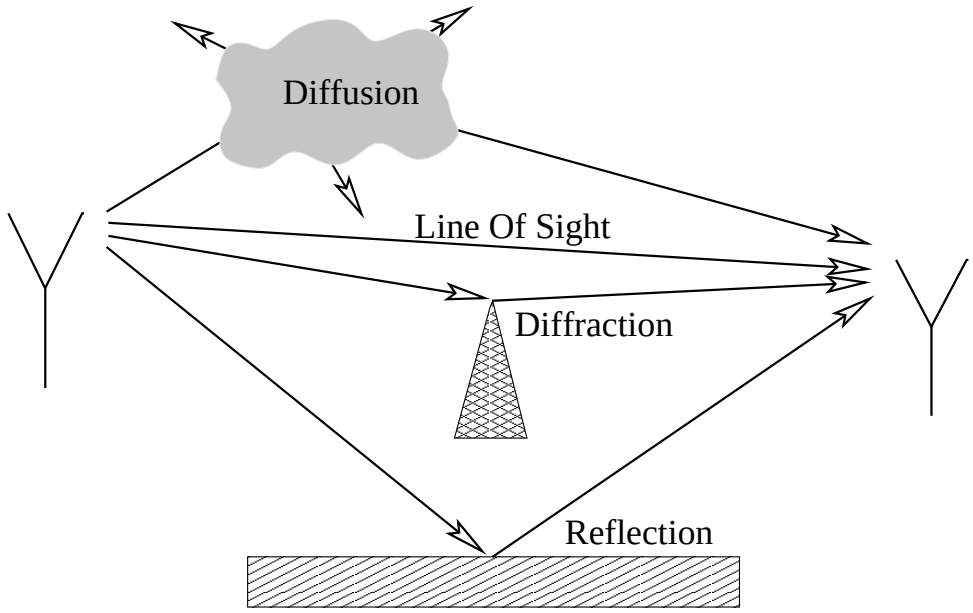


Figure 1.2 – Multipath Propagation of Radio Signals

$h(t,\tau)$. For an input signal $s(t)$, the channel output signal $r(t)$ is given by

$$r(t) = \int_{-\infty}^{+\infty} h(t,\tau)s(t-\tau)d\tau + n(t), \tag{1.1}$$

where $n(t)$ is the random noise process. However, we are interested in a characterization that is valid. That is, we recognize that the channel filter taps must be measured, but we want a statistical characterization of how many taps are necessary, how quickly they change and how much they vary. Such a characterization requires a probabilistic model of the channel tap values, perhaps gathered by statistical measurements of the channel. From this, we derived a discrete-time baseband model in terms of channel filter taps as

$$r[m] = \sum_{l} h[m,l]s[m-l] + n[m]. \tag{1.2}$$

Each tap $h[m,l]$ is the sum of a large number of such small independent circular symmetric random variables. This assures us that $h[m,l]$ is in fact circular symmetric $\mathcal{CN}(0,\sigma_l^2)$. With this assumed Gaussian probability density, we know that the magnitude $|h[m,l]|$ of the l^{th} tap is a Rayleigh random variable with density

$$\frac{x}{\sigma_l^2} exp\left\{\frac{-x^2}{2\sigma_l^2}\right\}, \qquad x \geq 0, \tag{1.3}$$

and the squared magnitude $|h[m,l]|^2$ is exponentially distributed with density

$$\frac{1}{\sigma_l^2} exp\left\{\frac{-x^2}{2\sigma_l^2}\right\}, \qquad x \geq 0. \tag{1.4}$$

This model called Rayleigh fading, is quite reasonable for scattering mechanisms where there are many small reflectors, but is adopted primarily for its simplicity in typical cellular situations with

a relatively small number of reflectors. Fading can then drastically reduce the received power. If reliable communication over wireless channels is to be achieved, measures must be put in place to counteract its effects. The most common technique against fading is to transmit many replicas of the original signal, in the hope that at least one of them will not fade ; this technique is known as diversity. Diversity has the drawback of causing inefficiencies in the system, since at least part of the available resources must be used to send the signal replicas. This introduces redundancy. In recent years, researchers have realized that multipath, as well as giving rise to fading, can help to combat it. Multiple refections are a natural phenomenon in wireless channels, and they can be harnessed to provide diversity ; no other system resource such as bandwidth needs to be employed. Furthermore, it has been determined that the capacity available in wireless channels can be enhanced using many antennas at the transmitter and the receiver side. Thus, the spatial dimension is a new aspect to be considered for the channel modeling.
Figure (1.3) illustrates different antenna configurations used in defining space-time systems. Single-input single-output (SISO) is the well-known wireless configuration, single-input multiple-output (SIMO) uses a single transmitting antenna and multiple (receive antennas, multiple-input single-output (MISO) has multiple transmitting antennas and one receive antenna and, finally, MIMO has multiple transmitting antennas and multiple receive antennas.

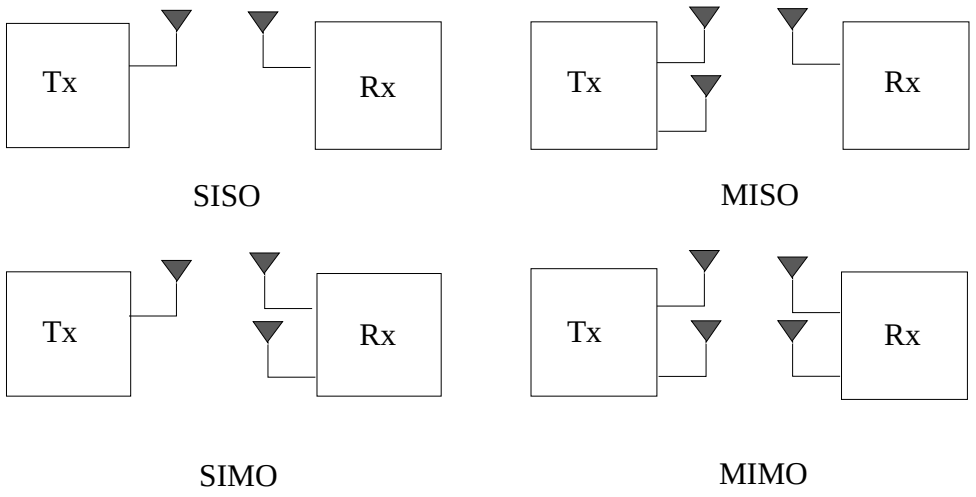


Figure 1.3 – Different antenna configurations in space-time systems.

Let us consider a MIMO system with M transmit and N receive antennas. Then the receive signal block is

$$\boldsymbol{Y_{N\times T}} = \boldsymbol{H_{N\times M}} \cdot \boldsymbol{X_{M\times T}} + \boldsymbol{W_{N\times T}}, \tag{1.5}$$

where $\boldsymbol{H}$ is the channel transfer matrix with complex entries h_{ij} representing the fading coefficients between the i^{th} receive and the j^{th} transmit antennas and are modeled by independent Gaussian random variables of zero-mean and variance 0.5 per component. The MIMO channel $\boldsymbol{H}$ at a given time instant may be represented as an $N \times M$ matrix

$$\boldsymbol{H_{N\times M}} = \begin{bmatrix} h_{11} & h_{12} & \cdots & h_{1M} \\ h_{21} & h_{22} & \cdots & h_{2M} \\ \vdots & \vdots & \ddots & \vdots \\ h_{N1} & h_{N2} & \cdots & h_{NM} \end{bmatrix}, \tag{1.6}$$

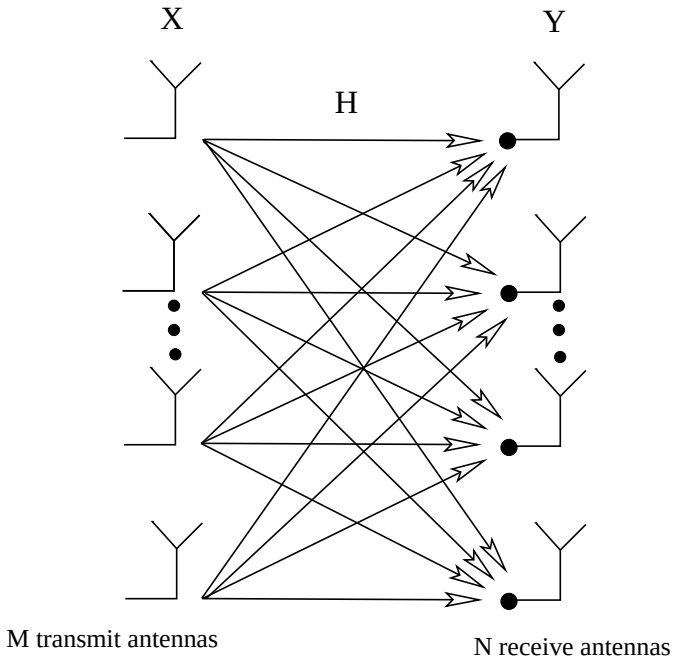
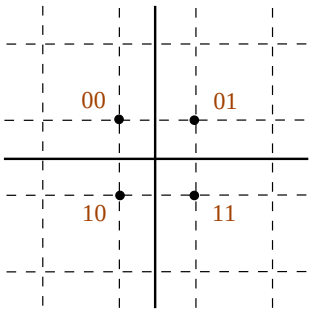
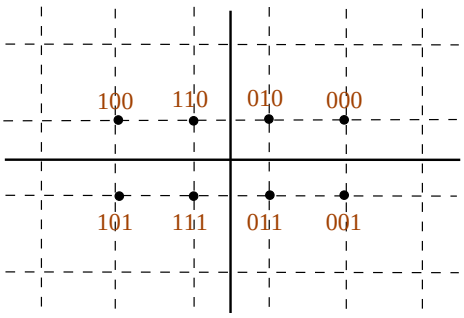


Figure 1.4 – MIMO System representation constituted of M transmit and N receive antennas

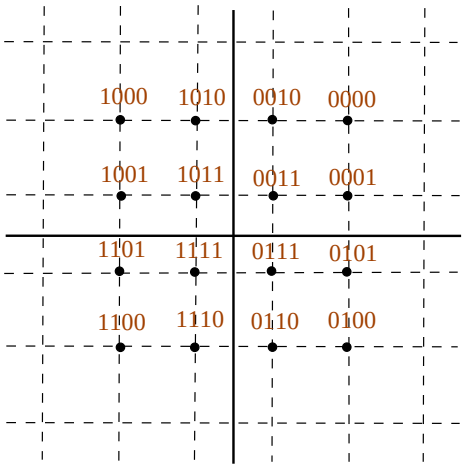
The figure (1.4) represents this system. $\boldsymbol{X_{M\times T}}$ is the transmit signal block over T symbol times and is a matrix of dimension $M \times T$. T is the temporal dimension, thus it's the number of channel use (cu). $\boldsymbol{W_{N\times T}} \in \mathcal{N}(0, \sigma^2.\boldsymbol{I})$ is an additive white gaussian noise. The channel is assumed to be constant during the transmission of a block (also sometimes referred to as a frame) and to change from one block to another. Also, it is assumed that the channel is memoryless between blocks ; that is, matrices associated with different blocks are statistically independent. Such a channel is known as a frequency-flat, slow fading channel,or simply as a block-fading channel. These characteristics are typical in fixed wireless applications, where some slow channel variations are expected ; an example would be an office environment where people constantly move around at walking speed. Matrix $\boldsymbol{H}$ is assumed to be full-rank. This is justified because the probability of a randomly generated matrix presenting non-independent rows and columns is very close to zero. In practice, this means that the receiver antennas must be adequately spaced. This requirement is not considered unreasonable in modern wireless applications where the carrier frequency is in the range of a few gigahertz and thus the required separation would be a few centimeters. Each receiver is assumed to have estimated $\boldsymbol{H}$ perfectly through the use of some appropriate method, such as a training sequence transmitted with each block. This situation is frequently described in the literature as the receiver having perfect channel-state information (CSI). This will be described later in this chapter.
The entries $|h_{ij}|$ are assumed to have a Rayleigh distribution. The entries of the channel matrix $\boldsymbol{H}$ are complex and gaussian. Each component lays out a gaussian real part and a gaussian imaginary part of null average and variance equal to 0.5. Thus, the channel $\boldsymbol{H}$ is considered as a Rayleigh channel.



(a) 4−QAM



(b) 8−QAM



(c) 16−QAM

FIGURE 1.5 – Example of a $\mathbb{Z}[i]$ lattice of dimension 2

1.1.2 Modulation and Demodulation

Modulation and demodulation are used in many kinds of data transmission, both analogue and digital. The choice of one type of modulation is based on bandwidth and signal-to-noise ratio. In digital modulation, an analog carrier signal is modulated by a digital bit stream. Digital modulation methods can be considered as digital-to-analog conversion, and the corresponding demodulation or detection as analog-to-digital conversion. The changes in the carrier signal are chosen from a finite number of alternative symbols (the modulation alphabet). If the alphabet consists of 2^{N_b} alternative symbols, each symbol represents a message consisting of N_b bits. Usually for our simulations, we will use the Quadrature Amplitude Modulation (QAM). The QAM modulation is simply a combination of amplitude modulation and phase shift keying. Its constellation points are usually arranged in a square grid with equal vertical and horizontal spacing. The set of constellations points is a final subset of $\mathbb{Z}[i]$. Since in digital telecommunications the data are usually binary, the number of points in the grid is usually a power of 2 (2, 4, 8 ...). Some examples of $q - QAM$ constellations with $q = 4,8,16$ are presented in figure (1.5).
By moving to a higher-order constellation, it is possible to transmit more bits per symbol. Ho-

wever, if the mean energy of the constellation is to remain the same (by way of making a fair comparison), the points must be closer together and are thus more susceptible to noise and other corruption ; this results in a higher bit error rate and so higher-order QAM can deliver more data less reliably than lower-order QAM, for constant mean constellation energy.

1.1.3 Fading Channels

The mobile radio channel experiences a lot of limitations on the performance of wireless systems. The transmission path can vary from line-of-sight to one severely obstructed by buildings and obstacles. Unlike wired channels, radio channels are extremely random and do not offer easy analysis. This modeling is therefore based more on statistics and requires the understanding of fading for an intended communication system. Fading is the deviation or the attenuation that a carrier-modulated signal experiences over certain propagation channel. The fading may vary with time, geographical position and/or radio frequency, and is often mathematically modeled as a random change in the amplitude and phase of the transmitted signal. A fading channel is a communication channel that experiences fading. In wireless systems, fading may either be due to multipath propagation, referred to as multipath induced fading, or due to shadowing from obstacles affecting the wave propagation, sometimes referred to as shadow fading.

A - Slow and Fast Fading
In the wireless communication literature, channels are often categorized as fast fading and slow fading. A channel is fast fading if the coherence time T_c is much shorter than the delay requirement of the application, and slow fading if T_c is longer. In a fast fading channel, one can transmit the coded symbols over multiple fades of the channel, while in a slow fading channel, one cannot. Thus, whether a channel is fast or slow fading depends not only on the environment but also on the application ; voice, for example, typically has a short delay requirement of less than 100 ms, while some types of data applications can have a laxer delay requirement. The important thing is to recognize that the major effect in determining time coherence is the Doppler spread D_s, and that the relationship is reciprocal ; the larger the Doppler spread, the smaller the time coherence :

$$T_c \simeq \frac{1}{D_s}. \tag{1.7}$$

B - Frequency Flat and Frequency-Selective Fading
Frequency selectivity is also an important characteristic of fading channel. As the carrier frequency of a signal is varied, the magnitude of the change in amplitude will vary. The coherence bandwidth B_c measures the separation in frequency after which two signals will experience uncorrelated fading. In flat fading, the coherence bandwidth of the channel B_c is larger than the bandwidth of the signal B_s. Therefore, all frequency components of the signal will experience the same magnitude of fading. Contrarily, in frequency-selective fading, the coherence bandwidth of the channel is smaller than the bandwidth of the signal. Different frequency components of the signal therefore experience decorrelated fading.
Since different frequency components of the signal are affected independently, it is highly unlikely that all parts of the signal will be simultaneously affected by a deep fade. Certain modulation schemes such as OFDM and CDMA are well-suited to employing frequency diversity to provide robustness to fading. OFDM divides the wideband signal into many slowly modulated narrowband subcarriers, each exposed to flat fading rather than frequency selective fading. This can be

combated by means of error coding, simple equalization or adaptive bit loading. Inter-symbol interference is avoided by introducing a guard interval between the symbols. CDMA uses the Rake receiver to deal with each echo separately.
Frequency-selective fading channels are also dispersive, in that the signal energy associated with each symbol is spread out in time. This causes transmitted symbols that are adjacent in time to interfere with each other. Equalizers are often deployed in such channels to compensate for the effects of the intersymbol interference.The echoes may also be exposed to Doppler shift, resulting in a time varying channel model.

C - Modeling of Channel Fading
Examples of fading models for the distribution of the attenuation are :

• The Rayleigh fading which is quite reasonable for scattering mechanisms where there are many small reflectors, but is adopted primarily for its simplicity in typical cellular situations with a relatively small number of reflectors. The word Rayleigh is almost universally used for this model, but the assumption is that the tap gains are circularly symmetric complex Gaussian random variables. The central limit theorem holds that, if there is sufficiently much scatter, the channel impulse response will be well-modeled as a Gaussian process irrespective of the distribution of the individual components. If there is no dominant component to the scatter, then such a process will have zero mean and phase evenly distributed between 0 and 2π radians. The envelope of the channel response will therefore be Rayleigh distributed.

• The Rice fading happens in the same conditions than the rayleigh fading and occurs when one of the paths, typically a line of sight signal, is much stronger than the others. In Rician fading, the amplitude gain is characterized by a Rician distribution.

The effects of fading can be combated by using diversity to transmit the signal over multiple channels that experience independent fading and coherently combining them at the receiver. The probability of experiencing a fade in this composite channel is then proportional to the probability that all the component channels simultaneously experience a fade, a much more unlikely event. Diversity can be achieved in time, frequency, or space

1.1.4 Diversity Techniques

The diversity techniques operate over time, frequency or space, but the basic idea is the same. By sending signals that carry the same information through different paths, multiple independently faded replicas of data symbols are obtained at the receiver end and more reliable detection can be achieved. There are three types of diversity schemes in wireless communications

• *Temporal diversity* : In this case replicas of the transmitted signal are provided across time by a combination of channel coding and time interleaving strategies. The key requirement here for this form of diversity to be effective is that the channel must provide sufficient variations in time. It is applicable in cases where the coherence time of the channel is small compared with the desired interleaving symbol duration. In such an event, we are assured that the interleaved symbol is independent of the previous symbol. This makes it a completely new replica of the

original symbol.

• *Frequency diversity* : This type of diversity provides replicas of the original signal in the frequency domain. This is applicable in cases where the coherence bandwidth of the channel is small compared with the bandwidth of the signal. This assures us that different parts of the relevant spectrum will suffer independent fades.

• *Spatial diversity* : This is also called antenna diversity and is an effective method for combating multipath fading. In this case, replicas of the same transmitted signal are provided across different antennas of the receiver. This is applicable in cases where the antenna spacing is larger than the coherent distance to ensure independent fades across different antennas.

Basically the effectiveness of any diversity scheme lies in the fact that at the receiver we must provide independent samples of the basic signal that was transmitted. In such an event we are assured that the probability of two or more relevant parts of the signal undergoing deep fades will be very small. The constraints on coherence time, coherence bandwidth, and coherence distance ensure this. The diversity scheme must then optimally combine the received diversified waveforms so as to maximize the resulting signal quality.
We can also categorize diversity under the subheading of spatial diversity, based on whether diversity is applied to the transmitter or to the receiver.

• *Receive diversity* : Maximum ratio combining is a frequently applied diversity scheme in receivers to improve signal quality. In cell phones it becomes costly and cumbersome to deploy. This is one of the main reasons transmit diversity became popular, since transmit diversity is easier to implement at the base station.
• *Transmit diversity* : In this case we introduce controlled redundancies at the transmitter, which

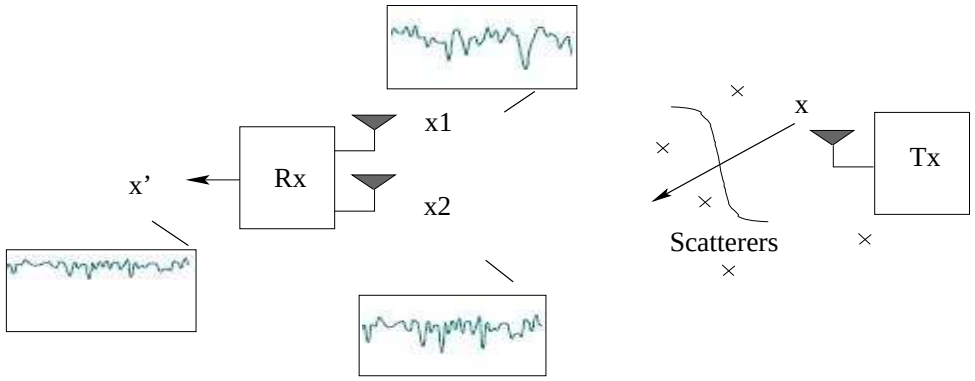


Figure 1.6 – Spatial Receive Diversity

can be then exploited by appropriate signal processing techniques at the receiver. Generally this technique requires complete channel information at the transmitter to make this possible. But with the invention of space-time coding schemes like Alamouti's scheme (4) it became possible to implement transmit diversity without knowledge of the channel. This was one of the fundamental reasons why the MIMO industry began to rise. Space-time codes for MIMO exploit both transmit as well as receive diversity schemes, yielding a high quality of reception.

Therefore, in MIMO we talk a lot about receive antenna diversity or transmit antenna di-

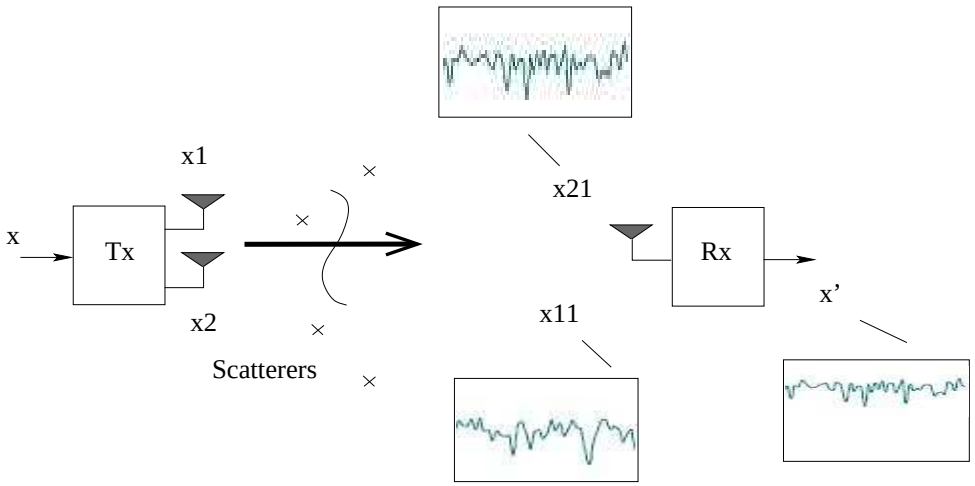


Figure 1.7 – Spatial Transmit Diversity

versity. In receive antenna diversity, the receiver that has multiple antennas receives multiple replicas of the same transmitted signal, assuming that the transmission came from the same source. This holds true for SIMO channels. If the signal path between each antenna pair fades independently, then when one path is in a fade, it is extremely unlikely that all the other paths are also in deep fade. If the number of receive antennas tends to infinity, the diversity order tends to infinity and the channel tends to additive white Gaussian noise (AWGN) (5).
In the category of spatial diversity there are two more types of diversity that we need to consider. These are :

• *Polarization diversity* : In this type of diversity horizontal and vertical polarization signals are transmitted by two different polarized antennas and received correspondingly by two different polarized antennas at the receiver. Different polarizations ensure that there is no correlation between the data streams, without having to worry about coherent distance of separation between the antennas.
• *Angle diversity* : This applies at carrier frequencies in excess of 10 GHz. At such frequencies, the transmitted signals are highly scattered in space. In such an event the receiver can have two highly directional antennas facing in totally different directions. This enables the receiver to collect two samples of the same signal, which are totally independent of each other.

Considering an $N \times M$ MIMO system, the maximum possible diversity gain is equal to $N \times M$. At high SNR, the error probability P_e decreases as the d^{th} power of SNR, corresponding to a slope of $-d$ in the error probability curve (in dB/dB scale).

$$P_e \propto \frac{1}{SNR^d} \tag{1.8}$$

Then, the diversity is

$$d = -\lim_{SNR\to\infty} \frac{\log(P_e)}{\log(SNR)}. \tag{1.9}$$

Thus, for high SNR, error probability average decreases asymptotically, therefore the probability of error will decrease if we send information on d independent paths.

1.1.5 Lattice Definition and Properties

A lattice Λ is a subset of rank p ; for $p < n$, of $\mathbb{R}^n$. Λ is then a lattice of dimension p and there exists p linearly independent n-dimensional vectors $\boldsymbol{v}_1,\boldsymbol{v}_2,\ldots,\boldsymbol{v}_p \in \mathbb{R}^n$ such that

$$\Lambda = \Lambda(\boldsymbol{S}) = \{a_1\boldsymbol{v}_1 + a_2\boldsymbol{v}_2 + \ldots + a_p\boldsymbol{v}_p : a_i \in \mathbb{Z}\} \tag{1.10}$$

where $\boldsymbol{S} = [\boldsymbol{v}_1,\boldsymbol{v}_2,\ldots,\boldsymbol{v}_p]$ is an $n \times p$ matrix. The set of column vectors $\{\boldsymbol{v}_1,\boldsymbol{v}_2,\ldots,\boldsymbol{v}_p\}$ and the matrix $\boldsymbol{S}$ are said to be the basis and the basis matrix of Λ, respectively. Thus, a lattice is an integer linear combination of the basis vectors. In the following, a lattice having $\boldsymbol{S}$ as a basis matrix will be denoted Λ_S.
Let's now consider some useful definitions (see figure (1.8)) :

• Gram matrix of a lattice Λ_S is $\boldsymbol{G} = \boldsymbol{S}^T\boldsymbol{S}$.

• The equivalent lattice
Let $\boldsymbol{Q} \in M_n(\mathbb{R})$, such that $\boldsymbol{Q}\boldsymbol{Q}^T = \boldsymbol{I}_n$.
Both lattices Λ_S and $\Lambda_{S.Q}$ are equivalent (same lattice).

• Fundamental volume of a lattice
The fundamental volume of a lattice Λ_S with a basis $\{\boldsymbol{v}_1,\boldsymbol{v}_2,\ldots,\boldsymbol{v}_p\}$ of $\mathbb{R}^n$ is given by

$$\boldsymbol{V} = \{\boldsymbol{x} \in \mathbb{R}^n \backslash \boldsymbol{x} = a_1\boldsymbol{v}_1 + a_2\boldsymbol{v}_2 + \ldots + a_p\boldsymbol{v}_p, 0 \leq a_i < 1, i = 1\ldots p\} \tag{1.11}$$

Geometrically, the determinant $det(\Lambda)$ of a lattice Λ is defined as the common content of the parallelepiped spanned by any lattice bases. Generally, a lattice has many possible bases but it has the same determinant.

• Voronoï Cell
The Voronoï Cell of a point $\boldsymbol{u}$ in a lattice Λ is the space region defined by

$$v(\boldsymbol{u}) = \{\boldsymbol{x} \in \mathbb{R}^n \backslash \|\boldsymbol{x} - \boldsymbol{u}\| \leq \|\boldsymbol{x} - \boldsymbol{y}\|, \boldsymbol{y} \in \Lambda\} \tag{1.12}$$

The Voronoï Cell is then a cell structure where each cell's interior consists of all points that are closer to a particular lattice point than to any other lattice point. Since lattice is uniform, all Voronoï Cells are identic. The fundamental volume of a lattice is equal to the Voronoï Cell volume.
The representation of MIMO systems as a lattice and decoding them with a lattice decoder was first explored by Damen et al. in (6). One can separate imaginary part and real part, then vectorizing MIMO systems (coded or non-coded) and get lattice representation of the channel model.

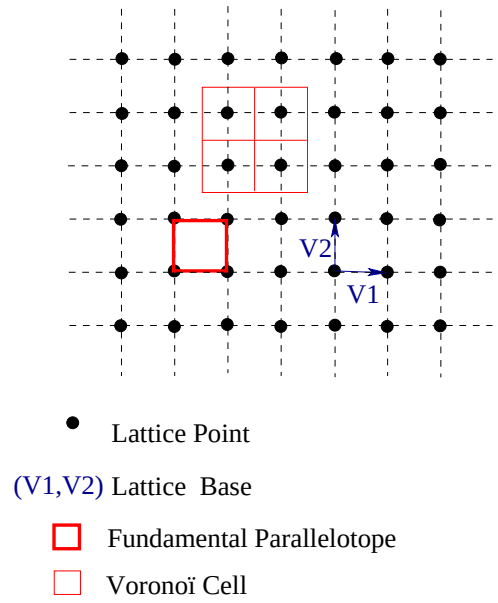


FIGURE 1.8 – Example of a $\mathbb{Z}[i]$ lattice of dimension 2

1.1.5.1 For the non-coded System case

Here we assume that T, the temporal dimension, is equal to 1. Let's the invertible mapping $\Psi : \mathbb{C}^M \rightarrow \mathbb{R}^{\underline{M}}$ from complex valued vector $\boldsymbol{v}$ to real valued vector $\underline{\boldsymbol{v}}$ by stacking the real part of $\boldsymbol{v}$ over its imaginary part, be defined as :

$$\underline{\boldsymbol{v}} = \Psi(\boldsymbol{v}) = \begin{bmatrix} \mathfrak{Re}(\boldsymbol{v}) \\ \mathfrak{Im}(\boldsymbol{v}) \end{bmatrix}, \tag{1.13}$$

where $\underline{M} = 2M$.
When $N \neq 1$, the mapping $\Psi : \mathbb{C}^{N \times M} \rightarrow \mathbb{R}^{\underline{N} \times \underline{M}}$ and $\underline{N} = 2N$ from complex valued matrix $\boldsymbol{A}$ to real valued matrix $\underline{\boldsymbol{A}}$ is defined as follows :

$$\underline{\boldsymbol{A}} = \Psi(\boldsymbol{A}) = \begin{bmatrix} \mathfrak{Re}(\boldsymbol{A}) & -\mathfrak{Im}(\boldsymbol{A}) \\ \mathfrak{Im}(\boldsymbol{A}) & \mathfrak{Re}(\boldsymbol{A}) \end{bmatrix}. \tag{1.14}$$

Then, one can rewrite equation (1.5) by separating real and imaginary parts as follows :

$$\underline{\boldsymbol{y}}_{\underline{N}} = \underline{\boldsymbol{H}}_{\underline{N} \times \underline{M}} \underline{\boldsymbol{x}}_{\underline{M}} + \underline{\boldsymbol{z}}_{\underline{N}}. \tag{1.15}$$

Since the matrix $\boldsymbol{H}_{\boldsymbol{N \times M}}$ is full rank , the matrix $\underline{\boldsymbol{H}}_{\underline{N} \times \underline{M}}$ is then of full rank. We get then a lattice representation of non-coded MIMO systems. The lattice dimension is $\underline{N} \times \underline{M}$ and the generating matrix is $\underline{\boldsymbol{H}}_{\underline{N} \times \underline{M}}$.

1.1.5.2 For the coded System case

In this case $\boldsymbol{X}$ is no longer an information symbols vector as described in the non coded case. But, it represents the codeword matrix to be sent. Then, the received codeword stays as described

in equation (1.5).

Two steps should be done to get a lattice representation of MIMO coded system :
1) represent the coded system as a non coded system,
2) separate real and imaginary parts.

The first step consists of vectorization. Thus, the equation (1.5) becomes therefore

$$\boldsymbol{y_{N\cdot T}} = \begin{pmatrix} \boldsymbol{H_{N\times M}} & & 0 \\ & \ddots & \\ 0 & & \boldsymbol{H_{N\times M}} \end{pmatrix} \cdot \underbrace{\begin{pmatrix} \phi_{11} & \dots & \phi_{1,M\cdot T} \\ \vdots & \ddots & \vdots \\ \phi_{M\cdot T,1} & \dots & \phi_{M\cdot T,M\cdot T} \end{pmatrix}}_{\boldsymbol{\phi_{M\cdot T\times M\cdot T}}} \cdot \underbrace{\begin{pmatrix} x_1 \\ \vdots \\ x_{M\cdot T} \end{pmatrix}}_{\boldsymbol{x_{M\cdot T}}} + \begin{pmatrix} z_1 \\ \vdots \\ z_{N\cdot T} \end{pmatrix},$$

where $\boldsymbol{\phi_{M\cdot T\times M\cdot T}}.\boldsymbol{x_{M\cdot T}}$ is the vectorization of $\boldsymbol{X_{M\times T}}$.

As a result, we get a system equivalent to (1.5)

$$\begin{aligned} \boldsymbol{y_{N\cdot T}} &= \boldsymbol{H_{1,N\cdot T\times M\cdot T}} \cdot \boldsymbol{\phi_{M\cdot T\times M\cdot T}} \cdot \boldsymbol{x_{M\cdot T}} + \boldsymbol{z_{N\cdot T}} \\ &= \boldsymbol{\mathcal{H}_{N\cdot T\times M\cdot T}} \cdot \boldsymbol{x_{M\cdot T}} + \boldsymbol{z_{N\cdot T}} \end{aligned} \tag{1.16}$$

The separation of the real and imaginary parts is applied on the former equation as in equation (1.15), and the coded system is therefore given by

$$\underline{\boldsymbol{y}}_{\underline{\boldsymbol{N}}\cdot\underline{\boldsymbol{T}}} = \underline{\boldsymbol{\mathcal{H}}}_{\underline{\boldsymbol{N}}\cdot\underline{\boldsymbol{T}}\times\underline{\boldsymbol{M}}\cdot\underline{\boldsymbol{T}}} \cdot \underline{\boldsymbol{x}}_{\underline{\boldsymbol{M}}\cdot\underline{\boldsymbol{T}}} + \underline{\boldsymbol{z}}_{\underline{\boldsymbol{N}}\cdot\underline{\boldsymbol{T}}} \tag{1.17}$$

where we defined by $\underline{\boldsymbol{\mathcal{H}}}_{\underline{\boldsymbol{N}}\cdot\underline{\boldsymbol{T}}\times\underline{\boldsymbol{M}}\cdot\underline{\boldsymbol{T}}}$ the equivalent lattice generator matrix.

1.2 Some Mathematical Tools for MIMO Decoding

Once the equivalent lattice generator matrix is defined, many different optimal and sub-optimal decoders algorithms can be developed and implemented. Among these decoders we can mention lattice decoders and sequential decoders. Those decoders can be seen as tree search decoders. In order to construct the tree search, we need to transform the lattice generator matrix using some mathematical tools for matrix decomposition like the QR decomposition, the Cholesky decomposition and the the singular value decomposition (SVD), etc. Here, we will present those three decompositions that will be used later for MIMO Decoding.

1.2.1 QR Decomposition

1.2.1.1 principle

A QR decomposition of a matrix is a decomposition of the matrix into an orthogonal and a upper triangular matrix. Let's $\underline{\boldsymbol{\mathcal{H}}}$ be a real matrix of dimension $n \times m$, there exists an orthogonal

matrix $\boldsymbol{Q}$ (its columns are orthogonal unit vectors meaning $\boldsymbol{Q^T \cdot Q = I}$) of dimension $n \times n$ and an upper triangular matrix $\boldsymbol{R}$ (also called right triangular matrix) of dimension $n \times m$ such that

$$\underline{\boldsymbol{\mathcal{H}}} = \boldsymbol{Q \cdot R} \tag{1.18}$$

1.2.1.2 Application

By applying this decomposition to the previously defined channel matrix $\underline{\boldsymbol{\mathcal{H}}}$, the MIMO system becomes

$$\underline{\boldsymbol{y}} \;=\; \boldsymbol{Q \cdot R \cdot \underline{x} + \underline{z}} \tag{1.19}$$

Since $\boldsymbol{Q}$ is orthogonal, the multiplication of the previous equation by $\boldsymbol{Q^T}$ gives an equivalent system since the noise stays white and gaussian.

$$\underline{\boldsymbol{u}} = \boldsymbol{Q^T \cdot \underline{y} = R \cdot \underline{x} + Q^T \cdot \underline{z}} \tag{1.20}$$

$\boldsymbol{R}$ is then considered as the new lattice generator matrix of $\Lambda_{\boldsymbol{R}}$. Making the lattice generator matrix triangular is used in many decoding algorithms and is generally used in the 'pre-decoding' phase.

1.2.1.3 Complexity

The complexity of this phase depends on the lattice dimensions. For a lattice of dimension n, the complexity of the algorithm is given by $\frac{2}{3}n^3$ (8) to which we should add the complexity of the multiplication of the matrix $\boldsymbol{Q^T}$ by the received vector $\underline{\boldsymbol{y}}$.

1.2.2 Cholesky factorization

1.2.2.1 Principle

Similarly to the QR decomposition, the Cholesky factorization is a decomposition of a symmetric, positive-definite matrix into the product of a lower triangular matrix and its conjugate transpose. Thus, for a lattice $\Lambda_{\underline{\boldsymbol{\mathcal{H}}}}$, it can be used to factorize the Gram matrix ($\boldsymbol{G = \underline{\mathcal{H}}^T \cdot \underline{\mathcal{H}}}$). It consists of determining a lower triangular matrix $\boldsymbol{L}$ with strictly positive diagonal entries such that

$$\boldsymbol{G = (LU) \cdot (LU)^T} \tag{1.21}$$

Algorithm 1 QR Decomposition

- Input : $\underline{\boldsymbol{\mathcal{H}}}$,$n$
- Initialization : For $i = 1 : n$, for $j = 1 : n$, $\boldsymbol{R}(i,j) = 0$.
- $\boldsymbol{R}(1,1) = \sum_{i=1}^{n} \underline{\boldsymbol{\mathcal{H}}}(i,1)$; $\boldsymbol{Q} = \underline{\boldsymbol{\mathcal{H}}}(:,1)/\boldsymbol{R}(1,1)$;
- For $k = 2 : n$
  $\boldsymbol{R}(1 : k-1,k) = \boldsymbol{Q}(:,1 : k-1)^T \cdot \underline{\boldsymbol{\mathcal{H}}}(:,k)$;
  $\boldsymbol{Q}(:,k) = \underline{\boldsymbol{\mathcal{H}}}(:,k) - \boldsymbol{Q}(:,1 : k-1) \cdot \boldsymbol{R}(1 : k-1,k)$;
  $\boldsymbol{R}(k,k) = \sum_{i=1}^{n} \boldsymbol{Q}(i,k)$
  $\boldsymbol{Q}(:,k) = \boldsymbol{Q}(:,k)/\boldsymbol{R}(k,k)$

or quite simply :

$$\boldsymbol{G} = \boldsymbol{L} \cdot \boldsymbol{L}^{\boldsymbol{T}} \tag{1.22}$$

$\boldsymbol{U}$ is a unitary matrix ($\boldsymbol{U} \cdot \boldsymbol{U}^{\boldsymbol{T}} = \boldsymbol{U}^{\boldsymbol{T}} \cdot \boldsymbol{U} = \boldsymbol{I}$).

1.2.2.2 Application

We can show that the Cholesky factorization and the QR decomposition are equivalent. Let's $\underline{\boldsymbol{\mathcal{H}}} = \boldsymbol{Q} \cdot \boldsymbol{R}$ be the QR decomposition of $\underline{\boldsymbol{\mathcal{H}}}$. The Gram matrix can be written as :

$$\begin{aligned} \boldsymbol{G} &= \underline{\boldsymbol{\mathcal{H}}}^{\boldsymbol{T}} \cdot \underline{\boldsymbol{\mathcal{H}}} \\ &= (\boldsymbol{QR})^{\boldsymbol{T}} \cdot (\boldsymbol{QR}) \\ &= \boldsymbol{R}^{\boldsymbol{T}} \cdot \boldsymbol{Q}^{\boldsymbol{T}} \cdot \boldsymbol{Q} \cdot \boldsymbol{R} \\ &= \boldsymbol{R}^{\boldsymbol{T}} \cdot \boldsymbol{R} \\ &= \boldsymbol{L} \cdot \boldsymbol{L}^{\boldsymbol{T}} \end{aligned} \tag{1.23}$$

We can remark that $\boldsymbol{R} = \boldsymbol{L}^{\boldsymbol{T}}$. $\boldsymbol{Q}$ is obtained by the equation $\boldsymbol{Q} = \underline{\boldsymbol{\mathcal{H}}} \cdot \boldsymbol{R}^{-\boldsymbol{1}}$. Thus, we conclude that both factorizations are equivalent. Another useful application for the Cholesky factorization is to calculate the inverse matrix of $\boldsymbol{G}$, calculate its determinant, etc. The determinant is equal to the square of the product of diagonal elements of $\boldsymbol{L}$. In fact :

$$\begin{aligned} \|\det(\boldsymbol{G})\| &= \left\|\det\left(\boldsymbol{LL}^{\boldsymbol{T}}\right)\right\| \\ &= \|\det(\boldsymbol{L})\| \quad \left\|\det\left(\boldsymbol{L}^{\boldsymbol{T}}\right)\right\| \\ &= \|\det(\boldsymbol{L})\|^2 \end{aligned} \tag{1.24}$$

And since the determinant of the triangular matrix is the product of diagonal elements :

$$\|\det(\boldsymbol{G})\| = \left(\prod_{i=1}^{n} \boldsymbol{L}_{ii}\right)^2 \tag{1.25}$$

1.2.2.3 Complexity

The complexity of the Cholesky decomposition is given by $\frac{1}{6}n^3 + n$ (8). It's inferior to the complexity of the QR method. But since the Cholesky factorization is done for the Gram matrix, this adds n^3 multiplications to the complexity. Moreover, the QR decomposition algorithm is numerically stable. Consequently, we chose this method for our pre-decoding phase.

Algorithm 2 Cholesky Factorization

- Input : $\boldsymbol{G}(i,j)_{1\leq j\leq i\leq n}$ representing the lower triangular part of $\boldsymbol{G}$ ($\boldsymbol{G}$ symmetric, positive-definite).
- Output : $\boldsymbol{G}(i,j)_{1\leq i\leq j\leq n}$ representing $\boldsymbol{L}$ satisfying $\boldsymbol{G} = \boldsymbol{L}\cdot\boldsymbol{L}^{\boldsymbol{T}}$.
- $\boldsymbol{G}(1,1) = sqrt(\boldsymbol{G}(1,1))$.
- For $i = 2 : n$
  $\boldsymbol{G}(i,1) = \boldsymbol{G}(i,1)/\boldsymbol{G}(1,1)$;
- For $k = 2 : n-1$
  $\boldsymbol{G}(k,k) = sqrt(\boldsymbol{G}(k,k) - \sum_{j=1}^{k-1} \boldsymbol{G}^2(k,j))$;
  For $i = k+1 : n$
  $\boldsymbol{G}(i,k) = (\boldsymbol{G}(i,k) - \sum_{j=1}^{k-1} \boldsymbol{G}(i,j)\cdot\boldsymbol{G}(k,j))/\boldsymbol{G}(k,k)$;
- $\boldsymbol{G}(n,n) = sqrt(\boldsymbol{G}(n,n) - \sum_{j=1}^{n-1} \boldsymbol{G}^2(n,j))$;

1.2.3 SVD Decomposition

1.2.3.1 Principle

The singular value decomposition is an important factorization of a rectangular real or complex matrix of dimensions $n \times m$. It consists of writing the matrix $\underline{\underline{\boldsymbol{\mathcal{H}}}}$ as :

$$\underline{\underline{\boldsymbol{\mathcal{H}}}} = \boldsymbol{U} \cdot \boldsymbol{D} \cdot \boldsymbol{V^T} \cdot \boldsymbol{s} \tag{1.26}$$

where $\boldsymbol{U}$ is a unitary matrix of dimensions $n \times n$, $\boldsymbol{D}$ is a $n \times m$ diagonal matrix with positive or null elements and $\boldsymbol{V}$ is an $m \times m$ unitary matrix. The matrix $\boldsymbol{D}$ contains the singular values of $\underline{\underline{\boldsymbol{\mathcal{H}}}}$.

