UNIVERSITAT
POLITÈCNICA
DE VALÈNCIA

Efficient algorithms for iterative detection and decoding in Multiple-Input and Multiple-Output Communication Systems

Doctoral Thesis

by

M. Ángeles Simarro Haro

Supervisors:

Dr. Alberto González Salvador
Dr. Francisco José Martínez Zaldívar

Valencia, Spain
July 2017

To my parents and Antonio

“El éxito es la suma de pequeños esfuerzos
repetidos día tras día.”

“Success is the sum of small efforts
repeated day in and day out”

Robert Collier

Abstract

This thesis fits into the Multiple-Input Multiple-Output (MIMO) communication systems. Nowadays, these schemes are the most promising technology in the field of wireless communications, since they let to take advantage of the spatial dimension together with the dimensions of frequency and time. In this way, the use of this technology allows to increase the rate and the quality of the transmission through the use of multiple antennas at the transmitter and receiver sides. Furthermore, the MIMO technology can also be used in a multiuser scenario, where a Base Station (BS) equipped with several antennas serves several users that share the spatial dimension causing interference. However, employing precoding algorithms the signal of the multiuser interference can be mitigated.

For these reasons, the MIMO technology has become an essential key in many new generation communications standards such as Wireless Local Area Network (WLAN), Worldwide interoperability for Microwave Acces (WiMAX), Long Term Evolution (LTE) or Next Generation Handheld (DVB-NGH). On the other hand, Massive MIMO technology or Large MIMO, where the BS is equipped with very large number of antennas (hundreds or thousands) serves many users in the same time-frequency resource, it is a promising candidate technology for next generations of wireless systems.

Nevertheless, the advantages provided by the MIMO technology entail a substantial increase in the computational cost. Therefore the design of low-complexity receivers is an important issue which is tackled throughout this thesis. To this end, one of the main contributions of this dissertation is the implementation of efficient soft-output detectors, since this stage is considered the most complex part of the communication process.

On the other hand, in a multiuser scenario the amount of computational cost is carried out by the precoding processing in the BS, allowing the development of small and inexpensive terminals. Therefore, other important contribution of this thesis is focused on improving the efficiency of precoding schemes.

First, the problem of efficient soft detection with no iteration at the receiver has been addressed. That is, detectors which only process the received vector. A detailed overview of the most employed soft detectors is provided. Furthermore, the complexity and performance of these methods are evaluated and compared. Additionally, two low-complexity algorithms have been proposed. The first algorithm is based on the efficient *Box Optimization Hard Detector* (BOHD) algorithm and provides a low-complexity implementation achieving a suitable performance. The second algorithm tries to reduce the computational cost of the *Subspace Marginalization with Interference Suppression* (SUMIS) algorithm.

Second, soft-input soft-output (SISO) detectors, which are included in an iterative receiver structure, have been investigated. A SISO detector processes the information received by the channel and also the information provided by the channel decoder in the feedback loop. An iterative receiver improves the performance with respect to no iteration, achieving a performance close to the channel capacity. In contrast, its computational cost becomes prohibitive. In this context, three algorithms are presented. Two of them achieve max-log performance reducing the complexity of standard SISO detectors. The last one achieves near max-log performance with low complexity.

The precoding problem has been addressed in the third part of this thesis. An analysis of some of the most employed precoding techniques has been carried out. The algorithms have been compared in terms of performance and complexity. In this context, the impact of the channel matrix condition number on the performance of the precoders has been analyzed. This impact has been exploited to propose an hybrid precoding scheme that reduces the complexity of the previously proposed precoders. In addition, in Large MIMO systems, an alternative precoder scheme is proposed. The proposed scheme reduces the computational cost with respect to the conventional precoding algorithms while good performance is maintained.

In the last part of the thesis, parallel implementations of the SUMIS algorithm are presented. Several strategies for the parallelization of the algorithm are proposed and evaluated on two different platforms: multicore central processing unit (CPU) and graphics processing unit (GPU). The parallel implementations achieve a significant speedup compared to the CPU version when the number of antennas or constellation order increase, that is the context of Large MIMO. Therefore, these implementations allow to simulate a scalable quasi optimal soft detector in a Large MIMO system much faster than by conventional simulation.

Keywords: MIMO detection, Sphere Decoding, GPU, Box Optimization, efficiency, Precoding.

Resumen

La presente tesis se enmarca dentro de los sistemas de comunicaciones de múltiples antenas o sistemas MIMO (*Multiple-Input Multiple-Output*). Hoy en día, estos sistemas presentan una de las tecnologías más prometedoras dentro de los sistemas comunicaciones inalámbricas, debido a que además de las dimensiones de frecuencia y tiempo también permiten aprovechar la dimensión espacial. De esta forma, a través del uso de múltiples antenas en ambos lados, transmisor y receptor, la tasa de transmisión y la calidad de la misma es aumentada. Por otro lado, la tecnología MIMO puede ser utilizada en un escenario multiusuario, donde una estación base (BS) la cual está equipada con varias antenas, sirve a varios usuarios al mismo tiempo, estos usuarios comparten dimensión espacial causando interferencias multiusuario. Sin embargo, empleando algoritmos de precodificación las interferencias multiusuario son mitigadas.

Por todas estas razones, la tecnología MIMO ha sido adoptada en muchos de los estándares de comunicaciones de nueva generación como por ejemplo W*ireless Local Area Network* (WLAN), *Worldwide interoperability for Microwave Acces* (WiMAX), *Long Term Evolution* (LTE) o *Next Generation Handheld* (DVB-NGH). Por otro lado, la tecnología MIMO Masivo, en la cual la estación base está equipada con un gran número de antenas (cientos o miles) que sirve a muchos usuarios en el mismo recurso de tiempo-frecuencia, se ha convertido en una candidata para ser implementada en los futuros estándares de comunicaciones. Sin embargo, las ventajas proporcionadas por los sistemas MIMO implican un aumento en el coste computacional requerido. Por ello, el diseño de receptores de baja complejidad es una cuestión importante en estos sistemas, la cual será abordada a lo largo de la tesis. Para conseguir esta finalidad, las principales contribuciones de la tesis se basan en la implementación de algoritmos de detección soft eficientes, debido a que esta etapa es considerada una de las más costosas en el proceso de comunicaciones. Por otro lado, en un escenario multiusuario el mayor coste computacional es llevado a cabo en el estado de precodificación el cual es implementado en la estación base, permitiendo de este modo el desarrollo de pequeños y económicos terminales. Por lo tanto, otras de las principales contribuciones de la tesis están centradas en mejorar los esquemas de precodificación.

En primer lugar, el problema de la detección soft eficiente en un sistema receptor sin iteración es abordado. Es decir, detectores que solamente procesan el vector recibido. Una descripción detallada sobre los detectores soft más empleados es presentada. Además, la complejidad y rendimiento de estos métodos han sido evaluados y comparados. Por otro lado, han sido propuestos dos algoritmos de bajo coste. El primer algoritmo está basado en el algoritmo *Box Optimization Hard Detector* (BOHD) y proporciona una baja complejidad de implementación logrando un buen rendimiento. El segundo de los algoritmos propuestos intenta reducir el coste computacional del conocido algoritmo *Subspace Marginalization with Interference Suppression* (SUMIS).

En segundo lugar, han sido investidados detectores de entrada y salida soft (SISO, *soft-input soft-output*) los cuales son ejecutados en estructuras de recepción iterativa. Un detector SISO además de procesar la información recibida por el canal también procesa la información proporcionada por el decodificador de canal a través del bucle de realimentación. El empleo de un receptor iterativo mejora el rendimiento del sistema con respecto a no realizar realimentación, pudiendo lograr la capacidad óptima. Por el contrario, el coste computacional se vuelve prohibitivo. En este contexto, tres algoritmos han sido presentados. Dos de ellos logran un rendimiento óptimo, reduciendo la complejidad de los detectores SISO óptimos que normalmente son empleados. Por el contrario, el otro algoritmo logra un rendimiento casi óptimo a baja complejidad. En la tercera parte de esta tesis, se ha abordado el problema de la precodificación. Se ha llevado a cabo un análisis de algunas de las técnicas de precodificación más usadas, prestando especial atención a su rendimiento y a su complejidad. En este contexto, se ha evaluado el impacto que el número de condición de la matriz de canal tiene en el rendimiento de los precodificadores. Además, se ha aprovechado este impacto para proponer un precodificador híbrido, con el fin de reducir la complejidad de algoritmos previamente propuestos. Por otro lado, en MIMO Masivo, se ha propuesto un esquema precodificador. El algoritmo propuesto reduce el coste computacional con respecto a precodificadores convencionales a la vez que el buen rendimiento es mantenido. En la última parte de la tesis, la implementación paralela del algoritmo SUMIS es presentada. Varias estrategias sobre la paralelización del algoritmo han sido propuestas y evaluadas en dos plataformas diferentes: Unidad Central de Procesamiento multicore (multicore CPU) y Unidad de Procesamiento Gráfico (GPU). Las implementaciones paralelas consiguen una mejora de

speedup cuando el número de antenas o el orden de la constelación incrementa, esto es en el contexto de MIMO Masivo. De este modo, estas implementaciones permiten simular para MIMO Masivo y de forma más rápida que por simulación convencional, un algoritmo soft, el cual presenta rendimiento casi óptimo.

Palabras Clave: detección MIMO, Decodificación Esférica, GPU, Optimización de caja, Eficiencia, Precodificación.

Resum

La present tesi s'emmarca dins dels sistemes de comunicacions de múltiples antenes o sistemes MIMO (multiple-input multiple-output). Avui dia, aquestos sistemes presenten una de les tecnologies més prometedora dins dels sistemes de comunicacions inalàmbriques, debut a que permeten aprofitar la dimensió espacial, a més de las dimensiós de freqüència i temps. D'aquesta forma, a través de l'ús de múltiples antenes en tots dos costats, transmissor y receptor, es pot augmentar la taxa de transmissió i la qualitat de la mateixa. D'altra banda, la tecnologia MIMO es pot utilitzar en un escenari multiusuari, on una estació base (BS) la qual està equipada amb diverses antenes serveix a diversos usuaris al mateix temps, aquests usuaris comparteixen dimensió espacial causant interferències multiusuari. No obstant aò, emprant algorismes de precodificació les interferències multiusuari poden ser mitigades

Per totes aquestes raons, la tecnologia MIMO ha sigut adoptada en molts dels estàndars de comunicacions de nova generació com per exemple *Wireless Local Area Network* (WLAN), *Worldwide interoperability for Microwave Acces* (WiMAX), *Long Term Evolution* (LTE) or *Next Generation Handheld* (DVB-NGH). D'altra banda, la tecnologia MIMO Massiu, en la cual l'estació base està equipada amb un gran nombre d'antenes (centenars o milers) que serveix a molts usuaris en el mateix recurs de temps-freqüència, s'ha convertit en una candidata per a ser implementada en els futurs estàndars de comunicacions. No obstant això, els avantatges proporcionats pels sistemes MIMO impliquen un augment en el cost computacional requerit. Per això, el disseny de receptors de baixa complexitat és una qüestió important en aquests sistemes, la qual serà abordada al llarg de la tesi. Per tal d'aconseguir esta finalitat, les principals contribucions de la tesi es basen en la implementació d'algoritmes de detecció soft eficients, debut a que aquesta etapa es considerada una de les més costoses en el procés de comunicacions. D'altra banda, en un escenari multiusuari el major cost computacional és dut a terme en l'estat de precodificació implementat en l'estació base, permetent d'aquesta manera el desenvolupament de xicotest i econòmics terminals. Per tant, unes altres de les principals contribucions de la tesi estan centrades en millorar els esquemes de precodificació. En primer lloc, és abordat el problema de la detecció soft eficient en un sistema receptor sense interacció. Es a dir, detectors que només processen el vector rebut. Una descripció detallada dels detectors soft més emprats és presentada. A més, la complexitat i rendiment d'aquests métodes han sigut avaluats i comparats. D'altra banda, han sigut proposats dos algorismes de baix cost. El primer algorisme està basat en l'algorisme *Box Optimization Hard Decoder* (BOHD) i proporciona una baixa complexitat d'implementació aconseguint un bon resultat. El segon dels algorismes proposats intenta reduir el cost computacional del conegut algoritme *Subspace Marginalization with Interference Suppression* (SUMIS).

En segon lloc, detectors d'entrada i eixidia soft (SISO, soft-input soft-output) els cuals són eixecutats en estructures de recepció iterativa han sigut investigats. Un detector SISO a més de processar la informació rebuda pel canal també processa la informació proporcionada pel decodificador del canal a través del bucle de realimentació. L'ocupació d'un receptor iteratiu millora el rendiment del sistema pel que fa a no realitzar realimentació, podent aconseguir la capacitat òptima. Per contra, el cost computacional es torna prohibitiu. En aquest context, tres algorismes han sigut presentats. Dos d'ells aconsegueixen un rendiment òptim, reduint la compleixitat dels detectors SISO òptims que normalment són emprats. Per contra, l'altre algorisme aconsegueix un rendiment quasi òptim a baixa complexitat. En la tercera part d'aquesta tesi, s'ha abordat el problema de la precodificació. S'ha dut a terme una anàlisi d'algunes de les tècniques de precodificació més usades, prestant especial atenció al seu rendiment i a la seua complexitat. Dins d'aquest context, l'impacte que el nombre de condició de la matriu de canal té en el rendiment dels precodificadors ha sigut avaluat. A més, aquest impacte ha sigut aprofitat per a proposar un precodificador híbrid , amb la finalitat de reduir la complexitat d'algorismes previament proposats. D'altra banda, en MIMO Massiu, un esquema precodificador ha sigut proposat. L'algorisme proposat redueix el cost computacional pel que fa a precodificadors convencionals alhora que el bon rendiment és mantingunt. En l'última part de la tesi, la implementació paral·lela de l'algorisme SUMIS és presentada. Divereses estratègies sobre la paral·lelizació de l'algorisme han sigut proposades i avaluades en dues plataformes diferents: Unitat central de processament multicore (multicore CPU) i unitat de processament gràfic (GPU). Les implementacions paral·leles aconsegueixen una millora de speedup quan el nombre d'àntenes o l'ordre de la constel·lació incrementen, això és en el context de MIMO Massiu. D'aquesta manera, aquestes implementacions permeten simular per a MIMO Massiu, i de forma més ràpida que la simulació convencional, un algorisme soft el cual presenta un rendiment quasi òptim.

Paraules Clau: detecció MIMO, decodificació esfèrica, GPU, Optimització de caixa, Eficiència, Precodificació.

Acknowledgements

It is a pleasure for me to thank those who made this thesis possible. First and foremost, I offer my sincerest gratitude to my supervisors, Dr. Paco Martínez and Prof. Alberto González, for supporting me throughout this thesis with their knowledge and advice, for the numerous hours of proofreading and for allowing me to work in my own way.

I would like to express my gratitude to the reviewers of this thesis, Prof. Erik G. Larsson from Linköping University, Dr. David Vargas from BBC Research and Development and Prof. Domingo Giménez from Universidad de Murcia. Their comments significantly improved the quality of this thesis. I am also very grateful to Dr. David Vargas from BBC Research and Development, Prof. Gema Piñero from Universitat Politècnica de València and Prof. Luis Castedo from Universidad de la Coruña for being members of the committee. Since time is a scarce resource for everyone, I deeply appreciate the time they dedicated to review this thesis.

Again, I would like to thank Prof. Erik G. Larsson for providing me the possibility to work as a visiting researcher in Linköping University during some months. I really appreciate the opportunity he gave me to work within his team.

I owe my deepest gratitude to Prof. Antonio Vidal and Dr. Victor García for their continuous support and collaboration, which have contributed in a very important way to the development of this thesis. Likewise, I would like to thank to Dr. Carmen Botella for her support.

I wish also to thank the Spanish Ministry of Education for the received financial support under the FPU program.

Very special thanks go to my colleagues of the Audio and Communications Signal Processing Group (GTAC). During these years, the people that had been part of this group have helped me to grow not only as a professional, but also as a person. In particular, thank you to Laura Fuster, Emmanuel Aguilera, Jose A. Belloch, Fernando Domene, Sandra Roger, Carla Ramiro, Pablo Gutiérrez, Christian Antoñanzas, Juan Estreder, Jorge Lorente, Maria de Diego, Miguel Ferrer, Jose J. Lopez and many others.

During the years the research for this thesis was done I was fortunate to have many good friends who are too numerous to be acknowledged individually. The activities, parties and numerous other shared moments with them were a lot of fun and helped me forget the research problems when needed.

Por supuesto, mi más profunda gratitud va hacía mi familia. Especialmente hacia mis padres, Gines y Antonia, por su apoyo y amor. La ayuda que me habéis dado a lo largo de mi vida es incalculable, soy lo que soy gracias a vosotros. A mi hermano, por los grandes momentos que hemos vivido juntos y los que nos quedan por vivir. A mi abuela Juanita por su amor y ternura. Y Por último, pero no por ello menos importante quisiera expresar mi gratitud a Antonio el cual me ha ofrecido apoyo y amor incondicional a lo largo de esta tesis. Gracias por mostrarme siempre el lado positivo de la vida y ayudarme a levantarme cuando los ánimos decaen. Eres, has sido y serás mi mas bonita casualidad. Te quiero.

Marian Simarro
July 2017

Contents

List of Figures

List of Tables

List of symbols

$\mathbf{X}$	Matrix
$\mathbf{x}$	Vector
x	Scalar
$X_{i,j}$	i,j component of matrix $\mathbf{X}$
$X_{i,:}$	i-th row of matrix $\mathbf{X}$
$X_{:,i}$	i-th column of matrix $\mathbf{X}$
$X_{i:j,k:l}$	Elements from i-th to j-th row and from k-th to l-th column of $\mathbf{X}$
$Tr(\mathbf{X})$	Trace of matrix $\mathbf{X}$
x_i	i-th component of vector $\mathbf{x}$
$(\cdot)^{\mathrm{T}}$	Transpose
$(\cdot)^{*}$	Complex conjugation
$(\cdot)^{\mathrm{H}}$	Conjugate transpose
$(\cdot)^{\dagger}$	Moore-Penrose pseudoinverse
$(\cdot)^{-1}$	Matrix inversion
$\mathbf{I}_N$	Identity matrix of size $N \times N$
$\lvert \cdot \rvert$	Absolute value
$\lVert \cdot \rVert_p$	p norm
$\lVert \cdot \rVert$	2 norm
$P(A)$	Marginal probability of A
$P(A \vert B)$	Conditional probability of A, given B
$\Re\{\cdot\}$	Real part of a complex number
$\Im\{\cdot\}$	Imaginary part of a complex number
$\max$	Maximum of a set
$\min$	Minimum of a set
$\arg\max$	Argument of the maximum of a set
$\arg\min$	Argument of the minimum of a set
$\log$	Natural logarithm
$\mathcal{Q}\{\cdot\}$	Quantization (slicing) operation
$\mathcal{O}(\cdot)$	Denotes an asymptotic complexity order of $\cdot$
$\mathbb{Z}^{m \times n}$	Set of $m \times n$ integer matrices
$\mathbb{R}^{m \times n}$	Set of $m \times n$ real matrices
$\mathbb{C}^{m \times n}$	Set of $m \times n$ complex matrices

$\mathbb{E}[\cdot]$	Mathematical expectation
$\mathrm{diag}\{\mathbf{x}\}$	Squared diagonal matrix with the elements of $\mathbf{x}$ on the main diagonal
$\lceil\cdot\rceil$	Ceil function (map a real number to the largest following) integer)
$\lfloor\cdot\rfloor$	Floor function (map a real number to the smallest previous integer)
$\lceil\cdot\rfloor$	Round function (map a real number to the nearest integer)

Abbreviations and Acronyms

a.k.a	also known as
APP	A posteriori Probabilities
API	Application Programming Interface
AWGN	Additive White Gaussian Noise
BCJR	Bahl, Cocke, Jelinek and Raviv
BD	Block Diagonalization
BER	Bit Error Rate
BICM	Bit-Interleaved Coded-Modulation
BLAS	Basic Linear Algebra Subprograms
BLAST	Bell-Labs Layered Space-Time
BO	Box Optimization
BOHD	Box Optimization Hard Detector
BOHD-SO	Box Optimization Hard Detector Soft Output
BORTS	Box Optimization Repeated Tree Search
BPSK	Binary Phase Shift Keying
BS	Base Station
CPU	Central Processing Unit
CAGR	Compounded annual growth rate
CSI	Channel state information
CUDA	Compute Unified Device Architecture
DF	Decision Feedback
DI	Distance Increments
DPC	Dirty Paper Coding
DTS	Double Tree Search
$\mathbf{E_b/N_0}$	Signal to noise ratio per bit
ED	Euclidean Distance
e.g.	for example (from the latin exempli gratia)
FE	Full Expansion
FER	Frame Error Rate
FSD	Fixed-complexity Sphere Decoder
GPU	Graphics Processing Unit
HPC	High Performance Computing
K-BEST	K-best sphere Detector

LDPC	Low-Density Parity-Check
ID-BICM	Bit-Interleaved Coded-Modulation with Iterative Decoding
i.e.	that is (from the Latin id est)
IoT	Internet of Things
LAPACK	Linear Algebra Package
LLL	Lentra, Lenstras and Lovász
LLR	Log-likelihood ratio
LR	Lattice Reduction
LRAP	Lattice-Reduction-Aided precoding
LSD	List based Sphere Decoder
LTE	Long Term Evolution
LTE-A	Long Term Evolution Advanced
MAP	Maximum-A-Posteriori
MF	Matched Filter
MIMO	Multiple-Input Multiple-Output
MKL	Math Kernel Library
ML	Maximum-Likelihood
MLE	Maximum-Likelihood Exhaustive
MMSE	Minimum Mean Squared Error
MMSE-PIC	Minimum Mean Squared Error with Parallel Interference Cancelation
MUI	Multiuser Interference
MU-MIMO	Multiuser Multiple-Input Multiple-Output
NGH	Next Generation Handheld
OSIC	Ordered Successive Interference Cancellation
PAM	Pulse Amplitude Modulation
PDF	Probability density function
PED	Partial Euclidean Distance
PINV	Pseudo Inverse
PSV	Partial Symbol Vector
QAM	Quadrature Amplitude Modulation
SD	Sphere Decoder
SDMA	Space Division Multiple Access
SE	Single Path Expansion
SESD	Schnorr-Euchner Sphere Decoder
SFSD	Soft-output Fixed-complexity Sphere Decoder
SIC	Successive Interference Cancellation

SISO	Soft-Input Soft-Output
SM	Spatial Multiplexing
SIMT	Single-Instruction Multiple-Threads
SOCA	Smart Ordering and Candidate Adding
SNR	Signal to Noise Ratio
SUMIS	Subspace Marginalization with Interference Suppression
THP	Tomlinson-Harashima Precoding
UT	User Terminal
WiMAX	Worldwide Interoperability for Microwave Access
WLAN	Wireless Local Area Network
ZF	Zero Forcing
VP	Vector Perturbation

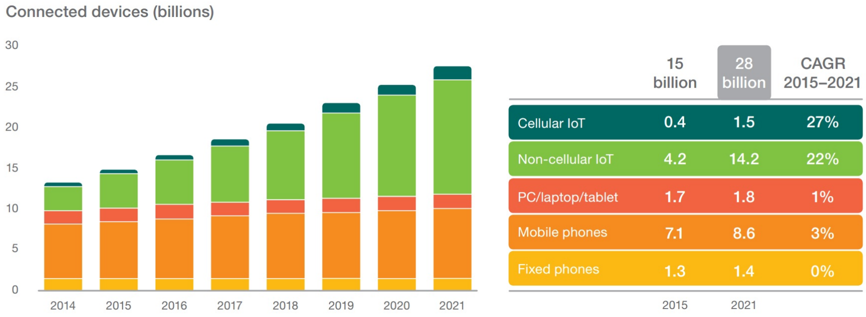
Introduction and Objectives 1

Introduction and Objectives 1

This chapter presents a brief introduction to wireless and mobile communications systems and describes the motivation of this thesis. Next, the objectives and contributions of the thesis are presented. Finally, the organization of this dissertation is described.

1.1 Background

During the past decades, many wired communication systems have been replaced by corresponding wireless services.Therefore wireless communications, especially mobile communications, are experiencing an exceptional growth and development. This implies that the demand in transferring large amounts of data rapidly and reliably [1] has been increasing drastically, at present we have the possibility of having internet access to any connected device from everywhere. The applications and services enabled by devices as smart-phones or tables have motivated this growth. Beside this, nowadays the Internet of Things (IoT) refers to the connection of devices (other than typical fare such as computers and smartphones) to the Internet. Cars, kitchen appliances, and even heart monitors can all be connected through the IoT. Mobile devices are getting smarter in their computing capabilities, beside this they are also evolving from lower-generation network connectivity (2G) to higher-generation network connectivity (3G, 3.5G, and 4G or LTE). The global mobile 4G connections will grow from 2.1 billion in 2016 to 6.1 billion by 2021. 5G connections will appear on the scene in 2020 and will grow more than a thousand percent from 2.3 million in 2020 to over 25 million in 2021 [2]. Combining device capabilities with higher bandwidth and more intelligent networks leads to wide adoption of advanced multimedia applications that contribute to increase mobile and Wi-Fi traffic. This demand is not likely to reduce but rather to increase. Actually, mobile phones are the largest category of connected devices, however in 2018 they are expected to be surpassed by the IoT, which includes connected cars, machines, remote metering or consumer electronics [3]. Figure 1.1 shows that between 2014 and 2021, IoT is expected to increase at a compounded annual growth rate (CAGR) of 23 percent, making up close to 16 billion of the total forecast of 28 billion devices connected in 2021.



	15 billion	28 billion	CAGR 2015–2021
Cellular IoT	0.4	1.5	27%
Non-cellular IoT	4.2	14.2	22%
PC/laptop/tablet	1.7	1.8	1%
Mobile phones	7.1	8.6	3%
Fixed phones	1.3	1.4	0%
	2015	2021	

Figure 1.1. Expected Connected devices (billions) [3].

Two of the main requirements for modern and future wireless communication systems are high spectral and energy efficiencies, due to the increase of the data traffic and the scarcity of other recourses such as bandwidth or transmit power. The Multiple-Input Multiple-Output (MIMO) technology has emerged as one of the most promising technologies [4][5], since this technology allows to increase the reliability, coverage and transmission rates without the need of extra bandwidth or power cost [4][6]. In fact, MIMO technology has been adopted by many wireless standards such as Long Term Evolution (LTE), LTE-Advanced (LTE-A), Worldwide interoperability for Microwave Access (WiMAX), Wireles Local Area Network (WLAN), IEEE 802.11n/ac, and certainly will be in 5G [7]-[10].

The benefits of MIMO technology can be categorized as follows [11]:

- *Diversity gain* is achieved by transmitting the same signal over all the transmit antennas. Thus, the signal is transmitted through independently fading links and combining all versions of the transmit signal in the receiver mitigates fading effect. The maximum amount of spatial diversity, with a MIMO channel with n_T transmitting antennas and n_R receiving antennas, corresponds to $n_T \times n_R$.

- *Array gain* refers to an increase in the average signal to noise ratio (SNR) at the receiver by coherent combination of all received signals. The increase is proportionally to the number of receiving antennas, n_R.

- *Multiplexing gain* can be achieved by the transmission of multiple data streams within the same frequency band. Thus, the data rate is increased without additional transmit power. With this technique, known as spatial multiplexing (SM), the system capacity is able to obtain a linear increase with the minimum value of n_T and n_R.