1.2.3.2 Application

This method can be used for MIMO decoding. For example, using the SVD decomposition for a MIMO system with $m < n$, we can prove that decoding of the initial system of dimensions $n \times m$ can be brought back to a decoding in lattice of dimensions $m \times m$. Thus, let's recall the equation

$$\underline{\boldsymbol{y}} = \underline{\underline{\boldsymbol{\mathcal{H}}}} \cdot \underline{\boldsymbol{x}} + \underline{\boldsymbol{z}} \tag{1.27}$$

where $\underline{\boldsymbol{y}}$, $\underline{\underline{\boldsymbol{\mathcal{H}}}}$, $\underline{\boldsymbol{x}}$ and $\underline{\boldsymbol{z}}$ are respectively of dimensions $n \times 1$, $n \times m$, $m \times 1$ and $n \times 1$. SVD decomposition applied to this system gives :

$$\underline{\boldsymbol{y}} = \boldsymbol{U} \cdot \boldsymbol{D} \cdot \boldsymbol{V^T} \cdot \underline{\boldsymbol{x}} + \underline{\boldsymbol{z}} \tag{1.28}$$

Since the matrix $\boldsymbol{U}$ is unitary, multiplication by $\boldsymbol{U^T}$ leads to an equivalent system :

$$\begin{aligned} \boldsymbol{U^T} \cdot \underline{\boldsymbol{y}} = &\quad \boldsymbol{D} \cdot \boldsymbol{V^T} \cdot \underline{\boldsymbol{x}} &&+\boldsymbol{U^T} \cdot \underline{\boldsymbol{z}} \\ \boldsymbol{y_1} = &\quad \boldsymbol{\Phi} \cdot \underline{\boldsymbol{x}} &&+\boldsymbol{z_1} \end{aligned} \tag{1.29}$$

Let's remind that the diagonal matrix $\boldsymbol{D}$ contains the singular values of $\underline{\underline{\boldsymbol{\mathcal{H}}}}$. But, since $\underline{\underline{\boldsymbol{\mathcal{H}}}}$ has at maximum m singular values, consequently, rows of $\boldsymbol{D}$ of index $> m$ are null.

$$\boldsymbol{D} = \left[\begin{array}{ccc} d_1 & & \\ & \ddots & \\ & & d_m \\ 0 & \cdots & 0 \\ \vdots & & \vdots \\ 0 & \cdots & 0 \end{array}\right] \Bigg\downarrow n$$

In the system (1.30), multiplication of $\boldsymbol{D}$ by $\boldsymbol{V^T}$ gives a matrix $\boldsymbol{\Phi}$ null for rows beyond m.

$$\begin{bmatrix} y_1 \\ \vdots \\ \\ \\ \\ y_n \end{bmatrix} = \begin{bmatrix} \Phi_{11} & \cdots & \Phi_{1m} \\ \vdots & & \vdots \\ \Phi_{m1} & \cdots & \Phi_{mm} \\ 0 & \cdots & 0 \\ \vdots & & \vdots \\ 0 & \cdots & 0 \end{bmatrix} \begin{bmatrix} \underline{x}_1 \\ \vdots \\ \underline{x}_m \end{bmatrix} + \begin{bmatrix} \underline{z}_1 \\ \vdots \\ \\ \\ \\ \underline{z}_n \end{bmatrix} \tag{1.30}$$

(1.31)

Consequently, we can estimate that the last $n-m$ elements of the vector $\boldsymbol{y_1}$ are also null. It's then unnecessary to apply decoding until the n^{th} order, the initial lattice of dimensions $n \times m$ will be reduced to a lattice of less dimensions $m \times m$.
The SVD decomposition permits also to calculate the rank of the matrix. It was shown in (8) that the rank of the matrix $\underline{\boldsymbol{\mathcal{H}}}$ is equal to the number of non-null singular values of $\boldsymbol{D}$. The decomposition into singular values permits also to calculate the pseudo-inverse of a matrix. In fact, the pseudo-inverse of the matrix $\underline{\boldsymbol{\mathcal{H}}}$ is given by :

$$\underline{\boldsymbol{\mathcal{H}}}^{+} = \boldsymbol{V} \cdot \boldsymbol{D}^{+} \cdot \boldsymbol{U^T} \tag{1.32}$$

where $\boldsymbol{D}^{+}$ represents the diagonal matrix $\boldsymbol{D}$ where each non-null component is replaced by its inverse.

1.2.3.3 Complexity

For a matrix $\underline{\boldsymbol{\mathcal{H}}}$ of dimensions $n \times m$, the complexity of this phase is of the order $8n^2m + 8nm^2 + 9m^3$. It's widely bigger than the complexity of the QR and the Cholesky decompositions. Nevertheless, by applying the SVD decomposition in the pre-decoding phase, we can limit the initial problem to a smallest lattice dimensions which means a significant gain in complexity of decoding.

1.3 Information Theory

1.3.1 Mutual Information and MIMO Channel Capacity

1.3.1.1 Mutual Information

A communication system can be modeled by a source and a destination. The source transmits a message which is detected by the receiver. If considering the transmitted signal $\boldsymbol{x}$ and the

received signal $\boldsymbol{y}$ as random processes, information theory introduces the the following notions :

Definition 1 (The entropy) *the entropy is a measure of the uncertainty associated with a random variable. The term by itself in this context usually refers to the Shannon entropy, which quantifies, in the sense of an expected value, the information contained in a message. The entropy of a random process $\boldsymbol{x}$ is*

$$\begin{aligned} \mathrm{H}(\boldsymbol{x}) &= \mathrm{E}_{\boldsymbol{x}}\left[-\log p_{\boldsymbol{x}}(x)\right] && (1.33)\\ &= \sum_{x\in\mathcal{X}} -p_{\boldsymbol{x}}(x)\log p_{\boldsymbol{x}}(x) && (1.34)\end{aligned}$$

with $p_{\boldsymbol{x}}(x)$ is the probability density function of the random variable $\boldsymbol{x}$ and $\mathcal{X}=\{x : p_{\boldsymbol{x}}(x)\neq 0\}$. The entropy measurement unit depends on the used logarithmic base (for the natural logarithm, we use the nat per symbol, and for the base 2 logarithm we use the bit per symbol).

Definition 2 (The mutual information) *the mutual information $\boldsymbol{I}(\boldsymbol{x},\boldsymbol{y})$ of two random variables $\boldsymbol{x}$ and $\boldsymbol{y}$ is a quantity that measures the mutual dependence of the two variables.*

$$\boldsymbol{I}(\boldsymbol{x},\boldsymbol{y}) = \sum_{x\in\mathcal{X},y\in\mathcal{Y}} p_{\boldsymbol{x},\boldsymbol{y}}(x,y)\log\frac{p_{\boldsymbol{x},\boldsymbol{y}}(x,y)}{p_{\boldsymbol{x}}(x)p_{\boldsymbol{y}}(y)} \tag{1.35}$$

where $\mathcal{X}=\{x : p_{\boldsymbol{x}}(x)\neq 0\}$ and $\mathcal{Y}=\{y : p_{\boldsymbol{y}}(y)\neq 0\}$.
The mutual information can also be expressed as a function of entropy

$$\begin{aligned} \boldsymbol{I}(\boldsymbol{x},\boldsymbol{y}) &= H(\boldsymbol{x}) - H(\boldsymbol{x}/\boldsymbol{y})\\ &= H(\boldsymbol{y}) - H(\boldsymbol{y}/\boldsymbol{x}) && (1.36)\\ &= H(\boldsymbol{x}) + H(\boldsymbol{y}) - H(\boldsymbol{x},\boldsymbol{y})\end{aligned}$$

The results of Shannon permitted to show that propagation channel can be assimilated to a function from the input process space to the output process space. This function includes the set of all deterministic and random transformations that occurred to the signal. Shannon showed also that there exists a theoretical limit for information throughput, exceeding this limit the information can't be transmitted without error. This maximum rate is called channel capacity.

1.3.1.2 MIMO Channel Capacity

The system capacity is defined as the maximum possible transmission rate such that the probability of error is arbitrarily small. ShannonŠs pioneering work showed that the capacity of a channel can be simply characterized in terms of the mutual information between the input and the output of the channel. Thus, the capacity of MIMO channel is defined as (7)(9)

$$C = \max_{p_{\boldsymbol{x}}(x)} \boldsymbol{I}(\boldsymbol{x},\boldsymbol{y}) \tag{1.37}$$

The ergodic capacity of a MIMO channel is the ensemble average of the information rate over the distribution of the elements of the channel matrix (13). It is the capacity of the channel when

every channel matrix is an independent realization [i.e., it has no relationship to the previous matrix but is typically representative of it class (ergodic)]. This implies that it is a result of infinitely long measurements. Since the process model is ergodic, this implies that the coding is performed over an infinitely long interval.

• **Channel State Information at Transmitter and at Receiver**
This case is generally denoted by the abbreviation CSITR (Channel State Information at Transmitter and Receiver). For a particular channel response (deterministic channel), the capacity can be expressed as

$$C = E_H \left\{ \max_{p_{\boldsymbol{x}|\boldsymbol{H}}: E[\boldsymbol{x}^H \boldsymbol{x}] \leq P_T} I(\boldsymbol{x}|\boldsymbol{H},\boldsymbol{y}|\boldsymbol{H}) \right\}. \tag{1.38}$$

Now, the mutual information $\boldsymbol{I}(\boldsymbol{x},\boldsymbol{y})$ is given by :

$$\boldsymbol{I}(\boldsymbol{x},\boldsymbol{y}) = \log\det\left(\boldsymbol{I}_N + \frac{1}{\sigma^2}\boldsymbol{H}\boldsymbol{R}_x\boldsymbol{H}^H\right) \tag{1.39}$$

The equation (1.38) assumes that $\boldsymbol{x}$ is a circular symmetric complex Gaussian vector[1]. Knowing $\boldsymbol{H}$, one should look for the covariance matrix of $\boldsymbol{x}$, $\boldsymbol{R}_x$, maximizing the mutual information and verifying $tr\,(\boldsymbol{R}_x) \leq P_T$ and $\boldsymbol{R}_x$ is definite positive .
Applying the singular value decomposition (SVD) factorization for the known channel matrix $\boldsymbol{H}$

$$\boldsymbol{H} = \boldsymbol{U}\Sigma\boldsymbol{V}^H, \tag{1.40}$$

where $\boldsymbol{U} \in \mathbb{C}^{N\times N}$ and $\boldsymbol{V} \in \mathbb{C}^{M\times M}$ are unitary complex matrices and Σ is a diagonal matrix constituted of $r = rank(\boldsymbol{H})$ channel singular values $\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_r$.
The mutual information can be rewritten as [2]

$$\boldsymbol{I}(\boldsymbol{x}|\boldsymbol{H},\boldsymbol{y}|\boldsymbol{H}) = \log\det\left(\boldsymbol{I}_N + \frac{1}{\sigma^2}\Sigma\boldsymbol{V}^H\boldsymbol{R}_x\boldsymbol{V}\Sigma\right). \tag{1.41}$$

The matrix $\tilde{\boldsymbol{Q}} = \boldsymbol{V}^H\boldsymbol{R}_x\boldsymbol{V}$ is definite positive, then

$$\det\left(\boldsymbol{I}_N + \Sigma\tilde{\boldsymbol{Q}}\Sigma\right) \leq \prod_{i=1}^{r}\left(1 + \frac{\tilde{Q}_{ii}\sigma_i^2}{\sigma^2}\right), \tag{1.42}$$

and the two quantities are equal if $\tilde{\boldsymbol{Q}}$ is diagonal.
The mutual information is then maximized if $\tilde{\boldsymbol{Q}}$ is diagonal with eigenvalues $\tilde{\boldsymbol{Q}}_{ii}$. Thus, the capacity can be written as

$$C_{CSITR} = \sum_{i=1}^{r=rank(\boldsymbol{H})} \log\left(1 + \frac{\tilde{\boldsymbol{Q}}_{ii}\sigma_i^2}{\sigma^2}\right) \tag{1.43}$$

The water-filling technique consists of optimal energy allocation over the r MIMO channel modes

$$\tilde{\boldsymbol{Q}}_{ii} = \left(\mu - \frac{\sigma^2}{\sigma_i^2}\right) \text{ where } \mu \text{ such that } \textstyle\sum_{i=1}^{r}\tilde{\boldsymbol{Q}}_{ii} = P_T$$

[1] A complex Gaussian vector $\boldsymbol{x}$ is circular symmetric if the covariance matrix of the vector $\underline{\boldsymbol{x}}$ has the form $E\left[(\underline{\boldsymbol{x}} - E[\underline{\boldsymbol{x}}])(\underline{\boldsymbol{x}} - E[\underline{\boldsymbol{x}}])^H\right] = 1/2.\underline{\boldsymbol{Q}}$, where $\boldsymbol{Q}$ is an hermitian definite positive matrix

[2] By applying the determinant identity : $\det(\boldsymbol{I} + \boldsymbol{A}\boldsymbol{B}) = \det(\boldsymbol{I} + \boldsymbol{B}\boldsymbol{A})$

P_T is the total transmitted energy over the M transmit antennas. Let's define the function $(.)^+$ as

$$(a)^+ = \begin{cases} a & \text{if } a > 0 \\ 0 & \text{else} \end{cases}$$

Finally, the capacity formula, when channel is known at the transmitter and the receiver, is given by

$$C_{CSITR} = \sum_{i=1}^{r} \left(\log \left(\frac{\mu \sigma_i^2}{\sigma^2} \right) \right)^+ \tag{1.44}$$

• Channel State Information at the Receiver Only

We assume in the following and throughout this report that channel knowledge is available only at the receiver. Thus, we speak about CSIR (Channel State Information at the Receiver Only). In these conditions, the channel matrix is assumed to be random and independent from the transmitted signal and the noise. In 1999, Telatar showed in (7) that the channel capacity is reached for a transmitted signal $\boldsymbol{x}$ which is circular symmetric of null average and with a covariance matrix $\boldsymbol{R}_x = \frac{P_T}{M} \boldsymbol{I}_M$. Then, the ergodic capacity can be expressed as

$$C_{CSIR} = E_{\boldsymbol{H}} \left[\log \det \left(\boldsymbol{I}_N + \frac{P_T}{M\sigma^2} \boldsymbol{H}\boldsymbol{H}^H \right) \right]. \tag{1.45}$$

Telatar results (7), Foschini and Gans results (9) permitted to establish the analytical capacity formulas in the case of Rayleigh channel with no spatial correlation.

Here, we will detail the channel capacity in the case of Rayleigh Channel (without spatial correlation). The stochastic Rayleigh model for a $N \times M$ MIMO system is defined by an $N \times M$ matrix where the components $h_{i,j}$ are random complex i.i.d gaussian centered and with unitary variance such that

$$h_{i,j} \sim \frac{1}{\sqrt{2}} \left[\mathcal{N}(0,1) + \sqrt{(-1)} \mathcal{N}(0,1) \right] \tag{1.46}$$

For an $M = N = 1$ channel, the capacity, taking into account equations (1.46) and (1.45), is

$$C_{CSIR}^{1\times 1} = \log \left(1 + \frac{P_T}{\sigma^2} \mathcal{X}_2^2 \right), \tag{1.47}$$

where $\mathcal{X}_2^2$ is a chi-square random variable with 2 degrees of freedom[3].

If we have a receive diversity ($M = 1, N > 1 : SIMOsystem$), the ergodic capacity is given by

$$C_{CSIR}^{SIMO} = \log \left(1 + \frac{P_T}{\sigma^2} \mathcal{X}_{2N}^2 \right). \tag{1.48}$$

Now, if we have transmit diversity ($N = 1, M > 1 : MISOsystem$), the ergodic capacity is expressed by

$$C_{CSIR}^{MISO} = \log \left(1 + \frac{P_T}{M\sigma^2} \mathcal{X}_{2M}^2 \right). \tag{1.49}$$

[3] A random variable is said to have a chi-square distribution $\mathcal{X}_2^k$ if it equals the sum of the squares of k statistically independent standard Gaussian random variables.

For a MIMO channel such that $M = N = N_a$, Telatar established the following analytical expression for systems of high dimensions (7) :

$$C \approx N_a \int_0^4 \log\left(1 + \frac{P_T}{\sigma^2}\tau\right) \frac{1}{4}\sqrt{\frac{1}{\tau} - \frac{1}{4}} d\tau. \tag{1.50}$$

Thus, for an identical and big antennas number at the transmission and the reception sides, the ergodic capacity for a Rayleigh channel with a flat frequency fading becomes linear in function of antennas number.
More generally, Oyman et al. established in (14) the ergodic capacity for a non-correlated Rayleigh channel with dimensions $M \times N$ with a high $SNR = \frac{P_T}{\sigma^2}$:

$$C \approx m \log_2\left(\frac{SNR}{M}\right) + \frac{1}{\ln 2}\left(\sum_{j=1}^{m}\sum_{p=1}^{n-1}\frac{1}{p} - \gamma m\right) \qquad b/s/Hz \tag{1.51}$$

where $m = \min(M,N)$, $n = \max(M,N)$ and $\gamma \approx 0.5772$ denotes the Euler constant. Thus, the ergodic capacity of the MIMO channel (where h_{ij} are random complex i.i.d variables) is an increasingly linear in function of the SNR (in dB).
The instantaneous capacity can be expressed as a function of the eigenvalues λ_i of the Wishart matrix $\boldsymbol{W}$:

$$C_{CSIR} = \sum_{i=1}^{min(M,N)} \log\left(1 + \frac{P_T}{M\sigma^2}\lambda_i(\boldsymbol{W})\right) \tag{1.52}$$

where

$$\boldsymbol{W} = \begin{cases} \boldsymbol{H}\boldsymbol{H}^H & \text{if } N \leq M \\ \boldsymbol{H}^H\boldsymbol{H} & \text{if} N > M \end{cases}$$

The MIMO system can be decomposed into an equivalent system of $n = \min(M,N)$ SISO subchannels whose path power gains are the eigenvalues of $\boldsymbol{W}$. Thus, the concept of effective degrees of freedom (EDOF) was introduced in (15). this concept represents the number of subchannels actively participating in conveying information under a given set of operating conditions. It is well known that for an SISO channel, at high SNR, a $G-fold$ increase in the transmitter power results in an increase in the channel capacity of $\log_2 G$ $b/s/Hz$. If a system is equivalent to EDOF SISO channels in parallel, the capacity of the system should increase by $(EDOF.\log_2 G)$ $b/s/Hz$ when the transmitter power is raised by a factor of G . In light of this, we define EDOF at a given SNR

$$\begin{aligned} EDOF &= \frac{d}{d\delta} C\left(2^\delta \cdot SNR\right)\Big|_{\delta=0} \\ &= \sum_{i=1}^{min(M,N)} \frac{1}{1 + \frac{M}{SNR\cdot\lambda_i}} \end{aligned} \tag{1.53}$$

The value of EDOF depends on correlation between different channel matrix components. For example, for a channel with a strong Line Of Sight path, EODF will tend to 1. In other hand, for a NLOS channel without spatial correlation, EDOF will tend to $min(M,N)$.

But, this Shannon capacity is valid for channels governed by a stationary ergodic [4] random process but is typically greater for general channels.

1.3.2 Outage Probability

In reality, the block lengths are finite. The common example is speech transmission. In such cases, we speak of outage capacity. Outage capacity is the capacity that is guaranteed with a certain level of reliability. Non-ergodic channels (i.e., the channel is randomly drawn but remains fixed for the whole channel use) often use the cumulative distribution function (CDF) to indicate the probability that the channel fails to support a given rate threshold. When ergodicity condition is not respected, the maximum rate that can be transmitted without error is random. Thus, the value of capacity using the previous capacity definition is null. As result, the outage probability is introduced and it represents the probability that the mutual information is less than a given rate R.

$$P_{out} = Pr(\boldsymbol{I}(\boldsymbol{x},\boldsymbol{y}) < R) \tag{1.54}$$

The exact value of outage probability is difficult to calculate, some approximations were proposed in (16). But, one can show that if N and M tend to infinity, the capacity $C(\boldsymbol{H})$ tends to a gaussian variable. Let's μ_C and σ_C^2 respectively the average and the variance of $C(\boldsymbol{H})$, an approximation of the outage probability can be given by [5]

$$P_{out} \approx Q\left(\frac{\mu_C - R}{\sigma_C}\right) \tag{1.55}$$

1.3.3 Multiplexing Gain

MIMO systems offer a linear increase in data rate through spatial multiplexing r (7)(17)(18)(19), i.e., transmitting multiple, independent data streams within the bandwidth of operation. Under suitable channel conditions, such as rich scattering in the environment, the receiver can separate the data streams. Furthermore, each data stream experiences at least the same channel quality that would be experienced by a SISO system, effectively enhancing the capacity by a multiplicative factor equal to the number of streams. In general, the number of data streams that can be reliably supported by a MIMO channel equals the minimum of the number of transmit antennas and the number of receive antennas. Thus, the spatial multiplexing gain increases the capacity of a wireless network.
The maximum multiplexing gain r_{max} that can be achieved over a MIMO channel is given by the asymptotic (in SNR) slope of the outage capacity (for fixed Frame Error Rate FER) plotted as a function of the SNR on a linea-log scale, i.e.,

$$r_{max} = \lim_{SNR\to\infty} \frac{C_{out}(SNR)}{\log_2 SNR} \tag{1.56}$$

[4] A stochastic process is said to be ergodic if its statistical properties (such as its mean and variance) can be deduced from a single, sufficiently long sample (realization) of the process.

[5] Formally, the Q-function is defined as $Q(x) = \frac{1}{\sqrt{2\pi}} \int_x^\infty exp\left(-\frac{u^2}{2}\right) du$

For $\boldsymbol{H}$ MIMO channel with optimal transceiver design (i.e., Gaussian code books, asymptotically large frame length, maximum-likelihood detection, ect..) $r_{max} = \min(M,N)$ indicating that for a fixed FER, the transmission rate may be increased by $\min(M,N)$ b/s/Hz for every 3dB increase in SNR.
For a high SNR and for a fixed rate R, we have :

$$R = r \log SNR, \qquad where \qquad 0 \leq r \leq min(M,N) \tag{1.57}$$

1.3.4 Diversity Multiplexing Tradeoff

A diversity gain $d^*(r)$ is obtained for a given multiplexing gain r. This diversity gain verifies :

$$d^*(r) = -\lim_{SNR \to +\infty} \frac{\log P_{out}(r \cdot \log SNR)}{\log SNR} \tag{1.58}$$

The curve of $d^*(r)$ represents the diversity-multiplexing tradeoff. Similarly, it's possible to characterize a diversity-multiplexing tradeoff for a space time coding scheme by just substituting the outage probability P_{out} by the average error probability P_e.
A space time codes family indexed on the SNR reaches a multiplexing gain r and a diversity gain d_C if :

$$d_C = -\lim_{SNR \to +\infty} \frac{\log P_e(r \cdot \log SNR)}{\log SNR} \tag{1.59}$$

For the i.i.d. Rayleigh-flat-fading channel, the optimal tradeoff turns out to be very simple for most system parameters of interest. Consider a slow-fading environment in which the channel gain is random but remains constant for a duration of l symbols. In (20), authors show that as long as the block length $l \geq M+N-1$, the optimal diversity gain $d^*(r)$ achievable by any coding scheme of block length l and multiplexing gain r (r integer) is precisely :

$$d^*(r) = (M-r)(N-r) \tag{1.60}$$

This suggests an appealing interpretation : out of the total resource of M transmit and N receive antennas, it is as though r transmit and r receive antennas were used for multiplexing and the remaining $M-r$ transmit and $N-r$ receive antennas provided the diversity. It should be observed that this optimal tradeoff does not depend on l as long as $l \geq M+N-1$; hence, no more diversity gain can be extracted by coding over block lengths greater than $M+N-1$ than using a block length equal to $M+N-1$. Thus, higher diversity comes at the price of sacrificing spatial multiplexing gain. The maximum diversity gain is obtained for $r = 0$:

$$d^*(r) = (M-r)(N-r) \tag{1.61}$$

In the figure (1.9) the diversity-multiplexing tradeoff is represented. Conceiving space time codes should lead to a diversity multiplexing tradeoff $d_C(r)$ near to the optimal channel tradeoff $d^*(r)$ (21).

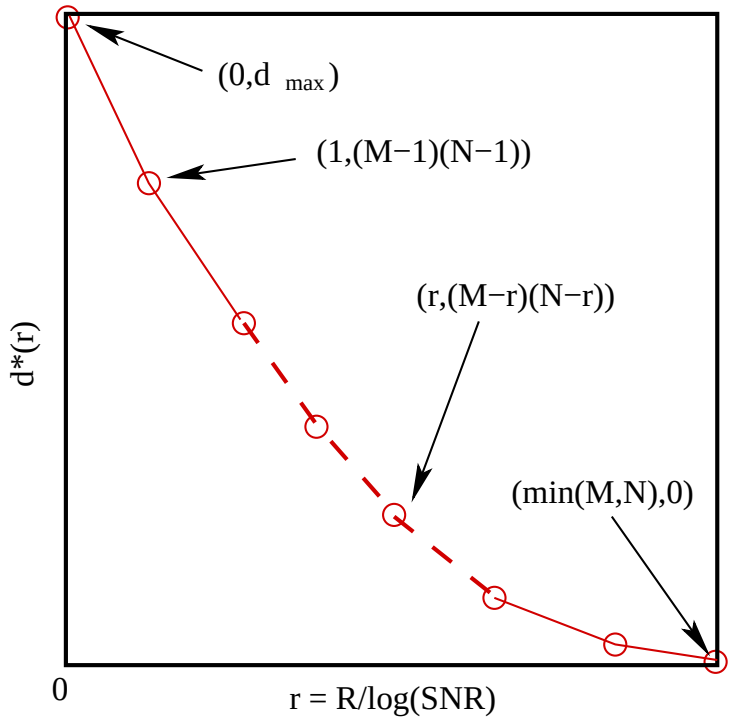


Figure 1.9 – Diversity Multiplexing Tradeoff for a Rayleigh Channel with M transmit and N receive antennas

1.4 Space Time Codes

Space-time codes aim to take advantage of the enormous potential of MIMO systems by combining space and time diversity. They have received considerable attention in academic and industrial circles due to their many advantages. The ST coding permits to introduce dependency between time and spatial domains. ST codes may be split into two main types : Space-time trellis codes (STTCs) (22) (23) and Space-time block codes (STBCs)(24). The space-time trellis codes are based on the generalization of trellis coded modulations (TCM) for MIMO systems. The joint decoding technique is usually based on the use of Viterbi's algorithm. However, identification of the most successful codes required an exhaustive search whose complexity quickly becomes prohibitive when the number of states of the encoder increases. In the following, we will only introduce Space-time block codes which are more interesting in practice.

1.4.1 Space Time Codes Construction Criteria

Let's the space-time codeword matrix $\boldsymbol{X}$ of dimensions $M \times T$ represented as

$$\boldsymbol{X} = \begin{bmatrix} x_1^1 & x_2^1 & \cdots & x_T^1 \\ x_1^2 & x_2^2 & \cdots & x_T^2 \\ \vdots & \vdots & \ddots & \vdots \\ x_1^M & x_2^M & \cdots & x_T^M \end{bmatrix}. \tag{1.62}$$

The t^{th} column $\boldsymbol{x}_t = \left[x_t^1, x_t^2, \cdots, x_t^M\right]^T$ is the symbols vector transmitted at the t instant throughout the M antennas.
We define the Pairwise error probability $\boldsymbol{Pr}\left(\boldsymbol{X}, \hat{\boldsymbol{X}}\right)$ as the probability of detecting one codeword $\hat{\boldsymbol{X}} = [\hat{\boldsymbol{x}}_1, \hat{\boldsymbol{x}}_2, \cdots, \hat{\boldsymbol{x}}_T]$ when another (different) codeword $\boldsymbol{X} = [\boldsymbol{x}_1, \boldsymbol{x}_2, \cdots, \boldsymbol{x}_T]$ is transmitted. The "pair" is the transmitted codeword and the detected codeword. In (25), Tarokh *et al.* showed that this pairwise error probability verified

$$\boldsymbol{Pr}\left(\boldsymbol{X}, \hat{\boldsymbol{X}}\right) \leq \exp\left(-\frac{E_s}{4N_0} d_h^2\left(\boldsymbol{X}, \hat{\boldsymbol{X}}\right)\right), \tag{1.63}$$

where E_s is the energy per symbol transmitted by each antenna and N_0 is the power spectral density of the AWGN. The euclidian distance $d_h^2\left(\boldsymbol{X}, \hat{\boldsymbol{X}}\right)$ between the two codeword matrices is defined by

$$d_h^2\left(\boldsymbol{X}, \hat{\boldsymbol{X}}\right) = \|\boldsymbol{H}\left(\boldsymbol{X} - \hat{\boldsymbol{X}}\right)\|^2 = \sum_{t=1}^{T}\sum_{j=1}^{N}\left|\sum_{i=1}^{M} h_{ji}(\hat{x}_t^i - x_t^i)\right|^2. \tag{1.64}$$

We define the difference between the matrices by $\boldsymbol{B}\left(\boldsymbol{X}, \hat{\boldsymbol{X}}\right)$

$$\begin{aligned} \boldsymbol{B}\left(\boldsymbol{X}, \hat{\boldsymbol{X}}\right) &= \boldsymbol{X} - \hat{\boldsymbol{X}} \\ &= \begin{bmatrix} x_1^1 - \hat{x}_1^1 & x_2^1 - \hat{x}_2^1 & \cdots & x_T^1 - \hat{x}_T^1 \\ x_1^2 - \hat{x}_1^2 & x_2^2 - \hat{x}_2^2 & \cdots & x_T^2 - \hat{x}_T^2 \\ \vdots & \vdots & \ddots & \vdots \\ x_1^M - \hat{x}_1^M & x_2^M - \hat{x}_2^M & \cdots & x_T^M - \hat{x}_T^M \end{bmatrix}. \end{aligned} \tag{1.65}$$

The distance matrix of the two codewords is defined by

$$\boldsymbol{A}\left(\boldsymbol{X}, \hat{\boldsymbol{X}}\right) = \boldsymbol{B}\left(\boldsymbol{X}, \hat{\boldsymbol{X}}\right)\left(\boldsymbol{B}\left(\boldsymbol{X}, \hat{\boldsymbol{X}}\right)\right)^H. \tag{1.66}$$

For a slow fading channel, matrix coefficients stay invariant during the T symbol periods, i.e $h_{ij}^t = h_{ij}\ \forall t \in 1,2,\ldots,T$. Thus, the euclidian distance between the two codewords can be expressed as

$$d_h^2\left(\boldsymbol{X}, \hat{\boldsymbol{X}}\right) = \sum_{j=1}^{N} \boldsymbol{h}_j \boldsymbol{A}\left(\boldsymbol{X}, \hat{\boldsymbol{X}}\right) \boldsymbol{h}_j^H, \tag{1.67}$$

where $\boldsymbol{h}_j = [h_{j,1}, h_{j,2}, \ldots, h_{j,M}]$. Then, the pairwise error probability can be upper bounded by

$$\begin{aligned} \boldsymbol{Pr}\left(\boldsymbol{X}, \hat{\boldsymbol{X}}\right) &\leq \prod_{j=1}^{N} \exp\left(-\frac{E_s}{4N_0} \boldsymbol{h}_j \boldsymbol{A}\left(\boldsymbol{X}, \hat{\boldsymbol{X}}\right) \boldsymbol{h}_j^H\right), \\ &\leq \prod_{j=1}^{N} \exp\left(-\frac{E_s}{4N_0} \sum_{j=1}^{M} \lambda_i |\beta_{ij}|^2\right) \end{aligned} \tag{1.68}$$

where λ_i is the i^{th} eigenvalue of $\boldsymbol{A}\left(\boldsymbol{X}, \hat{\boldsymbol{X}}\right)$ and $\beta_{ij} = \boldsymbol{h}_j \boldsymbol{v}_i^H$, the vector $\boldsymbol{v}_i$ is the i^{th} eigenvector of $\boldsymbol{A}\left(\boldsymbol{X}, \hat{\boldsymbol{X}}\right)$.

In the case of Rayleigh fading channel, the upper bound of the pairwise error probability is obtained by averaging $\boldsymbol{Pr}\left(\boldsymbol{X}, \hat{\boldsymbol{X}}\right)$ over $\boldsymbol{H}$. The inequality (1.68) becomes

$$\boldsymbol{Pr}\left(\boldsymbol{X}, \hat{\boldsymbol{X}}\right) \leq \left(\prod_{i=1}^{r} \left(1 + \lambda_i \frac{E_s}{4N_0}\right)\right)^{-N}, \qquad \text{where} \qquad r = rank\left(\boldsymbol{A}\left(\boldsymbol{X}, \hat{\boldsymbol{X}}\right)\right). \tag{1.69}$$

For a high SNR :

$$\boldsymbol{Pr}\left(\boldsymbol{X}, \hat{\boldsymbol{X}}\right) \leq \left(\prod_{i=1}^{r} \lambda_i\right)^{-N} \left(\frac{E_s}{4N_0}\right)^{-rN} = \left(G_c \frac{E_s}{4N_0}\right)^{-G_d} \tag{1.70}$$

with

$$G_c = \left(\prod_{i=1}^{r} \lambda_i\right)^{1/r}, \tag{1.71}$$

and

$$G_d = rN. \tag{1.72}$$

G_d is the space time code diversity gain. Graphically, it's the slope of the error probability in function of the SNR (in dB and for high SNR). The coding gain G_c can be viewed as an horizontal translation of the error probability curve (in function of SNR). Increasing diversity and coding gains reduce the pairwise error probability. Thus, we conclude that we can define two criteria for the construction of space time codes :

• **The rank criterion :** consists of maximizing the rank of the matrix $\boldsymbol{A}\left(\boldsymbol{X}, \hat{\boldsymbol{X}}\right)$ among all pairs of codewords.

• **The determinant criterion :** consists of maximizing the minimum determinant of $\boldsymbol{A}\left(\boldsymbol{X}, \hat{\boldsymbol{X}}\right)$among all pairs of codewords.

If the diversity gain rN is high (i.e $rN \geq 4$), the pairwise error probability can be upper bounded by (26)(27) :

$$\boldsymbol{Pr}\left(\boldsymbol{X}, \hat{\boldsymbol{X}}\right) \leq \frac{1}{2} \exp\left(-N \frac{E_s}{4N_0} \sum_{i=1}^{r} \lambda_i\right). \tag{1.73}$$

A new construction criterion is then introduced to minimize the pairwise error probability.

• **The trace criterion :** consists of maximizing the minimum trace $\sum_{i=1}^{r} \lambda_i$ of the matrix $\boldsymbol{A}\left(\boldsymbol{X}, \hat{\boldsymbol{X}}\right)$ among all pairs of codewords.

1.4.2 Space Time Block Codes

Space time codes have been proposed in the literature as an efficient means for improving the data rates over fading MIMO channels (28)(29)(30)(31)(32)(33). The most famous example was discovered by Alamouti (34). The Alamouti code is the simplest of all the STBCs. It was designed for a two transmit antenna system and has the coding matrix

$$\boldsymbol{X} = \begin{bmatrix} x_1 & x_2 \\ x_2^* & x_1^* \end{bmatrix}. \tag{1.74}$$

where * denotes complex conjugate.
For the alamouti code, the maximum likelihood detection can be reduced to a simple Zero Forcing decoding. Thanks to this simplicity in implementation, Alamouti code was adopted in wireless standards such as UMTS, IEEE 802.11n, and IEEE 802.16. In addition, the alamouti code is the only complex orthogonal space time code reaching the maximum diversity equal to 2 with a rate $R = 1 \quad symbol/cu$
From 1999, Tarokh et *al.* presented a generalization of the Alamouti code for any number of antennas at the transmitting side and one antenna at the reception side (32)(35). Unfortunately, these constructions are penalized by their rates lower than 1 $\quad symbol/cu$. To reach a rate equal to 1 for more than two antennas, it's necessary to sacrifice the orthogonality property (36) (37) (38)

1.4.3 Linear Dispersion Space Time Codes

Hassibi and Hochwald proposed a new concept Linear Dispersion Codes in (39). In fact, space time block codes are conceived to satisfy construction criteria of Tarokh *et al.* in order to take advantage of spatial diversity at the expense of spectral efficiency.
Linear Dispersion Codes optimize the mutual information but also with benefiting from spatial diversity of MIMO channel. In addition, these codes structure permits to benefit from decoding simplicity.
Space time codes with linear dispersion spread a vector of ℓ complex symbols $[x_1,x_2,\ldots,x_\ell]$ belonging to a linear constellation (q-QAM and q-PSK) over time and space. The codeword matrix verifies

$$\boldsymbol{X} = \sum_{q=1}^{\ell} \left(Re(x_q)\boldsymbol{A_q} + \sqrt{(-1)} Im(x_q)\boldsymbol{B_q} \right), \tag{1.75}$$

where $\boldsymbol{X}$, $\boldsymbol{A_q}$ and $\boldsymbol{B_q} \in \mathbb{C}^{M\times T}$. Linear Dispersion codes family is very big and includes many existing space time codes. Thus, spatial multiplexing scheme can be considered as a linear dispersion code with $T = 1$ and $\ell = M$ and can be formulated as

$$\boldsymbol{X}_{SM} = \sum_{q=1}^{\ell} \left(Re(x_q)\boldsymbol{e_q} + \sqrt{(-1)} Im(x_q)\boldsymbol{e_q} \right), \qquad where \quad \boldsymbol{A_q} = \boldsymbol{B_q} = \boldsymbol{e_q}. \tag{1.76}$$

Similarly, the Alamouti code can be written as

$$\boldsymbol{X}_{Ala} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} Re(x_1) + \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix} Im(x_1) + \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix} Re(x_2) + \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} Im(x_2) \tag{1.77}$$

To combine robustness criterion and mutual information maximization criterion, Sandhu and Paulraj proposed a new construction method (40). Later on, in (41), Gohary and Davidson provide a design technique for Linear Dispersion codes that generates codes which enable large capacities and perform well when decoded with a standard suboptimal detector. Their design technique is motivated by the observation that for an independent Rayleigh fading channel, as the number of transmit antennas grows, Linear Dispersion codes with a certain orthogonal structure simultaneously approach a maximized upper bound on the capacity, and a minimized lower bound on a certain mean square error performance measure. Recently, a new design scheme, that directly minimizes the block error rate in arbitrary fading statistics and various detection algorithm, was proposed (42). The proposed method can be applied to design the minimum-error-rate Linear Dispersion codes for a variety of detector structures including the maximum-likelihood (ML) detector and several suboptimal detectors. It can also design optimal codes under arbitrary fading channel statistics ; in particular, it can take into account the knowledge of spatial fading correlation at the transmitter and receiver ends.

1.5 Conclusion

In this chapter, we have introduced the MIMO Channel and system modeling. Then, we defined the different types of fading channels. Diversity techniques are then discussed and the huge increase in capacity that can be obtained in rich scattering environment thanks to these techniques is presented. In addition, we presented the diversity-multiplexing tradeoff introduced by Zheng and Tse. Next, we have recalled the fundamental concepts of information theory and we have given an overview of the main classes of space-time techniques recently developed in the literature.
In conclusion, the area of space-time coding and signal processing is new, active and full of challenges. In fact, one of the major issues in implementing a MIMO based system is the very high complexity of the detection algorithm at the receiver.

Chapitre 2

MIMO Decoding

2.1 Introduction

In this chapter, we present the state of the art of the main MIMO decoding algorithms existing in the literature. First, we will discuss the sub-optimal decoders such as the Zero Forcing (ZF) decoder, the Minimum Mean Square Error (MMSE) decoder, feedback decision decoders, etc. Then, we will present optimal MIMO decoders and particularly those based on lattice representation and sequential algorithms. We distinguish two categories : decoders using Pohst strategies like the Sphere Decoder algorithm and the Schnorr-Euchner algorithm and decoders using Dijkstra strategy (ex : best-first-search) like the stack decoder and the Fano decoder. We will show also that this last category of decoders can provide a Complexity-Performance tradeoff.

2.2 From a Channel Model to a lattice design

Lattice theory and coding theory are applied to efficiently encode and decode information in a digital transmission system with multiple antennas. Lattice theory is a powerful mathematical tool to represent the channel geometrically and help us understand its behavior in order to design a good modulator and its corresponding demodulator.

2.3 MIMO Decoders : Basic Principles and Structures

In this thesis, approached MIMO schemes are space time block codes schemes. Thus, it is possible to represent the system as an equivalent non-coded system. Seeking simplicity, we will consider the equivalent non-coded transmission scheme .Then, decoding algorithms that we will detail later stay true for systems using space-time coding and only system dimensions will change.
Let's recall the equation that gives the representation of MIMO system with M transmit and N receive antennas. Let's the codeword matrix $\boldsymbol{X}$ of dimension $M \times T$ and the received signal $\boldsymbol{Y}$ of dimension $N \times T$ verifying :

$$\boldsymbol{Y} = \boldsymbol{H}\boldsymbol{X} + \boldsymbol{W} \tag{2.1}$$

After vectorization, the system can be written as

$$\boldsymbol{y_{eq}} = \boldsymbol{H_{eq}}\boldsymbol{s} + \boldsymbol{w_{eq}} \tag{2.2}$$

where $\boldsymbol{y_{eq}}$ and $\boldsymbol{w_{eq}}$ are column vectors of NT elements obtained from $\boldsymbol{Y}$ and $\boldsymbol{W}$. $\boldsymbol{s}$ is the vector composed of the p symbols encoded by the codeword matrix $\boldsymbol{X}$. The equivalent channel matrix $\boldsymbol{H_{eq}}$ of dimensions $NT \times p$ includes the channel response and the space time coding operation. In the following, in order to simplify notations, we will not mention the index $_{\boldsymbol{eq}}$. Let's also $n = NT$. As a result, the new system dimension is $n \times p$.

We assume a coherent transmission (channel matrix known to the receiver). Then, the system becomes :

$$\boldsymbol{y} = \boldsymbol{H} \cdot \boldsymbol{s} + \boldsymbol{w}, \tag{2.3}$$

The purpose behind MIMO decoding is to find an estimation of the transmitted vector. The optimal decoding is the maximum likelihood (ML) decoding. It consists of finding the closest vector to $\boldsymbol{s}$ minimizing the metric :

$$\hat{\boldsymbol{s}} = \arg\min_{\boldsymbol{s}\in\mathcal{C}_s} \|\boldsymbol{y} - \boldsymbol{H} \cdot \boldsymbol{s}\| \tag{2.4}$$

where $\mathcal{C}_s$ is the set of symbol vectors of the constellation. The ML receiver searches through all the vector constellation for the most probable transmitted signal vector. Analyzing decoders structure, one can deduce that three phases can be distinguished in their construction. Every decoder can include some phases. This will depend on : The required *Complexity-Performance Tradeoff*. Decoders including much phases provide a more optimal estimation but they suffer from a huge complexity.

Phase Zero : Pre-processing

Pre-processing is an optional phase. Given a search problem, pre-processing phase is useful in order to make the most efficient use of Decoding. Preprocessing can be divided into two independent steps : left pre-processing and right pre-processing. This two steps will be detailed later.

Phase One : One First Point

This phase permits to obtain one first estimation. The advantage is to get a quick result but not efficient. Sometimes, the receiver need to get a fast estimation. In this case, one first point phase can be enough since it offers a very low complexity even if it's not optimal. This first point can be enhanced using other phases, but this will make the complexity cost more and more important.

Phase Two : One better Point

Generally, this phase is called : One better point phase, and it comes after phase one. The one first point gotten in phase one will be enhanced to get a more reliable one. This first point is generally used as an initialization for the phase two in order to start a more efficient search for the nearest point. This phase can not be independent from phase one .

2.4 MIMO Decoder Classes

2.4.1 Sub-Optimal MIMO Decoders

2.4.1.1 The ZF Decoder

The ZF receiver is a linear receiver. It behaves like a linear filter $\boldsymbol{F}$ and separates the data streams and thereafter independently decodes each stream. We assume that the channel matrix $\boldsymbol{H}$ is invertible and we estimate the transmitted data symbol vector as

$$\hat{\boldsymbol{s}} = \left(\boldsymbol{H}^H \boldsymbol{H}\right)^{-1} \boldsymbol{H}\boldsymbol{s} = \boldsymbol{H}^{\dagger}\boldsymbol{s} \tag{2.5}$$

where † represents pseudoinverse. Since an inverse of $\boldsymbol{H}$ can only exist if the columns of $\boldsymbol{H}$ are independent, it is assumed that the entries of $\boldsymbol{H}$ are i.i.d. Then

$$\boldsymbol{F_{ZF}} = \left(\boldsymbol{H}^H \boldsymbol{H}\right)^{-1} \boldsymbol{H} \tag{2.6}$$

and

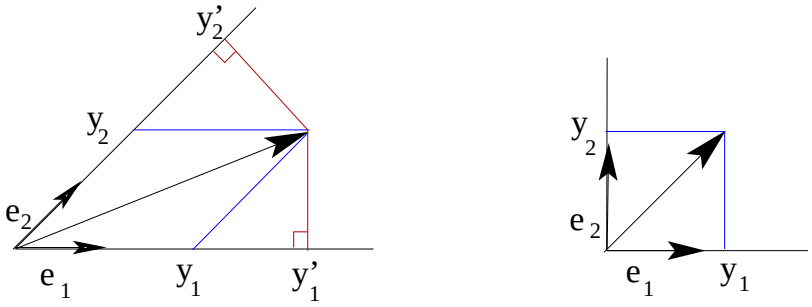
$$\boldsymbol{F_{ZF}} \cdot \boldsymbol{y} = \boldsymbol{s} + \boldsymbol{F_{ZF}} \cdot \boldsymbol{w}. \tag{2.7}$$

Then, a simple detection permits to estimate $\hat{\boldsymbol{s}}$ using a quantification in the QAM constellation thanks to the quantification function $\boldsymbol{Q}_{QAM}$:

$$\hat{\boldsymbol{s}} = \boldsymbol{Q}_{QAM}\left\{\boldsymbol{F_{ZF}} \cdot \boldsymbol{y}\right\} \tag{2.8}$$

The ZF decoding can be viewed as an orthogonal projection of the received vector on the base constituted of the raw vectors of $\boldsymbol{H}$. Figure (2.1) shows an example of projection in dimension 2. If the base is not orthogonal, the projection of $\boldsymbol{y}$ generates a decoding error. If the base is orthogonal, the projection doesn't induce a decoding error and the obtained solution is the ML one.

In practice, the channel matrix is not orthogonal. Many research works in literature permit



(a) random H (b) Orthogonal H

FIGURE 2.1 – Orthogonal Projection of the received vector

to obtain equivalent bases composed of the shortest and the most orthogonal vectors. These

techniques are called "Reduction Techniques" (43). Thus, applying those techniques like the pre-processing followed by the ZF decoding permit to obtain near-ML performances.
At the output of the ZF filter, the resulting noise is $\tilde{\boldsymbol{w}} = \boldsymbol{F_{ZF}} \cdot \boldsymbol{w}$. The covariance matrix is defined by :

$$\boldsymbol{R}_{\tilde{\boldsymbol{w}}\tilde{\boldsymbol{w}}} = E\left[(\tilde{\boldsymbol{w}}) \cdot (\tilde{\boldsymbol{w}})^H\right] = \sigma^2 \left(\boldsymbol{H}^H \cdot \boldsymbol{H}\right)^{-1} = \sigma^2 \boldsymbol{G}^{-1}. \tag{2.9}$$

Consequently, this noise is no longer white, $\boldsymbol{R}_{\tilde{\boldsymbol{w}}\tilde{\boldsymbol{w}}} \neq \boldsymbol{R}_{\boldsymbol{ww}} = \sigma^2 \boldsymbol{I}$. Moreover, if we apply the singular value decomposition (SVD) to the Gram matrix $\boldsymbol{G}$, we get $\boldsymbol{G} = \boldsymbol{U} \cdot \boldsymbol{D} \cdot \boldsymbol{V}^H$, where $\boldsymbol{U}$ and $\boldsymbol{V}$ are unitary matrices and $\boldsymbol{D}$ is a diagonal matrix containing the singular values of $\boldsymbol{G}$. Using the property that the singular values of the Gram matrix are equal to the square of the eigenvalues of $\boldsymbol{H}$, noted λ_1,λ_2,...,λ_n, the covariance matrix of $\tilde{\boldsymbol{w}}$ is given by

$$\boldsymbol{R}_{\tilde{\boldsymbol{w}}\tilde{\boldsymbol{w}}} = \sigma^2 \boldsymbol{V} \cdot \begin{bmatrix} \sqrt{\lambda_1}^{-1} & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & \sqrt{\lambda_n}^{-1} \end{bmatrix} \cdot \boldsymbol{U}^H \tag{2.10}$$

Thus, the well-known problem of the zero forcing is the noise amplification caused by the inverse of the eigenvalues of $\boldsymbol{H}$. These eigenvalues are big for badly conditioned matrix.

2.4.1.2 The Decision Feedback Decoder : ZF-DFE

The general idea of the ZF-DFE decoder is to process the received vector $\boldsymbol{y}$ to estimate the transmitted vector $\boldsymbol{s}$ estimating each component s_k, one at a time, canceling the effect of those symbols already decoded, and nulling those yet unknown. In practice, if a symbol $\hat{s}_k$ is estimated, the decoder exploits this decision to estimate $\hat{s}_{k-1}$, $\hat{s}_{k-2}$,..., $\hat{s}_1$. Thus, this non linear decoder is called a decision feedback decoder (DFE : Decision Feedback Equalization). The ZF-DFE decoder uses the ZF criterion to decode the symbol $\hat{s}_k$. The DFE incorporates a feedforward filter that operates on the received signal to suppress precursor ISI, with a feedback filter that operates on previously detected channel symbols to suppress postcursor ISI. The DFE generally outperforms the traditional linear equalizer. And since it's a successive symbol detection, the QR decomposition is very useful.

$$\begin{aligned} \boldsymbol{y} &= \boldsymbol{H} \cdot \boldsymbol{s} + \boldsymbol{w} \\ &= \boldsymbol{QR} \cdot \boldsymbol{s} + \boldsymbol{w} \end{aligned} \tag{2.11}$$

In order to exploit the upper triangular form of the matrix $\boldsymbol{R}$, we multiply both sides of equation (2.11) on the left by the transpose of $\boldsymbol{Q}$

$$\begin{aligned} \boldsymbol{y}_1 &= \boldsymbol{Q}^T \boldsymbol{y} \\ &= \boldsymbol{R} \cdot \boldsymbol{s} + \boldsymbol{Q}^T \boldsymbol{w} \end{aligned} \tag{2.12}$$

Since $\boldsymbol{R}$ is upper triangular, for the first iteration, the decoder estimates the symbol s_n using the following equation

$$\hat{s}_n = \boldsymbol{Q}_{QAM}\left\{\frac{y_{1,n}}{r_{nn}}\right\} \tag{2.13}$$

To decode the information symbol s_k, the decoder uses the information symbols estimated before $\hat{s}_j$, $j = k+1,\ldots,n$ using this equation

$$\hat{s}_k = \boldsymbol{Q}_{QAM}\left\{\frac{1}{r_{kk}}\left(y_{1,n} - \sum_{j=k+1}^{n} r_{kj} \cdot \hat{s}_j\right)\right\}, 1 \leq k \leq n \tag{2.14}$$

Unfortunately, ZF-DFE performance is hampered by error propagation. Degradation in DFE performance occurs when incorrectly detected symbols are fed through the feedback filter. Then instead of mitigating ISI from the cursor sample, the DFE enhances ISI. Error propagation may result that causes bursts of decision errors and a corresponding increase in the average probability of bit and symbol error.

2.4.1.3 The MMSE Decoder

The ZF receiver eliminates the interference but enhances noise. This might not be significant at high SNR, but at low SNR, it is both sensible and practical to design a filter maximizing the global signal to noise plus interference ratio (SNIR). One possibility is to minimize the total resulting noise, i.e. to find the suitable optimal filter $\boldsymbol{F_{MMSE}}$ minimizing the mean square error :

$$\begin{aligned} \boldsymbol{F_{MMSE}} &= \arg\min_{F} \left(E\left\{\|\hat{\boldsymbol{s}} - \boldsymbol{s}\|\right\}\right) \\ &= \arg\min_{F} \left(E\left\{\|\boldsymbol{F}\cdot\boldsymbol{y} - \boldsymbol{s}\|\right\}\right) \end{aligned} \tag{2.15}$$

Thus, the MMSE optimal filter can be written as :

$$\boldsymbol{F_{MMSE}} = \boldsymbol{H}^H \cdot \left(\boldsymbol{H}^H\boldsymbol{H} + \frac{\sigma^2}{\sigma_s^2}\boldsymbol{I}\right)^{-1}, \tag{2.16}$$

where σ_s^2 represents the average power of the $\boldsymbol{s}$ vector components, i.e $E\left[\boldsymbol{s}\boldsymbol{s}^H\right] = \sigma_s^2\boldsymbol{I}_p$ The MMSE criterion has better performances than zero-forcing for low SNR, but it has one major drawback : it requires the receiver to know the noise variance. Also, for large SNR, MMSE and zero-forcing are equivalent.
The MMSE receiver offers a compromise between interference suppression and reducing noise. For high SNR, the MMSE receiver becomes a ZF receiver. For low SNR, the MMSE receiver becomes similar to an adapted filter :

$$\boldsymbol{F}_{MMSE} = \begin{cases} \boldsymbol{F}_{ZF} & \text{if SNR is high} \\ \frac{\sigma^2}{\sigma_s^2}\boldsymbol{H}^H & \text{if SNR is low} \end{cases}$$

2.4.1.4 Comparison of Sub-Optimal MIMO Decoders

In the figure (3.13), we compare performances and complexities of different sub-optimal decoders that we have presented before. Thus, we consider a 2×2 MIMO system employing two transmit and two receive antennas with a spatial multiplexing technique. We used also a 4-QAM constellation. The considered channel is a quasi-static Rayleigh channel.
This MIMO scheme has a spectral efficiency equal to 4 b/s/Hz. Performances are calculated in terms of BER in function of SNR. The SNR is calculated using the following expression :

$$SNR = 10log_{10}\left(\frac{n\sum_{i=1}^{p} E_{s_i}}{2\sum_{i=1}^{p} log_2\left(q\right) N_0}\right) \qquad \text{dB}$$

where E_{s_i} is the average energy per dimension of complex information symbol belonging to the $q-QAM$ constellation and $\sigma^2 = 2N_0$. In the figure (3.13), we showed also complexities of the sub-optimal decoders in terms of the average number of multiplications per codeword. For all these decoders, operations are matrix operations applied on the received signal and are completely independent of noise variance . This explains constant complexities for all the values of SNR.
Even if these sub-optimal decoders offer low and constant complexities - which is very useful in practical implementation- , they don't allow good performances and they don't take advantages of diversity offered by the MIMO system. In other hand, using sub-optimal decoders can be very interesting if the number of receive antennas is very high compared to the number of transmit antennas because it allows a high receive diversity .
In order to recover the total diversity offered by the MIMO systems and the Space Time Codes, we should focus on the Optimal Decoders.

2.4.2 Lattice MIMO Decoders

ML decoding leads to the best performance in terms of error rate but it is extremely demanding in terms of complexity. For a constellation of size q, ML decoding involves searching over q^p possible candidates. This is affordable when q and p are small, but not for large spectral efficiency systems. The increasing complexity is caused by the search over all possible combinations, although many of them are most probably not the correct candidate : owing to the Gaussian distribution of noise, codewords that are far away from the received vector are much less probable than codewords close to the received vector. Lattice decoding allows for significant reductions in complexity, compared to maximum likelihood (ML) decoding, since 1) it avoids the need for complicated boundary control (44) and 2) It allows for using efficient preprocessing algorithms (e.g., the LLL algorithm (43)) which are known to offer significant complexity reduction. The search for the closest lattice point to a given point has been widely investigated in lattice theory. In general, the optimal search algorithm should exploit the structure of the lattice. For general lattices, that do not exhibit any particular structure, the problem was shown to be NP-hard. A common approach to the general closest-point problem is to identify a certain region within which the optimal lattice point must lie, and then investigate all lattice points in this region, possibly reducing its size dynamically. In general, the development of closest-point algorithms follows two main branches, inspired by two seminal papers : Phost (45) in 1981 examined lattice points lying inside a hyperspher, whereas Kannan (49) in 1983 used a rectangular parallelpiped. Both papers appeared later in revised and extended versions, Pohst's as (48) and Kannan's (following the work of Helfrich (47)) as (46). In (45), however, Pohst proposed an efficient strategy for enumerating all the lattice points within a sphere with a certain radius. Although its worst-case complexity is exponential in q, this strategy has been widely used ever since in closest lattice point search problems due to its efficiency in many useful scenarios (see (52) for a comprehensive review of related works). The Pohst enumeration strategy was first introduced in digital communications by Viterbo and Biglieri (50). In (51) , Viterbo and Boutros applied it to the ML detection of multi-dimensional constellations transmitted over single antenna fading channels, and gave a flowchart of a specific implementation. More recently, Agrell et al. (52) proposed the use of the Schnorr-Euchner refinement (53) of the Pohst enumeration in the closest lattice point search.

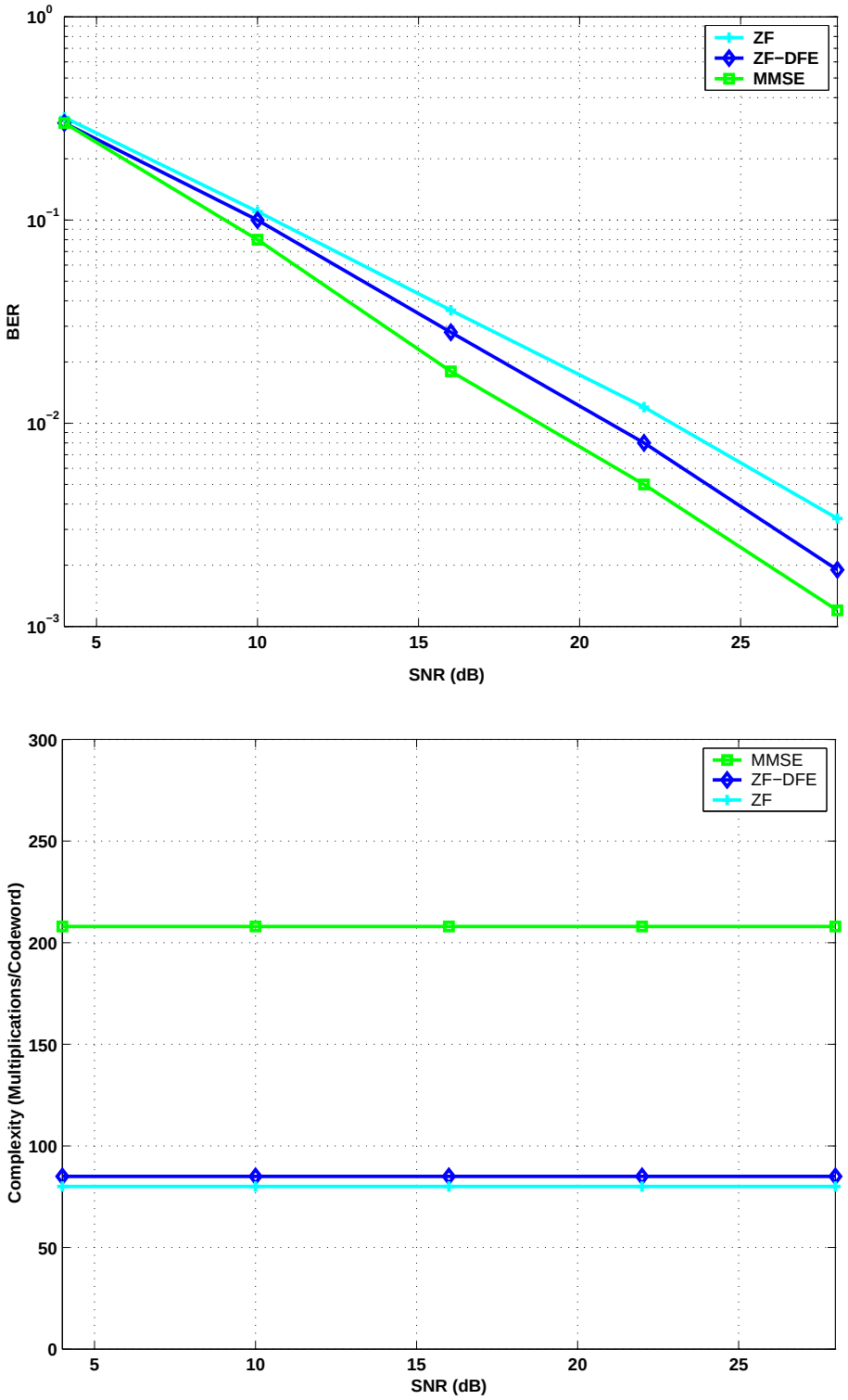


FIGURE 2.2 – Linear and Non-Linear Sub-Optimal Decoders Performance and Complexity Comparison for 2×2 MIMO System using Spatial Multiplexing with a 4-QAM Constellation

2.4.2.1 Sphere Decoding Algorithm

The sphere decoder algorithm was originally developed in the 1980s but has recently attracted much attention in the MIMO community thanks to its similar performance to the exhaustive ML decoder at a reasonable complexity.
The main idea is to limit the search among the possible candidates to those located within a sphere of radius $\sqrt{C}$ centered on the received vector (see figure (2.3)). In this part, we assume a symmetric MIMO system, $M = N$. Applying the mapping from complex valued matrix to real valued matrix of equation (2.3) as described in equations (1.13) and (1.14), we get

$$\underline{\boldsymbol{y}} = \underline{\boldsymbol{H}} \cdot \underline{\boldsymbol{s}} + \underline{\boldsymbol{w}}. \tag{2.17}$$

Then, we consider the QR decomposition of the matrix $\underline{\boldsymbol{H}} = \boldsymbol{Q}\boldsymbol{R}$. After multiplication of both sides of equation (2.17) by $\boldsymbol{Q}^T$, the system becomes

$$\begin{aligned} \boldsymbol{y_1} &= \boldsymbol{Q}^T \cdot \underline{\boldsymbol{y}} \\ &= \boldsymbol{R} \cdot \underline{\boldsymbol{s}} + \boldsymbol{w_1}. \end{aligned} \tag{2.18}$$

$\boldsymbol{Q}$ is orthogonal, and the multiplication by $\boldsymbol{Q}^T$ does not modify the previous system. The system is of dimension $2n$ since $M = N$ and since we have passed to the real domain representation. Now, finding the closest point inside the sphere is equivalent to resolving the following inequality

$$\min_{\underline{\boldsymbol{s}} \in C_{\underline{\boldsymbol{s}}}} \|\boldsymbol{y_1} - \boldsymbol{R} \cdot \underline{\boldsymbol{s}}\|^2 \leq C \tag{2.19}$$

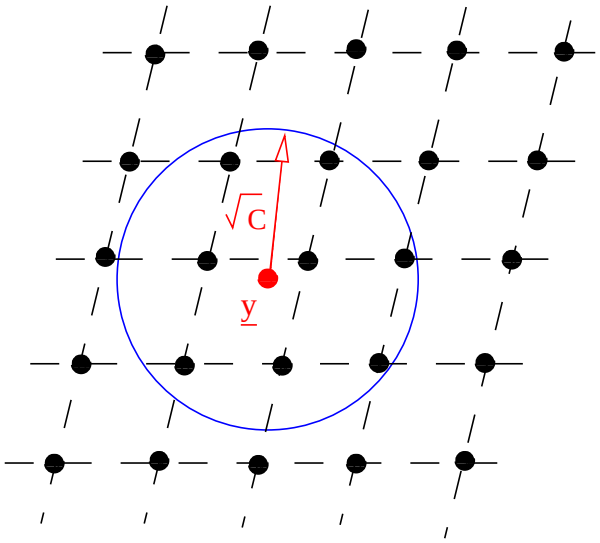


Figure 2.3 – Sphere Decoding

Let's the two vectors $\boldsymbol{\rho}$ and $\boldsymbol{\xi}$ defined as :

$$\boldsymbol{\rho} = \boldsymbol{R}^{-1} \cdot \boldsymbol{y}_1, \quad (\rho_1, \ldots, \rho_{2n}) \in \mathbb{R}^{2n}$$

$$\boldsymbol{\xi} = \boldsymbol{\rho} - \underline{\boldsymbol{s}}, \quad (\xi_1, \ldots, \xi_{2n}) \in \mathbb{R}^{2n}$$

$\boldsymbol{\rho}$ is the ZF point, it represents the coordinates of $\boldsymbol{y}_1$ in the lattice $\Lambda_{\boldsymbol{R}}$ and $\xi_i, \quad 1 \leq i \leq 2n$ are the coordinates of the vector $\underline{\boldsymbol{s}}$ in the new reference. By Substituting in the previous inequality, we get

$$\|\boldsymbol{R} \cdot \boldsymbol{\xi}\|^2 = \sum_{i=1}^{2n} \left(r_{ii}\xi_i + \sum_{j=1}^{2n} r_{ij}\xi_j \right)^2 \leq C \tag{2.20}$$

In the new coordinate system defined by $\boldsymbol{\xi}$, the sphere with radius $\sqrt{C}$ centered at the received point is transformed to an ellipsoid centered at the origin. In order to simplify the inequality (2.21), we note

$$q_{ii} = r_{ii}^2 \quad , \quad i = 1, \ldots, 2n$$

$$q_{ij} = \frac{r_{ij}}{r_{ii}} \quad , \quad j = i+1, \ldots, 2n$$

Then, the previous inequality becomes :

$$\|\boldsymbol{R} \cdot \boldsymbol{\xi}\|^2 = \sum_{i=1}^{2n} q_{ii} \left(\xi_i + \sum_{j=i+1}^{2n} q_{ij}\xi_j \right)^2 \leq C. \tag{2.21}$$

In order to determine the limits of the ellipsoid, it will be judicious to process first ξ_{2n} then the ξ_i $i = 2n-1, \ldots, 1$. Thus,

$$q_{2n,2n}\xi_{2n}^2 \leq C \leftrightarrow -\frac{\sqrt{C}}{\sqrt{q_{2n,2n}}} \leq \xi_{2n} \leq \frac{\sqrt{C}}{\sqrt{q_{2n,2n}}} \tag{2.22}$$

And since $\xi_{2n} = \rho_{2n} - \underline{s}_{2n}$, we can determine using the inequality (2.22) the interval to which belongs $\underline{s}_{2n}$

$$\left\lceil -\frac{\sqrt{C}}{\sqrt{q_{2n,2n}}} + \rho_{2n} \right\rceil \leq \underline{s}_{2n} \leq \left\lfloor \frac{\sqrt{C}}{\sqrt{q_{2n,2n}}} + \rho_{2n} \right\rfloor, \tag{2.23}$$

where $\lceil x \rceil$ is the smallest integer bigger than x and $\lfloor x \rfloor$ is the biggest integer less than x. Given the interval of $\underline{s}_{2n}$, we can determine the interval of $\underline{s}_{2n-1}$:

$$\left\lceil -\sqrt{\frac{C - q_{2n,2n}\xi_{2n}^2}{q_{2n-1,2n-1}}} + \rho_{2n-1} + q_{2n-1,2n}\xi_{2n} \right\rceil \leq \underline{s}_{2n-1} \leq \left\lfloor \sqrt{\frac{C - q_{2n,2n}\xi_{2n}^2}{q_{2n-1,2n-1}}} + \rho_{2n-1} + q_{2n-1,2n}\xi_{2n} \right\rfloor, \tag{2.24}$$

Similarly, we have for the i^{th} component :

$$\left\lceil -\sqrt{\frac{1}{q_{ii}} \left(C - \sum_{l=i+1}^{2n} q_{ll}(\xi_l + \sum_{j=l+1}^{2n} q_{lj}\xi_j)^2 \right)} + \rho_i + \sum_{j=i+1}^{2n} q_{ij}\xi_j \right\rceil \leq \underline{s}_i$$

$$\underline{s}_i \leq \left\lfloor \sqrt{\frac{1}{q_{ii}} \left(C - \sum_{l=i+1}^{2n} q_{ll}(\xi_l + \sum_{j=l+1}^{2n} q_{lj}\xi_j)^2 \right)} + \rho_i + \sum_{j=i+1}^{2n} q_{ij}\xi_j \right\rfloor \tag{2.25}$$

In order to simplify the inequality (2.25), we assume :

$$\begin{aligned} S_i &= \rho_i + \sum_{j=i+1}^{2n} q_{ij}\xi_j \\ T_i &= C - \sum_{l=1}^{2n} q_{ll}(\xi_l + \sum_{j=l+1}^{2n} q_{lj}\xi_j)^2 \\ &= T_{i-1} - q_{ii}(S_i - \underline{s}_i)^2 \end{aligned} \tag{2.26}$$

We can define bounds for each interval $I_i = [b_{inf,i}, b_{sup,i}]$ corresponding to the component $\underline{s}_i$ as :

$$b_{inf,i} = \left\lceil -\sqrt{\frac{T_i}{q_{ii}}} + S_i \right\rceil \leq \underline{s}_i \leq \left\lfloor \sqrt{\frac{T_i}{q_{ii}}} + S_i \right\rfloor = b_{sup,i} \tag{2.27}$$

Starting from the $(2n)^{th}$ order, the algorithm calculates all the components. As a result, the algorithm found a first candidate $\underline{s}^1 = (\underline{s}_{2n}, \underline{s}_{2n-1}, \underline{s}_{2n-2}, \ldots, s_1)$. The distance between this point and the received one is given by :

$$d^2 = C - T_1 + q_{11}(S_1 - \underline{s}_1)^2$$

The algorithm is then restarted by reducing the radius of the sphere to this distance.
The algorithm defines for each component $\underline{s}_i$ of the vector $\underline{\boldsymbol{s}}$ an interval I_i. In order to find the closest point, the algorithm visits the components of the interval I_i and the distance d^2 is evaluated for each visited combination of s_i. Each point verifying $d^2 \leq C$ is stored. And the search continues until all points belonging to the sphere are visited.
Each time a point is found inside the sphere, the algorithm updates the sphere radius and then the bounds $b_{inf,i}$ and $b_{sup,i}$, $i = 1, \ldots, 2n$. This induces a significant decrease of complexity for the search phase since each time we restart a search in a sphere with a smaller radius.
However, we note that the complexity depends on the choice of the initial radius. Clearly, if $\sqrt{C}$ is too large, we obtain too many points, but if $\sqrt{C}$ is too small, we obtain no points. Since we manipulate a white gaussian noise, choosing judiciously a radius taking into account the noise can accelerate the search for the closest point. We can claim intuitively that, for low SNR, the received point is deeply affected by the noise. But, for high SNR, the received point is slightly affected by the noise. This observation leads us to choose a large radius for low SNR and small radius for high SNR. For this reason, taking initially a small radius for high SNR limits the search time. In (54), Hassibi *et al.* proposed a method to calculate the optimal radius based on a probability calculation. This radius is a function of the noise variance and the size of the dimension of the lattice.

$$\sqrt{C} = 4 \cdot n \cdot \sigma^2 \tag{2.28}$$

This formula permits to define a sphere centered at the received point such that we can have at least one point inside the sphere with a probability equal to 99%. In our simulations, we will use this formula to calculate the initial radius.
The sphere Decoder permits to enumerate points inside the sphere and optimizes the search for the closest point. Nevertheless, it doesn't guarantee that these points belong to the used constellation. The algorithm given before is available when infinite lattices are used, but in our transmission scheme the information symbols belong generally to QAM constellations. Therefore,

there is no longer an infinite lattice but a finite lattice constellation, which means a finite subset of a lattice. Consequently the vector found by the SD must imperatively belong to this constellation. There are two ways to proceed :
1) Checking over the whole lattice, and keeping only vectors belonging to the constellation.
2) Seeking lattice constellation directly by checking only vectors belonging to the constellation. The first method is more expensive than the second in terms of operations. In our simulations we apply the second method. The second method consists of imposing a constraint on the search interval I_i so that only points belonging to the constellation will be visited. This is equivalent to calculating boundaries of the interval I_i taking into account the constellation boundaries. For a $q-QAM$ constellation, symbols $\underline{s}_i$ belong to the interval $I_c = \left[\pm 1, \pm 3, \ldots, \pm\left(\sqrt{q}-1\right)\right]$. Thus, the search interval that should be considered is the intersection of I_i and I_c. I_c is a finite subset of $2\mathbb{Z}+1$. In order to bring back the search to $\mathbb{Z}$, we can consider the following transformation :

$$u_i = \frac{\underline{s}_i + \left(\sqrt{q}-1\right)}{2}. \tag{2.29}$$

For example, for the 16-QAM constellation, the symbols $\underline{s}_i$ belong to the interval $I_{c,2\mathbb{Z}+1} = [\pm 1, \pm 3]$. The new interval is $I_{c,\mathbb{Z}} = [0,3] = \{0,1,2,3\}$. We noticed c_{min} and c_{max} the lower and the upper bounds of this interval. The search for the closest lattice point consists of enumerating points u_i in this interval.

2.4.2.2 Schnorr-Euchner Decoding Algorithm

The Schnorr-Euchner algorithm we are studying here was presented in (52). It was used in cryptography applications. This algorithm has the same principle as the SD : the search for the closest point. This algorithm is based on two stages. The first stage consists in searching the 'Babai point' (BP), which represents a first estimation, but is not necessarily, the closest point. Finding the BP gives us a bound on the error. In the second stage, we modify the BP until the closest point is reached. We zigzag around each BP component in order to build the closest point (unlike the sphere decoder, there is no minimum and maximum bound for each BP component). The time needed to find the closest point is closely related to BP, which means closely related to the SNR. In fact, if the BP is very far from the closest point, i.e for low SNRs, the algorithm takes much more time to converge. However, if the BP is close to the closest point, i.e for high SNRs, the algorithm converges rapidly.

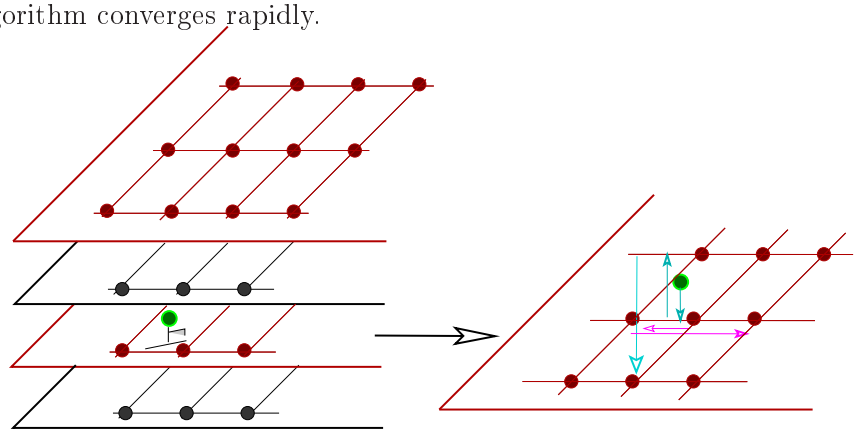


Figure 2.4 – SE Search Strategy

The key idea is to view the lattice as a superposition of hyperplanes and then start the search for the closest point in the nearest hyperplane (see figure 2.4).
Let's recall the equation (2.18). The upper triangular form of $\boldsymbol{R}$ permits to view the lattice as a multi-layered hyperplanes. Thus, the matrix $\boldsymbol{R}$ can be written as

$$\boldsymbol{R} = \begin{bmatrix} \boldsymbol{R_1} \\ \boldsymbol{r_{2n}} \end{bmatrix}, \tag{2.30}$$

where $\boldsymbol{R_1}$ is an $(2n-1) \times 2n$ matrix consisting of the top $2n-1$ rows of the matrix $\boldsymbol{R}$. The matrix $\boldsymbol{R}$ is triangular, the vector $\boldsymbol{r_{2n}} = (0,,0,r_{2n,2n})$ is orthogonal to the space generated by the matrix $\boldsymbol{R_1}$. Now, the search algorithm for the $2n$-dimensional lattice will be described recursively as a finite number of $2n-1$ dimensional search operations. The lattice $\Lambda_{\boldsymbol{R}}$ can be viewed as the infinite superposition of hyperplanes of dimensions $2n-1$ generated by the matrix $\boldsymbol{R_1}$:

$$\Lambda_{\boldsymbol{R}} = \cup \{\boldsymbol{c} + t_{2n}\boldsymbol{r_{2n}} / \boldsymbol{c} \in \Lambda_{\boldsymbol{R_1}}, t_{2n} \in \mathbb{Z}\}. \tag{2.31}$$

A successive projection on the different hyperplanes of the lattice permits to find a first estimation of the closest point. This is the 'Babai point' and it corresponds to the ZF-DFE point (55). Once this point is found, it constitutes the departure point to visit other lattice points. The purpose is to find the closest lattice point, it is then unnecessary to consider points having distance more than that of the 'Babai point'. Thus, the Schnorr-Euchner (SE) is a search algorithm inside a sphere centered at the received point and with initial radius equal to the distance between the received point and the BP.
In the following, we propose to calculate the BP. The projection of $\boldsymbol{y_1}$ onto the vector $\boldsymbol{r_{2n}}$ gives the $(2n)^{th}$ component of the Babai point :

$$\underline{\hat{s}}_{2n} = \frac{y_{1,2n}}{r_{2n,2n}} \tag{2.32}$$

Let's $\underline{s}_{2n} = [\underline{\hat{s}}_{2n}]$ such that $[x]$ is the nearest integer to x. The distance that separates $\boldsymbol{y_1}$ to the layer $2n$ is given by :

$$d_{2n} = |\underline{s}_{2n} - \underline{\hat{s}}_{2n}| \cdot |r_{2n,2n}|. \tag{2.33}$$

Considering the ZF vector, $\boldsymbol{\rho} = \boldsymbol{y_1} \cdot \boldsymbol{R^{-1}}$, we get :

$$\begin{aligned} \underline{\hat{s}}_{2n} &= \rho_{2n} \\ d_{2n} &= |[\rho_{2n}] - \rho_{2n}| \cdot |r_{2n,2n}| \end{aligned}$$

To calculate $\underline{\hat{s}}_{2n-1}$, we can start from this equation

$$\boldsymbol{y_1} = \boldsymbol{R} \cdot \boldsymbol{\rho}. \tag{2.34}$$

Then, we can write :

$$y_{1,2n-1} = \rho_{2n-1} r_{2n-1,2n-1} + \rho_{2n} r_{2n-1,2n}$$

And since the component $\underline{s}_{2n}$ was calculated in the previous layer, we can use this estimation to deduce ρ_{2n-1} :

$$\rho_{2n-1} = \frac{y_{1,2n-1} - r_{2n-1,2n} \cdot \underline{s}_{2n}}{r_{2n-1,2n-1}} \tag{2.35}$$

Similarly to the $(2n)^{th}$ order, the component $\underline{s}_{2n-1}$ can be calculated as :

$$\begin{aligned} \underline{s}_{2n-1} &= [\rho_{2n-1}] \\ d_{2n-1} &= |[\rho_{2n-1}] - \rho_{2n-1}| \cdot |r_{2n-1,2n-1}| \,. \end{aligned}$$

By a successive substitution, we can calculate all the components of the BP :

$$\begin{aligned} \underline{s}_i &= \left[\frac{y_{1,i} - \sum_{j \geq i} r_{ij} \cdot \underline{s}_j}{r_{ii}} \right] \\ d_i &= |[\rho_i] - \rho_i| \cdot \|r_{ii}\| \,. \end{aligned}$$

Once the BP is found, the distance between this point and the received point is :

$$d^2 = \sum_{i=1}^{2n} d_i^2 \tag{2.36}$$

As shown in figure (2.5), the idea of SE algorithm is to zigzag around components of the BP. For every detected point, the distance d^2 is recalculated. If the new distance is less than the previous one, this point is stored and the distance d^2 is updated. Then, the searching phase continues. In the original paper of the SE (52), authors considered a radius initialized to infinity and once the BP is found, the radius is updated and adjusted to the distance d^2. However, in (56), it was shown that if we consider a finite radius calculated with the same manner as for the SD, this permits to accelerate the decoding by 30%. In our simulations, we will consider a finite radius calculated as in equation (2.28).
The SE algorithm described above permits to decode lattices with $\underline{\boldsymbol{s}} \in \mathbb{Z}^{2n}$. In the case of finite constellations, one should check that the final solution belongs to the constellation.

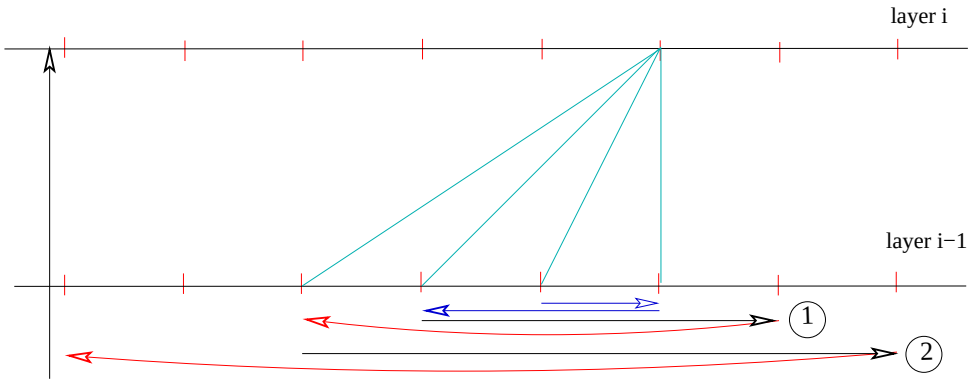


Figure 2.5 – SE zigzag Strategy

Let's $I_c = [c_{min}, c_{max}]$ the interval to which belong the real and the imaginary parts of the symbols constellation. In the algorithm, we should verify that BP components are correctly

chosen from this interval. That is why, we should take $\underline{s}_{2n} = \boldsymbol{Q}_{QAM}(\hat{\underline{s}}_{2n})$ instead of $\underline{s}_{2n} = [\hat{\underline{s}}_{2n}]$ such that :

$$\boldsymbol{Q}_{QAM}(\hat{\underline{s}}_{2n}) = \begin{cases} c_{min} & ,\text{if } [\hat{\underline{s}}_{2n}] < c_{min} \\ [\hat{\underline{s}}_n] & ,\text{if } \quad c_{min} \leq [\hat{\underline{s}}_{2n}] \leq c_{max} \\ c_{max} & ,\text{if } [\hat{\underline{s}}_{2n}] > c_{max} \end{cases} . (2.37)$$

The zigzag around every component of Babai point can probably go outside I_c. One should return to the interval each time it goes outside. Two possible cases can happen :
1) If the point components are inside I_c : Continue the search.
2) If the point components are not inside I_c, this proves that all points were tested. In this case, we should go up to the next component.

2.4.2.3 Comparison of the SE and the SD algorithms

Even if both strategies for visiting points inside the sphere are quite different, they lead to the same maximum likelihood solution. Many studies were realized to evaluate complexity of the two strategies (52)(57) and it was shown that the strategy of SE requires less time than the SD proposed by Viterbo *et al.*.
Here, we will compare the performance of SD and the SE algorithms using simulations. We present in the figure (2.6) performances in terms of BER in function of the SNR for a 4×4 MIMO system with a $4 - QAM$ constellation and using a spatial multiplexing. We will first check that the SD and the SE permit to obtain ML performances. Also compared to sub-optimal decoders, the SD and the SE offer a maximum diversity equal to N.
In the figure (2.6), we compare also complexities in terms of the average number of multiplications per codeword. We have considered here only the search phase. The Pre-decoding phase is similar for both decoders. As a result, we conclude that the complexity of the SD is more important than that of the SE for low SNRs. In fact, the complexity of the SD is especially related to the chose of the sphere radius. If the radius is very small, the decoder does'nt find any point and it will be obliged to restart search with higher radius. However, for the SE, the radius is initially equal to infinity or to a higher value containing at least the ZF-DFE point, the algorithm convergence is then guaranteed from the first iteration. For high SNRs, the complexity of both decoders converge and the complexity of the SD is slightly better than the complexity of the SE. This can be explained by the fact that the ZF-DFE point is far from the ML point as it can be seen through the performance curve of the ZF-DFE decoder. In this case, the SE zigzags around this point which requires more iterations to reach the ML point.

2.4.3 Sequential MIMO Decoders

Other types of decoders are sequential decoders. These decoders were initially introduced to decode binary trellis codes (58). They are based on some search algorithms inside a binary tree where tree branches represent the binary values of the code. Two principle algorithms were specially used : the Fano decoder (59) and the stack decoder (60). In (61), authors rediscovered these decoders and they applied them to MIMO systems. The tree search is no longer a binary tree but constituted of the different possible values of real and imaginary parts of symbols. Nevertheless, decoding main idea keeps the same. To apply tree-search algorithms, we need first to expose the tree structure. A QR or a Cholesky decomposition can be applied on the lattice generator matrix $\underline{\boldsymbol{H}}$. These two methods are quite equivalent, however the QR is more complex than the Cholesky decomposition but it allows a numerical stability (62).