The wide range of advantages provided by MIMO techniques comes at the expense of a substantial increase in the computational cost at the receiver and transmitter. One such major implementation difficulty of the MIMO technology is the signal detection process at the receiving side of the MIMO link. The detector is the responsible for recovering the received signals (which are affected by the channel, and the information from the different antennas which interfere with each other). The complexity of optimal detection problem increases exponentially with the number of transmitted antennas. To overcome this problem, many methods with a reasonable complexity have been proposed [12]- [19]. The goal is to achieve the accuracy of optimal methods with as low complexity as possible.

On the other hand, MIMO techniques can also be used in a multiuser scenario (MU-MIMO), such as in cellular systems. In this case, several users are sharing the spatial dimension, causing multiuser interference [20]. However, the multiuser interference can be mitigated, even when the users are equipped with only one antenna, due to the use of multiple antennas at the transmitter and the use of precoding. Thus, the computational complexity is moved to the transmitted side. Different precoding techniques for MU-MIMO have been proposed that vary in performance and computational complexity.

The number of transmit and receive antennas is another important factor that affects the performance and complexity of a MIMO system. Although the number of antennas currently allowed is not large (ten or less), there has been a shift in focus in the earlier research MIMO literature. A MIMO systems with very large number of antennas is a promising candidate technology for next generations of wireless systems [21]. The terminology Massive MIMO, very large MIMO, large multi-user MIMO, etc. refers to these emerging MIMO systems. The vast majority of the methods proposed for conventional MIMO system are not suitable for large dimensions. Thus, earlier work focuses on develop detection and precoding techniques that are suitable for all size of MIMO systems.

1.2 Motivation

As seen in the previous section, the use of MIMO systems has had enormous repercussion in nowadays communications systems. The benefits offered are achieved at the expense of an additional complexity, especially in the detection part. For this reason, the search for low-complexity MIMO detectors and precoding schemes have been the subject of deep study during the last decade and it will keep on in the near future. The goal is to achieve the accuracy of the optimal methods with as low complexity as possible. Then, this dissertation aims to contribute to meet this goal.

In the MIMO detection problem, if nearly optimal detection is desired the computational complexity at the receiver becomes expensive. Therefore, linear preprocessing techniques are employed for the simplest and computationally cheapest algorithms, in order to offer acceptable performance with low computational complexity. The main idea of these algorithms is to annul the effect of the MIMO channel matrix. Other non-linear approximations, which offer differents trade-offs between complexity and performance, have been proposed. This trade-off can be adjusted via some user parameters of the algorithms. If MIMO systems with Bit-Interleaved Coded-Modulation (BICM) are employed (soft detection), better performance can be reached. Moreover, BICM with iterative MIMO detection (ID-BICM) is believed to be the most promising approach for near optimum detection in coded MIMO systems. However, the complexity to perform optimum soft detection is much higher than the complexity of hard detection, thus requiring more complex algorithms.

On the other hand, precoding techniques in MU-MIMO systems allow to move the expensive hardware and software to the transmitter side and to develope smaller and cheaper terminals [20]. Different precoding techniques, which vary in performance and computational complexity, have been proposed. Dirty paper coding (DPC) is a theoretical optimal precoding technique that allows the cancellation of the interference without incurring a power penalty. However, its high complexity restricts its implementation in practical systems, therefore suboptimal precoding algorithms involving linear and non-linear techniques have been proposed [22]-[25]. The use of High Performance Computing (HPC) systems, such as multicore CPUs and Graphics Processing Units (GPUs), have become attractive for efficient implementation of parallel signal processing algorithms with high computational requirements [26] in the last years. The implementation of precoding and detection problems through HPC can accelerate the computation, by efficiently executing in parallel as many parts of the algorithms as possible. Hence, the implementation of MIMO detection and precoding algorithms by exploiting the potential of these architectures, is crucial in MIMO research. These implementations allow to reduce the execution time of computationally expensive problems and fit the proposed algorithms to the newest and future computation facilities.

1.3 Objectives

The problems that arise from the MIMO technology are computationally very complex. This complexity scales up with the MIMO system dimensions and constellation size. Thus, it is necessary to develop efficient algorithms in detection and precoding steps. Taking into account the above presented motivation, the main objective of this thesis is:

The search of efficient soft detector and precoding algorithms. By efficient we mean algorithms which provide a reasonable accuracy with low complexity. That is, a good trade-off between performance and complexity.

In order to achieve this major objective, the following particular targets should be met:

- To evaluate the performance and computational complexity of the existing precoding and detection techniques, including a fair comparison among them.
- To decrease the computational cost of existing soft-output MIMO detection algorithms.
- To contribute with new soft-output MIMO detection algorithms with lower computational cost than previously proposed approaches.
- To develope new precoding schemes to contribute to the reduction of the computational complexity.
- To implement efficient existing techniques on different parallel architectures.

1.4 Key Contributions

The common goal of this dissertation is to design and optimize algorithms in communication MIMO systems. For this purpose computational and efficient new approaches are presented in this thesis. The proposed methods have been described, analyzed and compared with other already proposed methods. The main contributions of this dissertation are summarized in this section.

Probability of the utility of the Box Optimization technique in detection algorithms.

Box optimization (BO) technique has been employed in combination with Sphere Decoding (SD) strategy to drastically decrease the computation required to detect the received signal in hard detection. The results were remarkably faster than other known hard-output algorithms. Throughout this thesis this technique has been used to design or improve different soft algorithms. For this reason, the utility of this technique has been analyzed by the probability of its use when a given received vector is being detected. An original mathematical derivation of this probability is given in this dissertation. This probability has been evaluated for different numbers of antennas, constellation orders and SNR values.

Design of low-complexity soft-output detectors

A comparison of the performance and computational complexity of some of the most employed soft detectors has been carried out. In addition, two different soft-output detectors have been proposed. One of them is called *Box Optimization Hard Detector with Soft Output* (BOHD-SO) and extends the efficient hard-output SD detector to the soft-output detection. The proposed algorithm has been compared with some of the most employed algorithms. Results show that the BOHD-SO algorithm achieves almost max-log performance at low complexity. On the other hand, the well-known *Subspace Marginalization with Interference Suppression* (SUMIS) algorithm has been optimized. The SUMIS algorithm achieves good performance and it is suitable to be implemented when the system size increased. The proposed algorithm reduces the complexity of the original SUMIS algorithm with negligible performance degradation.

Design of low-complexity SISO detectors

The improvement of known algorithms for soft-input soft-output (SISO) detection has been approached. These algorithms usually exhibit high computational complexity. Taking into account two well-known SISO algorithms (*Repeated Tree Search* (RTS) and *Single Tree Search* (STS)) as a starting point, a number of modifications (based mainly on BO technique) have been intended to improve the efficiency of the search. As a result, two new algorithms are proposed for optimal SISO detection. One of them is based on RTS and the other one is based on STS. The proposed algorithms have been compared with the STS algorithm, showing that the new algorithms are far more efficient than the STS algorithm when the constellation order and the system size increase. Furthermore, the BOHD-SO algorithm proposed for soft-output detection has been adapted to allow its implementation in an iterative receiver. The results show that the proposed algorithm achieves suitable performance at low complexity.

Design of low-complexity precoding algorithms

A comparison of the performance and computational complexity of some of the most employed precoding algorithms has been carried out. Furthermore the influence of the channel matrix condition number on the precoding performance has been analyzed. Then, an hybrid scheme based on the channel condition number has been proposed in order to achieve the performance of complex precoding algorithms with lower computational complexity. Results show that choosing an adequate value of a tradeoff parameter, the proposed scheme exhibits good performance at low complexity. Furthermore, for Large MIMO systems, a new scheme has been developed. This new scheme shows large improvements on computational complexity compared with other conventional algorithms.

Efficient implementation of existing detectors

The use of many-core processors has recently become attractive for the efficient implementation of signal processing algorithms in communication systems. In this work, two parallel implementations of the SUMIS detection algorithm have been proposed. One of them is based on multicore processors and the other one on Graphic processing units (GPU). The parallel implementations has been described and compared with its CPU counterpart, achieving significant speedups.

1.5 Organization

This thesis describes the research that has been undertaken to develop the previous aims. This dissertation is structured in seven chapters.

- **Chapter 2**: This chapter describes a large number of necessary concepts for the understanding of this dissertation. It contains an introduction to MIMO systems and overview of the MIMO detection techniques.
- **Chapter 3**: This chapter presents some efficient soft-output detector algorithms. First, a detailed description of some conventional soft-output algorithms is presented. Then, two methods to decrease the complexity of the soft MIMO detectors are proposed and evaluated.
- **Chapter 4**: This chapter is focused on the SISO MIMO detection. The chapter includes a detailed description of SISO detection and some conventional algorithms. In the last part, three efficient SISO algorithms are presented. Two of them are an optimal solutions and the other one presents a non-optimal low-complexity detector.
- **Chapter 5**: The precoding problem is adressed in this chapter.

Some conventional precoding are evaluated and compared. Then two schemes are proposed. One of them is based on the matrix condition number. The other proposed scheme is developed to be used in Large MIMO.

- **Chapter 6**: This chapter presents several parallel implementations of a soft-output detector algorithm. The parallel implementations allow to considerably decrease the computational time required for the data detection in Large MIMO.

- **Chapter 7**: Finally, the conclusions obtained throughout the dissertation are presented, including some guidelines for future research directions. In addition, a list of the publications related to this thesis is also included.

State of the Art 2

State of the Art 2

This chapter presents a brief overview of the topic of MIMO communications systems and describes many concepts necessary for the understanding of this dissertation. Throughout the first section, different schemes in MIMO communication systems have been presented, from the well-studied Bell-Labs layered space-time (BLAST) scheme to the Massive MIMO system among others. Next, the MIMO detection problem is studied beside the main hard demodulators proposed in the literature. Finally, a precoding description is given in the last section.

2.1 Multiple-Input Multiple-Output Systems

2.1.1 The BLAST System

Spatial multiplexing over multiple-antenna wireless communication systems is achieved by the well-known Bell-Labs Layered Space-Time system (BLAST) [27]. Such systems have multiple antennas at both, the transmitter and the receiver side. Figure 2.1 displays a block diagram of a MIMO-BLAST sytem equipped with n_T transmit antennas and n_R receive antennas, with $n_R \geq n_T$. The data stream is splitting into n_T substreams and each substream is sent simultaneously through a different transmit antenna, thus overlapping in time and frequency. At the receiver side, each antenna receives the signals transmitted from the entire transmit antennas. Therefore, the receiver has the task of recovering the transmitted data from the received signals.

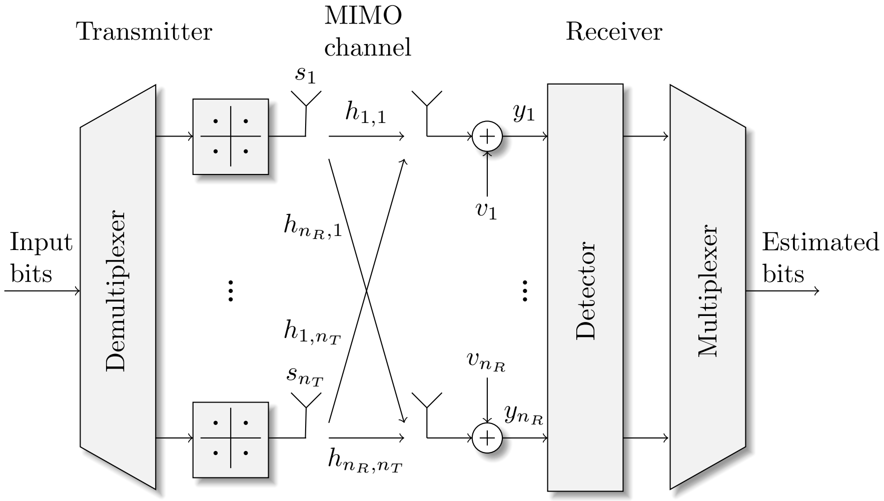


Figure 2.1. Block Diagram of a spatial multiplexing MIMO-BLAST system with n_T transmitting antennas and n_R receiving antennas.

Let us consider a complex-valued MIMO-BLAST system model as Fig. 2.1 shows. At the transmitter, groups of k bits in each n_T substream are mapped to a complex symbols s_i. Each symbol s_i is taken independently from a finite symbol alphabet or constellation denoted as Ω of size M, being $k = \log_2(M)$ the number of bits per symbol. The corresponding bits are denoted by $s_{i,b} \in \{0, 1\}$, where the indices refer to the bth bit associated with the ith symbol. The Quadrature Amplitude Modulation (QAM) constellations are usually employed in MIMO communications, which are known as M-QAM. Since a M-QAM constellation is a separable complex-valued constellation it can be defined as the cartesian product of a one-dimensional L Pulse Amplitude Modulation (L-PAM) constellation (denoted as P_M) with itself, being $L = \sqrt{M}$ the number of constellation points (constellation size) in the L-PAM constellation. Thus the Ω M-QAM constellation can be defined as $\Omega = \{a+bj \ldots a, b \in P_M\}$. Figure 2.2 shows the signal space representation of the most employed M-QAM and L-PAM constellation sizes.