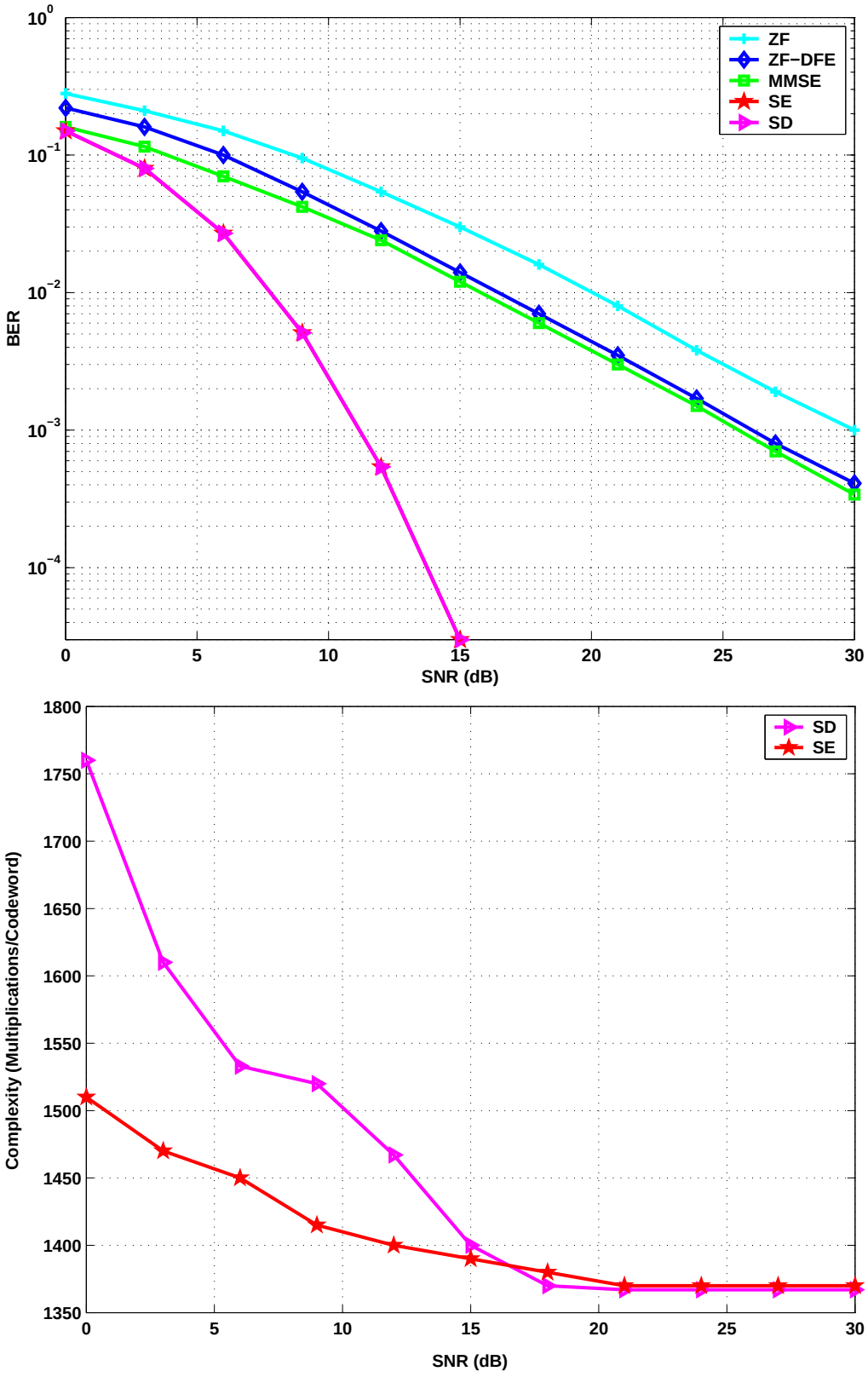


Figure 2.6 – SE and SD Performance and Complexity Comparison

Let's recall the equation (2.18) :

$$\boldsymbol{y_1} = \boldsymbol{R} \cdot \underline{\boldsymbol{s}} + \boldsymbol{w_1}. \tag{2.38}$$

Exploiting the upper triangular form of $\boldsymbol{R}$, one can solve the decoding problem using a tree search algorithm.
We consider a tree rooted at a fictive node ϖ_{root}. The node at level k is denoted by the vector $\underline{\boldsymbol{s}}^{(k)} = (\underline{s}_{2n},\underline{s}_{2n-1},...,\underline{s}_k)$ where $\underline{s}_j, j = 1,\ldots,2n$ are the components of $\underline{\boldsymbol{s}}$. Moreover, the branches of the tree at level k define all the possible values that can be taken by $\underline{s}_k$, and each node $\underline{\boldsymbol{s}}^{(k)}$ is associated with the squared distance

$$f(\underline{\boldsymbol{s}}^{(k)}) = \sum_{i=k}^{2n} f_i(\underline{s}_i), \tag{2.39}$$

where $f_i(\underline{s}_i) = \left| y_{1,i} - \sum_{j=i}^{2n} r_{i,j}\underline{s}_j \right|^2$.

We call $f(\underline{\boldsymbol{s}}^{(k)})$ the cost function of the node $\underline{\boldsymbol{s}}^{(k)}$. It represents the "sub-distance" between the received and the transmitted signal at the level k. As shown in figure (2.7), a node to be expanded is called a father node and its successors are called child nodes.

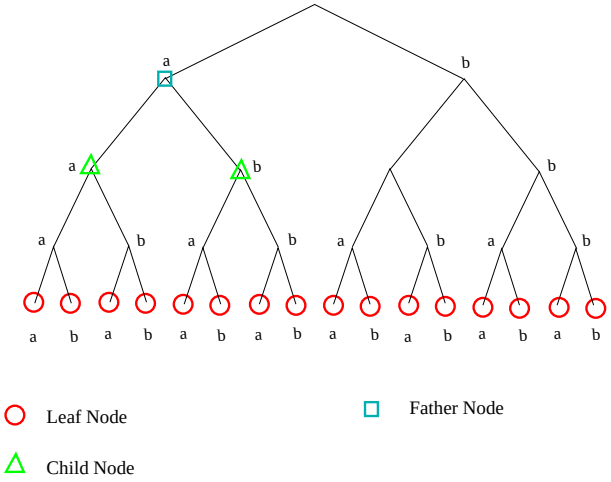


Figure 2.7 – Tree search for a point with tree depth = 4

The tree search consists in exploring the tree nodes in order to find the leaf node $(\underline{s}_{2n},\underline{s}_{2n-1},...,\underline{s}_1)$ with the least cost. In the literature, we find different tree search strategies. In the next paragraph, we will present the most known ones.

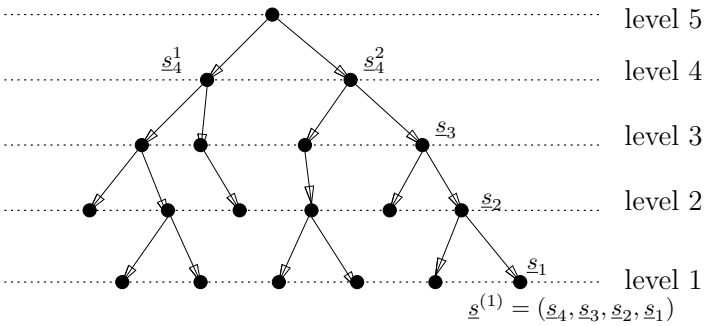


Figure 2.8 – Example of a tree structure with a depth = 4

2.4.3.1 Tree Search Strategies

A - Breadth-First Search(BrFS)

The breadth first search algorithm as given in figure (2.9) is a tree search algorithm that starts from a root node ϖ_{root} and explores all its children, the children of those and so on until it hits the end of the tree. From this point of view, the BrFS is like a tree of generations. It will visit the ancients, then the offspring of those and follow on with the descendants of those, and so forth.

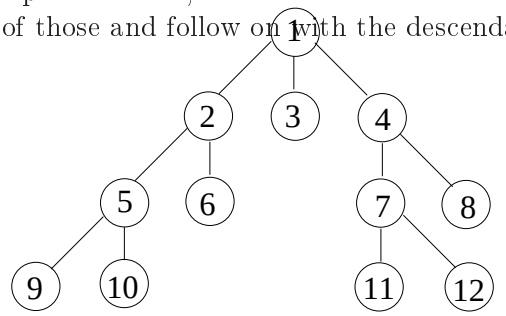


Figure 2.9 – Breadth-First Search Strategy

All we need to do is to store the current level and to increase it as we go deeper and deeper into the tree. The BrFS is then an exhaustive tree search algorithm. It moves from level $k+1$ to the level k until it explores all the nodes in the first one. The solution found is then the ML one. From the standpoint of the algorithm, all child nodes obtained by expanding a node are added to a FIFO queue. The algorithm works by putting the ending node in the queue then pull a node from the beginning of the queue and examine it. If the searched element is found in this node, then the algorithm quit the search and returns result. Otherwise we push all the successors of this node into the end of the queue. If there is a solution breadth first search will find it regardless of the kind of graph. However, if the graph is infinite and there is no solution breadth first search will diverge.

B-Depth-First Search(DeFS)

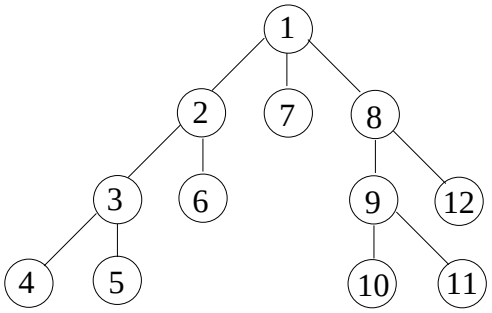


Figure 2.10 – Depth-First Search Strategy

Like the BrFS, our objective is to visit nodes of the tree by starting from a root item and

traveling through edges. The DeFS algorithm as given in figure (2.10) starts from the root node, explores its first child node and proceeds by going deeper and deeper until the end of the tree or until it hits a node that has no children. Then, the algorithm backtracks and returns to the most recent node being expanded and visits its second child and so on. We note that this algorithm needs more memory to 'remember' which nodes having been already visited. However, since it explores all the possible paths in the tree, the DeFS is an exact-ML algorithm. In a non-recursive implementation, all freshly expanded nodes are added to a LIFO stack for expansion. Space complexity of DeFS is much lower than BrFS. It also lends itself much better to heuristic methods of choosing a likely-looking branch. Time complexity of both algorithms are proportional to the number of nodes and the number of edges in the graphs they traverse. When searching large graphs that can be fully contained in memory, DeFS suffers from non-termination when the length of path in the search tree is infinite.

C-Best-First Search(BeFS)

One can see the BeFS as an optimization of the BrFS. In fact, the BeFS strategy aims to find the best path in the tree by expanding only the most promising nodes chosen according to some rule. In general, the BeFS uses a cost function and selects the next node to expand with the best score (the least one). In fact, starting from a given node, the algorithm evaluates first all its successors and selects the one to expand with the best score and so on until finding the final node.

D-Branch and Bound algorithm (BB)

Visiting all the tree nodes to find the one with the shortest path, using one of the three strategies described above, is prohibitively complex. However, this complexity can be reduced using the Branch and Bound algorithm (BB) which comes to establish constraints on the tree search by using a *bounding function* as explained in figure (2.11). This means that the algorithm chooses the nodes to expand by comparing their costs against this function. If the cost node is within the defined bounders, the node will be explored, else the node will be jumped, which allows to limit the expanding of some unnecessary nodes and advantages the most promising ones.
The Sphere Decoder and the Schnorr-Euchner algorithms can be viewed as BB algorithms using a depth-first-search strategy. In fact, they start from the upper level in the tree and first consider one possible value $\tilde{\underline{s}}_{2n}$ inside the bounded region and conducts a depth search over the sub-trees $\{\underline{\boldsymbol{s}}/\underline{s}_{2n} = \tilde{\underline{s}}_{2n}\}$ before going back to another sub-trees $\{\underline{\boldsymbol{s}}/\underline{s}_{2n} \neq \tilde{\underline{s}}_{2n}\}$, and so on.
The sequential decoders, that we study here, conducts also BB algorithms using a BeFS strategy. In the following, we will focus on the most known ones, namely the Fano and the stack algorithms. A description of these sequential decoding algorithms is therefore given.

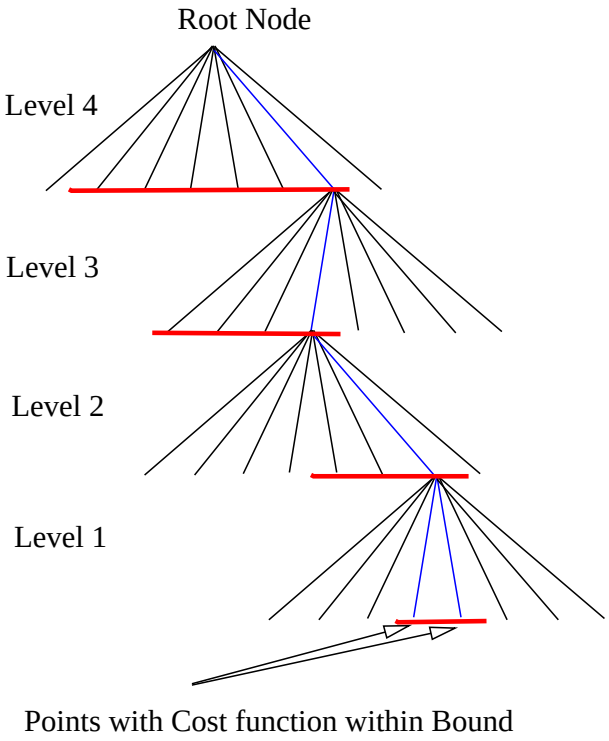


Figure 2.11 – Branch and Bound

2.4.3.2 Fano & Stack Decoding Algorithms

As given before, the sequential decoders were originally proposed to decode binary trellis codes (63). The most used one is the Fano decoder introduced in 1963 (64). Later, Zigangirov proposed in 1966 a sequential algorithm using a stack storage (or memory). In the 1960s, memory allocation represented a very important constraint. Thus, studies were oriented toward discovering the Fano decoder features which were more suitable for hardware implementation and so far the stack decoder has not been widely used. Nowadays, the price of memories is continually dropping and the stack decoder is therefore becoming of great interest.
In (61), the authors have rediscovered the Fano decoder and applied it to decode MIMO schemes. In this work, we will focus on the stack decoder and bring the necessary modifications to decode lattice and finite constellations. Let's first recall the principles of the original Fano and stack algorithms.

A - Fano decoder

Here, we will detail the search strategy of the Fano decoder. Let us suppose that the decoder is at a some node $\underline{\boldsymbol{s}}^{(k)}$ of a level k in the tree. The decoder can choose to proceed forward to a child node at level $k-1$ or to move back to the parent node $\underline{\boldsymbol{s}}^{(k+1)}$. At each step, the decoder looks forward to the best child node. The best node is the one having the least cost. The decoder can visit a node if its metric is smaller than a certain threshold Υ. Note that Υ is allowed to take values only in multiples of the step size Δ(i.e., 0, $\pm\Delta$, $\pm 2\Delta$, ...).

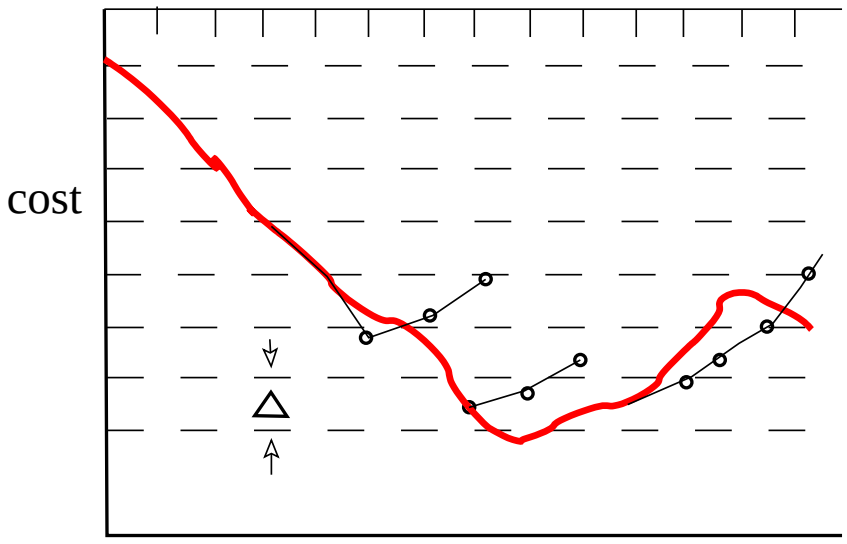


Figure 2.12 – Fano Decoder

Each time a new node is visited by the Fano Decoder for the first time, the threshold Υ is tightened to the least possible value higher than the cost function of the node. If the current node does not have a child node with cost function below the threshold, then the decoder moves back to the parent node if this parent node has a cost function below the threshold as shown in figure (2.12) . Then, the decoder attempts moving forward to the next best node. However, if the decoder cannot move back, the threshold is relaxed and attempt is made to move forward again, proceeding in this way until a leaf node is reached. We note that, from a current node, the Fano algorithm moves either to its parent or to a child but never jumps. For example, if we find the node $\underline{\boldsymbol{s}}^{(k)}$ such that $f(\underline{\boldsymbol{s}}^{(k)}) \leq \Upsilon$. In this case, the constraint is readjusted as follow :

$$\Upsilon = \Upsilon - \delta \cdot \Delta \tag{2.40}$$

where $\delta \in \mathbb{N}$ is the biggest integer such that

$$\Upsilon - (\delta + 1) \cdot \Delta \leq f(\underline{\boldsymbol{s}}^{(k)}) \leq \Upsilon - \delta \cdot \Delta. \tag{2.41}$$

In the original papers focusing on Fano decoders, Δ is usually chosen equal to 1. However, we can choose Δ in $\mathbb{R}$. The value of this parameter can have an influence on the speed of decoding and thus on complexity. The ideal choice is a step with the same order as $f(\underline{\boldsymbol{s}}^{(k)})$.
Based on the equation (2.21), the new constraint is chosen such that the successors nodes will have a less or an equal cost as the visited node. If there is not any node $\underline{\boldsymbol{s}}^{(k-1)}$ verifying this constraint, the algorithm will go upper to the higher level and generate the best $\underline{\boldsymbol{s}}^{(k)}$ except the one visited before. If the algorithm doesn't succeed to find any node, and this means that all nodes have a cost bigger than Υ. In this case Υ is incremented by Δ, $\Upsilon = \Upsilon + \Delta$ and the search algorithm continues. The algorithm ends when a full path is reached. However, since the decoder requires no memory, it may visit the same node several times which induces additional complexity.
Unlike the SD and SE which update the radius once a vector $\underline{\boldsymbol{s}}$ is found - which means a complete path in the tree -, the Fano decoder constraint is readjusted each time an intermediate node $\underline{\boldsymbol{s}}^{(k)}$

is visited. This is due to the difference in strategies between those algorithms.

B - Stack decoder

For the stack algorithm, the main search idea and strategy remain the same. However, the main difference is that the stack stores all the paths crossed by the algorithm in an ordered list called "stack" or "memory", whereas the Fano only retains the best path. In fact, starting from a root node ϖ_{root}, the stack algorithm generates all the children of ϖ_{root}. As given in figure (2.13), the algorithm computes the costs of those nodes according to the cost function given in equation (2.39) and stores them in an increasing order in the memory such that the top node of the stack is the one having the best cost. After that, the algorithm takes the top node, generates its children, computes their costs, places them in the memory and removes the top node being just expanded. The algorithm reorders the stack again, generates the child nodes of the current top node, and so on. Note also that some information concerning each node are also stored with this node, like its cost, its level in the tree, and its path. The algorithm terminates when a path of length $2n$ is found on the top of the stack, in other words, when a leaf node reaches the top of the list. The flowchart of the stack algorithm is presented in figure (2.15), where we define by $Gen(\varpi)$ the function that generates all the children nodes of ϖ and calculates their costs, $Sort(List)$ the function that reorders the nodes stored in the stack, $top(List)$ the function that selects the top of List and $store()$ the function that stores the children in List. In this flowchart, Γ is the set of generated children of the node ϖ. Note that, since the stack decoder stores potential candidates, it visits fewer nodes than the Fano decoder which may visit nodes having already been visited. Consequently, the stack decoder is faster and less complex than the Fano decoder.

One can here highlight a few of the stack decoder properties that are very important in the study of MIMO detection algorithms :
• Nodes are distributed over $2n$ levels, numbered from root node ϖ_{root} at level 0 to leaf nodes at level $2n$. Non-leaf nodes are those at levels 0 through $2n-1$
• Branches and nodes weights are non-negative.
• Each node weight is the sum of branch weights along its path from the root node or equivalently the sum of the weights of its parent node and the connecting branch.
• For any path from the root node to a leaf node the weights are non-decreasing.
Let's now ask this question "For a tree based tree search, what's the smallest number of nodes that should be expanded in order to obtain ML solution ? "
One Obvious remark is : given the weight of the smallest weight leaf node, it's clear that all non-leaf nodes having weights less than this weight must be expanded. Let's denote f_{best} the smallest weight leaf node. If we expand non-leaf nodes whose weights are greater than or equal to f_{best}, this can lead to the discovery of leaf nodes whose weights are also greater than or equal to f_{best}. Therefore, the SE and the SD may expand more nodes than necessary because they rely on a search radius to dictate whether or not a node should be expanded during the enumeration. Even though it may be adaptively reduced if the squared search radius is ever larger than f_{best}, then it's possible for nodes having weights greater than or equal to f_{best} to be expanded. In contrast, the stack decoding is designed to expand precisely the minimum number of nodes necessary to establish an ML solution. It's also able to do so without prior knowledge of the optimal leaf node f_{best}. The stack decoder explores nodes of the tree in order of increasing weight starting from the root. In the sequel, we will propose a modified stack decoder in order to enhance the original one and to reduce its complexity.

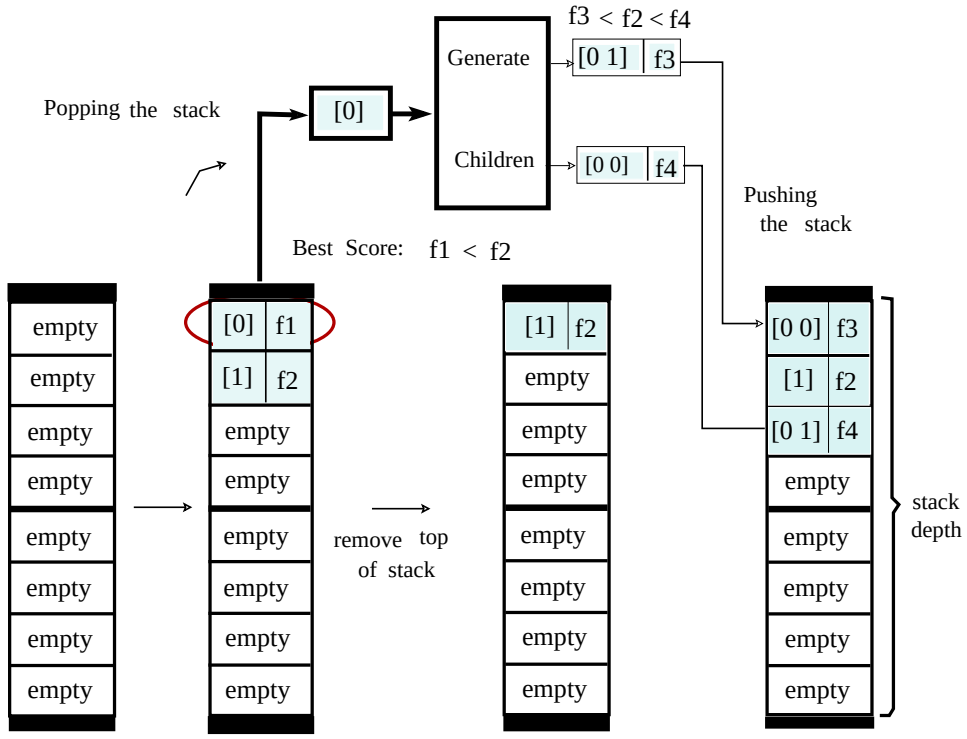


FIGURE 2.13 – Stack management for the Stack Decoding algorithm

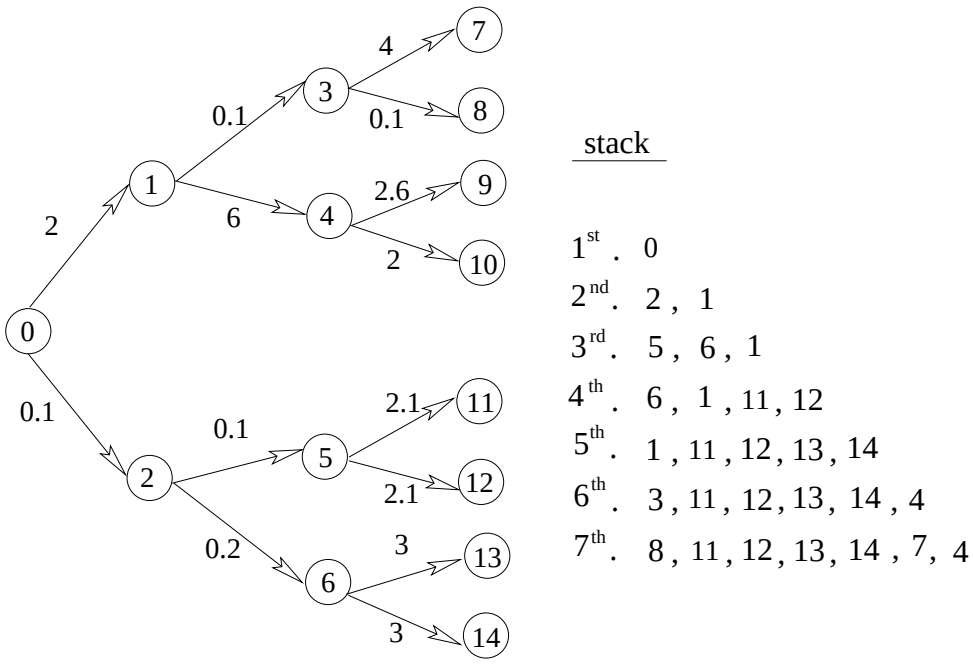


FIGURE 2.14 – Best-First Search Algorithm

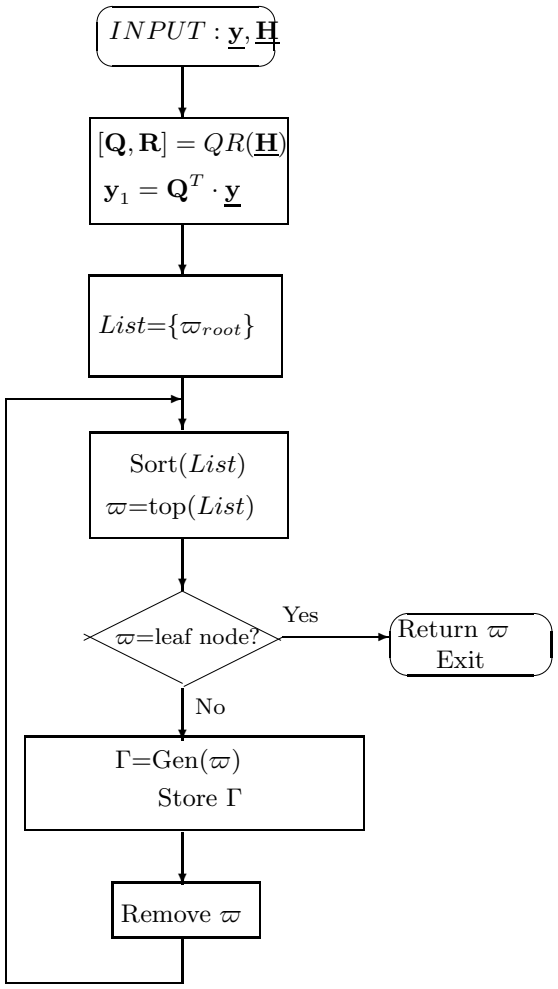


Figure 2.15 – Flowchart of the stack decoder

2.4.3.3 Complexity-Performance Tradeoff : The use of bias

Let's first recall the cost function given in equation (2.39)

$$f(\underline{\boldsymbol{s}}^{(k)}) = \sum_{i=k}^{2n} f_i(\underline{s}_i), \tag{2.42}$$

This gives the cost function of any node stored in memory at any instant. Rewriting this cost function by adding a bias value gives

$$f(\underline{\boldsymbol{s}}^{(k)}) = \sum_{i=k}^{2n} f_i(\underline{s}_i) - b \cdot k, \tag{2.43}$$

where we refer to $b \in \mathbb{R}$ as the bias coefficient. In (61), it was established that the efficiency of the stack decoder with $b = 0$ is an ML result. And since the SD and the SE can be viewed as tree search algorithms, it was also shown in (61) that the stack decoding algorithm with $b = 0$ generates the least number of nodes among all optimal tree search algorithms including the SE and the SD.
We report two advantages of stack decoding algorithm. First, it offers a natural solution for the problem of choosing an initial radius which is a problem faced in the design of sphere decoder. Second, it allows for a trading off performance complexity as shown in figure (2.16) where we plotted performances in terms of SER and complexities in terms of multiplications per codeword for different values of bias b for a 2×2MIMO system using a 16-QAM constellation. To illustrate this point, if we choose $b = 0$ we obtain the closest point with best performance but with very high complexity. On the other extreme, when $b \to \infty$, the stack decoder reduces to the MMSE-DFE point.

2.4.4 Soft MIMO Decoders

In the literature, soft-output MIMO decoding was studied. Some solutions to this issue have been proposed in (65),(66) and the so called 'list' or 'candidate list' was introduced. The most known soft-output lattice decoder for MIMO systems is the List Sphere Decoder (LSD).
Soft decoding can be realized using a posteriori probability techniques. A posteriori probability (APP) techniques are a judicious choice for high performance receivers with reasonable complexity. Maximizing the APP for a given bit minimizes the probability of making an error on that bit. The APP is usually expressed as a log-likelihood ratio (LLR) value. A decision is made from a LLR value by using its sign to tell whether the bit is one or zero. The magnitude of the LLR value indicates the reliability of the decision. LLR values near zero correspond to unreliable bits. In the following, the logical zero for a bit is represented by amplitude level $b_k = -1$ and logical one by $b_k = +1$. The modulator maps each layer of the bits into data symbols through the mapping

$$f : \{-1, +1\}^{1 \times B} \to \mathfrak{C}$$

where $\mathfrak{C}$ denotes the data symbol constellation and $B = \log_2 |\mathfrak{C}|$ is the number of bits represented by each data symbol.

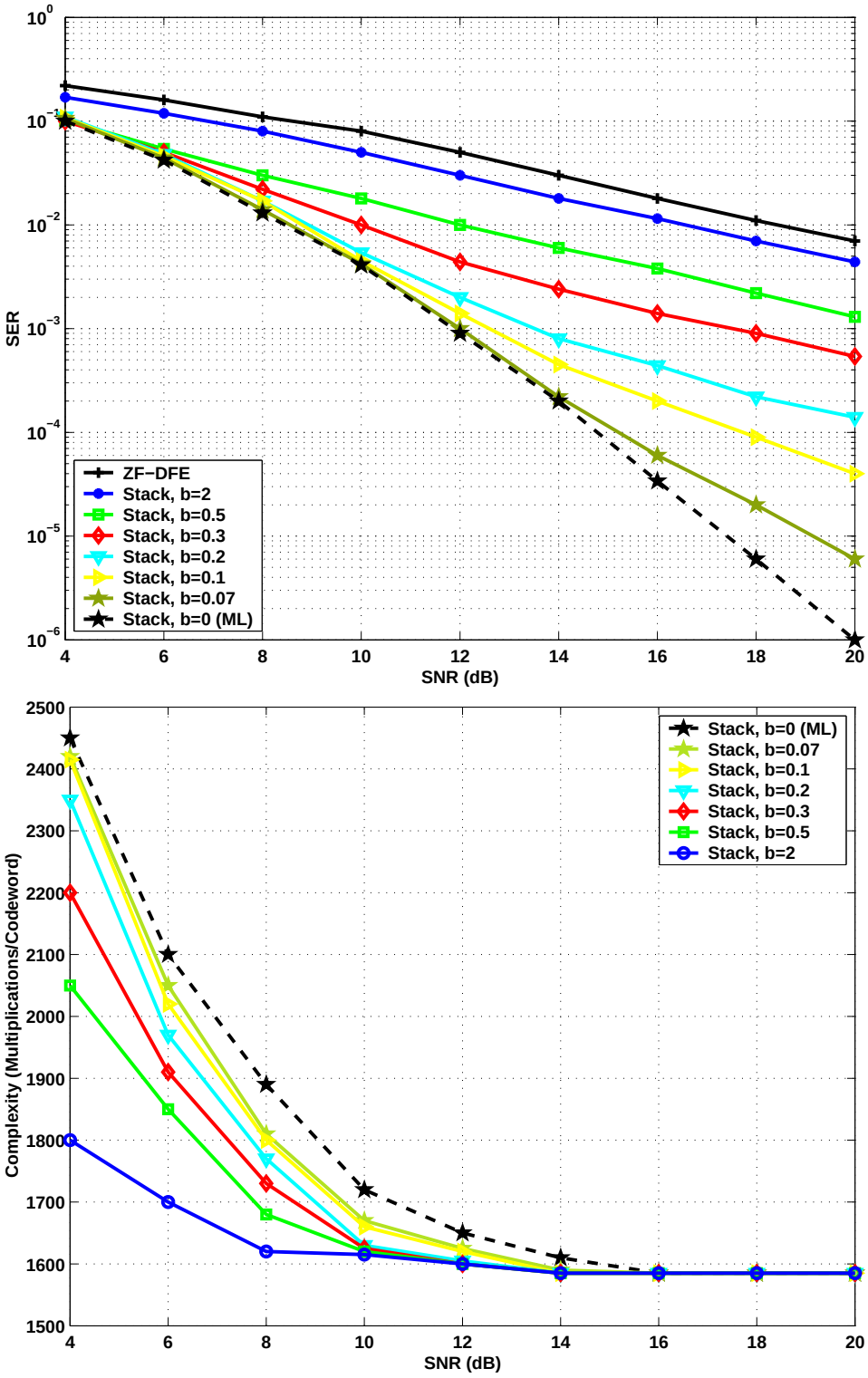


FIGURE 2.16 – Stack Decoding using bias : a Complexity-Performance Tradeoff

The LLR of the i^{th} bit, where $i \in [1,B \cdot p]$, is defined as

$$LLR(b_i) = \log \frac{\Pr\left(b_i = +1/\boldsymbol{y},\boldsymbol{H}\right)}{\Pr\left(b_i = -1/\boldsymbol{y},\boldsymbol{H}\right)}, \tag{2.44}$$

where p, as defined before, denotes the number of symbols belonging to each transmitted codeword. One can assume that data bits are independent (an interleaver at the encoder can be used to scramble the bits). Using Bayes theorem, the bit metric can be written as

$$LLR(b_i) = \log \frac{\sum_{\boldsymbol{b}\in\mathcal{D}_{i,+1}} \Pr\left(\boldsymbol{y}/\boldsymbol{b},\boldsymbol{H}\right)}{\sum_{\boldsymbol{b}\in\mathcal{D}_{i,-1}} \Pr\left(\boldsymbol{y}/\boldsymbol{b},\boldsymbol{H}\right)}. \tag{2.45}$$

where $\mathcal{D}_{i,+1}$ and $\mathcal{D}_{i,-1}$ are the set of $2^{B\cdot p-1}$ bit vectors $\boldsymbol{b}$ with b_i being respectively $+1$ and -1. If we assume an additive zero mean white circularly symmetric complex Gaussian noise, the equation (2.45) can be written as

$$LLR(b_i) = \log \frac{\sum_{\boldsymbol{b}\in\mathcal{D}_{i,+1}} e^{-\frac{1}{\sigma^2}\|\boldsymbol{y}-\boldsymbol{H}\cdot\boldsymbol{s}(\boldsymbol{b})\|^2}}{\sum_{\boldsymbol{b}\in\mathcal{D}_{i,-1}} e^{-\frac{1}{\sigma^2}\|\boldsymbol{y}-\boldsymbol{H}\cdot\boldsymbol{s}(\boldsymbol{b})\|^2}}. \tag{2.46}$$

In order to reduce the corresponding computational complexity, one can employ the *max-log* approximation (78) to get

$$\begin{aligned} LLR(b_i) &\approx \max_{\boldsymbol{b}\in\mathcal{D}_{i,+1}} \left\{-\frac{1}{\sigma^2}\left\|\boldsymbol{y}-\boldsymbol{H}\cdot\boldsymbol{s}\left(\boldsymbol{b}\right)\right\|^2\right\} - \max_{\boldsymbol{b}\in\mathcal{D}_{i,-1}} \left\{-\frac{1}{\sigma^2}\left\|\boldsymbol{y}-\boldsymbol{H}\cdot\boldsymbol{s}\left(\boldsymbol{b}\right)\right\|^2\right\} \\ &= \frac{1}{\sigma^2}\left[\min_{\boldsymbol{b}\in\mathcal{D}_{i,-1}} \left\|\boldsymbol{y}-\boldsymbol{H}\cdot\boldsymbol{s}\left(\boldsymbol{b}\right)\right\|^2 - \min_{\boldsymbol{b}\in\mathcal{D}_{i,+1}} \left\|\boldsymbol{y}-\boldsymbol{H}\cdot\boldsymbol{s}\left(\boldsymbol{b}\right)\right\|^2\right] \end{aligned}$$

Soft-output detection on MIMO channels can be achieved via an exhaustive list as in (79) or a limited size list of spherical shape as in (80) and (81).

The APP detector based on an exhaustive has a relatively large complexity exponential in the number of transmit antennas and the number of bits per modulated symbol. In other hand, a non-exhaustive list APP detector is sub-optimal but has a low complexity which is proportional to the list size. Several list decoders were already proposed. We recall in the following the most known ones and we propose a new soft-decoder based on the SB-Stack.

A. List Sphere Decoder (LSD)

An exhaustive search needs to examine all the constellation points. The sphere decoder avoids an exhaustive search by examining only the points that lie inside a sphere with a given radius $\sqrt{C}$. The performance of the algorithm is closely tied to the choice of the initial radius $\sqrt{C}$. If $\sqrt{C}$ is chosen too small, the algorithm could fail to find any point inside the sphere, requiring that $\sqrt{C}$ be increased. However, the larger $\sqrt{C}$ is chosen, the larger the search will spend time. In (66), a simple modification to the sphere decoder was introduced. The sphere decoder keeps in memory a list $\mathcal{L}$ of N_p points. These points make $||\boldsymbol{y}-\boldsymbol{H}\boldsymbol{s}||^2$ smallest among all the points inside the sphere. The list, by definition, must include the ML point. To create $\mathcal{L}$, the sphere

decoder needs to be modified in two ways : when a candidate is found inside the sphere, the radius $\sqrt{C}$ should not be reduced. In addition, the candidate is added to the list if one of the following conditions is satisfied : either the list is not full or at least one candidate in the list has a higher cost than the new candidate. In this last case, the new candidate replaces the one having the large euclidean distance to the received point. Thus, the constructed list contains the ML point and $N_p - 1$ neighbors for which the square error is the smallest. The soft information about any given bit b_k is essentially contained in $\mathcal{L}$: if there are more entries in $\mathcal{L}$ with $b_k = 1$ than those with $b_k = -1$, then it can be concluded that the likely value for b_k is $+1$, whereas if there are fewer entries in $\mathcal{L}$ with $b_k = 1$, then the likely value is -1. A larger radius $\sqrt{C}$ generally allows for a larger N_p which makes the list more reliable.

There is also a tradeoff between the accuracy and the decoding delay of the LSD. Finding N_p points is generally slower than just finding one point, because the search radius always stays fixed and does not decrease with each found point. One problem of this algorithm is the variable size of the list. In(66), a radius, function of the desired number of points, is proposed. The decoding error can be written as

$$\|\boldsymbol{y} - \boldsymbol{H} \cdot \boldsymbol{s}\|^2 = \|\boldsymbol{w}\|^2 \ \ \sigma^2 \chi^2_{2N_p}, \tag{2.47}$$

where $\chi^2_{2N_p}$ is a chi-square random variable with $2N_p$ degrees of freedom. The expected value of this random variable is $\sigma^2 E[\chi^2_{2N_p}] = 2\sigma^2 N_p$. One possible choice of the radius is

$$C = 2\sigma^2 \zeta N_p - \boldsymbol{y}^H \left(\boldsymbol{I} - \boldsymbol{H} \left(\boldsymbol{H}^H \boldsymbol{H} \right)^{-1} \boldsymbol{H}^H \right) \boldsymbol{y} \tag{2.48}$$

where $\zeta > 1$ is chosen so that one can be reasonably sure, as measured by a confidence interval for the $\chi^2_{2N_p}$ random variable, that the true transmitted $\boldsymbol{s}$ will be found.

The important weak point in the LSD is the instability of the list size. The number of visited points before reaching the ML point can not be fixed exactly, only an approximate number can be provided. The sphere radius is selected to give nearly the needed number. Moreover, the constructed list is not centered at the ML point. A Shifted Spherical List Decoder was proposed in (80) to resolve this problem.

B. Shifted Spherical Decoder (SSD)

The APP detector starts by applying a sphere decoder to find the ML point, then a spherical list centered around the ML point is built. This list depends on the ML point position and the channel state. The trick behind this idea is to center the spherical list $\mathcal{L}$ on the ML point instead of the ZF point. Fig.2.17 shows in two dimensional lattice the sphere centered on the ML point compared to the one centered on the ZF point.

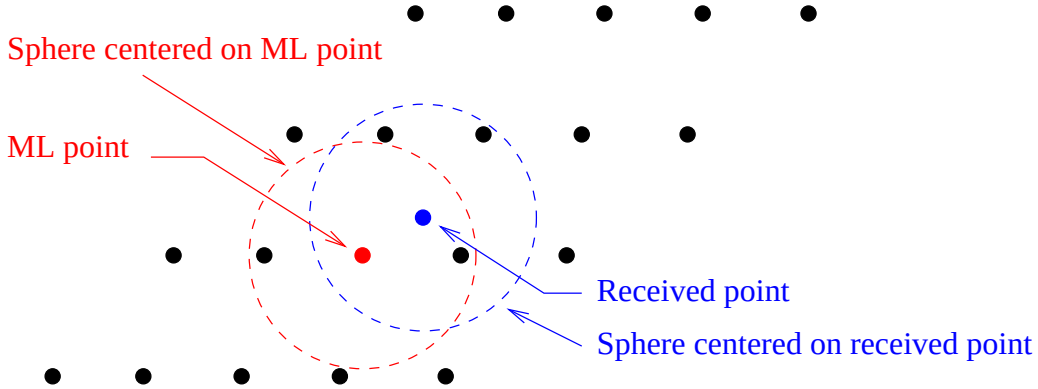


Figure 2.17 – Sphere Centered on the ML Point and the Sphere Centered on the Received point

Usually the received point $\boldsymbol{y}$ is outside the constellation, especially when considering large dimension. The sphere decoder centered at the received point visits a lot of lattice points to find a small number of constellation points. However, when the sphere is centered at the ML point, the number of enumerated points is reduced and higher likelihood constellation points are considered. But to guarantee a high stability for the number of points required in the list, one should be careful for the choice of the shifted list radius. This radius should take into account the number of points to create the list. In (66), an approximation is made : the volume of the sphere containing N_p points is equal to the volume of N_p fundamental parallelotopes. As a result, the radius $\sqrt{C}$ can be computed as

$$\sqrt{C} \approx \left(\frac{N_p \times vol(\Lambda)}{V} \right)^{\frac{1}{n}}, \tag{2.49}$$

where $vol(\Lambda) = |det(\boldsymbol{H})|$, $\boldsymbol{H}$ is the lattice generator matrix, n is the dimension of $\boldsymbol{H}$ and V is the unit radius sphere volume in the real space $\mathbb{R}^n$, $V = \frac{\pi^n}{n}$. This method has the disadvantage of being stable only for high values of N_p. If we assume N_e the effective number of points found inside the list $\mathcal{L}$, one can check that

$$\lim_{N_p \to \infty} \frac{N_e}{N_p} = 1 \tag{2.50}$$

But when considering a finite constellation, N_e will decrease because of the limited shape of the intersection between the sphere and the constellation. This depends on the ML point position inside the constellation and the shape of this constellation. As a result, the radius $\sqrt{C}$ of the shifted spherical list for the constellation can be given by

$$\sqrt{C} = \left(\frac{\alpha[n_{hyp}] \times \mu_\gamma \times N_p \times vol(\Lambda)}{V} \right)^{\frac{1}{n}}, \tag{2.51}$$

where α is an expansion factor of the list size which depends on the number of hyperplanes n_{hyp} at the constellation boundaries passing through the ML point. μ_γ is an additional expansion factor depending on the shape of the constellation (65).

C. K-Best Decoding

The K-Best decoding algorithm was proposed by Wong et al. in (72). This algorithm uses the breadth-first search strategy. At each level of the tree, the K best candidates regarding the optimization metric are memorized and expanded. Thus, in the following level, the N_b possible child nodes for each one of the K candidates are generated and only the K best candidates among the $k \cdot N_b$ child nodes are conserved for the next level, etc.
The advantage of such decoder is its constant complexity which doesn't depend on the reception conditions. However, to obtain near-ML results, we should use a sufficiently high number K which induces an augmentation of complexity. In fact, if the vector to decode contains p symbols and belongs to a constellation of dimension equal to $q = 2^{N_b}$, the number of metrics to calculate is equal to

$$N_b \cdot \left[\sum_{k=1}^{L} N_b^{k-1} + K(p-L)^+ \right] \qquad \text{with} \qquad L = \left\lfloor \frac{log(K)}{N_b} \right\rfloor + 1$$

In (72), the proposed algorithm adds a constraint which selects the K nodes belonging to the sphere of radius $\sqrt{C}$ and centered at the received point. This constraint reduces the complexity of the algorithm. It's also possible to combine the K-Best decoding with the Schnorr-Euchner algorithm (73). At the end of the search phase, the decoder gets $K \times N_b$ solutions which are used to provide a soft output. Nevertheless, these K solutions are not necessarily the nearest to the ML point. Guo and Nilsson proposed a modified K-Best algorithm using the paths in the tree which were not already expanded. Thus, a supplementary information can be recuperated to refine the soft output (73).

2.5 Pre-processing Techniques

The purpose of preprocessing is to transform the original constrained search problem into a form which is easier to the search algorithm. A friendly structure can be derived through two techniques : left preprocessing and right preprocessing. The left preprocessing, MMSE-GDFE, permits to have a well conditioned channel matrix. The right preprocessing consists of making the channel matrix the most orthogonal.
Thus, preprocessing followed by a sub-optimal decoder permits to have better performances. In the case of optimal decoding, preprocessing permits to accelerate decoding and reducing complexity of the search phase.

2.5.1 Left Pre-processing : MMSE-GDFE

Let's recall the equation of a MIMO system defined by :

$$\boldsymbol{y} = \boldsymbol{H} \cdot \boldsymbol{s} + \boldsymbol{w}$$

By multiplying the system by the matrix $\boldsymbol{H}^H$, we get

$$\boldsymbol{y_{LP}} = \boldsymbol{H}^H \boldsymbol{H} \cdot \boldsymbol{s} + \boldsymbol{w_{LP}}$$

The decoder can be a sub-optimal decoder like the ZF or the MMSE decoder , or optimal like the SD or the SE.
The MMSE-GDFE (Minimum Mean Square Error-Generalized Decision Feedback Equalizer) is a decoder with decision feedback. It permits to find information symbols using this equation :

$$\boldsymbol{z} = \boldsymbol{F_{LP}}\boldsymbol{y_{LP}} - (\boldsymbol{B_{LP}} - \boldsymbol{I})\hat{\boldsymbol{s}} \tag{2.52}$$

where $\boldsymbol{F_{LP}}$ and $\boldsymbol{B_{LP}} \in \mathbb{C}^{n\times n}$ and $\boldsymbol{B_{LP}}$ is upper triangular with diagonal elements equal to 1. $\boldsymbol{F_{LP}}$ and $\boldsymbol{B_{LP}}$ are called respectively the *forward* and the *backward* filter, $\boldsymbol{z}$ is the output of the MMSE-GDFE decoder and $\hat{\boldsymbol{s}}$ represents the estimation of the transmitted vector. Thanks to the triangular structure of $\boldsymbol{B_{LP}} - \boldsymbol{I}$, information symbols are recursively calculated starting by the n^{th} component.
Contrary to cases where MMSE-GDFE is used for decoding, it's used here as a preprocessing step. The idea is to find an equivalent system depending on $\boldsymbol{F_{LP}}$ and $\boldsymbol{B_{LP}}$. A decoder can be used after to calculate $\hat{\boldsymbol{s}}$. The forward and the backward MMSE-GDFE filters can be calculated based on the new channel matrix representation $\tilde{\boldsymbol{H}}$,

$$\tilde{\boldsymbol{H}} \triangleq \begin{bmatrix} \boldsymbol{H} \\ \frac{1}{\sqrt{SNR}}\boldsymbol{I} \end{bmatrix}$$

The QR decomposition of $\tilde{\boldsymbol{H}}$ gives :

$$\tilde{\boldsymbol{H}} = \tilde{\boldsymbol{Q}} \cdot \boldsymbol{R} = \begin{bmatrix} \boldsymbol{Q_1} \\ \boldsymbol{Q_2} \end{bmatrix} \cdot \boldsymbol{R}$$

with $\tilde{\boldsymbol{Q}} \in \mathbb{C}^{2n\times n}$, $\boldsymbol{R} \in \mathbb{C}^{n\times n}$, $\boldsymbol{Q_1} \in \mathbb{C}^{n\times n}$ represents the upper part of $\tilde{\boldsymbol{Q}}$. Taking :

$$\begin{aligned} \boldsymbol{B} &= \boldsymbol{R} \\ \boldsymbol{F} &= \boldsymbol{Q_1^H} \end{aligned}$$

We can verify that $\boldsymbol{B^H B} = \tilde{\boldsymbol{H}}^{\boldsymbol{H}}\tilde{\boldsymbol{H}} = \boldsymbol{H^H H} + \frac{1}{SNR}\boldsymbol{I} = \boldsymbol{\Sigma}$ and $\boldsymbol{F} = \left(\boldsymbol{B^{-1}}\right)^{\boldsymbol{H}} \boldsymbol{H^H}$. The new system becomes :

$$\begin{aligned} \boldsymbol{y'} &= \boldsymbol{Fy} \\ &= \boldsymbol{FHs} + \boldsymbol{Fw} \\ &= \boldsymbol{FHs} + \boldsymbol{Fw} + \boldsymbol{Bs} - \boldsymbol{Bs} \\ &= \boldsymbol{Bs} + \boldsymbol{Fw} - (\boldsymbol{B} - \boldsymbol{FH})\,\boldsymbol{s} \\ &= \boldsymbol{Bs} + \boldsymbol{w'} \\ &= \boldsymbol{Rs} + \boldsymbol{w'} \end{aligned}$$

Note that the transformed signal $\boldsymbol{y'}$ is not equivalent to the initial signal $\boldsymbol{y}$. Generally, $\boldsymbol{Q_1}$ is not orthogonal. Also, the resulting noise :

$$\boldsymbol{w'} = \boldsymbol{Fw} - (\boldsymbol{B} - \boldsymbol{FH})\,\boldsymbol{s} = \boldsymbol{Q_1^H w} - (\boldsymbol{R} - \boldsymbol{Q_1^H H})\,\boldsymbol{s}$$

contains some gaussian components given by $\boldsymbol{Q_1^H w}$ and some non-gaussian components given by $-(\boldsymbol{R} - \boldsymbol{Q_1^H H})\,\boldsymbol{s}$ and depending on information symbols. However, $\boldsymbol{w'}$ is white and gaussian, which doesn't affect system performances (67). In fact :

$$\boldsymbol{E}\left[\boldsymbol{w'w'^H}\right] = (\boldsymbol{B} - \boldsymbol{FH})\,(\boldsymbol{B} - \boldsymbol{FH})^{\boldsymbol{H}} + \boldsymbol{\sigma^2 FF^H}$$

Let's first calculate $\boldsymbol{FH}$:

$$\begin{aligned}
\boldsymbol{FH} &= \boldsymbol{Q_1^H H} \\
&= \left(\boldsymbol{B^{-1}}\right)^{\boldsymbol{H}} \boldsymbol{H^H H} \\
&= \left(\boldsymbol{B^{-1}}\right)^{\boldsymbol{H}} \left(\boldsymbol{H^H H} + \tfrac{\boldsymbol{I}}{\boldsymbol{SNR}}\right) - \tfrac{\left(\boldsymbol{B^{-1}}\right)^{\boldsymbol{H}}}{\boldsymbol{SNR}} \\
&= \boldsymbol{B} - \tfrac{\left(\boldsymbol{B^{-1}}\right)^{\boldsymbol{H}}}{\boldsymbol{SNR}}
\end{aligned}$$

And :

$$\boldsymbol{FF^H} = \left(\boldsymbol{B^{-1}}\right)^{\boldsymbol{H}} \boldsymbol{H^H H B^{-1}}$$

Thus, the covariance matrix of the noise $\boldsymbol{w'}$ is equal to :

$$\begin{aligned}
\boldsymbol{E}\left[\boldsymbol{w'w'^H}\right] &= \frac{1}{SNR^2}\left(\boldsymbol{B^{-1}}\right)^{\boldsymbol{H}} \boldsymbol{B^{-1}} + \frac{\boldsymbol{1}}{\boldsymbol{\sigma^2}} \left(\boldsymbol{B^{-1}}\right)^{\boldsymbol{H}} \boldsymbol{H^H H B^{-1}} \\
&= \left(\boldsymbol{B^{-1}}\right)^{\boldsymbol{H}} \left(\boldsymbol{\sigma^2 H^H H} + \frac{\boldsymbol{I}}{\boldsymbol{SNR^2}}\right) \boldsymbol{B^{-1}} \\
&= \frac{1}{SNR}\left(\boldsymbol{B^{-1}}\right)^{\boldsymbol{H}} \left(\boldsymbol{H^H H} + \frac{\boldsymbol{I}}{\boldsymbol{SNR}}\right) \boldsymbol{B^{-1}}
\end{aligned}$$

At the output of the MMSE-GDFE, the system to be resolved is $\boldsymbol{y'} = \boldsymbol{Rs} + \boldsymbol{w'}$. The decoding will be done in function of the new matrix $\boldsymbol{R}$ which still has a full rank. Moreover, $\boldsymbol{R}$ is better

conditioned than the original matrix $\boldsymbol{H}$. In fact, and since $\boldsymbol{R^H R} = \boldsymbol{H^H H} + \frac{1}{SNR}\boldsymbol{I}$, we can write :

$$\lambda^i_{R^T R} = \lambda^i_{H^T H} + \frac{1}{SNR} \tag{2.53}$$

such that λ^i_A represents the eigenvalues of the matrix $\boldsymbol{A}$. Considering the initial matrix $\boldsymbol{H}$, the eigenvalues can have very different values. Multiplying the constellation by the channel matrix $\boldsymbol{H}$ can induce a distortion to the constellation. Thus, it will be difficult to decode a constellation point. Nevertheless, by considering the next system, the second term $\frac{1}{SNR}$ permits to better equilibrate eigenvalues of $\boldsymbol{R}$ and better keeping the initial constellation.

Let's the SVD decomposition of the initial matrix $\boldsymbol{H}$:

$$\begin{aligned} \boldsymbol{y} &= \boldsymbol{H} \cdot \boldsymbol{s} + \boldsymbol{w} \\ &= \boldsymbol{U D V^H} \cdot \boldsymbol{s} + \boldsymbol{w} \end{aligned}$$

such as $\boldsymbol{U}$ and $\boldsymbol{V}$ are unitary of dimension $n \times n$ and $\boldsymbol{D}$ is diagonal of dimension $n \times n$. The elements of $\boldsymbol{D}$ represent the singular values of $\boldsymbol{H}$ or also the square root of eigenvalues of $\boldsymbol{H^H H}$, $\sqrt{\lambda^i}$,$i = 1$,...,r with $r = rank\left(\boldsymbol{H^H H}\right) \leq min(M,N)$. A multiplication by $\boldsymbol{V}$ applied to the signal in the transmission side permits to write :

$$\begin{aligned} \boldsymbol{U^H y} &= \boldsymbol{U^H (U D V^H) V} \cdot \boldsymbol{s} + \boldsymbol{U^H w} \\ \tilde{\boldsymbol{y}} &= \boldsymbol{D} \cdot \boldsymbol{s} + \tilde{\boldsymbol{w}} \end{aligned}$$

or also for a component i : $_i = \sqrt{\lambda^i} \cdot s_i + \tilde{w}_i$(2.-24)

The MIMO system can be viewed as r parallel sub-channels such as each eigenvalue is considered as a fading for one sub-channel. The sub-channels represent the number of symbols that can be transmitted simultaneously.

From the equation (2.5.1), a null eigenvalue λ^i breeds a loss of the information symbol s_i and if the eigenvalue is very low it induces a bad channel.

By considering the preprocessing MMSE-GDFE, the previous equation becomes

$$\tilde{y}_i = \sqrt{\lambda^i_{R^T R}} \cdot s_i + \tilde{w}_i$$

$\boldsymbol{R}$ is of full rank, moreover and based on equation (2.53), the $\lambda^i_{R^T R}$ is not null for $i = 1$,...,M. As a result, there's no loss of information. And for the initially low eigenvalues, the new values are necessarily higher. Thus, the channel is better conditioned.

2.5.2 Right Pre-processing

2.5.2.1 Definition

Let's Λ a lattice from $\mathbb{R}^n$. A lattice has an infinity of bases, let's $\boldsymbol{B_1}$ be a base of Λ. $\boldsymbol{B_2}$ is another base if $\boldsymbol{B_2} = \boldsymbol{B_1}.\boldsymbol{U}$, where $\boldsymbol{U}$ is an unitary matrix. In the figure (2.18), $\boldsymbol{B_1} = (\boldsymbol{u_1},\boldsymbol{u_2}) = ((\mathbf{3},\mathbf{2}),(\mathbf{2},\mathbf{1}))$ and $\boldsymbol{B_2} = (\boldsymbol{v_1},\boldsymbol{v_2}) = ((\mathbf{1},\mathbf{0}),(\mathbf{0},\mathbf{1}))$ are two bases from $\mathbb{Z}^2$. The relation between the two bases is given by the following transformation :

$$\boldsymbol{B_2} = \boldsymbol{B_1}.\boldsymbol{T} \tag{2.55}$$

with

$$\boldsymbol{T} = \begin{bmatrix} -1 & 2 \\ 2 & -3 \end{bmatrix}$$

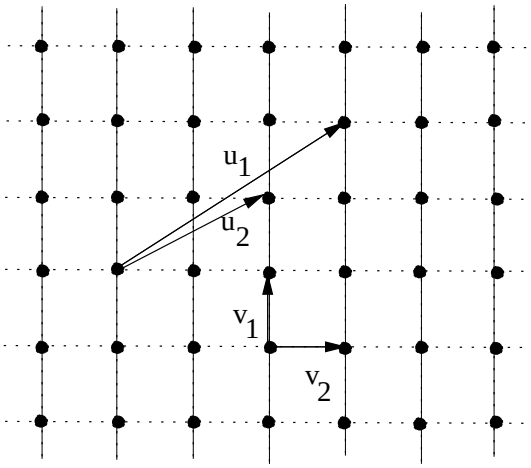


Figure 2.18 – Example of bases in $\mathbb{Z}^2$

$\boldsymbol{B_2}$ is a better base than $\boldsymbol{B_1}$, we say that $\boldsymbol{B_1}$ is reduced to $\boldsymbol{B_2}$. The aim of lattice reduction is to transform a given basis into a new basis with vectors of shortest length and into a basis consisting of roughly orthogonal basis vectors. Usually, the new basis is much better conditioned than the previous one and therefore leads to less noise enhancement for linear detection.
The matrix $\boldsymbol{T}$ that permits to find the reduction base is called the reduction matrix. Choosing the base is very important in decoding. In the paragraph 2.4.1 of this chapter, we illustrated the case of ZF decoding for an orthogonal base and a non-orthogonal base. It's clear that for the second base, the ZF decoding is optimal. Thus, if the base is composed of the shortest vectors and is highly orthogonal as possible :

- the sub-optimal decoding presents better performances.

- the optimal decoding is faster and converges rapidly.

The reduction consists of transforming a given channel matrix $\boldsymbol{H}$ to a matrix $\tilde{\boldsymbol{H}}$ having better structure :

$$\boldsymbol{H} = \tilde{\boldsymbol{H}}\boldsymbol{T}^{-1} \tag{2.56}$$

In the literature, many reduction methods methods were proposed.

2.5.2.2 Reduction Methods

In literature, three reduction methods exist : Minkowski reduction, Korkine-Zolotareff reduction and Lenstra-Lenstra-Lovasz (LLL) reduction. The two first methods permit to obtain an optimal reduced base but their complexities are not polynomial. The LLL reduction produces a base with vectors relatively short and reasonably orthogonal with low complexity. In the next part, we will present the strategy of each technique.

2.5.3 Minkowski Reduction

The purpose of the Minkowski reduction is to find the base with the shortest vectors in the lattice (68).

Let's Λ a lattice and $\boldsymbol{B}$ is the corresponding base. This base is the reduced base if it verifies the following conditions :

1. $\boldsymbol{b_1}$ is the shortest vector of Λ.
2. For $i = 1,...,n$, $\boldsymbol{b_i}$ is the shortest vector in Λ independent from vectors $\boldsymbol{b}_1,\ldots,\boldsymbol{b}_{i-1}$ such that the set $(\boldsymbol{b}_1,\ldots,\boldsymbol{b}_{i-1},\boldsymbol{b}_i)$ represents vectors linearly independent and can be completed to form a lattice base.

As a result, the search of the optimal base is done with a progressive manner by considering all possible linear combinations. It's evident that the obtained base contains the shortest vectors in the lattice.
To conclude, the use of the Minkowski reduction method in practice is very limited because of its huge complexity.

2.5.4 Korkine-Zolotareff Reduction

The KZ reduction is a variant of the Minkowski reduction. It permits to obtain the shortest vectors in the lattice which are also orthogonal and generated by the base obtained before.

The reduced base $\boldsymbol{B}$ verifies :

1. $\boldsymbol{b_1}$ is the shortest vector of Λ.
2. Let's Λ_i the lattice obtained by projecting Λ_{i-1} on the sub-space $\mathbb{R}^{n-(i-1)}$ perpendicular to $\boldsymbol{b_{i-1}}$. $\boldsymbol{b_i}$ is the shortest vector of Λ_i.

2.5.5 LLL Reduction

The LLL reduction is the classical one and is widely used. Contrarily to previous methods, it doesn't provide an optimal solution but a base which is enough good with polynomial complexity (69). For these reasons, it's very used in the number theory and in cryptography.
In (53) and (70), a practical reduction algorithm was proposed based on the orthogonality technique of 'Gram-Shmidt'.

Let's $\boldsymbol{B^\star} = \left(\boldsymbol{b_1^\star},\ldots,\boldsymbol{b_n^\star}\right)$ an orthogonal base obtained with this process. Vectors $\boldsymbol{b_1^\star},\ldots,\boldsymbol{b_n^\star}$ are calculated using the following formula :

$$\begin{aligned} \boldsymbol{b_i^\star} &= \boldsymbol{b_i} - \sum_{j=1}^{i-1} \mu_{ij} \boldsymbol{b_j^\star} \\ \mu_{ij} &= \frac{\left\langle \boldsymbol{b_i}, \boldsymbol{b_j^\star} \right\rangle}{\left\langle \boldsymbol{b_j^\star}, \boldsymbol{b_j^\star} \right\rangle} \end{aligned}$$

The base $\boldsymbol{B^\star}$ is orthogonal but not orthonormal.
Let's consider for example a lattice of dimension 2. The reduced base is constituted of two vectors $\boldsymbol{B} = (\boldsymbol{b_1},\boldsymbol{b_2})$. From equation (2.-26) and equation (2.-27), $\boldsymbol{b_1} = \boldsymbol{b_1^\star}$ and $\boldsymbol{b_2} = \mu_{21}\boldsymbol{b_1^\star} + \boldsymbol{b_2^\star}$ verifying the following conditions :

$$\begin{aligned} \frac{\langle \boldsymbol{b_1}, \boldsymbol{b_2} \rangle}{\|\boldsymbol{b_1}\|^2} &\leq \frac{1}{2} \\ \|\boldsymbol{b_1}\|^2 &\leq \|\boldsymbol{b_2}\|^2 \end{aligned}$$

In order to accelerate the search for the reduced base, the second inequality is replaced by a more selective constraint :

$$\|\boldsymbol{b_1}\|^2 \leq \frac{4}{3} \|\boldsymbol{b_2}\|^2 \tag{2.57}$$

Generally, for a lattice Λ of dimension n, we define by Λ_i the lattice generated by the vectors ($\boldsymbol{b_i}$,$\boldsymbol{b_{i+1}}$). An orthogonal projection of Λ_i on the hyperplane ($\boldsymbol{b_1}$,...,$\boldsymbol{b_{i+1}}$) gives a lattice with the base ($\boldsymbol{b_i}$,$\boldsymbol{b_{i+1}}$) such that :

$$\begin{aligned} \boldsymbol{b_i}(i) &= \boldsymbol{b_i^\star} \\ \boldsymbol{b_{i+1}}(i) &= \boldsymbol{b_{i+1}^\star} + \boldsymbol{\mu_{i+1,i}} \boldsymbol{b_i^\star} \end{aligned}$$

Applying the constraints in equation (2.-28) and equation (2.57) on ($\boldsymbol{b_i}$,$\boldsymbol{b_{i+1}}$), the reduced base $\boldsymbol{B}$ should verify the following conditions :

$$\begin{aligned} |\mu_{i,j}| &\leq \frac{1}{2}, \qquad 1 \leq i,j \leq n \\ \left\| \boldsymbol{b_{i+1}^\star} \right\|^2 &\leq \frac{4}{3} \left\| \boldsymbol{b_{i+1}^\star} + \mu_{i+1,i} \boldsymbol{b_i^\star} \right\|^2 \end{aligned}$$

Or also :

$$\left\| \boldsymbol{b_{i+1}^\star} \right\|^2 \geq \left(\frac{3}{4} - \mu_{i+1,i}^2 \right) \left\| \boldsymbol{b_i^\star} \right\|^2 \tag{2.58}$$

We note that the LLL algorithm apply the orthogonality process locally by considering each time two adjacent vectors. Consequently, the obtained base is quasi-orthogonal and vectors constituting this base are orthogonal two by two. In other hand, bases provided by Minkowski and KZ reductions are perfectly orthogonal.

2.6 The Diversity Multiplexing Tradeoff of MIMO Decoders

Sub-optimal receivers are an attractive low-complexity alternative to optimal processing for multi-antenna MIMO communications. In this section and for fixed number of antennas, we investigate the limit of their error probability in the high-SNR regime in terms of the Diversity-Multiplexing Tradeoff (DMT).
As far as the DMT is concerned, we report a negative result : we show that both linear Zero-Forcing (ZF) and linear Minimum Mean-Square Error (MMSE) receivers achieve the same DMT, which is largely suboptimal.
Let's first recall that the optimal DMT is the best possible error probability exponent $d^*(r)$ achievable by any space-time scheme at multiplexing gain r. While $d^*(r)$ is achievable under the optimal receiver, the following result characterizes the DMT of the MIMO channel when the linear receiver is either the ZF or the MMSE receiver defined before. As given in (71), the

DMT of the M-transmit, N-receive i.i.d Rayleigh MIMO channel with $N \geq M$, constrained to use Gaussian codes under either MMSE or ZF linear receivers is given by

$$d^*_{lin}(r) = (N - M + 1)\left(1 - \frac{r}{M}\right)^+, \tag{2.59}$$

for both the cases of coding across antennas or pure spatial multiplexing and where $(x)^+ \triangleq max\{x,0\}$. The use of lattice reduction (LR) aided versions (75) (76) of the sub-optimal (linear and no-linear) decoders can highly improve performances and diversity. The work in (74) showed that LR (lattice reduction)-aided ZF decoding can achieve maximal receive diversity for uncoded V-BLAST systems.
In (77), authors prove that MMSE regularized lattice decoding, as well as the computationally efficient lattice reduction (LR) aided MMSE decoder, allows for efficient and DMT optimal decoding of any approximately universal lattice based code. The result identifies an explicitly constructed encoder and a computationally efficient decoder that achieve DMT optimality for all multiplexing gains and all channel dimensions. Particularly, they proved that LLL based LR-aided decoding can in fact achieve the most general diversity-related optimality, by showing that the LLL based LR-aided MMSE decoder can, in the context of lattice codes, achieve the maximal diversity gain for all multiplexing gains r and fading statistics.

2.7 Conclusion

In this chapter, we have presented some well-known MIMO decoders available in literature. Thus, we highlighted the functioning of these decoders and their performances. We have showed that these decoders can be classified into three classes : the sub-optimal decoders, the optimal decoders and the decoders offering a complexity-performance tradeoff like the stack decoder. We remarked that optimal decoders have a prohibitive computational complexity and this later increases proportionally to constellation size which makes their use in practical applications very difficult especially for systems with high number of antennas. That's why the sequential decoding and particularly the stack algorithm using the bias parameter to tune and adjust decoder complexity can be a good alternative. However, it leads to sub-optimal results.

In order to improve the stack decoding capabilities, we propose in the next chapter a new sequential decoding algorithm based on the Stack decoding Algorithm and the SD algorithm. We will show that this decoder offers the lowest complexity when compared to all other known ML decoders and guarantees optimal ML performances.

Chapitre 3

Hard and Soft Spherical-Bound Stack decoder for MIMO systems

3.1 Introduction

ML decoding consists in looking for the closest point to the received one belonging to the lattice. Exhaustive search consists in visiting all the lattice points which is impossible to realize in practice. Thus, to decode the received vector, it's necessary to define a finite search region.
As stated before, Stack Decoding was originally designed to decode binary trellis codes, where the codeword is taken in a finite alphabet. However, considering a lattice, the codeword is taken in the infinite field $\mathbb{Z}^{2n}$ which leads to an infinite tree structure. Applying the stack decoder seems to be impossible in this case. Our purpose is then to propose a modified version of the stack algorithm in order to decode lattices and to reduce the prohibitive computational complexity. Next, we will propose a new algorithm combining the search region of the SD and the Stack Decoding Search strategy.

3.2 Spherical-Bound Stack decoder for MIMO systems

3.2.1 SB-Stack Decoding for lattices

A - 1^{st} approach

Applying the stack decoder, we look for the closest point in a finite region $\Lambda \subset \mathbb{Z}^{2n}$. Unfortunately, the truncation of the tree will affect the decoder performances. In fact, if the transmitted codeword belongs to Λ, the decoder will systematically decode it, however an error occurred if

the codeword is outside the search region. Thus, the main challenge is how to choose the optimal search region Λ.

Yet, the triangular form of the lattice basis reminds the Schnorr Euchner (SE) enumeration strategy (52). As detailed in the previous chapter, the key strategy of this algorithm is to consider the lattice as a superposition of $2n$ hyperplanes and to start the search by projecting the received vector on the nearest hyperplane. The resulting point is then recursively projected on the following $2n-1$ hyperplanes. The found point is the 'Babai Point' (BP) and it corresponds to the ZF-DFE point (44).

Our proposed search algorithm is similar to SE and is based on the BP, however the search strategy and the construction of the tree are quite different. In fact, the SE consists of enumerating all possible nodes inside a bounded region by zigzagging around the BP using a DeFS strategy.

In this first approach, we inspire from the SE algorithm and we propose a tree centered at the BP $\boldsymbol{u}$ (we considered the transformation given in equation (2.29)). At each level, it enumerates the neighbor lattice points defined as $\boldsymbol{u} \pm \boldsymbol{t} = (u_1 \pm t_1, u_2 \pm t_2, ..., u_{2n} \pm t_{2n})$ where $\boldsymbol{t}$ is a vector in $\mathbb{Z}^{2n}$, a BeFS strategy is further applied on this tree.

Applying this algorithm, we can delimit the size of the constructed tree by choosing the number of the neighboring lattice nodes of the BP that we will consider. However, the ML point is not guaranteed to be inside the considered tree. To reach it, we should enlarge the search region and that implies to have a denser tree which leads to a more complex decoding task.

In figure (3.1), we show the performances of the stacked decoding constrained to some search regions. Thus, we plot the symbol error rate as a function of the signal to noise ratio (SNR) , for a 4×4 MIMO scheme using a SM and in a quasi-static Rayleigh Channel. First, we proceed by considering the search region defined as $\Lambda_a = \{u_i - 1, u_i, u_i + 1, i = 1, \ldots, 2n\}$. This induces that the lattice points concerned with the search algorithm are only the immediate neighbors of the BP. Nevertheless, in bad channel conditions, the ML point may be far-off and unreachable. This is shown through curve (a), where the performances are sub-optimal and exhibit a loss of 2dB from ML. For the same system, we have progressively enlarged the search region and observed the algorithm's behavior. The curves (a)-(d) report the performances obtained by respectively considering the search regions $\Lambda_a = \{u_i - 1, u_i, u_i + 1\}$, $\Lambda_b = \Lambda_a \bigcup \{u_i - 2, u_i + 2\}$, $\Lambda_c = \Lambda_b \bigcup \{u_i - 3, u_i + 3\}$ and $\Lambda_d = \Lambda_c \bigcup \{u_i - 4, u_i + 4\}$. As shown in figure (3.1), the decoder provides sub-optimal performances, but it approaches the ML as well as we cover a larger search region. However, the complexity increases with performances. Therefore, a compromise may be established and this decoding algorithm can be of great interest. Thus, in the beginning of the algorithm, the complexity-performance tradeoff is fixed which defines the appropriate search region.

In figure (3.1) simulations, we have considered a uniform vector $\boldsymbol{t}$. We can also use a vector $\boldsymbol{t}$ with large t_i for first components and small t_i for the last ones. This choice may be efficient due to the problem of error propagation in the tree search.

B - 2^{nd}approach (SB-Stack decoder)

In the 1^{st} approach, the search region is centered at the BP. However, this latter is generally a rough estimation of the transmitted codeword, then centering the search region on it is not

optimal since the ML solution may not be inside, as shown in figure (3.2).

Therefore, we propose here a second approach for the lattice decoding inspired from the sphere decoder algorithm. The principle of the sphere decoder is to enumerate all the lattice points found in a sphere of a radius $\sqrt{C}$ centered at the received point. Each time a point is found, the radius is updated, which limits the number of the enumerated points but also guarantees the closest point criterion. The sphere decoder uses the DeFS strategy.
We call this second approach : the Spherical Bounds Stack decoder (SB-Stack). The SB-Stack algorithm explores only the lattice points inside the sphere with the radius $\sqrt{C}$ using the BeFS strategy, which leads to the definition of an upper and a lower bounds for each lattice point component. The purpose of the SB-Stack is to find the leaf node having the least cost and within the spherical search region. Starting from the root node, the algorithm computes the upper and lower bounds of the first component $\underline{s}_{2n}$, denoted respectively $b_{inf,2n}$ and $b_{sup,2n}$ and generates all the nodes within these bounds.

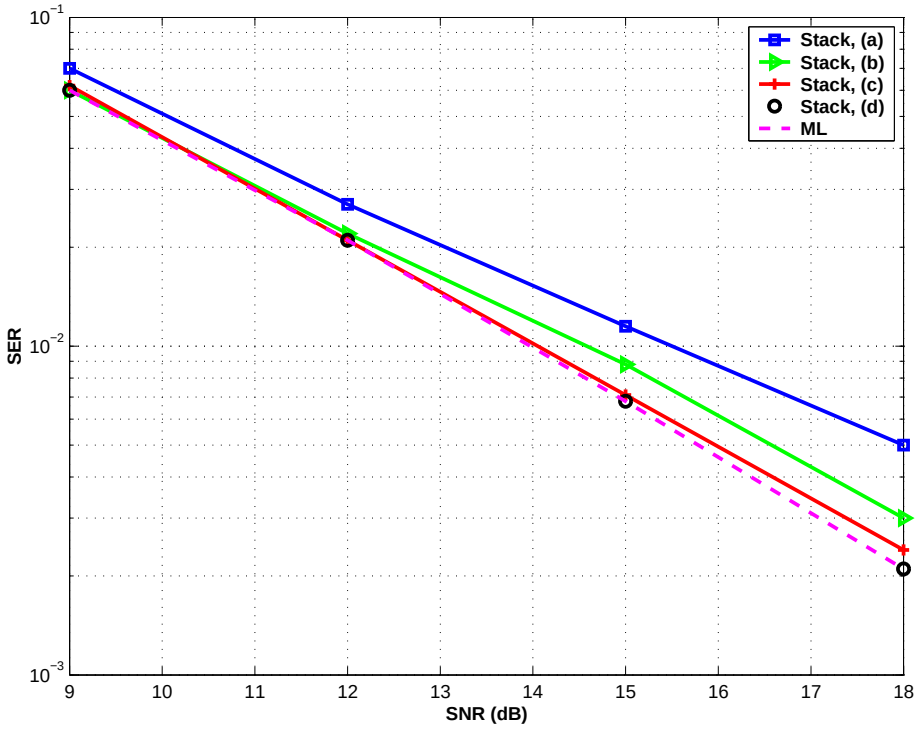


Figure 3.1 – Performance of a MIMO System using a SM with $M = N = 4$, obtained for different search regions

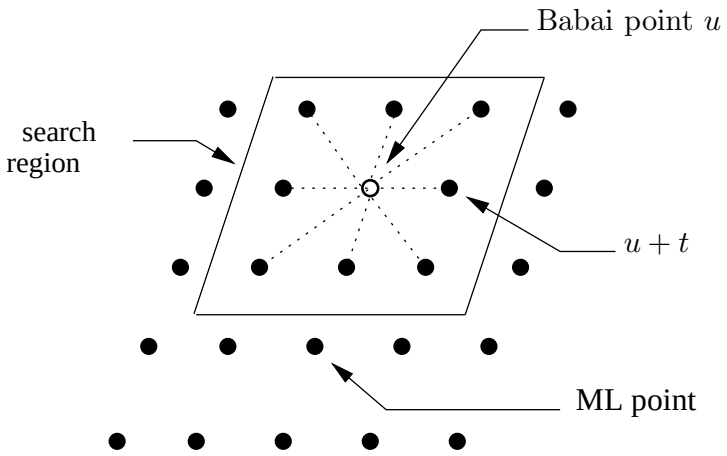


Figure 3.2 – Example of a Lattice defined in $\mathbb{Z}^2$; the Search Region does not contain the ML point

Generated nodes are stored with their respective costs in the stack memory. After that, the algorithm reorders the nodes in the memory in an increasing order according to their costs, selects the top node, then computes the bounds of the next level. Then, it generates all possible children of the top node and stores them in the memory. After that, the top node is removed from the stack. This procedure is repeated until a leaf node reaches the top of the memory.

Note that, although the apparent similarity between the traditional sphere decoder and the SB-Stack algorithm, these two search algorithms raise great differences. In this way, unlike the sphere decoder the radius in the SB-Stack decoder remains unchanged during all the decoding process, while for the sphere decoder the radius is being updated each time a point is found.

However, the search strategy for the stack algorithm is quite different. In fact, at each step the algorithm may backtrack to a higher node level having lower cost before reaching a leaf node which corresponds to a candidate solution. Thus, it's not possible to supervise the ML distance when the algorithm is in progress. Therefore, the radius should be fixed.

In the following, we will detail bounds calculation which is very similar to the case of the Sphere Decoder in (51).

3.2.1.1 Bounds calculation

First, remind that the distance to minimize is given by $\|\boldsymbol{y_1} - \boldsymbol{R}\cdot\underline{\boldsymbol{s}}\|^2$. Let us write $\boldsymbol{y_1} = \boldsymbol{R}\cdot\boldsymbol{\rho}$, where $\boldsymbol{\rho}$ is the ZF point. $\boldsymbol{\rho}$ represents the coordinate of the vector $\boldsymbol{y_1}$ in the new lattice generated by $\boldsymbol{R}$. The Euclidean distance is now written in the lattice system as :$\|\boldsymbol{R}\cdot(\boldsymbol{\rho}-\underline{\boldsymbol{s}})\|^2 = \|\boldsymbol{R}\cdot\boldsymbol{\xi}\|^2$, where $\boldsymbol{\xi}$ defines the coordinate of the translated point $\underline{\boldsymbol{s}}$.

In this case, the lattice points considered in the metric minimization are those within a

maximum distance $\sqrt{C}$ from the received point $\underline{\boldsymbol{s}}$. Thus, we can write

$$\|\boldsymbol{R}\cdot\boldsymbol{\xi}\|^2=\sum_{i=1}^{2n}\left(\sum_{j=i}^{2n}r_{ij}\xi_j\right)^2\leq C \tag{3.1}$$

Let's now define : $q_{ii}^1=r_{ii}^2$ for $i=1,\ldots,2n$, and $q_{ij}^1=\frac{r_{ij}}{r_{ii}}$ for $i=1,\ldots,2n$, and $j=i+1,\ldots,2n$. The equation (3.1) is rewritten as

$$\|\boldsymbol{R}\cdot\boldsymbol{\xi}\|^2=\sum_{i=1}^{2n}q_{ii}^1\left(\xi_i+\sum_{j=i+1}^{2n}q_{ij}^1\xi_j\right)^2\leq C$$

(3.3)

By working backward, we define the bounds at any level i by

$$b_{inf,i}=\left\lceil-\sqrt{\frac{T_i}{q_{ii}^1}}+S_i\right\rceil\leq\underline{s}_i\leq\left\lfloor\sqrt{\frac{T_i}{q_{ii}^1}}+S_i\right\rfloor=b_{sup,i} \tag{3.4}$$

where we refer to T_i and S_i by

$$\begin{aligned}T_{i-1}&=C-\sum_{l=i}^{2n}q_{ll}^1\left(\xi_l+\sum_{j=l+1}^{2n}q_{lj}^1\xi_j\right)^2=T_i-q_{ii}^1(S_i-x_i)\\S_i&=\rho_i+\sum_{l=i+1}^{2n}q_{il}^1\xi_l\end{aligned} \tag{3.4}$$

The flowchart of the SB-Stack decoder is given in figure (3.3).
For each generated node ϖ, many informations concerning this node are also stored with it like its cost, its level, the path to this node and parameters T_i, S_i, $b_{inf,i}$, $b_{sup,i}$if the node is at the level i . Programming with Matlab, we use data structure with several fields. Let's the function Gen : (ϖ,b_{inf}) $\rightarrow$ [b_{inf},ϖ] , that generates a child of ϖ by stacking b_{inf} over ϖ and $store(\varpi)$ the function that stores ϖ and all its parameters.

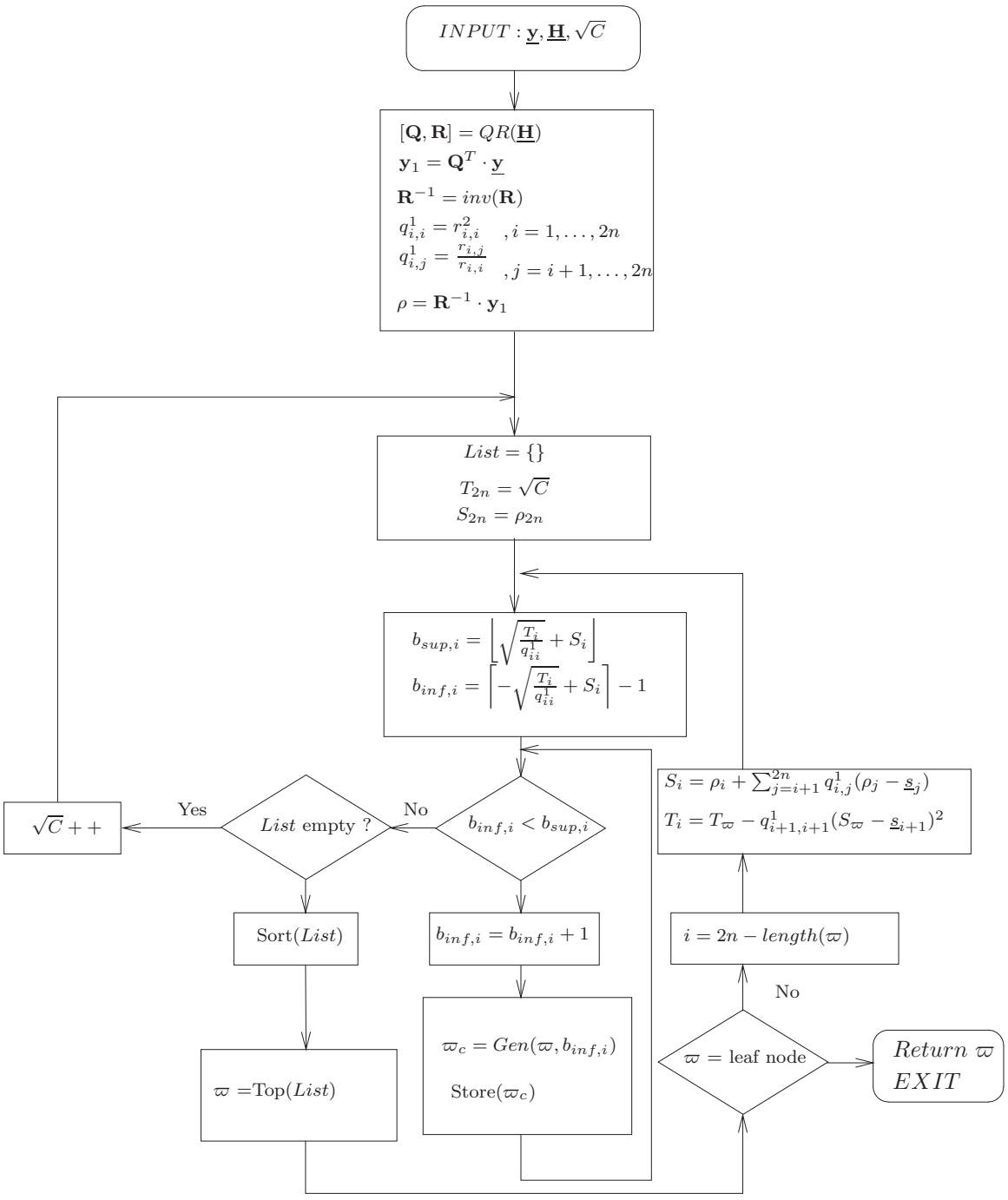


Figure 3.3 – Flowchart of the Spherical-Bound-Stack decoder

The SB-Stack stores more information than the classical stack algorithm and the price to pay is an increasing memory size. But the search region limitation allows visiting fewer nodes and converging quickly and ensures also obtaining the ML solution.

If the stack is empty, that means that no lattice point was found inside the sphere. In this case, the sphere radius must be increased and the algorithm is then restarted.

From this observation, it is clear that the complexity of the algorithm depends also on the sphere radius. In fact, if $\sqrt{C}$ is too large, we obtain too many points, and so a large tree search. But if $\sqrt{C}$ is too small, we obtain no points. Furthermore, for small SNR, the received signal is much affected by the noise and a large radius is needed, while for high SNR, the ML point is close to the received signal and a small radius is sufficient. As stated before, in (54), a formula to choose the optimal radius as a function of the SNR was first reported by Hassibi el *al.* as

$$C = 4.n.\sigma^2 \tag{3.5}$$

But in presence of a deep fading, the lattice can be much distorted and may be more stretched from some axes than others. It is then more suitable to compute the sphere radius taking into account the fading. Therefore, the following formula was proposed in (21)

$$C = \min(4.n.\sigma^2, \min(diag(\boldsymbol{H^T} \cdot \boldsymbol{H}))) \tag{3.6}$$

As for the Sphere Decoder, the SB-Stack offers ML performances. Thus, we will focus our comparison on the complexity which we count as the total number of multiplications of the search phase. Since the multiplications are the most expensive operations in terms of machine cycles compared to additions and comparisons, only multiplications will be taken into account to measure the complexity. The complexity of the algorithm is defined by the number of multiplications carried out until convergence. Both algorithms are composed of three stages : initialization, pre-decoding and searching. The complexity of the two first stages are quite similar, although to evaluate the complexity of the third stage simulation results is necessary. In Fig.3.4, we plot the complexity as a function of the SNR for a 2×2 and a 4×4 MIMO systems using SM.

We can remark that the SB-Stack offers a considerable complexity reduction for different lattice dimensions, which is about 40% less than the sphere decoder for the 2×2 system and 50% less for the 4×4 one for low SNRs. This important complexity reduction is due to the search strategy of the SB-Stack decoder which allows to look inside the sphere only for the most promising points and unlike the sphere decoder which checks all the lattice points inside the sphere.