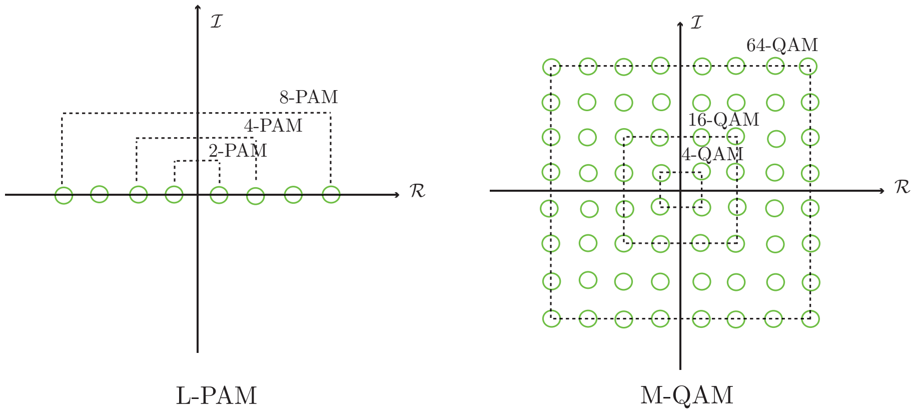


Figure 2.2. Signal space representation of the most employed M-QAM and L-PAM constellations.

At each signaling period, the transmitted vector carries $k \cdot n_T$ bits denoted as $\mathbf{s} = (s_1, s_2, \cdots, s_{n_T})^T$. The relation between the transmitted symbol vector $\mathbf{s} \in \mathbb{C}^{n_T}$ and the associated received vector $\mathbf{y} \in \mathbb{C}^{n_R}$ is given by

$$\mathbf{y} = \mathbf{H}\mathbf{s} + \mathbf{v}. \tag{2.1}$$

The $\mathbf{H} \in \mathbb{C}^{n_R \times n_T}$ denotes a fading channel matrix formed by independent complex elements, $h_{j,i}$, each element represents the complex fading gain from the $i-$th transmit antenna to the $j-$th receive antenna,

$$\mathbf{H} = \begin{pmatrix} h_{1,1} & h_{1,2} & \ldots & h_{1,n_T} \\ h_{2,1} & h_{2,2} & \ldots & h_{2,n_T} \\ \vdots & \vdots & \ddots & \vdots \\ h_{n_R,1} & h_{n_R,2} & \ldots & h_{n_R,n_T} \end{pmatrix}. \tag{2.2}$$

A Rayleigh fading model without correlation is often considered for the channel matrix, i.e., each complex $h_{j,i}$ element is a zero mean complex Gaussian random variable with variance $1/2$ per real dimension. The channel is commonly assumed to be perfectly known at the receiver and remains constant during a L_{ch} number of transmitted vectors. The complex vector $\mathbf{v} \in \mathbb{C}^{n_R}$ in (2.1) denotes an additive Gaussian noise (AWGN) with zero mean and σ_n^2 variance, being $\sigma_n^2/2$ the variance per real dimension. The complex-valued model in (2.1) can be represented through its equivalent real-valued form as

$$\begin{pmatrix} \Re(\mathbf{y}) \\ \Im(\mathbf{y}) \end{pmatrix} = \begin{pmatrix} \Re(\mathbf{H}) & -\Im(\mathbf{H}) \\ \Im(\mathbf{H}) & \Re(\mathbf{H}) \end{pmatrix} \begin{pmatrix} \Re(\mathbf{s}) \\ \Im(\mathbf{s}) \end{pmatrix} + \begin{pmatrix} \Re(\mathbf{v}) \\ \Im(\mathbf{v}) \end{pmatrix}, \tag{2.3}$$

obtaining a $(2n_T \times 2n_R)$-dimensional real-valued representation. The real-valued model can be more convenient for practical implementations or some detection algorithms. This model is denoted as

$$\mathbf{y}_r = \mathbf{H}_r \mathbf{s}_r + \mathbf{v}_r. \tag{2.4}$$

A certain signal to noise ratio (SNR) per received antenna is considered in the system. The SNR is defined as the ratio between the average transmitted symbol power E_s and the noise variance [28]

$$\text{SNR} = \frac{E_s}{\sigma_n^2} = \frac{E_b \cdot k \cdot n_T}{\sigma_n^2}, \tag{2.5}$$

where E_b is the transmitted energy per bit. Therefore the SNR can be related to the E_b/N_0 (SNR per bit) as:

$$\frac{E_b}{N_0} = \frac{\text{SNR}}{k \cdot n_T}. \tag{2.6}$$

As it said before, the most common constellation employed is the M-QAM constellation. Therefore, the average energy of each symbol constellation (E_M), with $E_b = E_M/k$, can be easily calculated as:

$$E_M = 2 \cdot \frac{(M-1)}{3}. \tag{2.7}$$

For a polar L-PAM, the E_M average energy per symbol constellation can be computed as:

$$E_M = 2 \cdot \frac{(L^2-1)}{3}. \tag{2.8}$$

2.1.2 Bit-Interleaved Coded-Modulation Systems

Bit-Interleaved Coded-Modulation (BICM) has received special attention in wireless communications due to its power and bandwidth efficiency. For single-antenna, BICM systems can achieve almost the channel capacity [29]. These advantages have led the extension of BICM to MIMO systems [30].

A block diagram of a MIMO-BICM system is represented in Fig. 2.3. The information bits are encoded using an error-correcting code and passed through a bitwise interleaver $\prod$. The encoded and interleaved bits are then demultiplexed into n_T substreams and mapped to symbols. The baseband equivalent model for the received vector in this scheme is the same than the model (2.1).

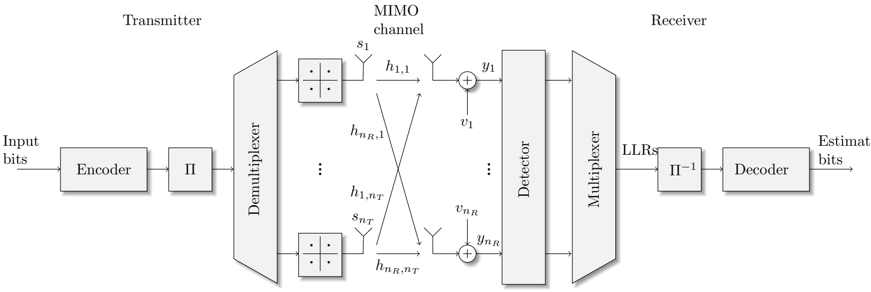


Figure 2.3. Block Diagram of a MIMO-BICM system with n_T transmitting antennas and n_R receiving antennas.

At the receiver side, a soft MIMO detector is employed to compute soft information about the code bits in terms of log-likelihood ratios (LLRs). The LLRs are deinterleaved and employed by the channel decoder to make a final decision about the transmitted bits. As in the MIMO-BLAST system, in this model, the corresponding bits are denoted by $s_{i,b}$. The LLRs are computed by the detector employing the received vector $\mathbf{y}$ in (2.1) and the channel matrix $\mathbf{H}$ known at the receiver:

$$L_{i,b} = \log \frac{P(s_{i,b} = 1|\mathbf{y}, \mathbf{H})}{P(s_{i,b} = 0|\mathbf{y}, \mathbf{H})}, \tag{2.9}$$

where $P(s_{i,b} = u|\mathbf{y}, \mathbf{H})$ denotes the conditional probabilities that the coded bit $s_{i,b}$ will be equal to u conditioned on $\mathbf{y}$ and $\mathbf{H}$.

When an error control coding of rate R is employed, the SNR expression in (2.5) is modified as

$$\mathrm{SNR} = \frac{E_s}{\sigma_n^2} = \frac{E_b \cdot k \cdot n_T \cdot R}{\sigma_n^2}. \tag{2.10}$$

Thus, the SNR can be related to the E_b/N_0 as

$$\frac{E_b}{N_0} = \frac{\mathrm{SNR}}{k \cdot n_T \cdot R}. \tag{2.11}$$

2.1.3 Iterative Decoding BICM Systems

In [31] the turbo principle is applied to BICM, known as BICM with iterative decoding (ID-BICM). Soft information is exchanged between the detector and the decoder through interleaving and deinterleaving in this scheme, as it is shown in Fig. 2.4. In summary, soft-detection estimates the bits that are mapped to the transmit vector and gives information about how reliable these estimates are. This supplementary information can be exploited by a channel decoder to achieve a better decoding performance. Moreover the performance can be further improved using iterative receiver structures as shown in Fig. 2.4, so called turbo receivers. In this scheme, the soft-input soft-output MIMO detector computes intrinsic LLRs ($L_{i,b}$) according to (2.9), then the extrinsic LLRs are computing based on the intrinsic LLRs as:

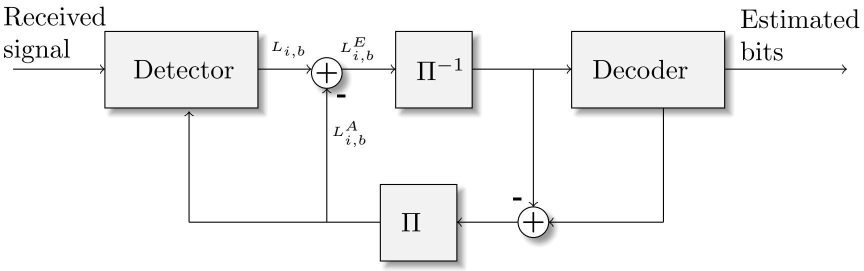


Figure 2.4. Block Diagram of a receiver ID-BICM system.

$$L^E_{i,b} = L_{i,b} - L^A_{i,b} \quad \forall i, b. \tag{2.12}$$

The $L^A_{i,b}$ values in (2.12) are the a priori LLRs provided by the channel decoder.

2.1.4 Multiuser Systems

MIMO techniques can also be used in a multiuser scenario, multiuser MIMO (MU-MIMO). MU-MIMO systems have the potential to combine the high throughput achievable with MIMO processing with the benefits of space division multiple access (SDMA). In a MU-MIMO system, a Base Station (BS) provided by multiples antennas communicates with multiple user's terminal (UTs), each with one or more antennas. In MU-MIMO the different users share the spatial dimension, causing multi-user interference (MUI). On the downlink, known as the MIMO broadcast channel, the BS sends different information streams to the users. On the uplink, the BS receives information from the different users.

In this thesis we consider the downlink of a MU-MIMO system, as seen in Fig. 2.5, where the BS is equipped with n_T antennas that serves to K UTs, equipped with a single antenna.

In the downlink scenario, the detection process becomes more complex due to each user receives the information of the K users and must process its own information. In order to reduce the complexity at the UTs, the BS exploits the channel state information (CSI) available at the transmitter in order to mitigate or ideally completely eliminate the MUI Due to the computational complexity at the UTs several preprocessing techniques, such as precoding algorithms, are employed to reduce the MUI, moving the computational complexity to the BS.

The received signal at the $m-$th UT in the Fig. 2.5 can be expressed as

$$y_m = \mathbf{h}_m^T \mathbf{x} + v_m, \tag{2.13}$$

where $\mathbf{x} \in \mathbb{C}^{n_T}$ includes the precoded information symbols, $\mathbf{h}_m$ is the downlink channel matrix for the m-th UT, and v_m is the received noise at the mth user. This equation can be expressed in a compact way by aggregating the received signal of the different users in a vector similar to the one of the BLAST system (2.1) as:

$$\mathbf{y} = \mathbf{H}\mathbf{x} + \mathbf{v}. \tag{2.14}$$

The $\mathbf{y} \in \mathbb{C}^K$ vector contains the received signal for the K UTs, and $\mathbf{H} = [\mathbf{h}_1 \cdots \mathbf{h}_K]^T \in \mathbb{C}^{K \times n_T}$ is the aggregated downlink channel matrix.

The type of precoding sets the way in which $\mathbf{x}$ is built from the $\mathbf{s}$ vector constructed by the constellation points. Moreover the real-valued form of

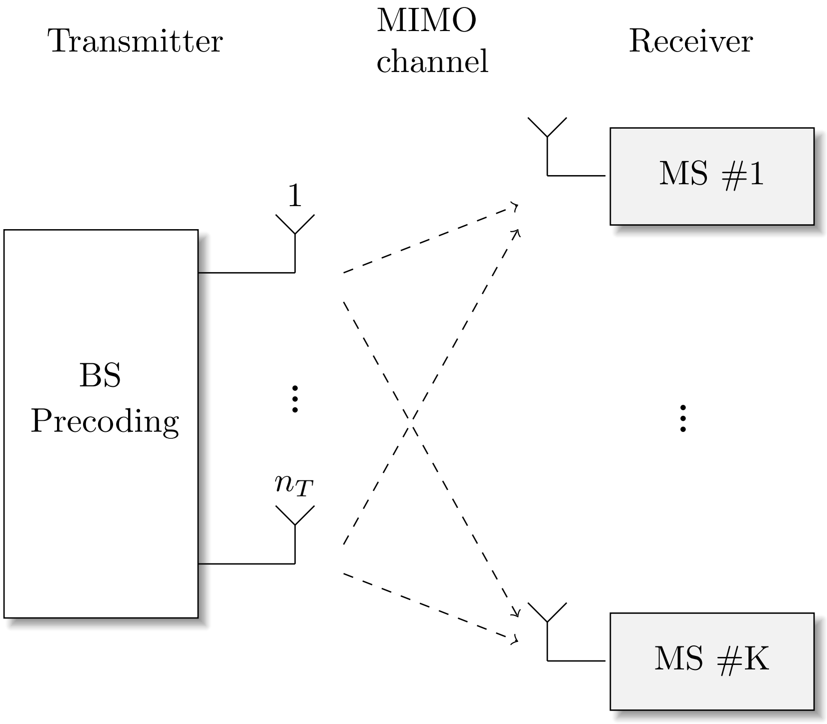


Figure 2.5. Block Diagram of a downlink MU-MIMO system.

the system can be employed as in Section 2.1.1.

2.1.5 Massive MIMO Sytems

In Massive MIMO, a.k.a Large MIMO or Very Large MIMO, the BS is provided with a large number of antennas (e.g. hundreds or thousands) and simultaneously serves a large number of users in the same time and frequency resource (see Fig.). The number of UTs is of the order of tens to hundreds.