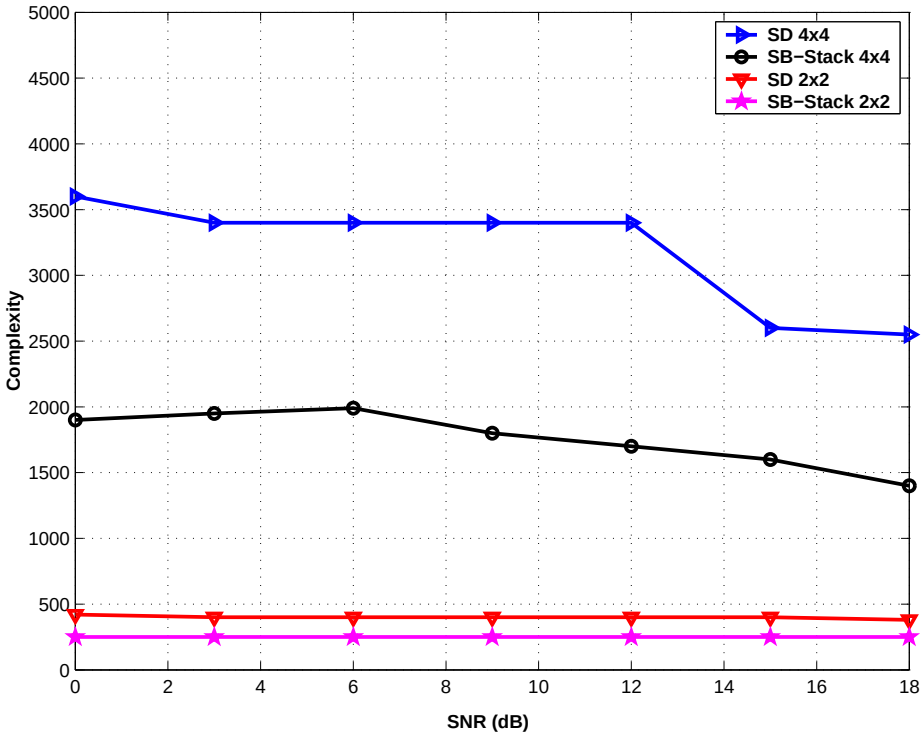


Figure 3.4 – Performance and complexity of the SB-Stack decoder for a 2×2 and a 4×4 MIMO systems using a SM

In practical transmission schemes, we do not consider information symbols in $\mathbb{Z}^{2n}$ but in finite constellations. The most used ones for MIMO schemes are the QAM constellations. We propose in the sequel to adapt the SB-stack decoder to take into account the finite QAM constellations.

3.2.2 SB-Stack Decoding for constellations

In this section, we will focus on the decoding of finite constellations using the stack decoder. As in the previous paragraph, we propose two approaches : the first approach is largely inspired from the original stack decoder, while the second one is a readjustment of the proposed SB-Stack algorithm described before.

A - 1^{st}approach

Using finite constellations tasks, the decoding problem is nearly in its original context where the stack decoder was applied to decode binary codes. In our case, the tree is no longer binary even though it remains finite. As a first and a natural approach, we propose to use the original stack decoder, but instead of the binary values of the tree nodes, we consider the correspondent interval of the $q-QAM$ constellation. For example, for a $16-QAM$ constellation, each tree edge belongs to the set $I_c = \{\pm 1, \pm 3\}$. Then, we have a $4-ary$ tree. More generally, for a $q-QAM$, the nodes to consider are in $\left\{\pm 1, \pm 3, \ldots, \pm\sqrt{q}-1\right\}$. Consequently, the tree structure is directly linked to the used constellation. For large constellation sizes, the information to which belong

the symbols is too large which leads to an excessively dense tree and thus to a very complex decoding. To illustrate this, we represent in figure (3.5) the complexity of the stack decoder for 16 and $64-QAM$ constellations as a function of SNR for a 4×4 MIMO system using SM. We conclude that the complexity increases as the constellation size increases. We also note that, for large sizes, the stack decoder is much more complex than the sphere decoder. In fact, for each level the stack decoder generates all possible children, while the sphere decoder selects only the closer ones. Moreover, this complexity is especially high for low SNR where the decoder crosses more nodes to reach the optimal solution.

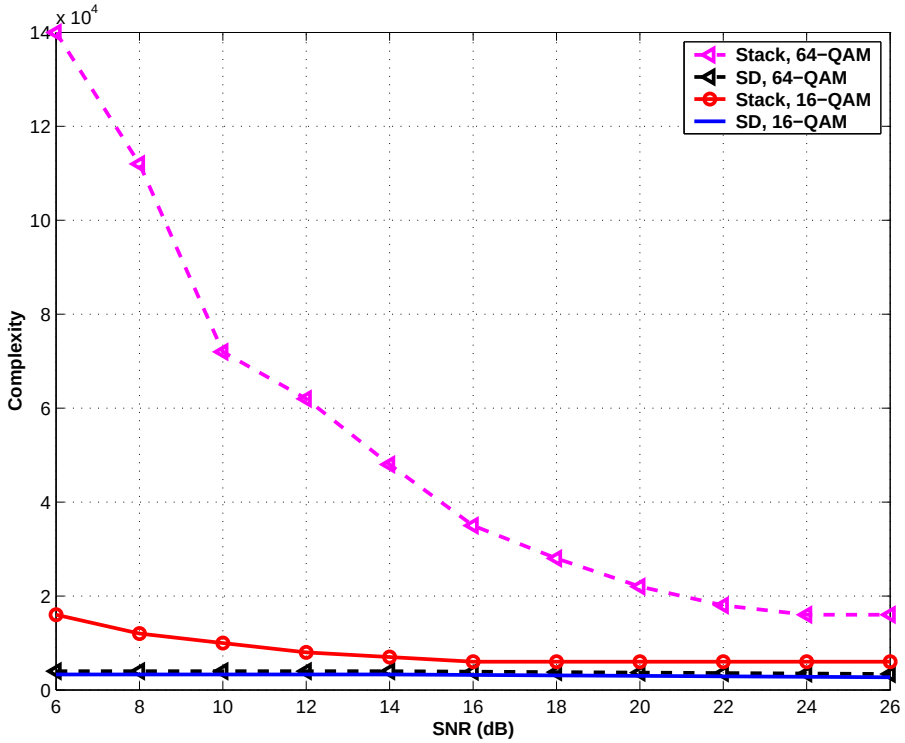


Figure 3.5 – Complexity Comparison of the Stack Decoder and the Sphere Decoder using different QAM Constellations, for a MIMO System with SM, $M = N = 4$

The complexity therefore comes from the high number of visited nodes. In (82), a tree search algorithm was proposed. This one performs a stack algorithm but it limits the size of the stack so that it only retains η nodes at each level of the tree (with $\eta \leq \sqrt{q}$). Nevertheless, the nodes selected in the stack may not lead to the shortest path but conversely nodes with the smallest metric may be discarded in high levels in the tree. Consequently, the ML solution will be rarely reached. So, this algorithm allows to reduce complexity by limiting the number of the generated nodes but at the cost of a performance loss.

We propose in the sequel to use the SB-Stack decoder described before. We will detail in the following the necessary modifications needed to adapt this algorithm for finite constellations decoding.

B - 2^{nd} approach using the SB-Stack algorithm

The SB-Stack algorithm as presented before is conceived to decode lattices. For information symbols taken in a $q-QAM$ constellation, each component of $\underline{\boldsymbol{s}}$ belongs to the finite interval $I_c = \left[\pm 1, \pm 3, \ldots, \pm\sqrt{q}-1\right] \subset \mathbb{Z}^{2n}$. The nodes concerned with the search algorithm are inside the sphere and belong to the constellation. Furthermore, in order to restrict the search to a set of $\mathbb{Z}^{2n}$, we consider the transformation $u_i = \underline{s}_i + \sqrt{q}-1)/2$. The new bounders of the constellation are then given by $I_{c,\mathbb{Z}} = \left[0,1,2,\ldots,\sqrt{q}-1\right]$. Consequently, the nodes that we look for are taken in the interval $\left[sup\left(b_{inf,i},0\right), inf\left(b_{sup,i},\sqrt{q}-1\right)\right]$ instead of the interval $[b_{inf,i},b_{sup,i}]$ computed in (3.4).

Remark 1 :
Once at a given level, the algorithm can be unable to find a valid node $\underline{\boldsymbol{s}}^{(k)}$ if $b_{inf,k} \geq b_{sup,k}$, this means that the path $(\underline{s}_{2n},\underline{s}_{2n-1},\ldots,\underline{s}_{k+1},\underline{s}_k)$ doesn't exist inside the tree and this node doesn't correspond to a point in the sphere. In this case, the algorithm deletes the node $\underline{\boldsymbol{s}}^{(k+1)}$ from the stack and the algorithm continues by considering a new node (the first one in the stack).

Remark 2 :
If the stack is empty, this implies that the algorithm is not able to find a valid path in the tree. In this case, the radius is increased and the search is restarted. But, in order to guarantee the convergence, the initial radius should be chosen as given in equation (radius) which guarantees at least one point in the sphere.

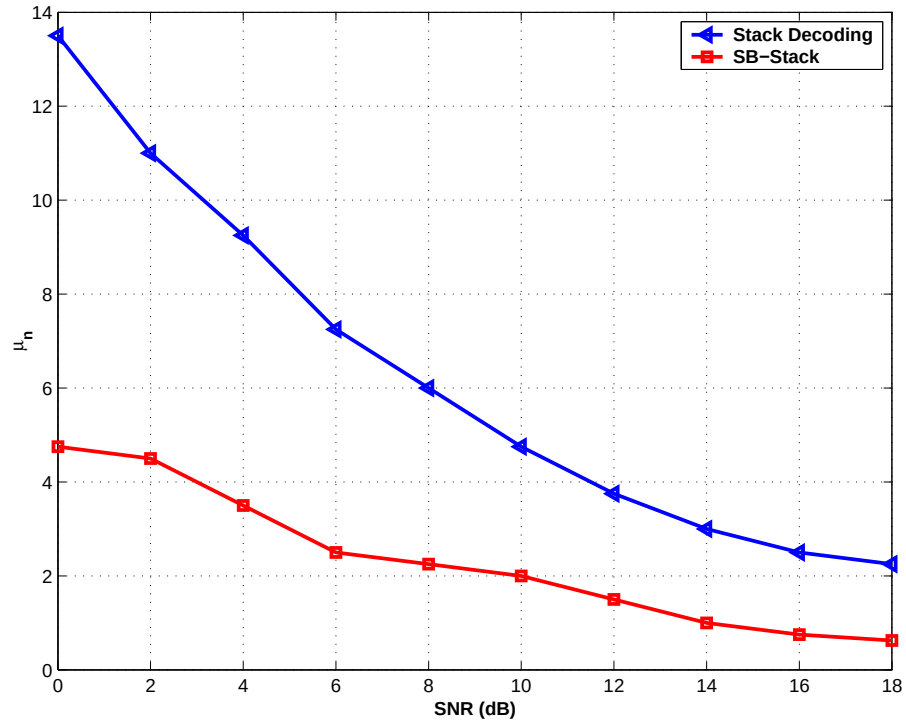
Remark 3 :
Unlike the SD, the radius of the sphere for the SB-Stack can not be adjusted during the search phase since we use the BeFS strategy. In the beginning of the search phase, we can't get a finite path to a leaf node contrary to the SD which uses a DeFS and looks for a finite path at the beginning of the algorithm.

3.2.3 Comparison of the SB-Stack Decoder and the original Stack Decoder

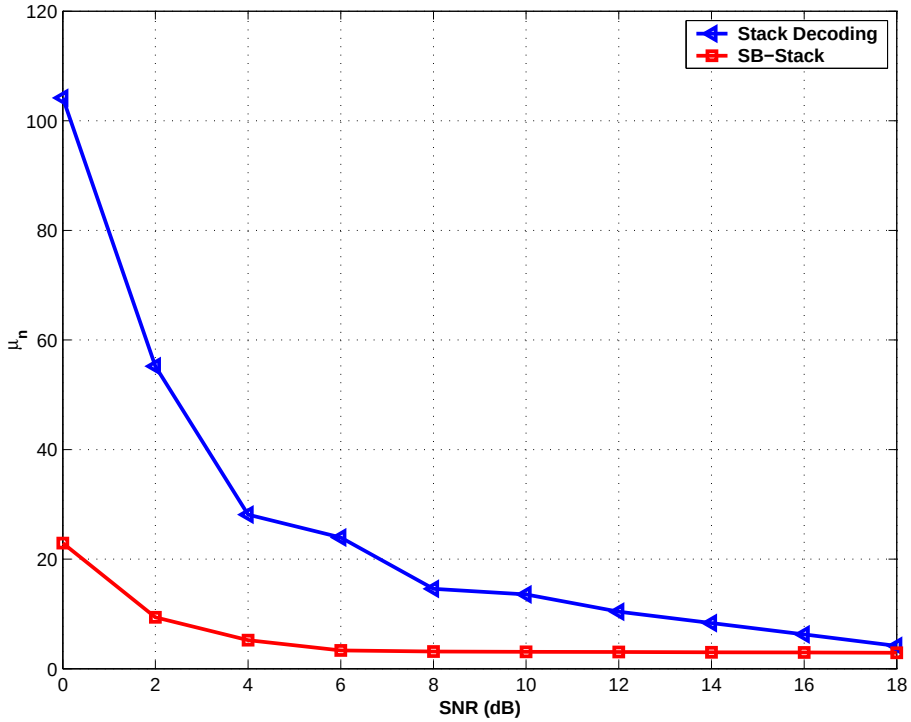
The stack and the SB-Stack decoders adopt the same BeFs strategy and provide the same ML performances (for a null bias). However, the considered trees are quit different. In fact, for the stack decoder, the tree is constituted of all possible combinations of information symbols of the used constellation. At each level of the tree, the vector components are defined by

$$c_{min} \leq \underline{s}_k \leq c_{max} \tag{3.6}$$

The size of the tree and consequently the complexity of the search phase depends on the constellation size. For example, for a q-QAM constellation, each tree node generates $\log_2(q)$ child nodes. In other hand, the SB-Stack decoder constraints the search to a sphere instead of all the constellation. Taking into account this constraint, the SB-Stack algorithm discarded all points that don't belong to the sphere. As a result, for a same constellation, the tree is less dense and the search is less complex.



16-QAM



64-QAM

Figure 3.6 – Comparison of the Stack and the SB-Stack decoding in terms of visited nodes for a 4×4 system with spatial multiplexing

In figure (3.6), we plotted the average of decoding complexities for both decoders where

$$\mu_n = \frac{\text{average number of visited nodes to decode one ST codeword}}{\text{number of bits per ST codeword}}.$$

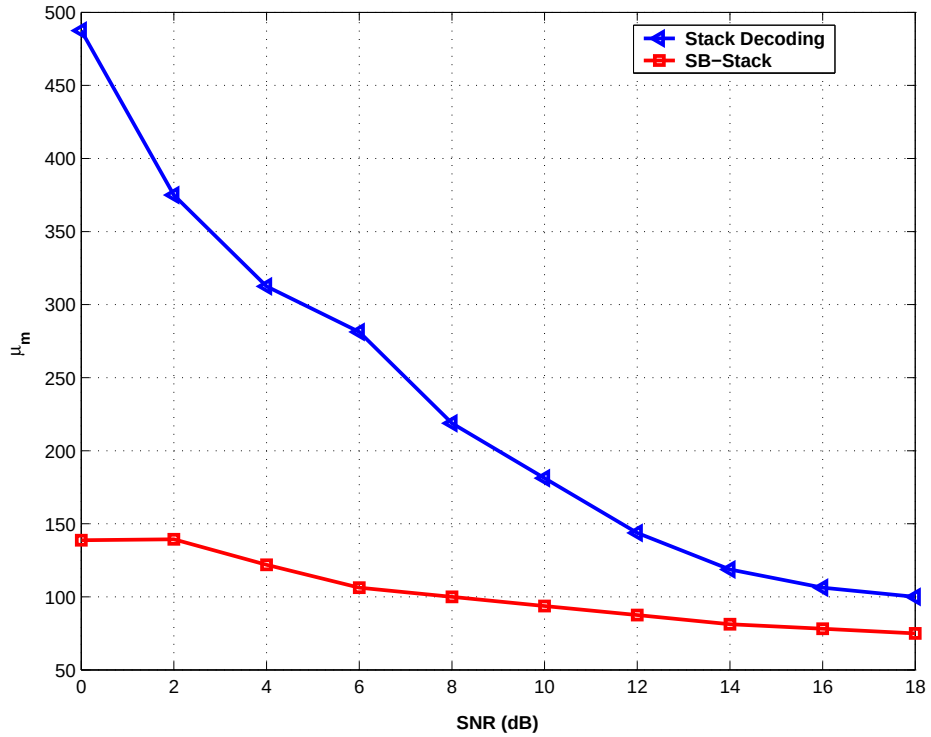
Here, we considered a 4×4 MIMO system using spatial multiplexing with 4-QAM and 16-QAM constellations. Thus, we verified that the SB-Stack decoder expands less nodes than the original Stack decoder. The gain in terms of the average number of visited nodes is nearly equal to 60 % for the 16-QAM and 80% for the 64-QAM (average over the considered SNR). We conclude that by increasing the size of the constellation, it becomes more interesting to use the SB-Stack.
But, by adding the sphere constraint, the SB-Stack decoder should calculate the upper and lower bounds for each expanded node which increases in practice the number of needed operations for each node compared to the original Stack decoder. In fact, bounds are also stored in the stack in order to calculate intervals for the following levels. As a result, the SB-Stack decoder stores more information and requires more memory. However, the huge reduction of the number of the expanded nodes compensates the added complexity. In order to explore complexity in terms of the number of multiplications, we plot in figure (3.7) the complexity of the same 4×4 system such that

$$\mu_n = \frac{\text{average number of multiplications to decode one ST codeword}}{\text{number of bits per ST codeword}}.$$

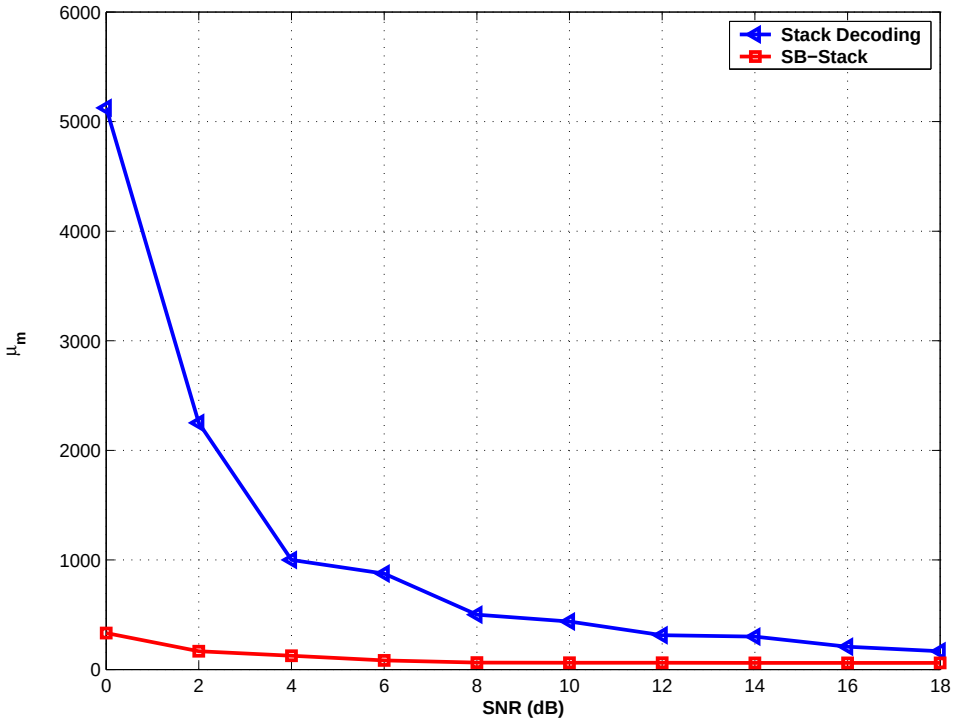
We note that the gain in complexity of the SB-Stack is 46% for the 16-QAM constellation and is 78% for the 64-QAM constellation (average over the considered SNR).

3.2.4 Comparison of the SB-Stack Decoder and the Sphere Decoder

Similarly to the SD, the SB-Stack algorithm constraints the search to a spherical region centered at the received point. Thus, both algorithms share the same search region but they don't adopt the same search strategy.
For each component $\underline{s}_k$, the SB-Stack browses all the elements of $I_k = [b_{inf,k}, b_{sup,k}]$. The node with the least metric is selected and all its children are generated. The selected node can be at a level different from the current one and the algorithm can return back to a higher level in the tree.
The SD browses the same interval I_k and selects the first valid component corresponding to $b_{inf,k}$. This generated node will be expanded in the next iteration and this continues until reaching a leaf node (level 1). Once a leaf node is reached, the algorithm updates the sphere radius and the algorithm continues with a smaller radius. This corresponds to the DeFS strategy. As shown in figure (3.8), we plotted the complexity in terms of the number of visited nodes for MIMO systems using 64-QAM constellation. We notice that the SB-Stack decoder allows a minimum average gain of 57% compared to the SD. We note also that this gain increases if the system dimension increases. It's equal to 63% and 70% for respectively the 4×4 and 6×6 MIMO systems. The gain is particularly high for low SNR.
Considering the same system, we present in figure (3.9) the complexity calculated in terms of the number of multiplications. The SB-Stack decoder shows a great enhancement in complexity compared to the SD.

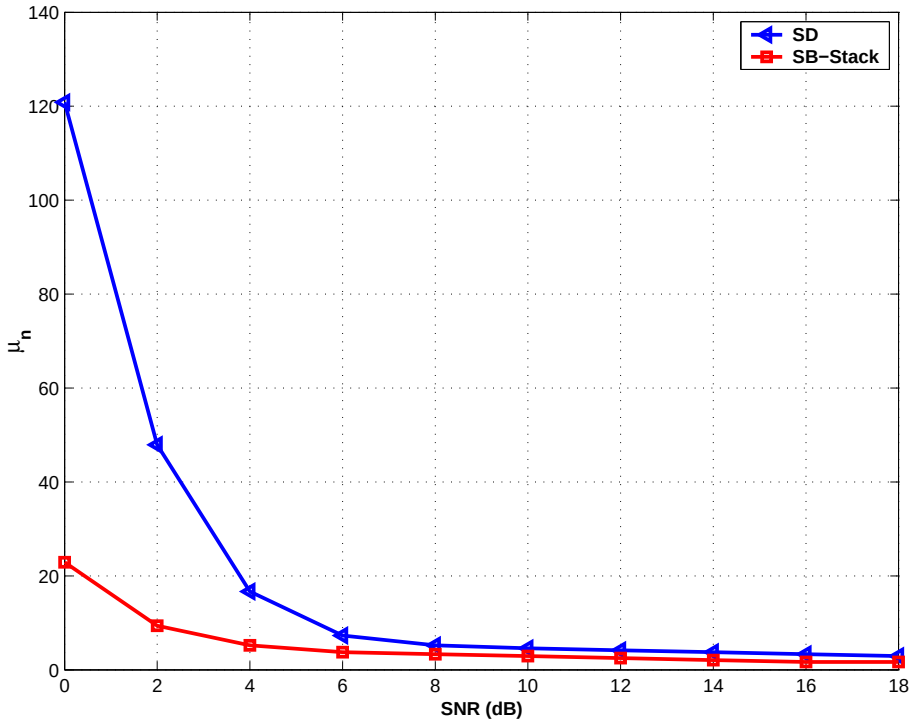


16-QAM

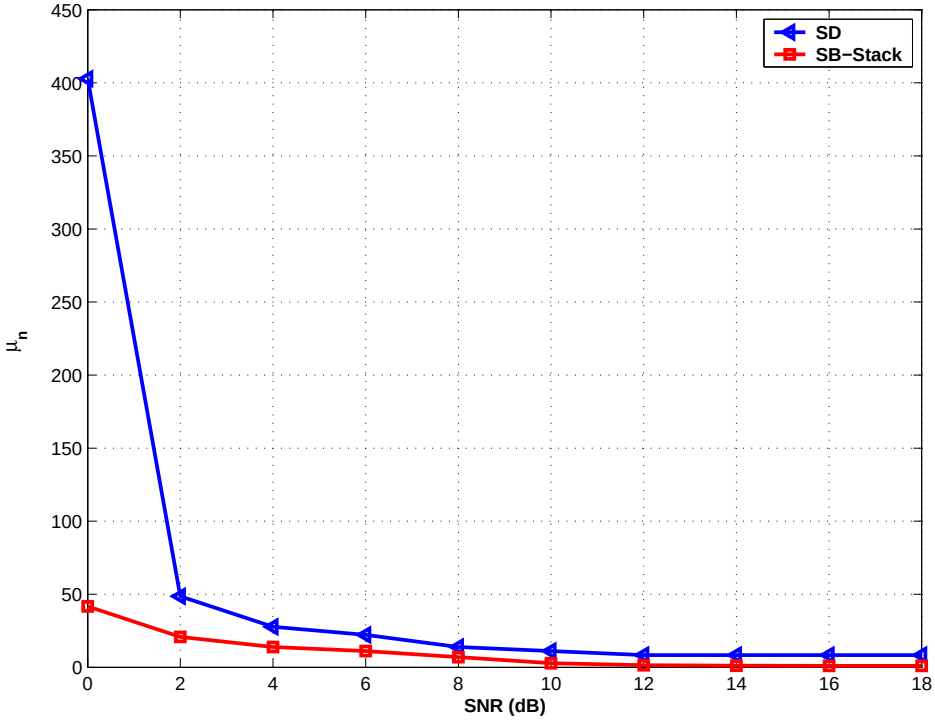


64-QAM

Figure 3.7 – Comparison of the Stack and the SB-Stack decoding in terms of the number of multiplications for a 4×4 system with spatial multiplexing

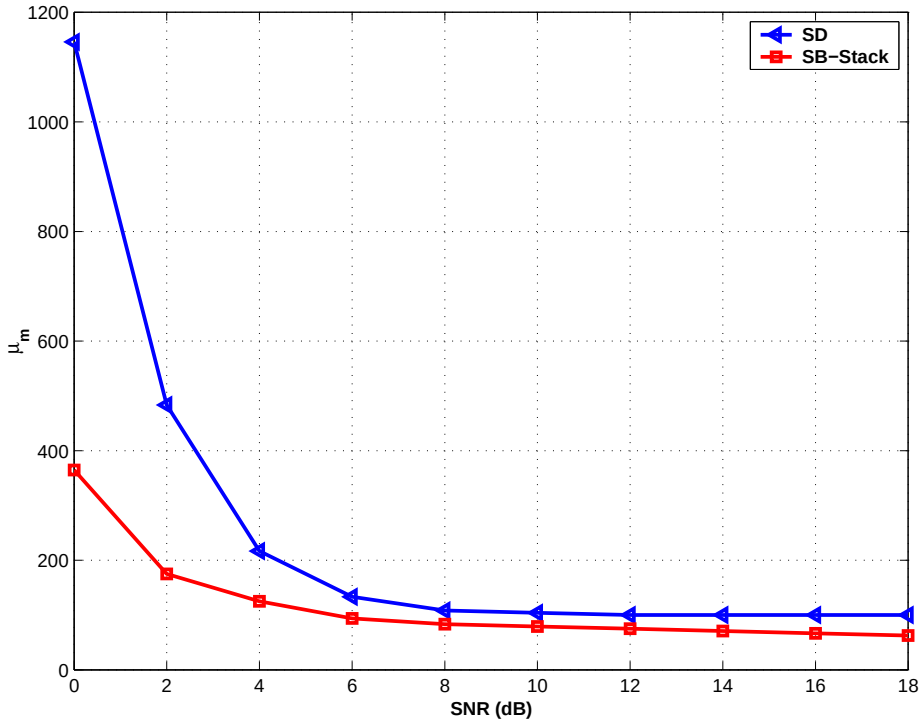


4x4

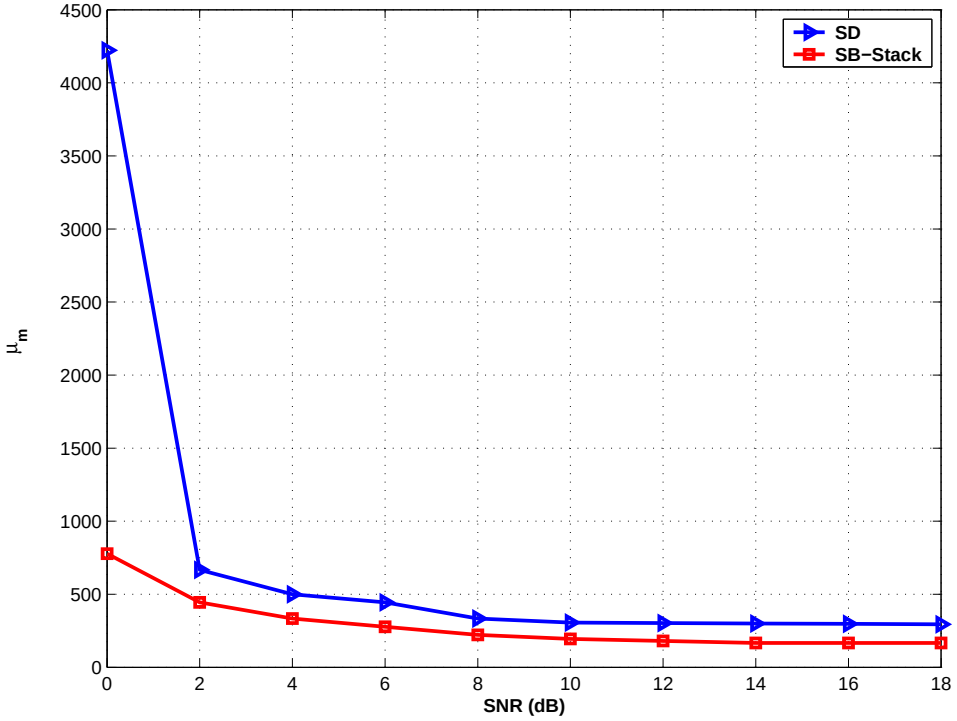


6x6

Figure 3.8 – Comparison of the SD and the SB-Stack decoding in terms of the average number of visited nodes for a MIMO system with spatial multiplexing and with a 64-QAM constellation



4x4



6x6

Figure 3.9 – Comparison of the SD and the SB-Stack decoding in terms of the average number of multiplications for a MIMO system with spatial multiplexing and with a 64-QAM constellation

3.2.5 Sub-optimal SB-Stack decoder

The SB-Stack algorithm searches for the shortest path $(\underline{s}_{2n},\underline{s}_{2n-1},\ldots,\underline{s}_1)$ that minimizes the metric $f(\underline{\boldsymbol{s}}^{(2n)})$. The final solution provided by this algorithm is then the ML one. However, by introducing a parameter in the cost function as defined in section 2.4.3.3, we can rewrite (2.39) as

$$f(\underline{\boldsymbol{s}}^{(k)}) = \sum_{i=2n-k+1}^{n} f_i(\underline{s}_i) - b \cdot k, \tag{3.5}$$

where $b \in \mathbb{R}^+$ is called the bias. Then, a negative weight is added to the cost of each visited node depending on its level in the tree. Under this constraint, the algorithm advantages the deepest nodes in the tree. Hence, this biased version allows the SB-Stack decoder to visit less nodes. Consequently, the complexity is reduced and the decoder converges rapidly, however the solution is not guaranteed to be ML. The performance decreases as b increases, and for a high value of b, it approaches the ZF-DFE. In fact, since the decoding rule is no longer the minimization of the euclidean distance as in (2.39), the performances obtained are not optimal. Though, at very small values of b, the second term $-b \cdot k$ is negligible in equation (3.5), the cost function is approximately equal to (2.39) and then near-ML performances are achieved. The complexity however decreases continuously with b. This parameterized version of the SB-Stack decoder is therefore very interesting since it allows a complexity-performance tradeoff by only adapting the value of the bias.

3.3 Soft Decoding using the stack decoder with Spherical Bounds

3.3.1 Soft Stack Decoding Strategy

We propose here an extension of the new proposed SB-Stack decoder to support soft information outputs. We have modified this algorithm to generate soft-output information in the form of LLR. Stack decoders have the capability of generating a candidate list in their original algorithm. In each iteration, children nodes are generated and stored in the stack ordered in function of their costs. At the end of the algorithm, the first leaf node reaching the top of the stack is the ML point. In this work, we improve the SB-Stack algorithm to make it suitable for a soft output by constructing a list instead of selecting only the ML point. In fact, after the end of the process, one can remark that stack is still full of nodes with different sizes (with different levels in the tree) and no one among them is reaching the top of the stack. The most straightforward idea is to extract the ML point from the original stack, to put it in another stack and to continue the searching phase. The next node reaching the top of the stack is also removed and putted in the second stack with its corresponding cost and so on. There are two possibilities to stop the algorithm :

- either we fix the number of points in the list (the size of the second stack). In this case the algorithm continues in this manner until the second stack will be full.
- another possible criterion is to fix a lower bound on the node costs (worst cost to be admissible), and when the cost falls below this limit value, the algorithm gives up.

Thus, only the nodes stored in the second stack will contribute to the soft decision. This new Soft SB-Stack decoder is an extension of the first one and aims at generating more leaf nodes. The second stack is used later to generate the LLR. The main advantages of this algorithm are

- the stability : the algorithm will stop as soon as the number of candidates is reached. The issue regarding the computation or estimation of the ideal radius value is removed.
- the list is centered at the ML point. In other words, the list is filled up with the closest points only in an ascending cost order, leading to an optimal LLR computation for a given list size.
- a low complexity since we only pursue the stack algorithm with no additional search method and exploit the nodes being already computed and still in the stack.

The disadvantage of all the previous soft decoders is their inability to provide soft outputs with low complexity, and the worst case corresponds to the exhaustive search (66; 78; 89).

The Soft SB-Stack decoder provides less complexity than these latter. Moreover, we can apply the bias parameter as in the equation (3.5). This leads to a complexity-performance tradeoff with soft outputs and one can even impose an aggregate run time constraint.

A straightforward conclusion is the flexibility of the new Soft SB-Stack decoder with practical constraints that system engineers can be faced with the design of the receiver.

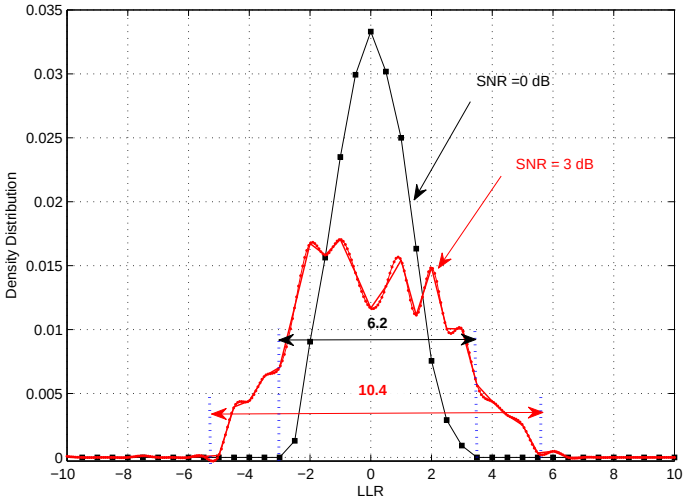


Figure 3.10 – LLR Density Distribution for SNR=0 dB and SNR=3 dB

3.3.2 Soft Decoders Comparison

In this part, we illustrate the application of the Soft SB-Stack decoder for MIMO space time transmission. The binary information is encoded using a rate R-convolutional code. The coded

bits are fed to a q-QAM mapper (Gray mapping) that generates symbols. The spectral efficiency in the V-BLAST case is $R \times N_b \times M$ bits per channel use, $N_b = \log_2(q)$.

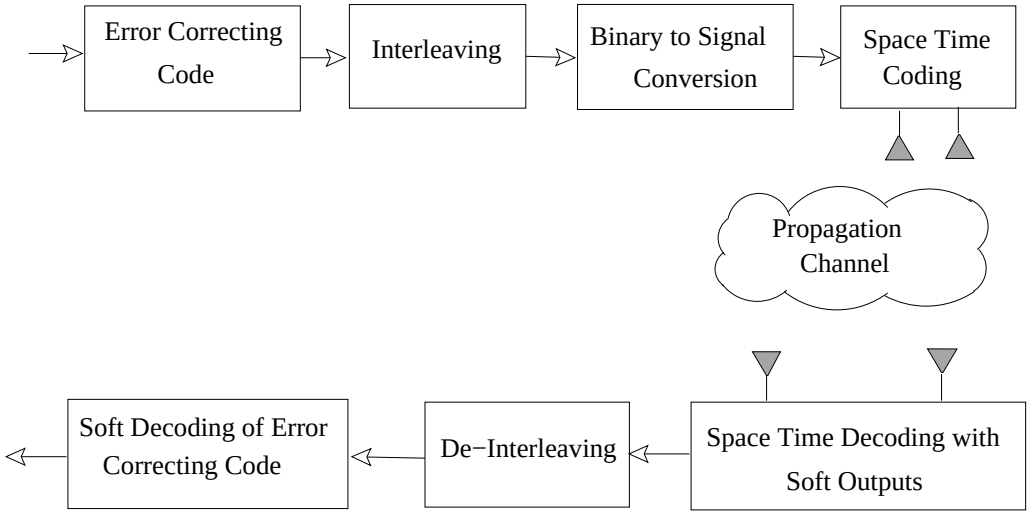


Figure 3.11 – Communication Chain

The figure (3.10) shows the LLR distribution of the candidates found inside the stack for $SNR = 0dB$ and $SNR = 3dB$. It can be observed that when SNR increases, the LLR distribution curve is going to get a concave shape with a cavity around zero. This can be expected since $zero - LLR$ means ambiguity in the decision which is diminished when SNR increases.

For high SNR, the LLR values stretch to infinity and in practice they are saturated to a high chosen value. The LLR distribution curves provide us with information about the intervals to which LLR belong. LLR will be sampled into $2^m - 1$ levels of their interval distribution and then quantized to m bits to serve as input for the soft Viterbi decoder. The considered transmission chain is given in figure (3.11). This transmission chain enhances performances of MIMO systems and space time codes that will be used. We associate it to an error correcting code followed by an interleaver juste before the space time coding bloc. This transmission scheme was introduced by Tonello (90) in 2000, and is an extension to space time codes of bit-interleaved coded modulation (BICM). The idea of concatenating an error correcting code with a binary interleaver and the operation of binary to signal conversion comes from Zehavi who shows experimentally (91) that the interleaving of bits conduces to better performances in terms of BER that interleaving symbols, for an equivalent complexity, in a rayleigh channel. In (92), Caire et al. confirmed these results theoretically for BICM. Thus, the figure (3.11) represents a communication chain ST-BICM constituded of an error correcting code, a binary interleaver, a binary to signal converter, and STBC (or just a ST multiplexing).
In error correcting codes, we associate a codeword $\mathcal{C}$ constituted of N_c bits to the information codeword constituted of N_d bits. The rate of the code is equal to $R_c = \frac{N_d}{N_c}$. In the paper of Tonello (90) , the used error correcting code is the convolutional code. However, other types of codes can be used in the ST-BICM scheme, like for example the turbo-codes (93) and the LDPC (Low Density Parity Check) codes (94) (90) .
The binary interleaver is the principal element of the BICM. It has a double role. In one hand, it allows a decorrelation of channel fading subjected to coded bits and maximize the diversity order

of the system. In other hand, it guarantees a decorrelation of bits belonging to the same codeword of the error correcting code. This last property is very important if we want in future works to upgrade the system to iterative decoding. Here, the considered interleaver is a pseudo-random one.

The simulated MIMO system is constituted of 2 transmit and 2 receive antennas. The propagation channel is modeled by a quasi-static Rayleigh channel model. The error correcting code is a convolutional code with a rate $R_c = 1/2$, the constraint length is $L = 7$ and the generator polynomial is $[133,171]_{\mathcal{O}}$. The interleaver length is equal to 9216 bits and the length of codewords of the error correcting code is also equal to 9216 coded bits (and thus 4608 information bits per codeword). The considered space-time coding scheme is a spatial multiplexing ($T = 1$) scheme and symbols belong to a 16-QAM constellation. The spectral efficiency of this system is equal to 4b/s/Hz. Performances in terms of bit error rate are given in figure (3.12) in function of E_b/N_0. Concerning the Stack Decoder, the LSD, the Shifted SD and the SB-Stack, we limited the number of generated nodes to a maximum number $N_{max} = 148$ which corresponds to the constant complexity of the K-BEST Decoder with $K = 16$. The size of the candidates list is fixed to $N_p = 16$. In figure (3.12), we observe that the best performance is attained by the Stack Decoder, the SB-Stack Decoder and the K-BEST Decoder with K=16. In other hand, the spherical decoders (LSD and Shifted SD) give degraded performances. This degradation is related to the spherical limits constraining these decoders. In fact, if the chosen radius is too small, the number of the required candidates is not reached. In this case, the algorithm can stop the search and content with candidates in the list, it can also update the sphere radius in order to obtain the needed number of candidates. In our results, we have chosen to implement this latter one. However, since we have limited the number of visited nodes to $N_{max} = 148$, the SD can be unable to reach the required number of candidates N_p. This can happen if the radius of the sphere is badly calibrated in the beginning which makes a big disadvantageous for the SD especially that it can influence the complexity of the decoder. It can be remarkable that the SB-Stack which has spherical bounds is not sensitive to these performances degradation. This can be explained by the fact that we realize a quick termination when the maximum authorized number of nodes is reached. This termination consists of calculating the ZF-DFE for the most $\min(N_s, N_p - N_L)$ advanced nodes in the tree, where N_s is the present size of the stack and N_L is the present size of the soft List. We conclude also that the K-Best decoder offers the same performances as the SB-Stack but with a sufficiently high K. The average and the standard deviation of decoding complexities for different decoders are given in figures (3.13). Complexities are given here as follows :

$$\mu_n = \frac{\text{average number of visited nodes to decode one ST codeword}}{\text{number of bits per ST codeword}}$$

$$\sigma_n = \frac{\text{standard deviation of the number of visited nodes to decode one ST codeword}}{\text{number of bits per ST codeword}}$$

As shown in those figures, the K-Best Decoder with $K = 16$ is the most complex. The SB-Stack decoder offers the best performance-complexity tradeoff with an average complexity less than all other decoders. The Stack Decoder presents a complexity inferior or similar to the Shifted SD for high E_b/N_0, but its performances are better than the Shifted SD.

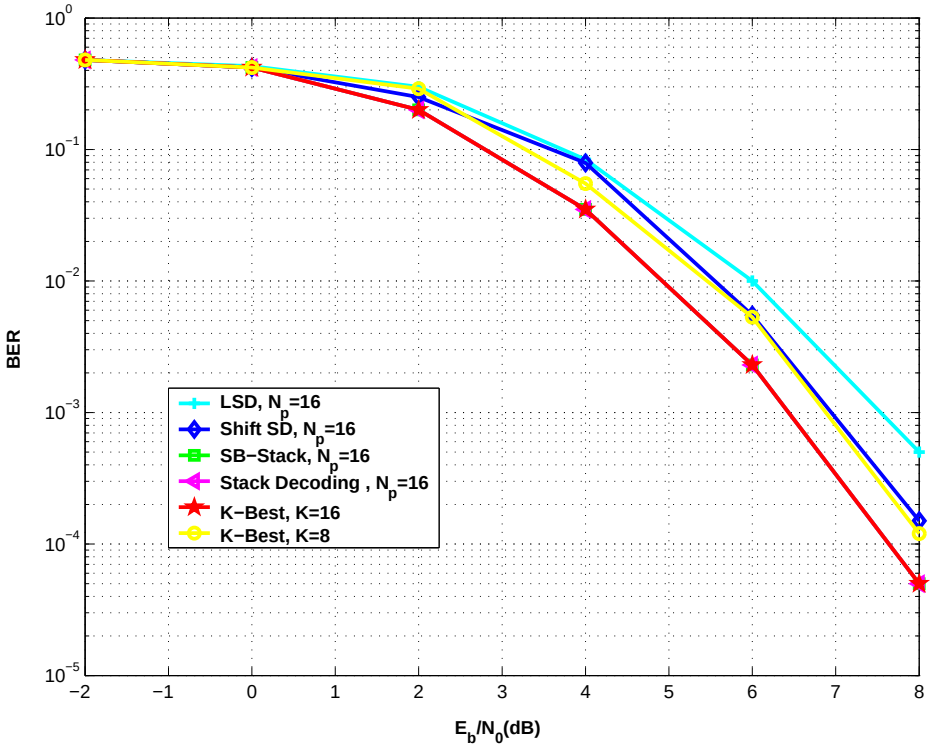


FIGURE 3.12 – Comparison of the Performance of Soft Decoders for a 2×2 MIMO System with Spatial Multiplexing, 16-QAM, $CC[133,171]_{\mathcal{O}}$,$R_c = 1/2$, in a Rayleigh Channel

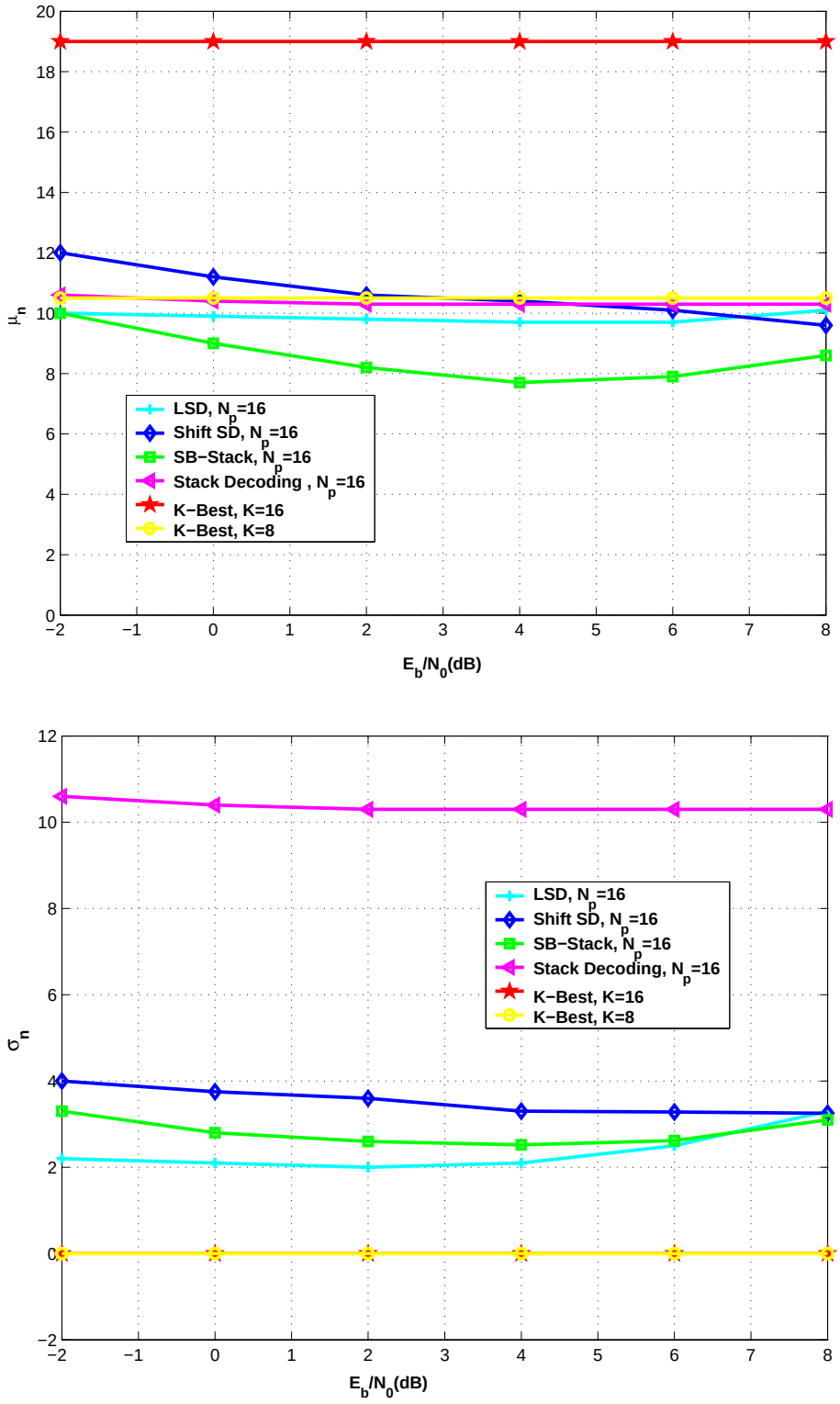


FIGURE 3.13 – Comparison of the Complexity of Soft Decoders for a 2×2 MIMO System with Spatial Multiplexing, 16-QAM, $CC[133, 171]_{\mathcal{O}}$, $R_c = 1/2$, in a Rayleigh Channel

3.4 Conclusion

The main purpose of this chapter was to apply sequential decoders, especially the stack one, for multi-antenna systems. We saw that the MIMO decoding can be tasked on a CLPS problem. Toward this end, we proposed approaches that apply the stack decoder to decode lattice. Our main contribution was to introduce a novel version of the stack decoder for MIMO systems with reduced complexity. The proposed decoder, that we called the Spherical Bound Stack decoder, consists in a modified stack decoder combining both the stack algorithm search strategy and the sphere decoder properties. In a first time, this modified decoder was introduced to decode lattice. In a second time, we brought the necessary modifications to apply it in the case of constellations and we showed by simulations that the SB-Stack decoder outperforms the SD decoder and the original stack decoder in terms of complexity.

In the second part, we extended the SB-Stack decoder to support soft output MIMO detection. By exploiting the advantage of the memory use to deliver a soft output, a good improvement in performance is distinguished. The simulation results show that the Soft SB-Stack decoder outperforms the known List Sphere Decoder and the Shifted Sphere Decoder. Moreover, due to the stack properties, the Soft SB-Stack decoder is also easily implemented and is more flexible to an increase in the size of the list of candidates

Chapitre 4

On Reducing Complexity of MIMO Stack Decoding

4.1 Introduction

The present chapter is intended to improve the stack decoding in terms of complexity. As shown before the stack decoding algorithm is a good candidate for resolving the complexity-performance tradeoff while giving an excellent structure that provides soft outputs. Our purpose is then to propose a modified version of the stack algorithm in order to reduce complexity and time consumption.

This chapter is organized as follows. First, we exploit a specific lattice representation to define stack decoding with parallel processing. Thus, many instructions can be carried out simultaneously which is not possible with SE and SD. This accelerates the decoder and permits to reach the ML point in less time. In the second part, we give some different strategies for expanding tree nodes and we show that those strategies can scale down the decoder complexity. At the end of this chapter, we propose some new techniques to resolve the high noise effect problem that can slow down the speed of the stack decoder.

4.2 Parallel Stack Decoding

Stack decoding shows good features to be implemented and improves complexity compared to ML decoders SE and SD. But it still suffers from big complexity. Parallel processing can reduce time by executing instructions simultaneously. We propose here a new stack decoder based on a parallel processing. SD was parallel processed in (91) but with a loss in performance.
Parallelism is a form of computing in which many instructions are carried out simultaneously. However, parallel programs are harder to write than sequential ones.
To solve a problem, an algorithm is constructed which produces a serial stream of instructions.

These instructions are executed on a central processing unit on one computer. Only one instruction may execute at a given time - after that instruction is finished, the next is executed. Parallel processing uses multiple compute resources simultaneously to solve a problem. The problem is broken into parts which are independent so that each compute resource can execute its part of the algorithm simultaneously with others. Frequency scaling was the dominant reason for computer performance increases from the mid 1980s until our days. The total runtime of a program is proportional to the total number of instructions multiplied by the average time per instruction.
Everything else constant, increasing the clock frequency decreases the average time it takes to execute an instruction. An increase in frequency thus decreases runtime for all computation-bounded programs.
In this part, we improve the stack decoding complexity efficiency by proposing a new structure performing parallel processes. This structure enables decoding the real and imaginary parts of each symbol independently and at the same time. The lattice representation given before by Ψ function imposes a major restriction on the tree search algorithm. Specifically, the search has to be executed serially from one level to another on the tree. Processing each level to estimate symbols needs to have estimation of previous symbols which are necessary to calculate costs for each child. The standard stack decoding using the tree search starts at the top level and traverse down the tree with one level at a time, and computes for each step the costs of child nodes.

4.2.1 New Lattice Representation

According to the Ψ lattice representation, it's impossible, for instance to calculate the cost for a node in level k without assigning an estimate for the levels before.
This approach means that decoding of any $\underline{s}_k$ requires an estimate value for all preceding $\underline{s}_k$ where $j = k-1,...,2n$.
The idea behind this work is to relax the tree search structure making it more flexible for parallelism. Thus, one can decode each pair of adjacent levels in the tree, and each level of this pair is independent from the other. For that, one should start by giving a second shape to the channel matrix representation. Instead of the Ψ function defined as :

$$\begin{aligned} \underline{\mathbf{H}} &= \Psi(\mathbf{H}) \\ &= \begin{bmatrix} \mathfrak{Re}(\mathbf{H}) & -\mathfrak{Im}(\mathbf{H}) \\ \mathfrak{Im}(\mathbf{H}) & \mathfrak{Re}(\mathbf{H}) \end{bmatrix} \end{aligned} \tag{4.0}$$

we use another function Ω and we give another lattice representation defined in (91) as :

$$\begin{aligned} \underline{\tilde{\mathbf{H}}} &= \Omega(\mathbf{H}) \\ &= \begin{bmatrix} \Psi(H_{1,1}) & \cdots & \Psi(H_{1,n}) \\ \vdots & \ddots & \vdots \\ \Psi(H_{n,1}) & \cdots & \Psi(H_{n,n}) \end{bmatrix}, \end{aligned} \tag{4.0}$$

where we assume that Ψ can be applied to a complex component as for the matrices case.
Using this channel representation changes the order of detection of the received symbols to the

following form

$$\begin{aligned} \underline{\tilde{\mathbf{y}}} &= \Omega(\mathbf{y}) \\ &= \begin{bmatrix} \mathfrak{Re}(\mathbf{y}_1) \\ \mathfrak{Im}(\mathbf{y}_1) \\ \vdots \\ \mathfrak{Re}(\mathbf{y}_n) \\ \mathfrak{Im}(\mathbf{y}_n) \end{bmatrix}, \end{aligned} \tag{4.0}$$

which means that the first and the second levels of the tree search correspond to the real and imaginary parts of $\mathbf{s}_n$.

4.2.2 Overview of Parallel Decoding technique

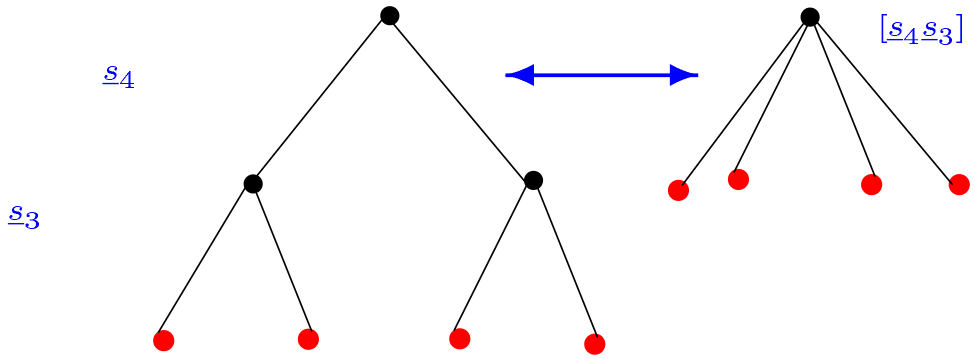
After applying the QR decomposition, this structure becomes very interesting. In fact, due to the orthogonality between the columns of each set, all the elements $\underline{r}_{k,k+1}$ for $k = 1,3,\cdots,2n$ in the upper triangular matrix are null. The localizations of these zeros are very important since they introduce orthogonality between the real and imaginary parts of every detected symbol. For example, for the 4×4 $\underline{\tilde{\mathbf{R}}}$ upper triangular matrix, we got the following form

$$\underline{\tilde{\mathbf{R}}} = \begin{bmatrix} \underline{r}_{1,1} & 0 & \underline{r}_{1,3} & \underline{r}_{1,4} \\ 0 & \underline{r}_{2,2} & \underline{r}_{2,3} & \underline{r}_{2,4} \\ 0 & 0 & \underline{r}_{3,3} & 0 \\ 0 & 0 & 0 & \underline{r}_{4,4} \end{bmatrix}. \tag{4.1}$$

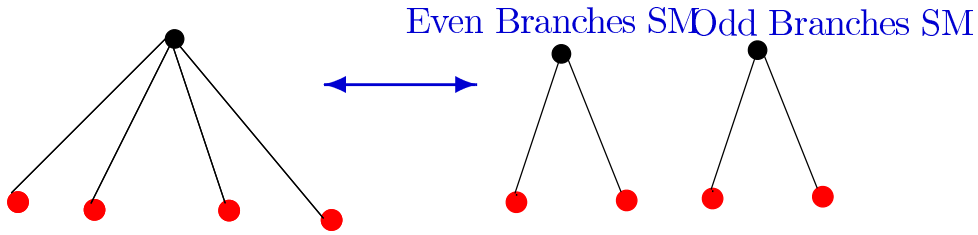
Using this example, figure (4.2) defines the parallel stack decoder which will treat two layers in each step by duplicating the treatment and keeping only one memory to store children.

Throughout figure(4.1), we can understand the importance of parallel processing in speeding execution of decoding. The new lattice representation permits to calculate two dimensions in only one step thanks to the independence between the two last layers of the generating lattice matrix. Then, the run time is twice lower since two dimensions are crossed over for each node computation. Figure (4.3) shows the first step in parallel decoding. we use : ϖ to design node, b to design branch, f is the node cost and ϱ is the branch weight. It consists on generating two layers simultaneously. Paths $[\underline{s}_3\underline{s}_4]$ are then generated, and one gets 4 candidates which are the combination of two possible estimation for each tree, costs are also combined (the sum). In the following step, the algorithm will select the best candidate among these four and will proceed similarly.

Assume the decoder chooses $[\underline{s}_3\underline{s}_4] = [1, 1]$. In the next step the decoder acts as follows : the list of new candidates found in the step described in figure (4.4) is added to the previous list. The algorithm continues to run by selecting the most promising node among all nodes in the complete list stored in the stack. This example gives an illustration overview of the general algorithm to understand the idea behind parallel processing for stack decoding. The proposed structure is optimal in performance and keeps the same ML results.



(a) The new lattice permits to calculate the four nodes simultaneously



(b) The four nodes calculation is distributed on two state machines

Figure 4.1 – Parallel processing principle

In figure (4.5), a hardware architecture of the standard stack decoder is proposed. The stack architecture is composed of 3 main state machines(SM) :
- the first one performs the nodes computation, especially the cost function,
- the second one ensures the stack management, i.e. the reordering of its elements according to their cost.
- The last one masters the overall processing, including the loading of the input data and the output of the decoding processing.

Due to the specificity of the stack algorithm, where the number of reading/writing operations in memory is large, it has been decided :
- to allow a concurrent operation of the 2 first state machines,
- to scale down the arithmetic processing capability so that both SMs operates in approximately the same duration for each new node computation.

The other components of the stack are memories or registers. In the figure(4.6), we propose a parallel stack decoder hardware architecture. More state machines are in use for the architecture of the parallel stack decoder in figure (4.6) exploiting the parallelism which enhances the decoder speed.

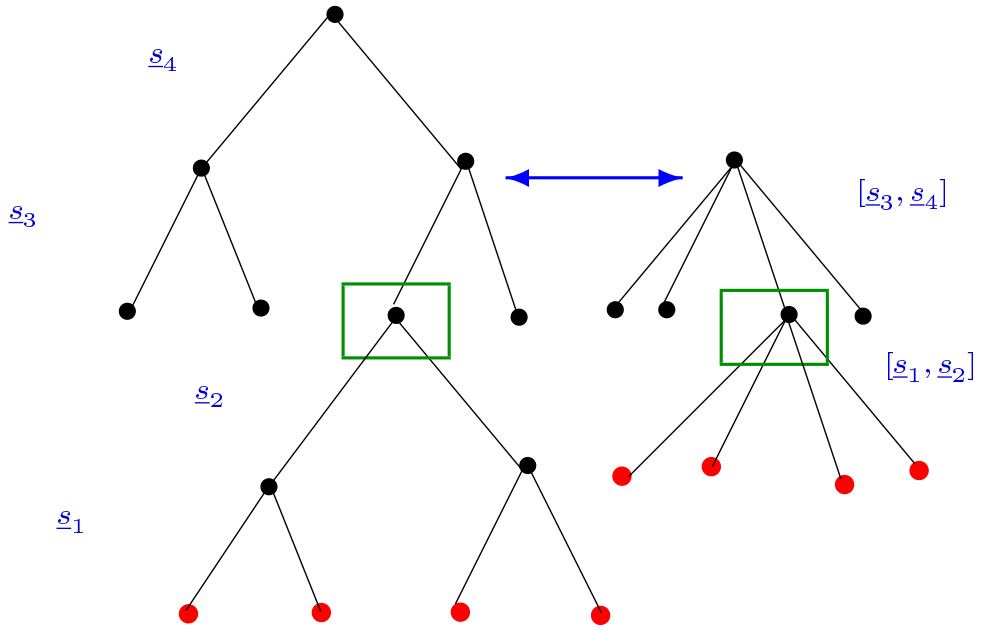


Figure 4.2 – Parallel processing : Four dimension in two steps

4.3 Optimized Stack Decoding Strategies

4.3.1 Child - Sibling Stack Decoding Strategy

In this strategy, in each step of stack decoding, one child and one sibling are generated instead of all possible childs of one node. The decoder decomposes the equivalent cost function as follows

$$
\begin{aligned}
f(2n,\underline{\mathbf{y}}) &\triangleq \left|\underline{\mathbf{y}} - \underline{\mathbf{R}}\underline{\mathbf{s}}\right|^2 \qquad (4.2)\\
&= \left|\begin{bmatrix} y_1 \\ \vdots \\ y_{2n-1} \\ y_{2n} \end{bmatrix} - \begin{bmatrix} \underline{r}_{1,1} & \cdots & \underline{r}_{1,2n-1} & \underline{r}_{1,2n} \\ 0 & \ddots & \ddots & \vdots \\ \vdots & \ddots & \underline{r}_{2n-1,2n-1} & \underline{r}_{2n-1,2n} \\ 0 & \dots & 0 & \underline{r}_{2n,2n} \end{bmatrix} \begin{bmatrix} \underline{s}_1 \\ \vdots \\ \underline{s}_{2n-1} \\ \underline{s}_{2n} \end{bmatrix}\right|^2 \\
&= \left|\underline{y}_{2n} - \underline{r}_{2n,2n}\underline{s}_{2n}\right|^2 + \left|\begin{bmatrix} \underline{y}_1 - \underline{r}_{1,2n}\underline{s}_{2n} \\ \vdots \\ \underline{y}_{2n-1} - \underline{r}_{2n-1,2n}\underline{s}_{2n} \end{bmatrix} - \begin{bmatrix} \underline{r}_{11} & \cdots & \underline{r}_{1,2n-1} \\ 0 & \ddots & \vdots \\ 0 & \ddots & \underline{r}_{2n-1,2n-1} \end{bmatrix} \begin{bmatrix} \underline{s}_1 \\ \vdots \\ \underline{s}_{2n-1} \end{bmatrix}\right|^2 \\
&= \left|\underline{y}_{2n} - \underline{r}_{2n,2n}\underline{s}_{2n}\right|^2 + f(2n-1,(\underline{\mathbf{y}} - \underline{\mathbf{r}}_{2n}\underline{s}_{2n})_{\setminus 2n}) \qquad (4.1)
\end{aligned}
$$

This first stage of decomposition leads to write the overall cost function as the sum of a partial cost associated with the optimization variable $\underline{s}_{2n}$, i.e., the squared distance $\left|\underline{y}_{2n} - \underline{r}_{2n,2n}\underline{s}_{2n}\right|^2$, and a cost sub-function $f(.,.)$ whose second argument is the residual target where $\underline{\mathbf{r}}_{2n}$ is the $2n^{th}$ column of the matrix $\underline{\mathbf{R}}$. For the more general case, i.e., the subsequent stages of the decomposition, a residual target may be of dimension $\underline{d} = 2n,...,0$ and respectively encapsulate

$[\ \underline{s}_3\underline{s}_4]$	$\mathrm{f}(\varpi_{kl})=f_1(\varpi_k)+f_2(\varpi_l)+f(\varpi_0)$
[-1,1]	$\mathrm{f}(\varpi_{21})=f_1(\varpi_2)+f_2(\varpi_1)+f(\varpi_0)$
[-1,-1]	$\mathrm{f}(\varpi_{11})=f_1(\varpi_1)+f_2(\varpi_1)+f(\varpi_0)$
[1,1]	$\mathrm{f}(\varpi_{22})=f_1(\varpi_2)+f_2(\varpi_2)+f(\varpi_0)$
[1,-1]	$\mathrm{f}(\varpi_{12})=f_1(\varpi_1)+f_2(\varpi_2)+f(\varpi_0)$

FIGURE 4.3 – Parallel Processing Stack Decoding : estimation of $\underline{s}_4$ and $\underline{s}_3$

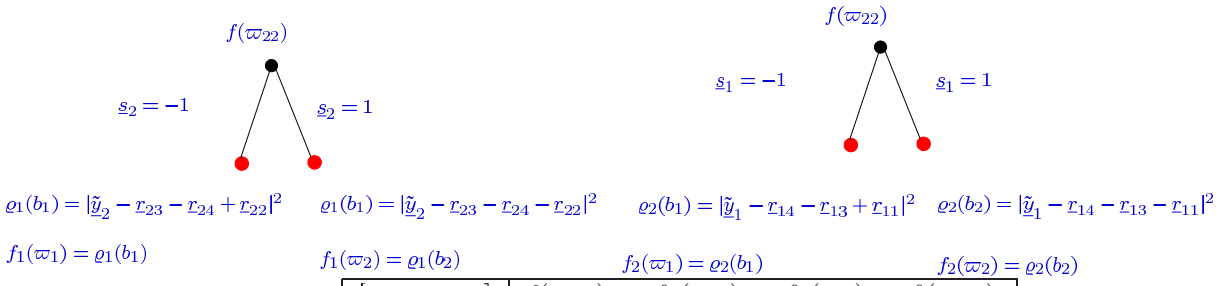
the dependence of its associated cost sub-function on the last 0,...,$2n$ optimization variables. Thus leading to the following recursive definition of residual targets :

$$\underline{\tilde{\mathbf{v}}}\left(\underline{d},\mathbf{s}_{\underline{d}+1}^{2n}\right)\triangleq\begin{cases}\underline{\mathbf{y}} & \underline{d}=2n\\ (\underline{\mathbf{y}}-\underline{\mathbf{r}}_{2n}\underline{s}_{2n})_{\setminus 2n} & \underline{d}=2n-1\\ \vdots & \vdots\\ (\underline{\tilde{\mathbf{v}}}\left(\underline{d}+1,\mathbf{s}_{\underline{d}+2}^{2n}\right)-\underline{\mathbf{r}}_{\underline{d}+1}\underline{s}_{\underline{d}+1})_{\setminus\underline{d}+1} & \underline{d}=2n-2,\ldots,1\\ \vdots & \vdots\\ \emptyset & 0\end{cases}\tag{4.2}$$

$\underline{\tilde{\mathbf{v}}}$ resides in the same orthogonal space as $\underline{\mathbf{y}}$. Equation 4.1 can then be also written as follows

$$\begin{aligned}\left|\underline{\mathbf{y}}-\underline{\mathbf{R}}\underline{\mathbf{s}}\right|^2 &= \left|\underline{\tilde{v}}(2n,\emptyset)_{2n}-\underline{r}_{2n,2n}\underline{s}_{2n}\right|^2+\left|\underline{\tilde{\mathbf{v}}}(2n-1,\underline{s}_{2n})-\underline{\mathbf{R}}_{\setminus\underline{2n},2n}\underline{s}_{\setminus 2n}\right|^2\\ &= \sum_{\underline{d}=2n}^{1}\left|\underline{\tilde{v}}(\underline{d},\mathbf{s}_{\underline{d}+1}^{2n})_{\underline{d}}-\underline{r}_{\underline{d}\underline{d}}\underline{s}_{\underline{d}}\right|^2\end{aligned}\tag{4.2}$$

In the summation (4.2), each term can be interpreted as the partial cost incurred by assigning a particular value to each variable $\underline{s}_{\underline{d}}$ in the dimension $\underline{d}$. Therefore, the accumulation of these costs over all $2n$ variables is precisely the value of the overall cost function.
In this sequel, one can observe that it's often the case that only one or perhaps two children per set of siblings are selected from the nodelist and themselves expanded. In other words, at the termination of the algorithm, many more nodes than needed remain in the Stack. One can reduce space and time complexity by modifying the expansion procedure so that unnecessary children are not placed in the border nodelist. That's why , the expansion procedure can be optimized by computing only the first sibling of each set in the first instance and only the first child of a parent node instead of all children. The next sibling nodes are then computed as each node in the set is expanded. Siblings are enumerated in order of non decreasing weight. The smallest weight sibling is generated first. In order to execute such a node expansion procedure, the node



$[\ \underline{s}_1\underline{s}_2\underline{s}_3\underline{s}_4]$	$\mathrm{f}(\varpi_{kl}) = f_1(\varpi_k) + f_2(\varpi_l) + f(\varpi_{22})$
[-1,1,1,1]	$\mathrm{f}(\varpi_{21}) = f_1(\varpi_2) + f_2(\varpi_1) + f(\varpi_{22})$
[-1,-1,1,1]	$\mathrm{f}(\varpi_{11}) = f_1(\varpi_1) + f_2(\varpi_1) + f(\varpi_{22})$
[1,1,1,1]	$\mathrm{f}(\varpi_{22}) = f_1(\varpi_2) + f_2(\varpi_2) + f(\varpi_{22})$
[1,-1,1,1]	$\mathrm{f}(\varpi_{12}) = f_1(\varpi_1) + f_2(\varpi_2) + f(\varpi_{22})$

FIGURE 4.4 – Parallel Processing Stack Decoding : estimation of $\underline{s}_1$ and $\underline{s}_2$

data structure should be augmented by the addition of four components :
♢ the position with respect to its siblings $\iota \in \{1, \ldots, N_b\}$
♢ the weight f' of its parent node
♢ the unconstrained target $\underline{\tilde{v}}'$ of its parent node.
♢ the constrained value $\underline{s}^-$ of its previous sibling.
These additional components are required to determine the value associated with the next sibling in the set, as well as to compute the properties of the next sibling node. This new expansion can then be defined as follows :

Definition

Given upper triangular matrix $\underline{\mathbf{R}}$, alphabet $\mathcal{A} = \{\underline{x}_1, \ldots, \underline{x}_q\}$ of size q and non leaf node ϖ, let expanding ϖ be defined as generating its first child node ϖ_c and its next sibling ϖ_s if it exists. The time and space complexities of the stack decoding can be significantly reduced because of a smaller number of node. But, in other hand, there's an increase in the amount of data to be stored for each node. Also, an increase in the computations is required for each expansion.
The elements of $\mathcal{A}$ can be written as (for the 4-ASK modulation case)

$$\underline{x}_k = \underline{x}_{min} + (k-1)\Delta\underline{x} \tag{4.2}$$

for some lower bound $\underline{x}_{min}$, inter-element separation $\Delta\underline{x}$, and indices $k = 1, \ldots, q$. For an unconstrained value $\underline{x}_0 \in \mathbb{R}$, the nearest element is

$$\underline{x}_1 = \arg\min_{\underline{x} \in \mathcal{A}} |\underline{x} - \underline{x}_0|^2 \tag{4.2}$$

Thus, the algorithm generated the second and subsequent elements of $\mathcal{A} = \{\underline{x}_2, \ldots, \underline{x}_q\}$ in order of non decreasing squared distance from $\underline{x}_0$.

Through Figures (4.7), an illustration of the way the algorithm of stack decoding with child sibling generation, visited symbols. The argument values $\underline{x}_k$ are shown as filled dots, those representing previously processed values as circled crosses, and the computed next values are designed by the arrow. Others crosses represent values that are valid but that have not been selected.

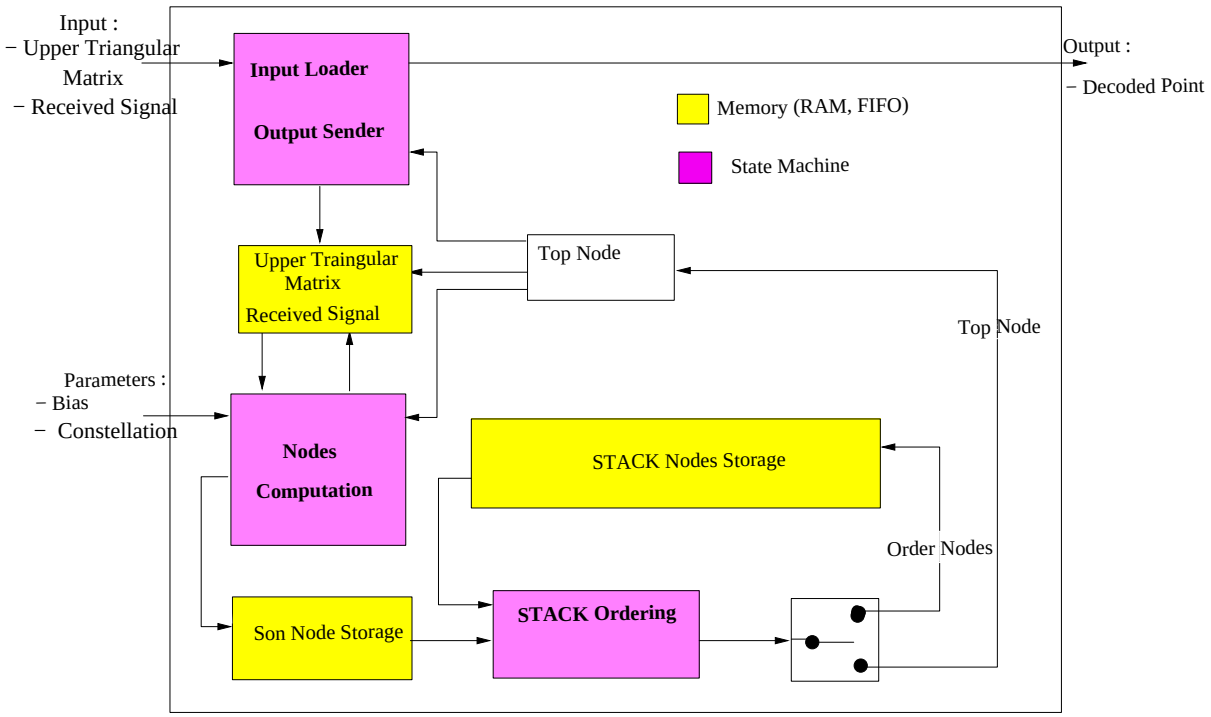


Figure 4.5 – The Standard Stack Decoder Hardware Architecture

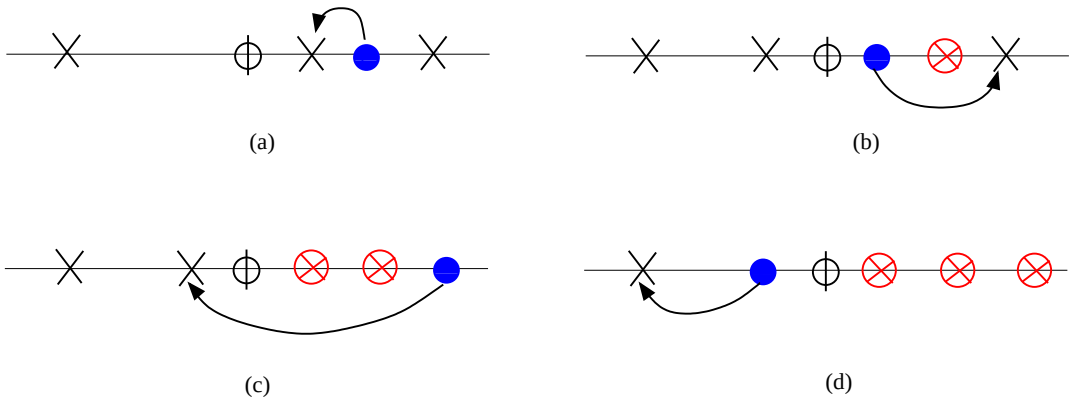


Figure 4.7 – 4-ASK order of visited symbols

4.3.2 Complex Stack Decoding Strategy

The idea is to perform detection over complex symbol alphabets. As before, branch weights are non negative and node weights are monotonically non decreasing along rooted paths. Therefore, the basic ordered traversal detection philosophy could be extended directly to operations over complex alphabets. However, the expansion procedure would then involve generating all N_b^2 complex child nodes and the additional complexity incurred by such an approach would be far greater than the savings afforded by the complex QR factorization. A more desirable idea is to

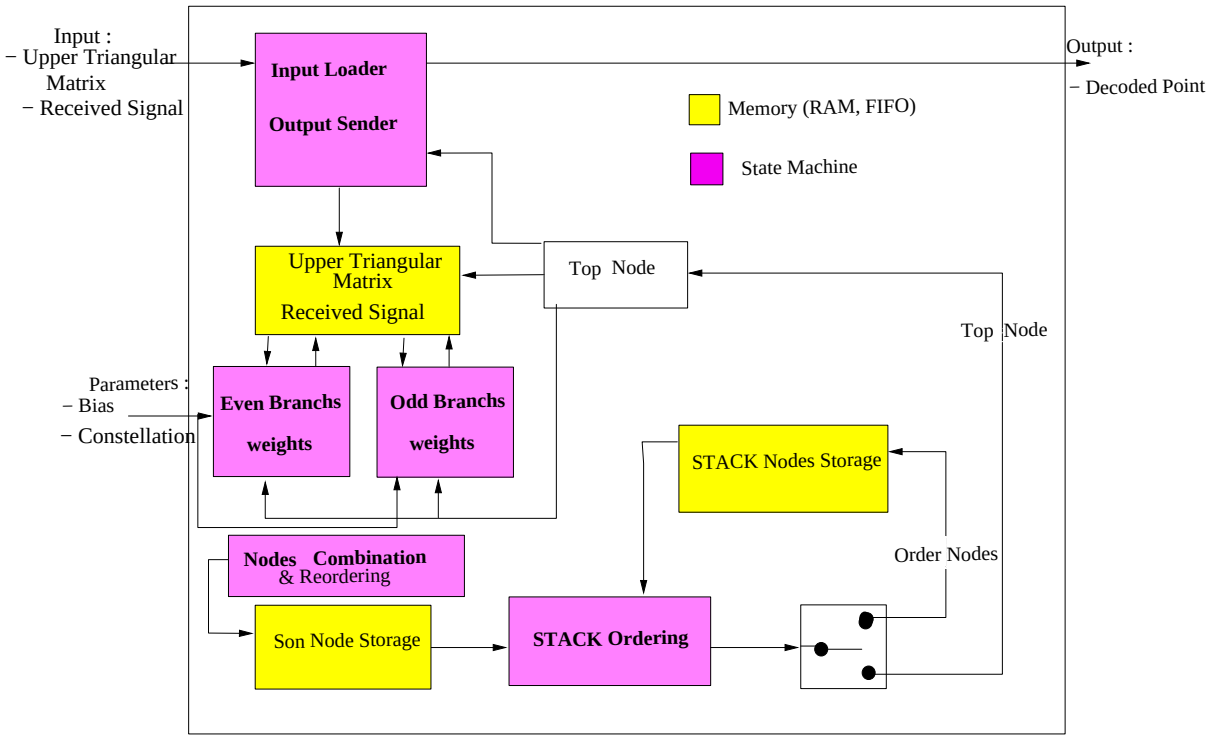


Figure 4.6 – Parallel Stack decoder hardware architecture

extend the Child-Sibling Algorithm to complex symbol alphabets. But unlike the real numbers, the complex number field lacks a natural notion of ordering. The root node is associated with the unconstrained value x_0 and its first child with the nearest element

$$\begin{aligned} x_1 &= \arg\min_{\underline{x}\in\mathcal{A}} \left|\underline{x} - \underline{x}_0^R\right| + j \arg\min_{\underline{x}\in\mathcal{A}} \left|\underline{x} - \underline{x}_0^I t\right| \\ &= \underline{x}_1^R + j\underline{x}_1^I \end{aligned} \tag{4.2}$$

which is the combination of the nearest real and imaginary values. We define the weight of each node as the sum of the weight of the root node and the squared distance from its associated constraint value to the unconstrained value. Thus, the node weights are monotonically non-decreasing along root paths as well as across sets of siblings. Therefore, we can enumerate the nodes of the tree in order of non decreasing weight by maintaining a border nodelist and removing the smallest weight elements and generating its first child and its next sibling nodes.
Figures (4.9)and (4.10) show the way the complex stack visited symbols in 4-dimension(ex 4×4 BLAST) 16-QAM constellation.

In Figure (4.11), we distinguish two kinds of symbols : type 1 and type 2 symbols. We introduce this notion of type so that the algorithm will not visit twice the same node. For type 1 nodes, generating both siblings in the two dimension is permitted. In other hand, type 1 nodes are only permitted to generate nearest sibling in only one dimension. An additional node component called type is required. The expansion procedure may produce one or two new border nodes in the same level

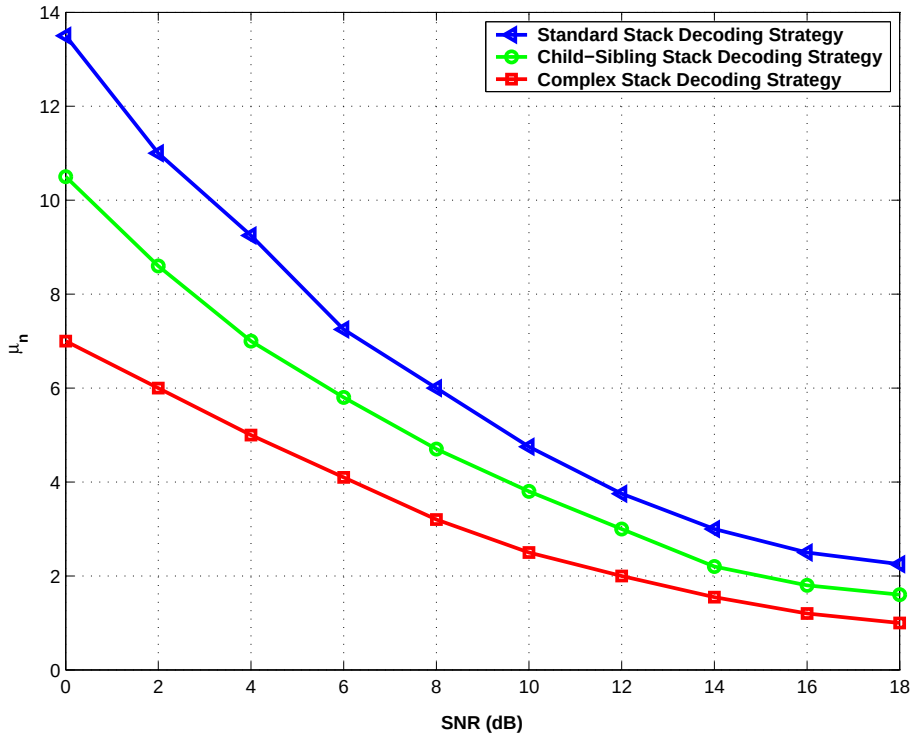


Figure 4.8 – Comparison of different Stack Decoding Strategies in terms of visited nodes for a 4×4 system with spatial multiplexing and 16-QAM constellation

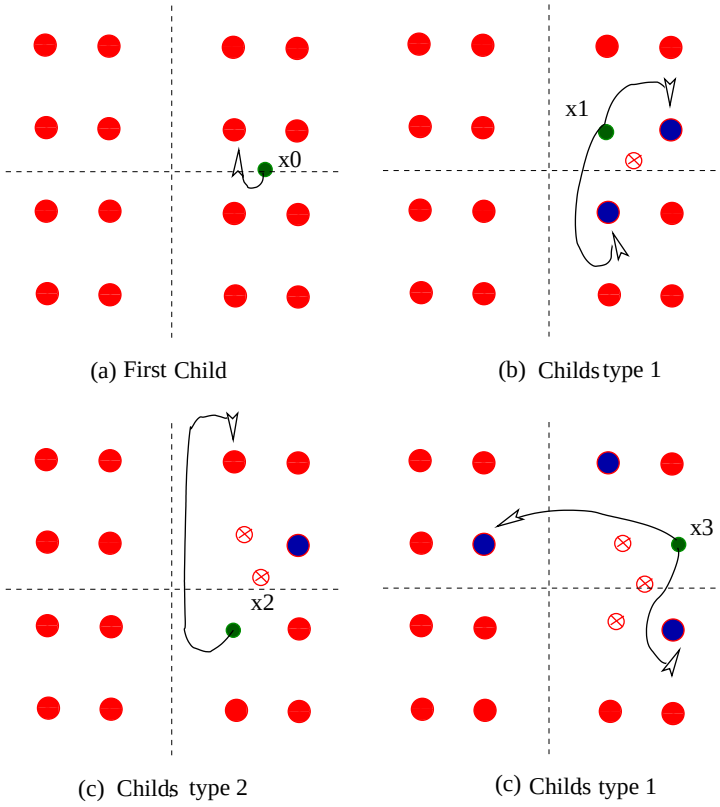


Figure 4.11 – 16-QAM order of visited symbols

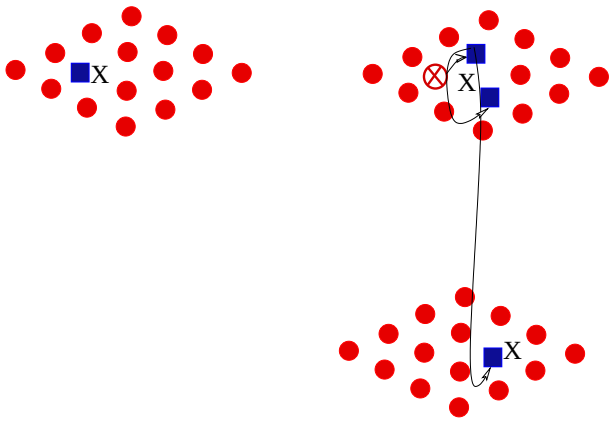


Figure 4.9 – Complex Stack decoding : (a) First step : x_4 symbol detection , (b) Second step x_3 symbol detection

Figure (4.8) shows the improvement introduced by the use of complex stack decoding strategy and the child-sibling strategy compared to the original stack decoding. The number of visited nodes is reduced. One can't also negligate the reduced complexity of QR decomposition when using the complex stack decoding strategy, since this decomposition is held in less dimension.

4.4 Early Termination Stack Decoding

A drawback of the sphere decoders family (SD and SE) is that, for close-to-ML performance, complexity remains high in the low signal-to-noise ratio (SNR) regime or when there's correlation. Clipping can be an evident solution but it induces a big loss in performance. For severe channels, these decoders seem to be converted to exhaustive search decoders visiting all constellation points. Then, clipping for a fixed time provide a non-optimized point.

The question involving this issue is : How to resolve a slow decoding for the Stack Decoder ? Slow decoding is the major problem for actual used ML decoders. All proposed ML decoders suffer from big complexity for low SNR and for Correlated Channels. Providing ML solution in these conditions with usual time constraint seems to be unreachable for the moment. Researchers focused on how to stop algorithm when it exceeds fixed time. This constraint of time is important in practical realizations. In the following, we will detail two techniques that we propose for the case of hard and soft decoding and one last technique that can be used for both cases.

4.4.1 ZF-DFE and K-BEST early termination for hard and soft decoding

For this case, we propose to finish the stack decoding by a ZF-DFE decoding. To put this idea into practice, we should add a control system to compute time elapsed for decoding one codeword.