Massive MIMO systems offer huge improvements in throughput and energy efficiency, accommodating more users at higher data rates with better reliability and consuming less power. Because of that, massive MIMO presents even more benefits than MIMO [21][33]--[35]:

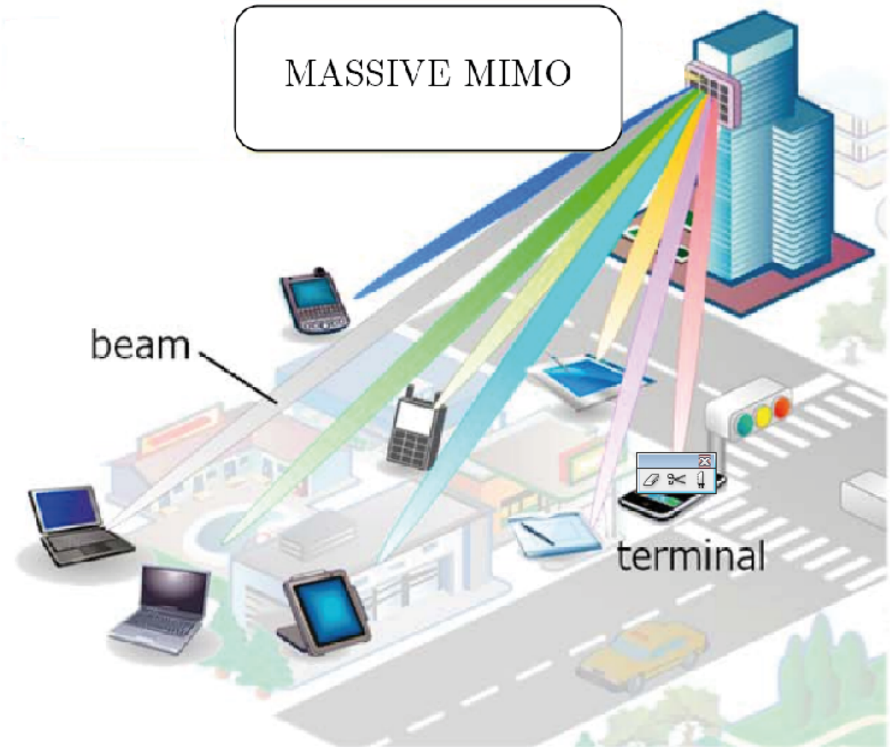


Figure 2.6. Massive MIMO overview [32].

- *Increasing data rate and communication reliability:* The increase in the number of antennas (n_T and K) allows a huge spectral efficiency leading a much greater level of data to be transferred within a given time.

- *High energy efficiency:* Keeping the total transmitted power and increasing the number of antennas, each antenna can use extremely low power. Thus, low power components can be used resulting less expensive equipment.

- *Simple signal processing:* Since the number of antennas at both ends is large, the signal processing at the UTs should be simple, due to the large dimension of the matrix and vector employed. When the number of BS antennas is much longer than the number of users, the effects of MUI and noise can be eliminated using simple linear processing, due to the channel vectors among the users become more orthogonal as the number of BS antennas increases.

The demand for wireless throughput and communication reliability as well as the user density will always increase, and Massive MIMO is a promising candidate technology for future wireless communication systems. For this reason, a great interest in this technology has been noted recently [21][33]-[36].

2.2 MIMO Detection

The detection step is often the most computationally expensive part in a MIMO receiver, especially when nearly optimal data detection is desired. The detector, a.k.a demodulator, preprocesses the received signals in order to recover the transmitted data with the accuracy required by the considered application. This issue has motivated the search for MIMO demodulators capable to be reconfigurable and scalable with the system parameters in performance and complexity. As shown in Fig. 2.7, there are two types of detectors depending on the type of output:

- *Hard-output detectors*: The detector makes a hard decision based on symbols, which are at last directly demapped onto bits. Thus, the detector decides whether a bit is either zero or one, delivering a hard decisions.

- *Soft-output detectors*: The detector decides the probability that a particular bit takes the zero or one values, providing soft information in terms of LLRs. These values are used by the channel decoder to make final decisions on the received coded bits. Furthermore, for the soft-output detection two different cases can be considered:

    - Receiver with no feedback: the detector does not process any a priori information provided by the channel decoder. In this scheme, a soft-output detector is considered where only the received vector is processed by the detector in order to compute the soft information.

- Receiver with feedback: soft extrinsic information between a soft-input soft-output (SISO) detector and a SISO channel decoder is considered in an iterative loop.

The complexity is particularly high when soft detection is performed. Nevertheless soft detection improves the performance compared to hard detection and is a prerequisite for iterative receiver structures (turbo receivers). This categorization defines very important input-output properties of detectors and thus, aspects of the receiver structure. Furthermore, depending on the performance achieved by the detector we can differentiate between optimal or suboptimal methods. The realization of efficient and effective detection algorithms is a distinguishing aspect common to all the variants of demodulators.

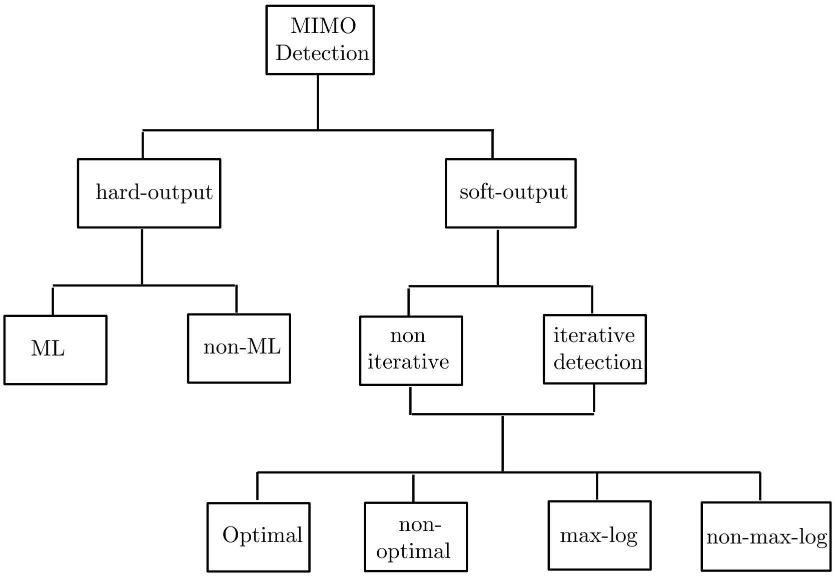


Figure 2.7. Main types of MIMO detection.

Figure 2.8 represents the performance in terms of Bit Error Rate (BER) for the different types of detection (hard, soft without iteration and SISO). The performance has been represented for a 4 × 4 MIMO system and $4-$QAM constellation. The number of iterations in the SISO detection has been set to 4. The channel code used is a convolutional encoder of rate 1/2 and codeword length 4096. The generator polynomials $[133_o, 171_o]$, constraint length 7 and max-log Bahl, Cocke, Jelinek and Raviv (BCJR) [37] algorithm channel decoder based on the min-sum method with the max-log approximation have been employed. This figure illustrates how soft MIMO detection attains significantly better performance than hard detection. Furthermore, employing an iterative structure and increasing the number of iterations the performance achieved outperforms hard-output and soft-output without iteration.

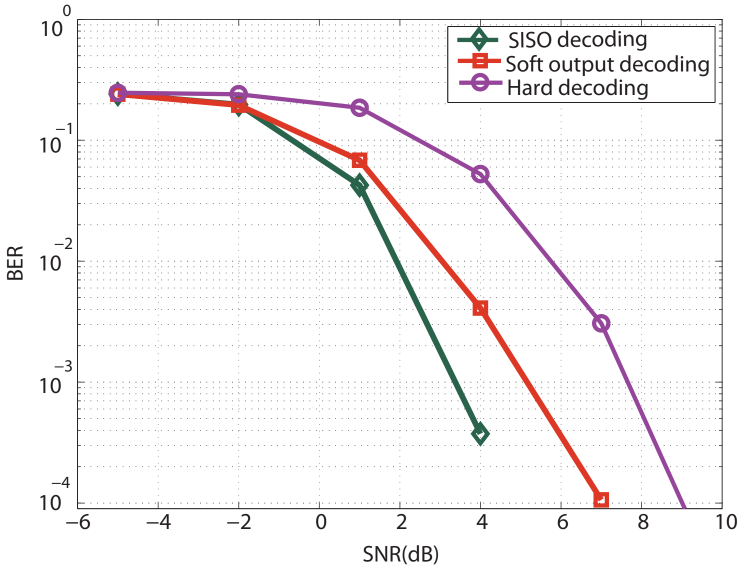


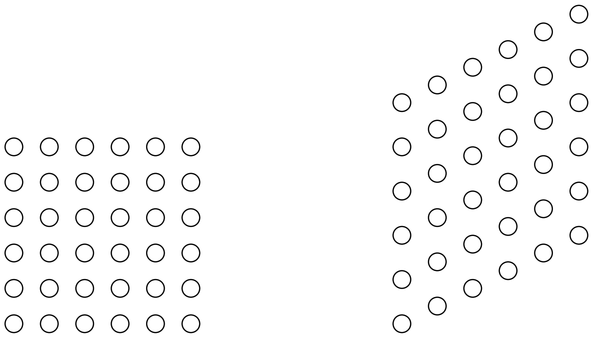
Figure 2.8. BER of a Hard-Output ML detector and the MAP soft-output detection without feedback and with feedback employing a SISO detector. The parameters employed are a 4×4 MIMO system and $4-$QAM constellation.

2.2.1 Hard-Output MIMO Detection

The optimal hard-output MIMO detector (in terms of minimizing the probability that the estimated vector does not correspond to the transmitted symbol vector) is referred as the *maximum-a-posteriori* (MAP) detector and is defined by

$$\hat{\mathbf{s}}^{MAP} = \arg \max_{\mathbf{s} \in \Omega^{n_T}} P(\mathbf{s}|\mathbf{y}, \mathbf{H}), \qquad (2.15)$$

where $\hat{\mathbf{s}}^{MAP}$ is referred to as the MAP estimate, and Ω^{n_T} denotes the set of possible transmitted *lattice points*. Then, the detection problem in the model (2.1) can be described in terms of *lattices* [15], being a periodic arrangements of discrete points. Here, all the possible n_T-dimensional transmitted vectors with symbols belonging to a known finite Ω constellation can be represented as a *lattice*, as shown Fig. 2.9(a). The *lattice points* are multiplied by the channel matrix $\mathbf{H}$, which implies a deformation in the *lattice* (see Fig. 2.9(b)).



(a) Initial lattice points.

(b) Deformed lattice points.

Figure 2.9. Example of lattices for two transmitting antennas.

After applying Bayes′ theorem to (2.15) an equivalent formulation of MAP detection is achieved by

$$\begin{aligned}\hat{\mathbf{s}}^{MAP} &= \arg\max_{\mathbf{s}\in\Omega^{n_T}} \left(p(\mathbf{y}|\mathbf{s},\mathbf{H})\frac{P(\mathbf{s})}{p(\mathbf{y})}\right) \\ &= \arg\max_{\mathbf{s}\in\Omega^{n_T}} \left(p(\mathbf{y}|\mathbf{s},\mathbf{H})P(\mathbf{s})\right),\end{aligned} \tag{2.16}$$

where the second equality results from the fact that the probability density function (PDF) $p(\mathbf{y})$ does not depend on $\mathbf{s}$. On the other hand, if AWGN is considered, the PDF $p(\mathbf{y}|\mathbf{s},\mathbf{H})$ corresponds to

$$p(\mathbf{y}|\mathbf{s},\mathbf{H}) = \frac{1}{(\pi\sigma_n^2)^{n_T}} \exp\left(-\frac{\|\mathbf{y}-\mathbf{Hs}\|^2}{\sigma_n^2}\right). \tag{2.17}$$

Maximizing equation (2.16) is equivalent to maximizing the logarithm of $p(\mathbf{y}|\mathbf{s},\mathbf{H})$, due to $\log(\cdot)$ is a monotonically increasing function. This leads to the well-known MAP detection rule for MIMO systems [10]

$$\begin{aligned} \hat{\mathbf{s}}^{MAP} &= \arg\max_{\mathbf{s}\in\Omega^{n_T}} \left(\log\left(\exp\left(-\frac{\|\mathbf{y}-\mathbf{Hs}\|^2}{\sigma_n^2}\right)P(\mathbf{s})\right)\right) \\ &= \arg\min_{\mathbf{s}\in\Omega^{n_T}} \left(\frac{\|\mathbf{y}-\mathbf{Hs}\|^2}{\sigma_n^2} - \log P(\mathbf{s})\right). \end{aligned} \tag{2.18}$$

If a priori inputs are assumed to be equally likely, the MAP rule turns into the *maximum-likelihood* (ML) detection rule as follows [4] [12]:

$$\hat{\mathbf{s}}^{ML} = \arg\min_{\mathbf{s}\in\Omega^{n_T}} \|\mathbf{y}-\mathbf{Hs}\|^2. \tag{2.19}$$

Therefore, the ML detector solves optimally the so-called closest *lattice point* problem by calculating the Euclidean distances (EDs) between the received signal $\mathbf{y}$ and the lattices points $\mathbf{Hs}$, selecting the lattice point that minimizes the ED to the received vector. An exhaustive search, checking all the Ω^{n_T} lattice points and selecting the one that minimizes (2.19) can be carried out to solve the ML detection problem. This strategy called *Maximum-Likelihood Exhaustive* (MLE) leads to an algorithm with prohibitively high computational complexity, growing exponentially with the number of transmit antennas and the number of bits per symbol ($|\Omega|^{n_T} = M^{n_T} = 2^{k\cdot n_T}$). The high complexity of the MLE detection motivates the research on efficient detection schemes for MIMO systems [19][38]. The most efficient and relevant contributions in hard-output detection are detailed below.

Zero-Forcing Detector

One of the simplest approximation of the problem in (2.19) is the *Zero-Forcing* (ZF) detector [39], being a linear technique for recovering the transmitted signal at the receiver. The ZF detector considers the signal from each transmit antenna as the target signal and the rest of signals as interferences. This method corresponds to the multiplication of the Moore-Penrose pseudoinverse of the channel matrix ($\mathbf{H} \in \mathbb{C}^{n_R\times n_T}$ with $n_R \geq n_T$) to the received vector such that

$$\hat{\mathbf{s}}^{ZF} = \mathcal{Q}\{\mathbf{H}^{\dagger}\mathbf{y}\} = \mathcal{Q}\{(\mathbf{H}^H\mathbf{H})^{-1}\mathbf{H}^H\mathbf{y}\}, \tag{2.20}$$

where $\mathcal{Q}\{\cdot\}$ assigns the closest constellation symbol and it is known as quantization operation. The ZF detector has the next drawback: after the product by the pseudoinverse channel, the noise variance can be significantly amplified [39], so the algorithm gets good performance only when $\mathbf{H}$ is well-conditioned.

Matched Filter Detector

Matched Filter (MF) is the most simple method in (2.19), which is carried out just by multiplying the received vector by the $\mathbf{H}^H$ and quantizing the result

$$\hat{\mathbf{s}}^{MF} = \mathcal{Q}\{\mathbf{H}^H\mathbf{y}\}. \tag{2.21}$$

MF works well only for scenarios that yield orthogonal (or very close to orthogonal) columns in $\mathbf{H}$, making MF equivalent to ZF since the matrix $\mathbf{H}^H\mathbf{H}$ becomes diagonal (or very close to diagonal).

Minimum Mean Square Error Detector

The *Minimum Mean Square Error* (MMSE) detector, minimizes the error due to the noise and the interference [39], removing the noise enhancement problem of the ZF detector by using

$$\hat{\mathbf{s}}^{MMSE} = \mathcal{Q}\{(\mathbf{H}^H\mathbf{H} + \sigma_n^2\mathbf{I})^{-1}\mathbf{H}^H\mathbf{y}\}, \tag{2.22}$$

Hence, the MMSE detector requires knowledge of σ_n^2 as opposed to ZF.

Successive Interference Cancelation Detector

Successive Interference Cancelation (SIC) [40] is performed in order to improve the ZF and MMSE methods. Similar to the *decision feedback* (DF) method, this technique leads to the well-known nulling and cancelation detectors, ZF-SIC and MMSE-SIC. In this cases, the quantization is done successively for each component of $\hat{\mathbf{s}}^{SIC}$ and not jointly as in the ZF and MMSE. This method has the drawback of error propagation, since an already detected and quantized symbol is employed to cancel out interference in the subsequent detection steps. To carry out the SIC method, the QR decomposition of the channel matrix is performed $\mathbf{H} = \begin{bmatrix}\mathbf{Q} & \mathbf{Q}^0\end{bmatrix}\begin{bmatrix}\mathbf{R} \\ \mathbf{0}\end{bmatrix} = \mathbf{QR}$, where $\tilde{\mathbf{Q}} = \begin{bmatrix}\mathbf{Q} & \mathbf{Q}^0\end{bmatrix} \in \mathbb{C}^{n_R \times n_R}$ is unitary ($\tilde{\mathbf{Q}}\tilde{\mathbf{Q}}^H = \tilde{\mathbf{Q}}^H\tilde{\mathbf{Q}} = \mathbf{I}$), with $\mathbf{Q} \in \mathbb{C}^{n_R \times n_T}$ and $\mathbf{Q}^0 \in \mathbb{C}^{n_R \times (n_R - n_T)}$. $\mathbf{R} \in \mathbb{C}^{n_T \times n_T}$ is an upper triangular matrix and $\mathbf{0}$ a $(n_R - n_T) \times n_T$ zero matrix. Multiplying (2.1) by $\tilde{\mathbf{Q}}^H$ and calling $\tilde{\mathbf{z}} = \tilde{\mathbf{Q}}^H\mathbf{y}$, the model in (2.1) can be represented as:

$$\tilde{\mathbf{z}} = \begin{bmatrix}\mathbf{R}\\ \mathbf{0}\end{bmatrix}\mathbf{s} + \begin{bmatrix}\tilde{\mathbf{n}}\\ \tilde{\mathbf{n}}^0\end{bmatrix}. \tag{2.23}$$

The $\tilde{\mathbf{z}}$ vector can be divided as $\tilde{\mathbf{z}} = \begin{bmatrix}\mathbf{z}\\ \mathbf{z}^0\end{bmatrix}$ with $\mathbf{z} = \mathbf{Q}^H\mathbf{y}$ and $\mathbf{z}^0 = \mathbf{Q}^0\mathbf{y}$, thus:

$$\mathbf{z} = \mathbf{R}\mathbf{s} + \tilde{\mathbf{n}}. \tag{2.24}$$

Then, to detect the i-th component in $\hat{\mathbf{s}}^{SIC}$ according to the MIMO detection based on SIC, the next expression is employed

$$\hat{s_i}^{SIC} = \mathcal{Q}\left\{\frac{(z_i - \sum_{t=i+1}^{n_T} R_{i,t}\hat{s_t}^{SIC})}{R_{i,i}}\right\}, \tag{2.25}$$

where i varies from n_T to 1. To prevent error propagation, e.g., it is advantageous to detect first the symbols with the highest SNR, i.e., the most reliable ones. Thus, the use of channel matrix ordering techniques before detection is an interesting strategy. In [41] an optimal ordering was proposed to employ for SIC detection. The resulting scheme was named *ordered SIC* (OSIC) detector. To implement the MMSE-SIC method, the QR decomposition will be performed on the inverse of the matrix $\mathbf{W} = (\mathbf{H}^H\mathbf{H} + \sigma_n^2\mathbf{I})^{-1}\mathbf{H}^H$.

Tree Search and Sphere Decoding detection

The *sphere decoding* (SD) MIMO detection algorithm intends to reach the ML solution with lower complexity than MLE. The main idea of the SD method is to transform the ML problem in (2.19) into a tree-search problem, which can be solved more efficiently. The SD method was initially developed by Pohst and Fincke in [42][43] for computation of minimal-length lattice vectors. In [44] Schnorr-Euchner presented an improvement to the method which leads to lower computational complexity than the Pohst variant. The *Schnorr-Euchner Sphere Decoding* (SESD) was applied to detection problem in [45].

Similarly to SIC detection, a QR decomposition of the channel matrix is performed, $\mathbf{H} = \mathbf{QR}$. The properties and dimensions of $\mathbf{Q}$ and $\mathbf{R}$ are the same that for the SIC case. Then, the ML problem in (2.24) can be computed as

$$\hat{\mathbf{s}}^{ML} = \arg \min_{\mathbf{s}\in\Omega^{n_T}} \left\{ \|\mathbf{z} - \mathbf{R}\mathbf{s}\|^2 + \|\mathbf{z}^0\|^2 \right\}. \tag{2.26}$$

The term $\|\mathbf{z}^0\|^2$ does not depend on $\mathbf{s}$, so the ML problem can be solved as:

$$\hat{\mathbf{s}}^{ML} = \arg \min_{\mathbf{s}\in\Omega^{n_T}} \left\{ \|\mathbf{z} - \mathbf{R}\mathbf{s}\|^2 \right\}. \tag{2.27}$$

The previous expression can be represented as:

$$\hat{\mathbf{s}}^{ML} = \arg \min_{\mathbf{s}\in\Omega^{n_T}} \left\{ \sum_{i=n_T}^{1} \left| z_i - \sum_{j=i}^{n_T} R_{i,j} s_j \right|^2 \right\}, \tag{2.28}$$

where the triangular structure of $\mathbf{R}$ has been exploited. Each level in the tree corresponds to a single transmit antenna, starting with antenna $i = n_T$ below the root node and ending with antenna $i = 1$ at the leaf nodes. Each node of such tree is a scalar symbol candidate $s_i \in \Omega$ for antenna i. Thus, the tree structure contains all the candidate *lattice points* associated to the problem to be solved (see Fig. 2.10). The tree search starts from the root (level $n_T + 1$) and descends from a node in level i to the nodes in level $i-1$ that are connected to it. Thus, a tree path contains the selected symbols from the root down to the node i, thus the partial symbol vector (PSV) of a tree path is defined as

$$\mathbf{s}^{(i)} = \left[s_i, s_{i+1}, \cdots, s_{n_T}\right]^T, \tag{2.29}$$

with $\mathbf{s}^{(1)} = \mathbf{s}$.

The ED, $d(\mathbf{s}) = \|\mathbf{z} - \mathbf{R}\mathbf{s}\|^2$, associated with (2.27) can be computed recursively by employing (2.28). Then, by defining $d(\mathbf{s}^{(i)}) = d_i$, the Partial Euclidean distances (PEDs) are denoted as

$$d_i = d_{i+1} + |e_i|^2, \quad i = n_T, \cdots, 1, \tag{2.30}$$

where it is assumed that the root node starts with a PED equal to zero ($d_{n_T+1} = 0$). The term $|e_i|^2$ denotes the distance increments (DIs) and is computed as:

$$|e_i|^2 = \left| z_i - \sum_{j=i}^{n_T} R_{i,j} s_j \right|. \tag{2.31}$$

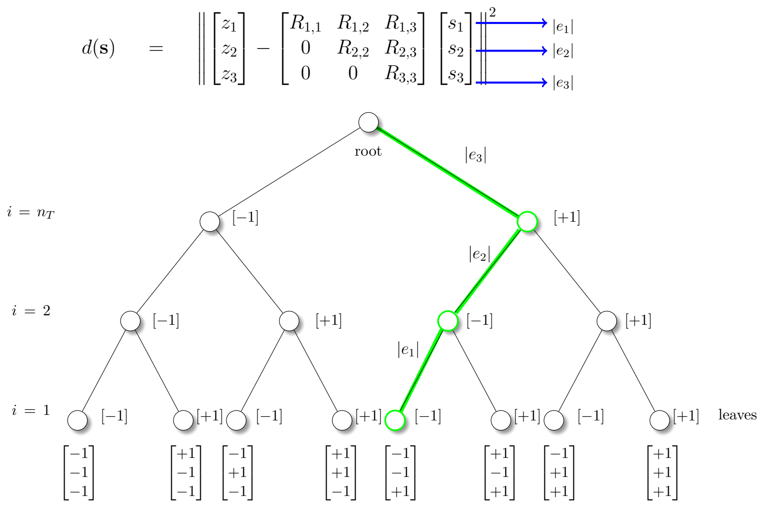

$d(\mathbf{s}) = \left\| \begin{bmatrix} z_1 \\ z_2 \\ z_3 \end{bmatrix} - \begin{bmatrix} R_{1,1} & R_{1,2} & R_{1,3} \\ 0 & R_{2,2} & R_{2,3} \\ 0 & 0 & R_{3,3} \end{bmatrix} \begin{bmatrix} s_1 \\ s_2 \\ s_3 \end{bmatrix} \right\|^2$
$|e_1|$
$|e_2|$
$|e_3|$
root
$|e_3|$
$i = n_T$
[−1]
[+1]
$|e_2|$
$i = 2$
[−1]
[+1]
[−1]
[+1]
$|e_1|$
$i = 1$
[−1]
[+1]
[−1]
[+1]
[−1]
[+1]
[−1]
[+1]
leaves
$\begin{bmatrix} -1 \\ -1 \\ -1 \end{bmatrix}$ $\begin{bmatrix} +1 \\ -1 \\ -1 \end{bmatrix}$ $\begin{bmatrix} -1 \\ +1 \\ -1 \end{bmatrix}$ $\begin{bmatrix} +1 \\ +1 \\ -1 \end{bmatrix}$ $\begin{bmatrix} -1 \\ -1 \\ +1 \end{bmatrix}$ $\begin{bmatrix} +1 \\ -1 \\ +1 \end{bmatrix}$ $\begin{bmatrix} -1 \\ +1 \\ +1 \end{bmatrix}$ $\begin{bmatrix} +1 \\ +1 \\ +1 \end{bmatrix}$

Figure 2.10. Example of decoding tree for a MIMO system with $n_T = 3$ and BPSK constellation. The green path represents an example of selected path to a leaf node. The $|e_i|$ values are the distances increments.

Then, the tree search type decoding algorithm descends through the tree paths and at each i-th level computes (2.30). Thus, each node is associated with a PED and each branches with a DI. The resulting tree structures is illustrated in Fig. 2.10 for a 3×3 MIMO system using a Binary Phase Shift Keying (BPSK) constellation ($M = 2$). As a result, the decision tree depicted in Fig. 2.10 becomes a weighted tree with non-negative weights for each edge.

The tree search structure enables efficient branch and bound algorithms by applying radius reduction strategy. That is, if the sum of PED along a path is larger than a metric constraint r^2, every node in the subtree does not need to be visited since any leaf metric $d(\mathbf{s})$ can only be equal or larger than $d(\mathbf{s}^{(i)})$. That means that when the accumulated metric of a node is higher than the radius, the metric constraint

$$d\left(\mathbf{s}^{(i)}\right) < r^2, \tag{2.32}$$

is violated, therefore the remaining subtrees can be pruned, accelerating the search process.