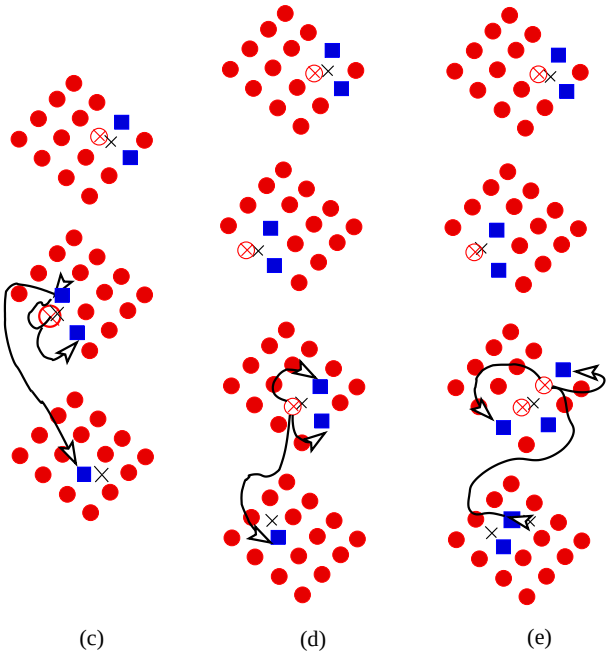


Figure 4.10 – Complex Stack decoding : (c) Third step : x_2 symbol detection , (d)(e) Fourth step x_1 symbol detection

When time is exceeded, a message will be sent to the stack decoder block to inform it to move to the second regime : ZF-DFE.

This move from the standard execution to the ZF-DFE is simple to put in practice especially with the easy structure of stack decoding. Stack Decoding can be transformed to a ZF-DFE decoder keeping the same execution architecture and only limiting the size of memory to one candidate. We take the best candidate in the list and we erase all others. Once generating all children, we pick only the best one and reject others. This process will lead to get rapidly a solution which is not optimal. The solution is not ML but it's better than the ZF-DFE point because the DFE process started later with an intermediate point in the tree. Nevertheless, the structure of this algorithm permits to get also ZF-DFE point directly if we fix time control to zero. In this case, the algorithm will use the stack decoder structure to get exactly the same ZF-DFE solution. Figure (4.14) shows the transition from Stack decoding to ZF-DFE for the case of 4-QAM (two children) and 3 symbols to be decoded (6 in real dimension). In the first phase, stack decoding spent a lot of time to generate nodes. The SNR is low ; the stack decoder met difficulties to pursue a strict direction and is lost in visiting nearly all nodes. Once the time constraint is elapsed, the decoder will pick the best candidate and reject others. A quick result is then provided with the ZF-DFE process.

For the case of soft decoding, Figure (4.15) shows the transition from Stack decoding to K-BEST for the case of 16-QAM (four children) and 3 symbols to be decoded (6 in real dimension). In the first phase, stack decoding spent a lot of time to generate nodes. The SNR is low ; the stack decoder met difficulties to pursue a strict direction and is lost in visiting nearly all nodes. Once the time constraint is elapsed, the decoder will pick the best candidate and reject others.

A quick result is then provided with the K-BEST process. K-BEST provides k candidates at the end of decoding.

Since Time constraint is related to the number of operations which are performed during this time. Time constraint can be translated also into complexity constraint. Thus, we can set a time or a complexity constraint. The multiplicative complexity constraint is for example the maximal number of multiplications allowed to decode the signal. In the sequel, for our simulations we will set a complexity constraint in terms of multiplications. We will compare the stack decoding using the ZF-DFE early termination and the SD using a clipping. Decoding clipping can be defined as stopping the algorithm when the configured constraint is reached. Concerning the SD algorithm, clipping consists of selecting the actual visited point inside the sphere as an output of the decoding process. Figure (**??**) shows that stack decoding using the ZF-DFE early termination outperforms the SD algorithm with clipping under a constraint of complexity in terms of multiplications.

4.4.2 Bias update for early termination

We propose here to accelerate the stack decoder by increasing the bias. As for other methods, one should control time elapsed for decoding one codeword. When time is exceeded, a message will be sent to the stack decoder block to inform it to update the bias (increase). This implies that there is a small optimistic bias in the beginning and it's not enough to accomplish decoding in needed time.

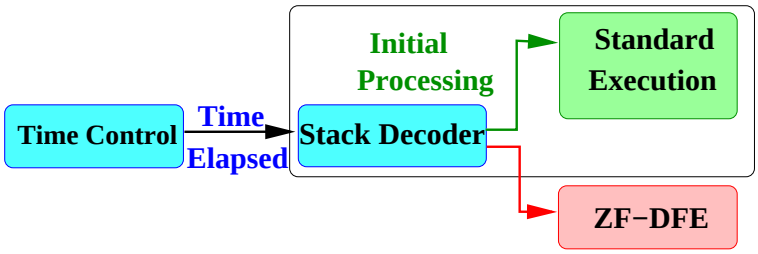


Figure 4.12 – Early Termination Control for hard decoding

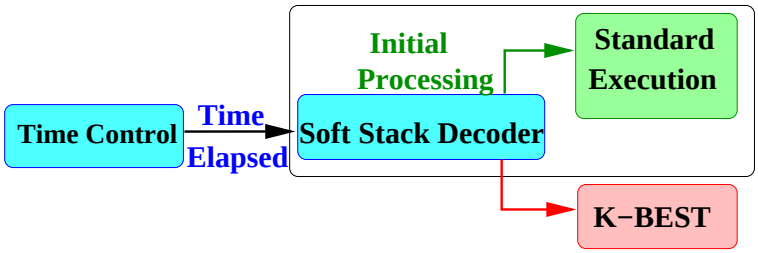


Figure 4.13 – Early Termination Control for soft decoding

Big bias favor more advanced paths in the tree since bias is the product of dimension times a constant value b. Thus, advanced paths will reach a leaf node quicker. To accomplish this task, stack nodes should be arranged in order of dimension. Dimension can be stored for each node

during stack decoding process or also can be determined using the path. The number of allocated values in each node path is equal to the dimension. Once arranged, a supplement bias will be subtracted for each node as function of its dimension. A node at dimension k will see its cost decreasing of k*b. The reduction in costs will not be uniform for all stack nodes. Nodes with higher dimension will get a big decrease and will gain some steps in stack cost ranking. The stack will be arranged another time with cost values criterion, and one can remark that nodes with higher dimensions advanced in the stack rank and are fortunate to reach a leaf node of the tree with a first rank in the stack.

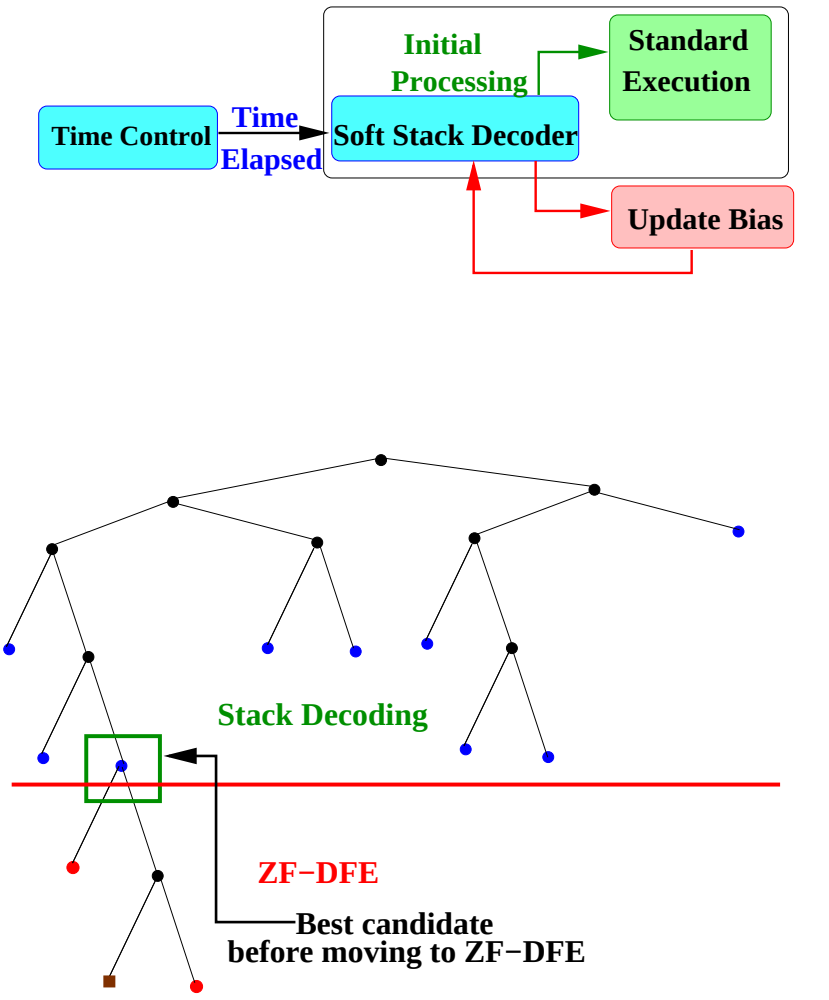
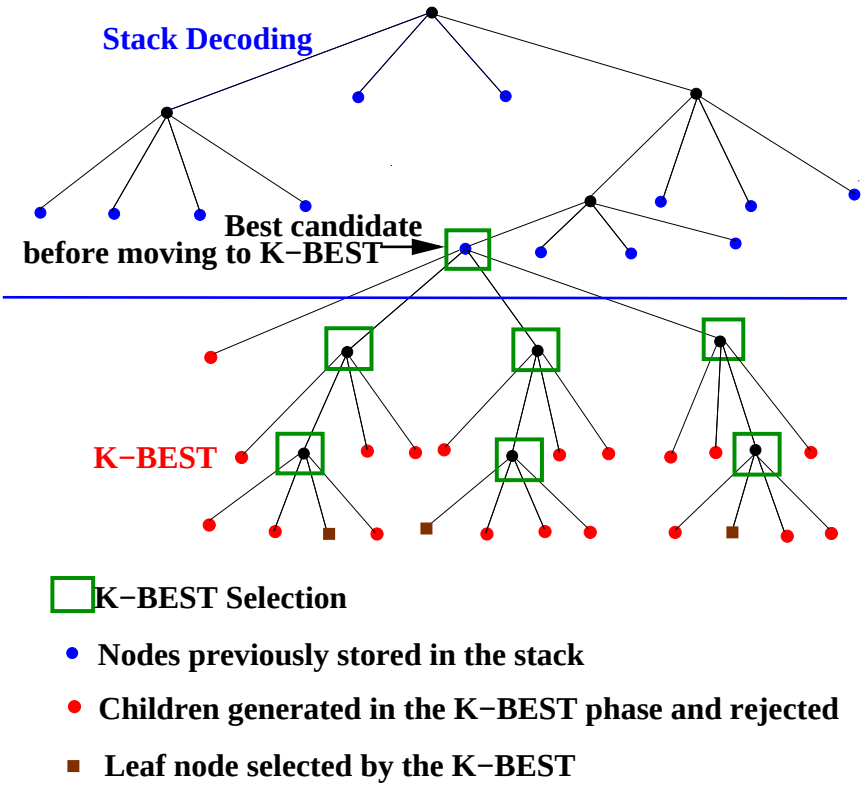


Figure 4.14 – Stack decoding to ZFE-DFE transition



K−BEST Selection

- Nodes previously stored in the stack
- Children generated in the K−BEST phase and rejected
- Leaf node selected by the K−BEST

Figure 4.15 – Stack decoding to K-BEST transition (k=3)

4.5 Conclusion

In this chapter, stack decoder is shown to be capable of supporting parallel processing. Our main contribution was to exploit a novel lattice matrix representation to make the parallel processing possible. By exploiting the advantage of the new sparser matrix a good improvement in run time was distinguished. The simulation results show that the proposed parallel decoder outperforms the classical one in terms of complexity, keeping the same ML performances. Moreover, due to the stack decoder properties, the parallel processing is also easily implemented in practice. This algorithm can further be enhanced by looking for space time codes that introduce more zeros in the lattice representation matrix. Thus, parallel processing can be done for 3 layers or even more in only one step.
Then, we detailed the child-sibling strategy and the complex domain strategy. We showed also that using these two techniques we can enhance the stack decoding complexity and reduce the number of visited nodes.

At the end of this chapter, we focused on the early termination in order to avoid clipping which can be an evident solution but which induces a big loss in performance. Thus, we proposed to terminate the stack decoding by a ZF-DFE in the case of hard decoding and with a K-Best in the case of soft decoding.

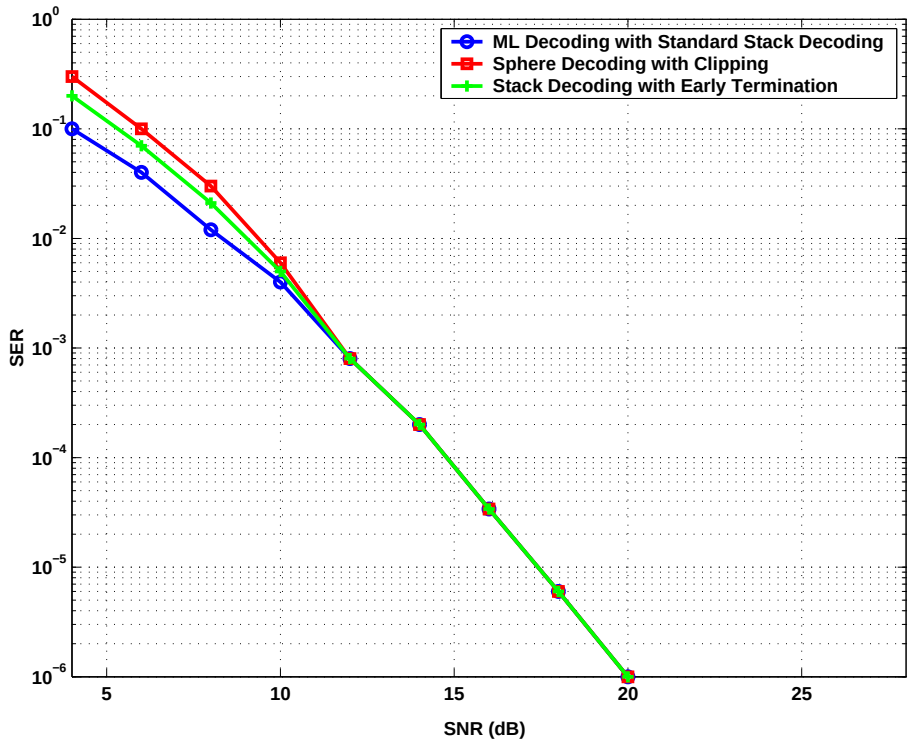


Figure 4.16 – Comparison of Sphere Decoding with clipping and Stack Decoding with ZF-DFE early termination with a multiplication complexity constraint of 1700 operations per codeword for a 2×2 MIMO system with spatial multiplexing and using a 16-QAM constellation.

Conclusions and perspectives

Conclusions

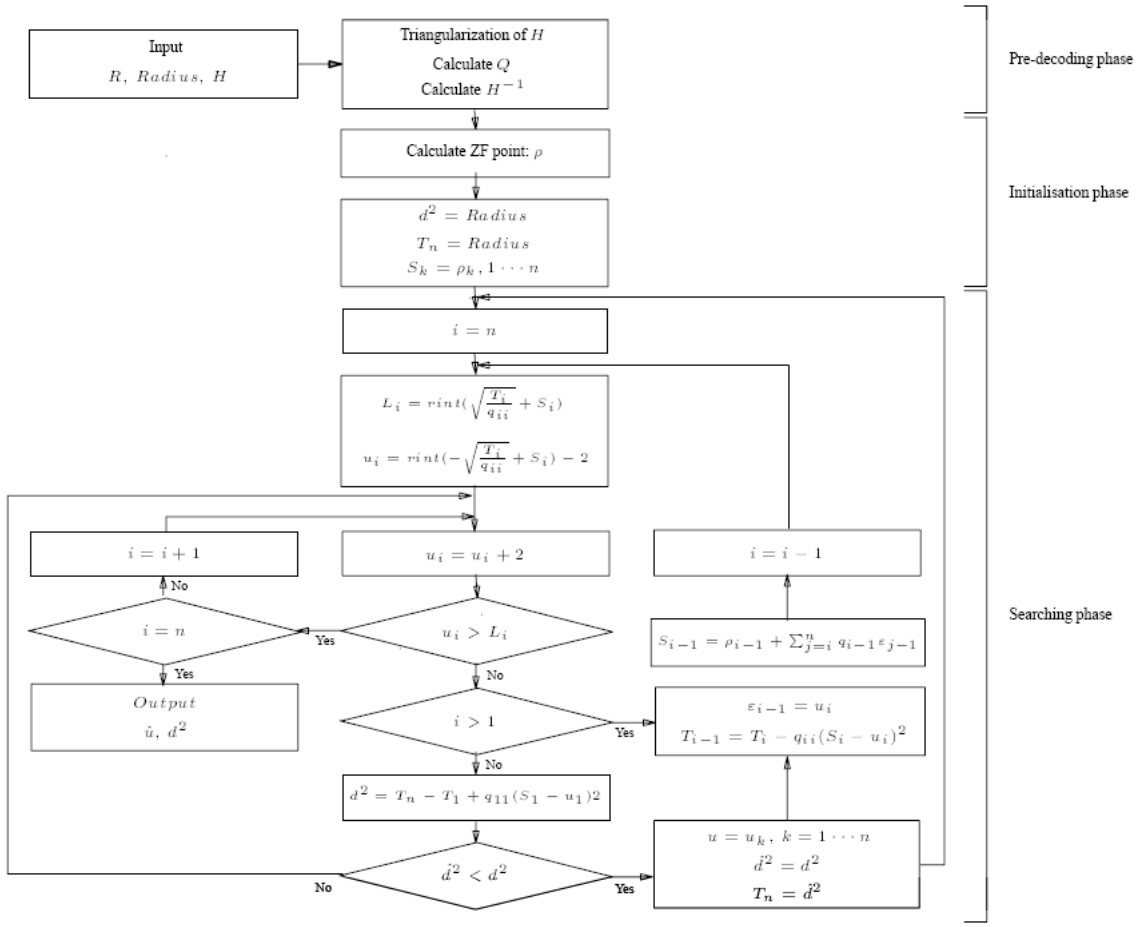
At the outset, the objective for this thesis was to explore the performance and complexity of MIMO decoders for communication systems using a linear dispersion space-time coding in the case of coherent transmission with quasi-static channel. After analyzing, implementing, simulating and comparing a large number of algorithms, all the while gathering big quantities of data, it is time to look at the results and draw conclusions. The complexity analysis presented in this thesis are useful, because they provide information that is needed for future practical implementations of MIMO decoders that meet economic constraints of cost, power consumption, and reliability.
The first two chapters allowed to establish a state of the art of MIMO systems, the necessary prerequisite to answer the problematic of this thesis. So, the first chapter begins with a description of the MIMO channel. The purpose of this chapter is to present the various techniques of modeling the MIMO radio channel. Fading phenomenon and diversity techniques are also described. A plethora of information theory concepts and techniques for transmission over MIMO channels are given. The second chapter shows that lattice representation of linear dispersion codes allows the use of lattice decoding algorithms on the reception side. The most known of them are the sphere decoder and the Schnorr-Euchner decoder. Thus, we compared lattice decoders with suboptimal decoders like the ZF, ZF-DFE and the MMSE. Then, we introduced sequential decoders and different tree search strategies. Recently, decoders based on sequential search inside a tree, like Fano and the Stack decoders, were generalized for the decoding of MIMO systems. These decoders offer a natural solution for the problem of choosing an initial radius which is a problem faced in the design of sphere decoding. Second, they allow for complexity-performance tradeoff by using the bias parameter to tune and adjust decoder complexity .
After that, soft-output MIMO decoding was studied and some well-known soft decoders in the literature were presented. Pre-processing techniques are also presented. The purpose of preprocessing is to transform the original constrained search problem into a form which is easier to the search algorithm. The left preprocessing, MMSE-GDFE returns a well conditioned channel matrix. The right pre-processing consists in making the channel matrix the most orthogonal. In the third chapter, we propose a new algorithm combining the search region of the SD and the Stack Decoding Search strategy. Our purpose is then to propose a modified version of the stack algorithm in order to reduce the prohibitive computational complexity of the SD and the stack decoder. The proposed decoder enables ML performance with complexity lower than the original decoders. Then, we proposed an extension of the new proposed SB-Stack decoder to support soft information outputs. We have modified this algorithm to generate soft-output information

in the form of LLR. We showed that the soft SB-Stack outperforms other soft decoders in terms of performance and complexity. In the fourth chapter, we exploit a specific lattice representation to define stack decoding with parallel processing. Thus, many instructions can be carried out simultaneously, which is not possible with SE and SD. Then, we introduced some different strategies for expanding tree nodes and we showed that those strategies can enhance the decoder complexity. At the end of this chapter, the limitation of ML decoder due to their inability to provide quick solution for bad channels has been exposed and an early termination control technique was proposed to resolve this problem.

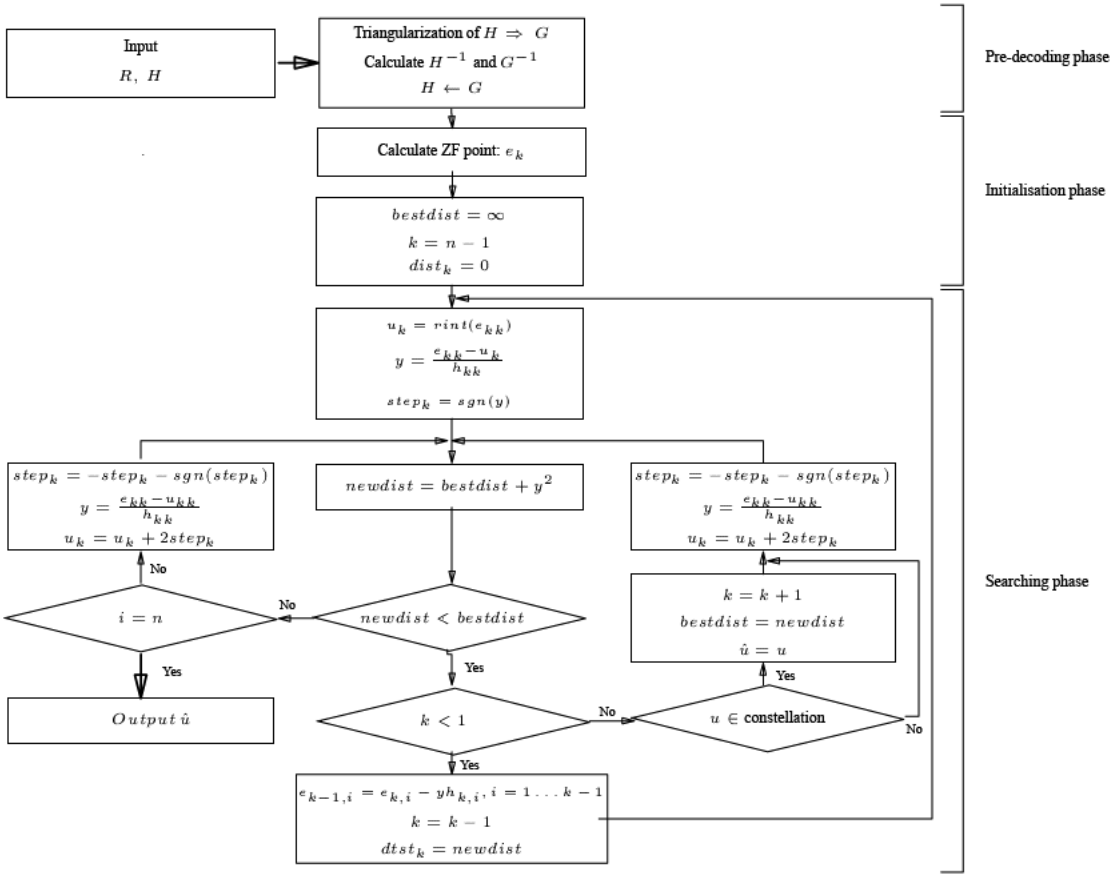
Perspectives

The decoders which we have proposed and studied can be applied to any ST linear dispersion codes. However, we propose in future work to exploit some structures of certain ST codes to conceive adapted, less complex decoders with better performances. Another very interesting idea consists in proposing ST codes with an easy decoding algorithms like the Silver code and its variants (92) (93) (94) and their appropriate decoders.
In the short term, it will be also interesting to pursue comparison of different decoders with other types of channels by introducing correlation and evaluate its effect on complexity and performance of MIMO receivers. Also, the presence of a LOS component is a particular case of radio propagation and it is important to test it. In fact, for professional radio systems, there's no particular propagation case that can be neglected and systems should guarantee a minimum quality of service. A study of all the proposed algorithms can be realized for realistic channel models like TGn channels. Concerning soft decoding, it will be interesting to compare MIMO decoders with an error correcting code which is more powerful than the convolutional code by using for example LDPC codes or turbo-codes which can conduct to different conclusions than obtained here.
The proposed MIMO decoders can also be optimized and improved by integrating the 'turbo' concept to the reception scheme. Thus, we can imagine an iterative Stack decoding and using the bias parameter to tune the complexity-performance tradeoff. One can think also of developing a complexity control algorithm by theoretically establishing a formula for the bias for a targeted complexity and given SNR.
In summary, much remains to be done in this area. Even though work in MIMO systems and space-time codes can be traced back to at least 1998, it still has not left research laboratories. The perennial problem of receiver complexity is still present. Code design and construction is still in large part an open field, as is constellation design. The hypothesis that receivers have perfect channel state information needs to be tested, as well as all the assumptions about the channel model.

Annex A : Flowchart of the SD Algorithm



Annex B : Flowchart of the SE Algorithm



Bibliographie

[1] Lamy.C and Boutros.J, "On Random Rotations Diversity and Minimum MSE Decoding of Lattices," *IEEE Transactions on Information Theory*, July 2000.

[2] Jafarkhani.H, "Space-Time Coding : Theory and Practice," *Cambridge University Press*

[3] Proakis.J.G, "Digital Communications," *McGraw-Hill,* 4th éd., 2001.

[4] Alamouti.S.M, "A Simple Transmit Diversity Technique for Wireless Communications," *IEEE Journal Select. Areas Commun.,* Vol. 16, No. 8, October 1998, pp. 1451-1458.

[5] Jankiraman.M, "Space-Time Codes and MIMO Systems," Norwood, MA, USA : Artech House, Incorporated, 2004

[6] Damen.M.O , Chkeif.A, and Belfiore.J.C, "Lattice Code Decoder for Space-Time Codes," *IEEE Communications letters.,* vol.4 , pp.161-163 , May 2000.

[7] Telatar. E,"Capacity of Multi-Antenna Gaussian Channels," *European Transactions on Telecommunications,* Vol. 10, No. 6, November/December 1999, pp. 585-595.

[8] Wolniansky.P.W, Foschini.G.J, Golden.G.D and Valenzuela.R.A, "V-BLAST : an architecture for realizing very high data rates over the rich-scattering wireless channel," *Bell Labs, Lucent Technologies,* Crawford Hill Laboratory 791 Holmdel-Keyport RD.,Holmdel, NJ07733.

[9] Foschini.G. J and M. J. Gans,"On Limits of Wireless Communications in a Fading Environment When Using Multiple Antennas," *Wireless Personal Communications,*Vol. 6, 1998, pp. 311-335.

[10] Sawahashi.M, Kishiyama.Y and Taoka.H, "UMTS radio access techniques for IMTAdvanced, " *Wireless signal processing and networking workshop : emerging wireless multiaccess technologies at Tohoku University,*fév. 2008.

[11] ITRManager.com, ń Demonstration of mobile telephony of 4th generation in Samsung 4G Forum 2006 ż. http ://www.itrmanager.com/articles/56073/56073.html, sept. 2006.

[12] Mobiledia, ń Samsung to demonstrate 4G mobile technology ż. http ://www.mobiledia.com/news/50638.html, août 2006.

[13] Arogyaswami.P, Nabar.R, and Gore.D, "Introduction to Space-Time Wireless Communications," *Cambridge, UK : Cambridge University Press,* ,2003.

[14] Oyman.O, Nabar.R.U, Bölcskei.H and Paulraj.A.J, "Characterizing the statistical properties of mutual information in MIMO channels," *CIEEE Transactions on Signal Processing,*vol. 51, p. 2784-2795, nov. 2003.

[15] Shiu.D.S, Foschini.G. J, Gans.M. J and Kahn.J. M, "Fading correlation and its effect on the capacity of multi-element antenna systems," *IEEE Transactions on Communications,*vol. 48, no. 3, p. 502-512, 2000.

[16] Biglieri.E and Taricco.G, "How far away is infinity ? Using asymptotic analyses in multiple-antenna capacity calculations," *JWCC Barolo,* Italy, Novembre 2002.

[17] Bölcskei.H, Gesbert.D and Paulraj.A.J, "On the capacity of OFDM-based spatial multiplexing systems," *IEEE Trans. Commun.* vol. 50, no. 2, pp. 225-234, Feb.

[18] Paulraj.A.J and Kailath.T, "Increasing capacity in wireless broadcast systems using distributed transmission/directional reception," U.S. Patent, 1994, no. 5,345,599.

[19] Foschini.J.G, "Layered space-time architecture for wireless communication in a fading environment when using multi-element antennas," *Bell Labs Tech. J.,* pp. 41-59, 1996.

[20] Zheng.L and Tsee.D.N.C, "Diversity and multiplexing : a fundamental trade-off in multiple-antenna channels," *IEEE Transactions on Information Theory,* vol. 49, p. 1073- 1096, mai 2003.

[21] Rekaya.G, "Nouvelles constructions algébriques de codes spatio-temporels atteignant le compromis multiplexage-diversité," *Thèse doctorat,* Telecom Paris, 2004.

[22] Naguib.A, Tarokh.V, Seshadri.N and Calderbank.A.R, " A space-time coding modem for high-data-rate wireless communications," *IEEE Journal on Selected Areas in Communications,* vol. 16, p. 1459-1478, 4th edition 1998.

[23] Tarokh.V, Seshadri.N and Calderbank.A.R, " Space-time codes for rate wireless communication : performance criterion and code construction ," *IEEE Transactions On Information Theory,* vol. 44, p. 744-765, March 1998.

[24] Tirkkonen.O and Hottinen.A," Square-matrix embeddable space-time block codes for complex signal constellations," *IEEE Transactions on Information Theory,* vol. 48, p. 384- 395, February 2002.

[25] Tarokh.V, Seshadri.N and Calderbank.A.R, " Space-time codes for high data rate wireless communication : performance criterion and code construction," *IEEE Transactions on Information Theory,* vol. 44, p. 744-765, mars 1998.

[26] Vucetic.B and Yuan.J, "Space-Time Coding," *John Wiley & Sons Ltd,*2003.

[27] Yuan.J, Chen.Z, Vucetic.B and Firmanto.W, " Performance and design of space-time coding in fading channels ," *IEEE Transactions on Communications,* vol. 51, p. 1991-1996, déc. 2003.

[28] Damen.M, Abed-Meraim.K and Belfiore.J.-C, " Diagonal algebraic space-time block codes ," *IEEE Transactions on Information Theory,* vol. 48, p. 628-636, March 2002.

[29] ElGamal.H and Damen.M.O, " Universal space-time coding," *IEEE Transactions On Information Theory,* May 2003.

[30] Damen.M.O, Tewfik.A and Belfiore.J.-C, "A construction of a space-time code based on number theory," *IEEE Transactions on Information Theory,* vol. 48, p. 753-760, March 2002.

[31] Aktas.D, Gamal.H.E and Fitz.M.P, "Towards optimal space-time coding," *in Proceedings of Asilomar Conference on Signals, Systems and Computers,* vol. 2, p. 1137-1141, November 2002.

[32] Tarokh.V, Jafarkhani.H and Calderbank.R.A, "Space-time block codes from or- thogonal designs," *IEEE Transactions on Information Theory,* vol. 45, p. 1456-1467, July 1999.

[33] Hassibi.B and Hochwald.B.M "High-rate codes that are linear in space and time," *IEEE Transactions on Information Theory,* vol. 48, p. 1804-1824, July 2002.

[34] Alamouti.S, "Space-time block coding : A simple transmitter diversity technique for wireless communications," *IEEE Journal On Select Areas In Communications,* vol. 16, p. 1451-1458, October 1998.

[35] Tirkkonen.O and Hottinen.A, "Complex space-time block codes for four Tx," *in Proceedings of IEEE Global Telecommunications Conference,* vol. 2, p. 1005-1009, November 2000.

[36] Jafarkhani.H, "A quasi-orthogonal space-time block code," *IEEE Transaction on Communication,* vol. 49, p. 1-4, 2000.

[37] Uysal.M and Georghiades.C, "Non-Orthogonal Space-Time Bmock Codes for 3-TX Antennas," *IEEE Electronics Letters,* vol. 38, p. 1689-1691, December 2002.

[38] Sharma.N and Papadias.C, "Full-rate full-diversity linear quasi-orthogonal space-time codes for any number of transmit antennas," *EURASIP Journal on Applied Signal Processing,* vol. 2004, p. 1246-1256, January 2004.

[39] Hassibi.B and Hochwald.B, "High-Rate Codes That are Linear in Space and Time," *IEEE Trans. Inf. Theory,*Vol. 48, No. 7, pp. 1804-1824.

[40] Sandhu.S and Paulraj.A, "Unified design of linear space-time block codes," *in IEEE Global Telecommunications Conference,*vol. 2, p. 1073-1077, nov. 2001.

[41] Gohary.R.H and Davidson.T.N, "Design of linear dispersion codes : Asymptotic guidelines and their implementation," *IEEE Transactions on Wireless Communications,*vol. 4, p. 2892-2906, nov. 2005.

[42] Wang.X, Krishnamurthy.V and Wang.J, "DStochastic gradient algorithms for design of minimum error-rate linear dispersion codes in MIMO wireless systems," *IEEE Transactions on Signal Processing,*vol. 54, p. 1242-1255, avril 2006.

[43] Lenstra.A.K, Lenstra.H and Lovasz.L, "Factoring polynomials with rational coeffi- cients," *Matematische Annalen,*vol. 261, p. 515-534, 1982.

[44] Damen.M.O, El Gamal.H and Caire.G, "On maximum likelihood decoding and the search of the closest lattice point," *IEEE Trans. Info. Theory,*Oct. 2003.

[45] Pohst.M, "On the computation of lattice vectors of minimal length, successive minima and reduced basis with applications," *in ACM SIGSAM,*Vol. 15, pp. 37-44, Bull, 1981.

[46] Kannan.R, "Minkowski's convex body theorem and integer programming," *in Proc, of the ACM Symposium on Theory of Computing,*pp. 193-206, Boston, MA., Apr.1983.

[47] Helfrich.B, "Algorithms to construct Minkowski reduced and Hermite reduced lattice bases," *Theoretical Computer Science,* vol.41, nos.2-3, pp.125-139, 1985.

[48] Fincke.U and Pohst.M, "Improved methods for calculating vectors of short length in a lattice, including a complexity analysis," *Mathematics of Computation,* vol.44, pp.463-471, Apr.1985.

[49] Kannan.R, "Improved algorithms for integer programming and related lattice problems," *in Proc. of the ACM Symposium on Theory of Computing,* pp.193-206, Boston, MA., Apr.1983.

[50] Viterbo.E and Biglieri.E, "A universal lattice decoder," *in GRETSI 14- 'eme Colloque,* Juan-les-Pins, Sept. 1993.

[51] Viterbo.E and Boutros.J, "A universal lattice code decoder for fading channel," *IEEE Trans. Inform. Theory,* Vol. 45, pp. 1639-1642, July 1999.

[52] Agrell.E, Eriksson.T, Vardy.A and Zeger.K, "Closest point search in lattices," *IEEE Trans. Inform. Theory,* Vol. 48, pp. 2201-2214, Aug. 2002.

[53] Schnorr.C.P and Euchner.M, "Lattice basis reduction : improved practical algorithms and solving subset sum problems," *Mathematical Programming,* Vol. 66, pp. 181-191, 1994.

[54] Hassibi.B and Vikalo.H,"On the expected complexity of sphere decoding ," *in Proceedings of Asilomar Conference on Signals, Systems, and Computers,* vol. 2, p. 1051-1055, 2001.

[55] Damen.M.O, ElGamal.H and Caire.G, "On maximum likelihood detection and the search for the closest lattice point ," *IEEE Transactions on Information Theory,* p. 2389- 2402, October 2003.

[56] Zimmermann.E, Rave.W and Fettweis.G, "On the complexity of sphere decoding ," *in Proceeding International Symposium on Wireless Pers,* September 2004.

[57] Rekaya.G,"Nouvelles constructions algébriques de codes spatio-temporels atteignant le compromis multiplexage-diversité," *Thèse doctorat, Telecom Paris,* 2004.

[58] Johannesson.R and Zigangirov.K.S, "Fundamentals of convolutional coding," *IEEE series on digital and mobile communication,* p. 269-315, 1999.

[59] Fano.R.M, "A heuristic discussion of probabilistic decoding," *IEEE Transactions On Information Theory,* vol. 9, p. 64-74, April 1963.

[60] Zigangirov.K.S, "Some sequential decoding procedures," *Probl. Peredach. Inform.,* vol. 2, p. 13-25, 1966.

[61] Murugan.A.D, ElGamal.H, Damen.M. O and Caire.G, "A unified framework for tree search decoding : rediscovering the sequential decoder," *IEEE Transactions On Information Theory,* vol. 52, p. 933-953, March 2006.

[62] "Numerical Recipes," *http ://www.nr.com.*

[63] Johannesson.R and Zigangirov.K. S,"Fundamentals of convolutional coding," *IEEE series on digital and mobile communication,* pp. 269-315, 1999.

[64] Fano.R. M, "A heuristic discussion of probabilistic decoding," *IEEE Transactions On Information Theory,* vol. 9, pp. 64- 74, April 1963.

[65] Boutros.J, Gresset.N, Brunel.L, and Fossorier.M, "Soft-input soft-output lattice sphere decoder for linear channels," *IEEE Global Telecommunications Conference GLOBECOM,* 2003.

[66] Hochwald.B.M and ten Brink.S, "Achieving near-capacity on a multiple-antenna channel," *IEEE Transactions on Communication,* vol. 53, pp. 389-399, March 2003.

[67] ElGamal.H, Caire.G et Damen.M, "Lattice coding and decoding achieve the op- timal diversity-vs-multiplexing tredeoff of MIMO," *submitted on IEEE Transactions On Information Theory,* November 2003.

[68] Minkowsk.H, "Geometrie der zahlen," *Teubner-Verlag,* 1896.

[69] Lenstra.A.K, Lenstra.H and Lovasz.L, "Factoring polynomials with rational coeffi- cients," *Matematische Annalen,* vol. 261, p. 515-534, 1982.

[70] Schnorr.C.P, "A hierarchy of polynomial time lattice basis reduction algorithms," *Theoretical Computer Science,* vol. 53, p. 201-224, 1987.

[71] Kumar.K.R, Caire.G and Moustakas.A.L, "Asymptotic Performance of Linear Receivers in MIMO Fading Channels," *submitted for publication to IEEE Trans. Inform. Theory.*

[72] Wong.K.-w, Tsui.C.-y, Cheng.R. S.-k and Mow.W.-h, "A VLSI architecture of a K-best lattice decoding algorithm for MIMO channels," *in Proceedings of the IEEE International Symposium on Circuits and Systems,* vol. 3, p. 273-276, 2002.

[73] Guo.Z and Nilsson.P, "Algorithm and implementation of the K-best sphere decoding for MIMO detection," *IEEE Journal on Selected Areas in Communications,* vol. 24, p. 491-503, mars 2006.

[74] Taherzadeh.M, Mobasher.A, and Khandani.A. K, " LLL reduction achieves the receive diversity in MIMO decoding," *IEEE Trans. Inform. Theory,* vol. 53, no. 12, pp. 4801-4805, Dec. 2007.

[75] Yao.H and Wornell.G. W, " Lattice-reduction-aided detectors for MIMO communication systems," *in Proc. IEEE Global Communications Conference (GLOBECOM),* vol. 1, Nov. 2002, pp. 424-428.

[76] Wubben.D, Bohnke.R, Kuhn.V, and Kammeyer.K.-D, " Nearmaximum-likelihood detection of MIMO systems using MMSE-based lattice reduction," *in Proc. IEEE International Conference on Communications (ICC),* vol. 2, June 2004, pp. 798-802.

[77] Jalden.J and Elia.P, " LR-aided MMSE lattice decoding is DMT optimal for all approximately universal codes ," *ISIT 2009, IEEE International Symposium on Information Theory,* June 28-July 3rd, 2009, Seoul, Korea , pp 1263-1267

[78] Hagenauer.J and Offer.E and Papke.L " Iterative decoding of binary block and convolutional codes ," *IEEE Transactions on Information Theory,* March 1996, vol.42, no.2, pp.429-445.

[79] Boutros.J , Boixadera.F and Lamy.C, " Bit-interleaved coded modulations for multiple-output channels ," *IEEE 6th Int. Symp. on Spread Spectrum Techniques and App.,* September , 2000, pp 13-25.

[80] Boutros.J, Gresset.N and Brunel.L and Fossorier.M, " Soft-input soft-output lattice sphere decoder for linear channels ," *IEEE Global Telecommunications Conference GLOBECOM,* September , 2003.

[81] Hochwald.B.M and ten Brink.S, " Achieving near-capacity on a multiple-antenna channel," *IIEEE Transactions on Communication,* March, 2003, pp.389-399.

[82] Chin.W.H," QRD Based Tree Search Data Detection for MIMO Communication Systems," *Vehicular Technology Conference, VTC 2005-Spring. 2005 IEEE 61st,* June, 2005, volume3, pp.1624-1627.

[83] Seethaler.D, Matz.G and Hlawatsch.F," An efficient MMSE-based demodulator for MIMO bit-interleaved coded modulation," *IEEE GLOBECOM,* December, 2004, pp.2455-2459.

[84] Tonello.A.M," Space-time bit-interleaved coded modulation with an iterative decoding strategy," *in IEEE Conference on Vehicular Technologies,* vol. 1, pp. 473-478, sept. 2000.

[85] Zehavi.E,"8-PSK trellis codes for a rayleigh channel," *IEEE Transactions On Communications,* vol. 40, pp. 873-884, mai 1992.

[86] Caire.G, Taricco.G and E. Biglieri.E," QBit-interleaved coded modulation ," *IEEE Transactions On Information Theory,* vol. 44, pp. 927-946, mai 1998.

[87] Berrou.C, Glavieux.A and Thitimajshima.P," Near shannon limit error-correcting coding and decoding : Turbo-codes," *in IEEE International Conference on Communications,* vol. 2, pp. 1064-1070, mai 1993.

[88] Gallager.R.G," Low-density parity-check codes," *Thèse doctorat,* 1963.

[89] Seethaler.D, Matz.G and Hlawatsch.F," An efficient MMSE-based demodulator for MIMO bit-interleaved coded modulation," *IEEE GLOBECOM,* December, 2004, pp.2455-2459.

[90] Mac Kay.D and R. M. Neal.R.M," Near shanon limit performance of low density parity-check codes," *Electronic Letters,* vol. 32, p. 1645, août 1996.

[91] Azzam.L, Ayanoglu.E"Reduced Complexity Sphere Decoding for Square QAM via a New Lattice Representation," *IEEE GLOBECOM 2007*, Washington, DC, November 2007.

[92] Hottinen.A and Tirkkonen.O,"Precoder designs for high rate space-time block codes," *in Proc. Conference on Information Sciences and Systems*, Princeton, NJ, March 2004.

[93] Biglieri.E, Hong.Y and Viterbo.E,"On fast-decodable space-time block codes," *IEEE Transactions On Information Theory*, vol. 55, p. 524-530, February 2009.

[94] Hollanti.C, Lahtonen.J, Ranto.K, Vehkalahti.R and Viterbo.E,"On the algebraic structure of the silver code," *IEEE Information Theory Workshop*, Porto, Portugal, May 2008.

.