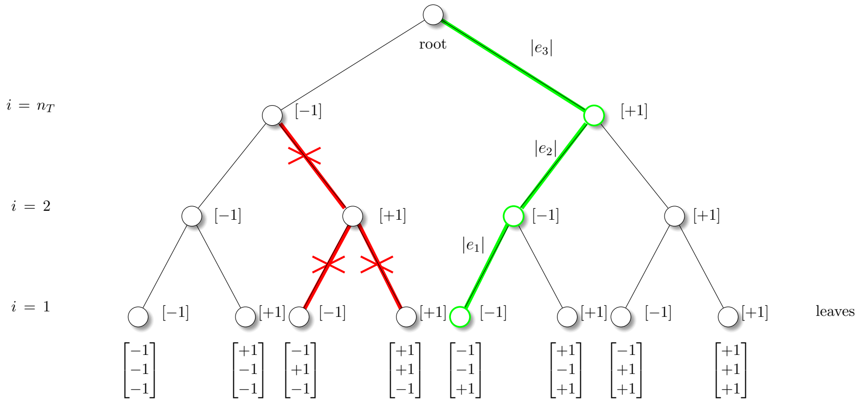

root
$|e_3|$
$|e_2|$
$|e_1|$
$i = n_T$
$i = 2$
$i = 1$
[−1]
[+1]
leaves
$\begin{bmatrix}-1\\-1\\-1\end{bmatrix}$
$\begin{bmatrix}+1\\-1\\-1\end{bmatrix}$
$\begin{bmatrix}-1\\+1\\-1\end{bmatrix}$
$\begin{bmatrix}+1\\+1\\-1\end{bmatrix}$
$\begin{bmatrix}-1\\-1\\+1\end{bmatrix}$
$\begin{bmatrix}+1\\-1\\+1\end{bmatrix}$
$\begin{bmatrix}-1\\+1\\+1\end{bmatrix}$
$\begin{bmatrix}+1\\+1\\+1\end{bmatrix}$

Figure 2.11. Example of tree pruning for a MIMO system with $n_T = 3$ and BPSK constellation. The red path represents an example of pruned subtree and the green path is an example of selected path to a leaf node.

An example for a pruned subtree can be visualized in Fig. 2.11 by red edges. Since the metrics corresponds to ED, the search in the tree is constrained to nodes which lie within a radius r around $\mathbf{z}$, as shown Fig. 2.12. This analogy is the reason for defining r^2 as *sphere radius* and naming this MIMO detector as SD.

A suitable sphere radius is generally needed for getting the ML solution expanding as few nodes as possible. If a too small radius is chosen, the decoder might not find any solution. On the other hand, if a too large sphere radius is selected, too many candidate points may be found leading to high complexity. There are several methods to estimate the sphere radius [12]. One useful technique is to set the radius to the distance between the received vector and the solution provided by a low-complexity detection method (ZF or MMSE), due to we guarantee that at least one point will be inside the sphere. On the other hand, in order to avoid the problem of choosing a suitable radius, SESD performs a search from the top to the bottom of the tree with radius reduction. That is, the initial radius is set to ∞ and is updated any time a leaf node is visited. Thereby, every time a leaf node is reach with $d(\mathbf{s}^{(1)}) < r^2$, the detector updates the radius as $r = d(\mathbf{s}^{(1)})$, reducing the sphere radius. With this approach the ML solution will be found in an efficient way.

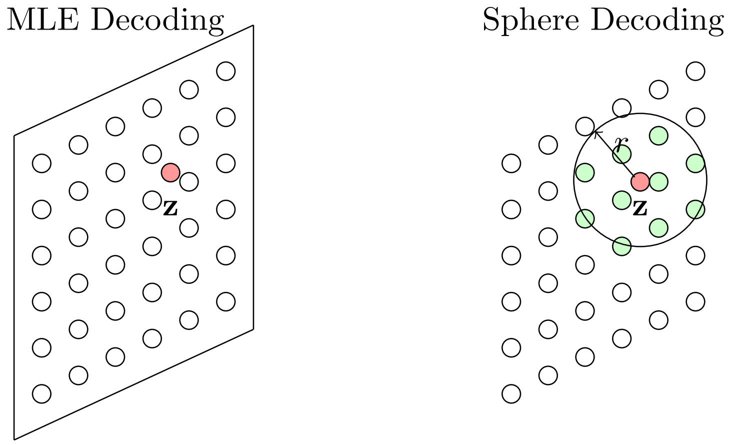


Figure 2.12. Idea behind the MLE and the Sphere Decoder with radius reduction.

Many different SD tree search strategies have been proposed during the last years, some of which can be found in [12][46][47]. The main advantages of the SD method is that it is simple to implement and it guaranties to deliver the optimal solution. On the other hand, one of the main drawbacks of the SD method is the variable complexity depending on the radius, the noise and the effective channel conditions. In order to attain low and fixed computational complexity using tree-search based MIMO detection, different methods have been proposed in the literature. It is important to note that, the complexity of the tree-search detectors can be measured in number of expanded nodes, allowing a fair comparison among different algorithms. Recently, a new hard-output SD ML algorithm was proposed in [48], where the SD algorithm is combined with Box Optimization (BO). The results obtained are remarkably faster than other known hard-output ML detectors. This algorithm has been employed to proposed efficient soft-output algorithms throughout this dissertation and it will be detailed in the next chapter.

K-BEST Sphere Detector

The *K-best Sphere Detector* (K-BEST)[49] algorithm explores the tree descending level by level up to the leaf nodes. In each level the K-BEST algorithm expands only the nodes with the smallest K accumulated PED.

Thereby, the detected signal vector $\hat{\mathbf{s}}$ is given by the path from the root down to the leaf node with the smallest ED. The main advantage is that the number of expanded nodes is known in advance performing fixed-complexity. However, this method does not guarantee to find the ML solution, due to it can be discard it in early decoding layers.

Fixed-Complexity Sphere Detector

In [50] a method called *Fixed-Complexity Sphere Decoder* (FSD) was proposed in order to fix the complexity of the SD method. The FSD algorithms works in two stages: In the first stage a full expansion (FE) is carried out. That means, in the first n_E tree levels all the nodes are expanded. In the second stage a single path expansion (SE) is done. That is, for the remaining $n_T - n_E$ levels the solution is computed using a SIC problem (2.25).

The symbols are detected following a specific order proposed in [50], so a preprocessing stage is performed before running the method. After the preprocessing stage, the symbols that suffer the largest noise additions are detected in the first n_E levels, since no candidates are being discarded. On the other hand, the symbols that suffer the smallest noise addition are placed at the levels where SE is performed. As for the K-BEST algorithm, the main advantage is the fixed complexity, however the solution might not coincide with the ML solution.

2.2.2 Soft-Output MIMO Detection

The optimal soft information can be achieved by computing exact a posteriori probabilities (APPs) in form of instrinsic a posteriori LLR values by employing (2.9) [51]. The sign of an LLR value indicates whether the correspoding bit $s_{i,b}$ is more likely to be 1 or 0, while the magnitude $|L_{i,b}|$ denotes the reliability of the estimate $s_{i,b}$. Large values of magnitude indicate high reliability, whereas low values correspond to estimates with low reliability. By using Bayes$'$ theorem, the probability in (2.9) can be written as [51][52]

$$L_{i,b} = \log\left(\frac{p(\mathbf{y}|s_{i,b}=1,\mathbf{H})}{p(\mathbf{y}|s_{i,b}=0,\mathbf{H})}\right). \tag{2.33}$$

The probability of a transmitted bit $s_{i,b} = 1$ is equal to the sum of all the probability combinations containing a $s_{i,b}$ equal to 1 is represented in (2.33), and can be modified by taking the expectation of $p(\mathbf{y}|\mathbf{s})$ over $\chi^1_{i,b} = \{\mathbf{s}|s_{i,b} = 1\}$. Then, the a posteriori probability LLRs $L_{i,b}$ can be computed by [53]

$$L_{i,b} = \log\left(\frac{\sum_{\mathbf{s}\in\chi^1_{i,b}} p(\mathbf{y}|\mathbf{s},\mathbf{H})P(\mathbf{s}|s_{i,b})}{\sum_{\mathbf{s}\in\chi^0_{i,b}} p(\mathbf{y}|\mathbf{s},\mathbf{H})P(\mathbf{s}|s_{i,b})}\right), \tag{2.34}$$

where $\chi^u_{i,b} = \{\mathbf{s}|s_{i,b} = u\}$ is the set of Ω^{n_T} vectors $\mathbf{s}$ having $s_{i,b} = u$. Employing the corresponding PDF for an AWGN noise, represented in (2.17), the intrinsic a posteriori probability LLR $L_{i,b}$ is set to

$$\begin{aligned} L_{i,b} &= \log\frac{\sum_{\mathbf{s}\in\chi^1_{i,b}} \exp\left(-\frac{\|\mathbf{y}-\mathbf{Hs}\|^2}{\sigma_n^2}\right)P(\mathbf{s}|s_{i,b})}{\sum_{\mathbf{s}\in\chi^0_{i,b}} \exp\left(-\frac{\|\mathbf{y}-\mathbf{Hs}\|^2}{\sigma_n^2}\right)P(\mathbf{s}|s_{i,b})} \qquad (2.35)\\ &= \log\left(\sum_{\mathbf{s}\in\chi^1_{i,b}} \exp\left(-\frac{\|\mathbf{y}-\mathbf{Hs}\|^2}{\sigma_n^2}\right)P(\mathbf{s}|s_{i,b})\right)\\ &- \log\left(\sum_{\mathbf{s}\in\chi^0_{i,b}} \exp\left(-\frac{\|\mathbf{y}-\mathbf{Hs}\|^2}{\sigma_n^2}\right)P(\mathbf{s}|s_{i,b})\right). \end{aligned}$$

The exact calculation of (2.35) requires the computation of $|\Omega|^{n_T}$ ED per LLR value, which leads to a prohibitively high computational complexity. Using the max-log approximation $\log\sum a_j \approx \max(\log(a_j))$ [54] the intrinsic max-log MAP LLRs can be calculated according to [51]

$$\begin{aligned} L_{i,b} &= \min_{\mathbf{s}\in\chi^1_{i,b}}\left(\frac{\|\mathbf{y}-\mathbf{Hs}\|^2}{\sigma_n^2} - \log P(\mathbf{s}|s_{i,b})\right) \qquad (2.36)\\ &- \min_{\mathbf{s}\in\chi^0_{i,b}}\left(\frac{\|\mathbf{y}-\mathbf{Hs}\|^2}{\sigma_n^2} - \log P(\mathbf{s}|s_{i,b})\right). \end{aligned}$$

The max-log approximation does not lead to a complexity reduction by itself. Eq. (2.36) requires the computation of the same metrics than (2.35). Nonetheless, it can be exploited to design lower complexity algorithms.

The a priori information (a.k.a soft-input) is considered and it is for example delivered by an outer channel decoder as Fig. 2.4 shows. The a priori information represents the probability of the transmitted bit being more likely 0 or 1 and can be computed as

$$L_{i,b}^{A} = \log\left(\frac{P(s_{i,b}=1)}{P(s_{i,b}=0)}\right), \qquad \forall i, b. \tag{2.37}$$

From (2.37) it follows that

$$\begin{aligned} P(s_{i,b}=1) &= \frac{\exp(L_{i,b}^{A})}{1+\exp(L_{i,b}^{A})} \qquad (2.38)\\ P(s_{i,b}=0) &= \frac{1}{1+\exp(L_{i,b}^{A})}. \qquad (2.39) \end{aligned}$$

With the previous definition based on the a priori information and the independence of the bits $s_{i,b}$ (employing BICM the bits are independent among bits and among spatial substreams) we can write

$$P(s_i|s_{i,b}) = \prod_{b:s_{i,b}=1} \frac{\exp(L_{i,b}^{A})}{1+\exp(L_{i,b}^{A})} \prod_{b:s_{i,b}=0} \frac{1}{1+\exp(L_{i,b}^{A})}. \tag{2.40}$$

Therefore $P(\mathbf{s}|s_{i,b}) = \prod_{i=1}^{n_T} P(s_i|s_{i,b})$. The equation (2.40) can be reformulated in more compact form as

$$P(s_i|s_{i,b}) = \prod_{b=1}^{k} \frac{\exp\left(s_{i,b}L_{i,b}^{A}\right)}{1+\exp(L_{i,b}^{A})}. \tag{2.41}$$

From (2.41) can be easily derived the contribution of the a priori information to the $-\log P(\mathbf{s}|s_{i,b})$ term in (2.36):

$$\log P(\mathbf{s}|s_{i,b}) = \sum_{i=1}^{n_T} \left(-\tilde{B}_i + \sum_{b=1}^{k} s_{i,b}L_{i,b}^{A}\right), \tag{2.42}$$

where the $\tilde{B}_i$ constant does not depend on $\mathbf{s}$:

$$\tilde{B}_i = \sum_{b=1}^{k} \log\left(1+\exp(L_{i,b}^{A})\right). \tag{2.43}$$

Note that intrinsic LLR-values $L_{i,b}$ are based on the received vector and the a priori LLRs $L_{i,b}^{A}$. In order to prevent that “old” information will be contained in the new a posteriori output, only "new" information has to be feeded back. The common approach is to compute intrinsic LLRs according to (2.36), and then subtracting the corresponding a priori value as in (2.12).

The hard-output MAP detection problem provides one of the two minima in (2.36), i.e.,

$$\begin{aligned} \hat{\mathbf{s}}^{MAP} &= \arg\min_{\mathbf{s}\in\Omega^{n_T}} \left(\frac{\|\mathbf{y}-\mathbf{H}\mathbf{s}\|^2}{\sigma_n^2} - \log P(\mathbf{s}) \right) && (2.44) \\ d^{MAP} &= \frac{\|\mathbf{y}-\mathbf{H}\hat{\mathbf{s}}^{MAP}\|^2}{\sigma_n^2} - \log P(\mathbf{s}^{MAP}). && (2.45) \end{aligned}$$

For each (i, b), the second minimum in (2.36) can be computed as

$$\bar{d}_{i,b} = \min_{\mathbf{s}\in\chi_{j,b}^{\left(\overline{s_{i,b}^{MAP}}\right)}} \left(\frac{\|\mathbf{y}-\mathbf{H}\mathbf{s}\|^2}{\sigma_n^2} - \log P(\mathbf{s}) \right), \tag{2.46}$$

where $\overline{s_{i,b}^{MAP}}$ denotes the complement of bit $s_{i,b}$ in $\hat{\mathbf{s}}^{MAP}$. Note that $\mathbf{s} \in \chi_{j,b}^{(\overline{s_{i,b}^{MAP}})}$ represents the counter hypotheses to the MAP solution for bit b in substream i. Once (2.44) and (2.46) have been computed, the intrinsic LLR in (2.36) is obtained as

$$L_{i,b} = L_{i,b}^{A} + (d^{MAP} - \bar{d}_{i,b})(1 - 2s_{i,b}^{MAP}), \tag{2.47}$$

where the term $(1 - 2s_{i,b}^{MAP})$ adjusts the sign depending on whether d^{MAP} corresponds to the first or the second minimum in (2.36).

On the other hand, if a priori information is not considered by the detector, considering a receiver without iteration (e.g. the one showed in Fig. 2.3), the a priori LLRs are not considered and the max-log ML LLRs in (2.36) are computed using

$$L_{i,b} = \min_{\mathbf{s}\in\chi_{i,b}^{1}} \left(\frac{\|\mathbf{y}-\mathbf{H}\mathbf{s}\|^2}{\sigma_n^2} \right) - \min_{\mathbf{s}\in\chi_{i,b}^{0}} \left(\frac{\|\mathbf{y}-\mathbf{H}\mathbf{s}\|^2}{\sigma_n^2} \right). \tag{2.48}$$

In this case, the hard-output ML solution provides one of the two minima in (2.48) obtaining

$$
\begin{aligned}
\hat{\mathbf{s}}^{ML} &= \arg\min_{\mathbf{s}\in\Omega^{n_T}} \left(\|\mathbf{y}-\mathbf{Hs}\|^2\right) && (2.49)\\
d^{ML} &= \|\mathbf{y}-\mathbf{Hs}^{\hat{M}L}\|^2 && (2.50)\\
\bar{d}_{i,b} &= \min_{\mathbf{s}\in\chi_{j,b}^{\left(\overline{s_{i,b}^{ML}}\right)}} \left(\|\mathbf{y}-\mathbf{Hs}\|^2\right). && (2.51)
\end{aligned}
$$

Therefore, the LLRs values when none iteration is considered can be computing as

$$L_{i,b} = (d^{ML} - \bar{d}_{i,b})(1 - 2s_{i,b}^{ML}). \tag{2.52}$$

Employing the max-log approximation instead of computing all 2^{kn_T} possible transmitted vectors it would be enough to calculate an hypothesis (the MAP solution for soft-input and ML solution for hard-input) as well as those $k \cdot n_T$ counter-hypothesis distances. Generally, to find exactly these distances with reasonable complexity will be difficult. For this reason, there are numerous suboptimal alternatives to avoid an exhaustive search over the entire range of possibilities.

2.3 Precoding

The precoding method sets the way in which a vector $\mathbf{x}$ is built on (2.14). The precoder processes the symbols in $\mathbf{s}$ before transmission from the antennas. At the other side, the receiver decodes the noise-corrupted received signal to recover the data bits, considering the combination of the precoder and the channel as an effective channel. Dirty paper coding (DPC) [55][56] is a theoretical precoding technique that allows to cancel the interference that is known to the transmitter without incurring a power penalty. However, the implementation of DPC requires significant complexity at both, transmitter and receiver, and suboptimal techniques have to be used.

Vector-perturbation (VP) method [24] is an interesting suboptimal technique that can be considered as a general form of precoding. This technique employs a channel inversion precoding matrix and applies a perturbation on the transmitted symbols in order to reduce the power of the transmitted signal or to maximize the SNR in case the transmit power is fixed. The VP precoded can be expressed as

$$\mathbf{x} = \mathbf{H}^{\dagger}(\mathbf{s} + \mathbf{p}). \tag{2.53}$$

Using the real-valued equivalent model, the precoded signal in (2.53) can be expressed by

$$\mathbf{x}_r = \mathbf{H}_r^{\dagger}(\mathbf{s}_r + \mathbf{p}_r). \tag{2.54}$$

The perturbation vector $\mathbf{p}$ has to be known at the K UTs, in order to allow that the UTs remove the perturbation. If the perturbation is not known at the UT, it will cause decoding errors. To avoid this situation, the perturbation vector is set to

$$\mathbf{p} = \tau\gamma \tag{2.55}$$

where τ is a positive real number and γ is a K-dimensional complex vector $\gamma_r + j\gamma_I$, with $\gamma_r, \gamma_i \in \mathbb{Z}^K$. Thus, $\mathbf{p}_r = \tau\gamma_r$, where $\gamma_r \in \mathbb{Z}^{2K}$. The τ value is chosen such that the points from the signal constellation can be uniquely recovered. If the scalar is chosen large enough, the receivers may apply the modulo operation to remove the perturbation:

$$y_j \mod \tau := y_j - \tau \left\lfloor \frac{y_j + \tau/2}{\tau} \right\rfloor, \quad j = 1, \cdots, K, \tag{2.56}$$

where the function $\lfloor \cdot \rfloor$ achieves the largest integer less than or equal to its argument. The modulo operation (2.56) is applied to the real and imaginary parts separately when the complex-valued model is employed. Assuming a M-QAM constellation, a possible value of τ is given by $\tau = 2\sqrt{M}$. A more detailed discussion about the value of τ can be found in [24]. The optimal choice of the perturbation vector is such that it minimizes the power of the transmitted signal

$$\mathbf{p}_r = \arg \min_{\mathbf{p}'_r \in \tau\mathbb{Z}^{2K}} \|\mathbf{H}_r^{\dagger}(\mathbf{s}_r + \mathbf{p}'_r)\|^2. \tag{2.57}$$

Note that (2.57) can be seen as a search for the point $\mathbf{H}_r^{\dagger}\mathbf{p}'_r$ that is closest to $-\mathbf{H}_r^{\dagger}\mathbf{s}_r$ in the lattice given by $\tau\mathbb{Z}^{2K}$, which is a search for a $2K$-dimensional lattice points. An exhaustive search over all the equivalent symbols can be done to found the optimal perturbation. However, the search can be computationally expensive, thus, Sphere search techniques are typically employed to solve the minimization. This search can also be computationally expensive and therefore, other alternative approaches, which differ in performance and complexity are usually employed to calculated the VP method. This kind of algorithms will be further detailed in a later chapter of this dissertation together with the related contributions developed in this thesis.

2.4 Performance Comparison

Numerical simulations have been performed in order to characterize the performance of the described detection and precoding algorithms. The channel matrix used within the simulation had entries from independent Gaussian random variables with unit variance and zero mean. The numerical results have been computed by Monte Carlo simulations with high enough channel realizations to provide statistically reliable results. To this end a number of minimum number of simulated frames has been defined. Unless explicitly stated otherwise, each channel realization remained constant during a block of 16 transmitted vectors. Bit Interleaved Coded Modulation (BICM) with random interleaved has generally been considered. However an uncoded system has been employed to study the performance of the different MIMO precoding schemes. Specific information about other parameters (such as, for example: frame size, constellation order or encoder type) is provided previously to each evaluation result.

3 Efficient Soft-Output Algorithms

Efficient Soft-Output Algorithms 3

The use of soft detection in MIMO-BICM systems can substantially improve the performance with respect to the use of hard detection. In contrast, higher computational requirements are needed. In this chapter the soft-output detection is approached. This chapter contains a brief description of several soft-output algorithms and their comparison. Furthermore, several contributions are presented. First, the Box Optimization Hard Detector (BOHD) has been employed to implement an efficient non-optimal soft detector. Our approach achieves almost max-log performance with reduced complexity. In the last part of the chapter, an efficient modification of the Subspace Marginalization with Interference Suppression (SUMIS) algorithm via BO strategy has been proposed. The results show that the proposed scheme reduces the computational cost of the original algorithm, in contrast the performance worsens slightly.

3.1 Soft MIMO Detection Problem

Section 2.1.2 of chapter 2 describes a generic MIMO-BICM systems with n_T transmit antennas, n_R receive antennas (with $n_R \geq n_T$) and a certain SNR.

As it was already presented, in a BICM-MIMO scheme the soft detection and channel decoding are not performed jointly. First, the demodulator provides reliability information about the transmitted coded bits in form of real-valued LLRs. Next, these values are used by the channel decoder to make final decisions on the transmitted coded bits. The considered scheme throughout this section is represented in Fig. 2.3, note that no information is exchanged between the soft detector and the decoder. As seen in Section 2.2.2, the optimal soft information searched by the channel detector is the a posteriori log-likelihood ratio

$$L_{i,b} = \log \frac{P(s_{i,b} = 1|\mathbf{y}, \mathbf{H})}{P(s_{i,b} = 0|\mathbf{y}, \mathbf{H})}, \tag{3.1}$$

where $s_{i,b}$ is the $b-$th bit of the $i-$th symbol of the transmitted vector $\mathbf{s}$. The equality (3.1) tell us, given some $\mathbf{y}$ and conditioned on $\mathbf{H}$, how likely the $b-$th bit of the $i-$th symbol of $\mathbf{s}$ is zero or one. Section 2.2.2 proves that by using Bayes$'$ rule and assuming equal a priori probabilities, the LLRs can be computed as

$$L_{i,b} = \log \frac{\sum_{\mathbf{s}\in\chi_{i,b}^{1}} \exp\left(-\frac{\|\mathbf{y}-\mathbf{Hs}\|^2}{\sigma_n^2}\right)}{\sum_{\mathbf{s}\in\chi_{i,b}^{0}} \exp\left(-\frac{\|\mathbf{y}-\mathbf{Hs}\|^2}{\sigma_n^2}\right)}. \tag{3.2}$$

The equality of the a priori probabilities is assumed since no iteration is performed. M^{n_T} terms need to be evaluated and added in (3.2). This involves a significant complexity, which grows exponentially with n_T. Therefore, many different approximate methods have been proposed.

3.1.1 Max-log Soft Detection

Meaningful estimation of (3.2) is the so called max-log approximation, presented in Section 2.2.2. Using this approximation the additions of the numerator and denominator are replaced by their corresponding largest term and (3.2) can be approximated to

$$L_{i,b} \approx \min_{\mathbf{s}\in\chi_{i,b}^{1}} \left(\frac{\|\mathbf{y}-\mathbf{Hs}\|^2}{\sigma_n^2}\right) - \min_{\mathbf{s}\in\chi_{i,b}^{0}} \left(\frac{\|\mathbf{y}-\mathbf{Hs}\|^2}{\sigma_n^2}\right). \tag{3.3}$$

The max-log approximation does not lead to a computational reduction by itself, since the same number of terms has to be evaluated. However, (3.3) can be used by different strategies to reduce the computational cost.

As Subsection 2.2.2 showed, one of the two minima in (3.3) is the distance associated to the ML solution. Thus, equation (3.3) can be computed as

$$L_{i,b} = (d^{ML} - \bar{d}_{i,b})(1 - 2s_{i,b}^{ML}). \tag{3.4}$$

Therefore, the soft-output max-log algorithms must compute the hard ML solution $\mathbf{s}^{\mathrm{ML}}$, its associated ML distance d^{ML}, and the counter-hypothesis distance $\bar{d}_{i,b}$ for all $i = 1, \ldots, n_T$, $b = 1, \ldots, k$.

There are several soft-output detection algorithms that use the max-log approximation to compute the LLRs. Some of these soft-output algorithms are *Repeated Tree Search* (RTS) [57], a modified RTS algorithm [58], *Single Tree Search* (STS) [59], the *List-based SD* (LSD) scheme [51], *Soft-output Fixed-complexity* (SFSD) [60], the *Smart Ordering and Candidate Adding* (SOCA) algorithms [17], and *Soft-output K-BEST* [61][62].

Some of these algorithms provide exact max-log LLRs, such us RTS and STS. However there are other soft-output methods (like LSD or the SFSD algorithms) that do not guarantee finding the exact distance, providing approximations to the max-log LLRs, this entails a certain loss of performance. Thereby, some accuracy is usually lost in order to obtain better computational complexity.

On the other hand, clipping strategy [63] can be applied to reduce the complexity of the tree search algorithms. Given a clipping parameter L_{clip}, it is assumed that any counter-hypothesis distance larger than $d^{\mathrm{ML}} + L_{clip}$ does not need to be computed exactly and can be set to the value $d^{\mathrm{ML}} + L_{clip}$. When clipping is applied to a soft-output max-log method, the resulting method cannot strictly be called max-log because the LLRs are no longer exact. However, it is important to note that any max-log soft-output algorithm that is applied with a given clipping parameter L_{clip}, must compute exactly the same LLRs as any other max-log soft-output method that is applied with the same clipping parameter. In other words, all of the counter-hypothesis distances that are larger than $d^{\mathrm{ML}} + L_{clip}$ are set to $d^{\mathrm{ML}} + L_{clip}$, and all of the counter-hypothesis distances that are smaller than $d^{\mathrm{ML}} + L_{clip}$ are computed exactly. Therefore, the accuracies obtained by any two max-log soft-output algorithms that use the same clipping parameter are the same, and the performance obtained by these algorithms would be the same.

3.1.2 Partitioned Model

To address the drawbacks of the exact detection in (3.2) an alternative approximation based on a partitioned model can be employed. This approximation uses and adjustable parameter denoted as n_s, with $n_s \in 1, \cdots n_T$. Thus, based on (2.1) the partitioned model can be defined as

$$\mathbf{y} = \mathbf{H}\mathbf{s} + \mathbf{v} = \begin{bmatrix} \bar{\mathbf{H}} & \tilde{\mathbf{H}} \end{bmatrix} \begin{bmatrix} \bar{\mathbf{s}}^T & \tilde{\mathbf{s}}^T \end{bmatrix}^T + \mathbf{v} = \bar{\mathbf{H}}\bar{\mathbf{s}} + \tilde{\mathbf{H}}\tilde{\mathbf{s}} + \mathbf{v}, \tag{3.5}$$

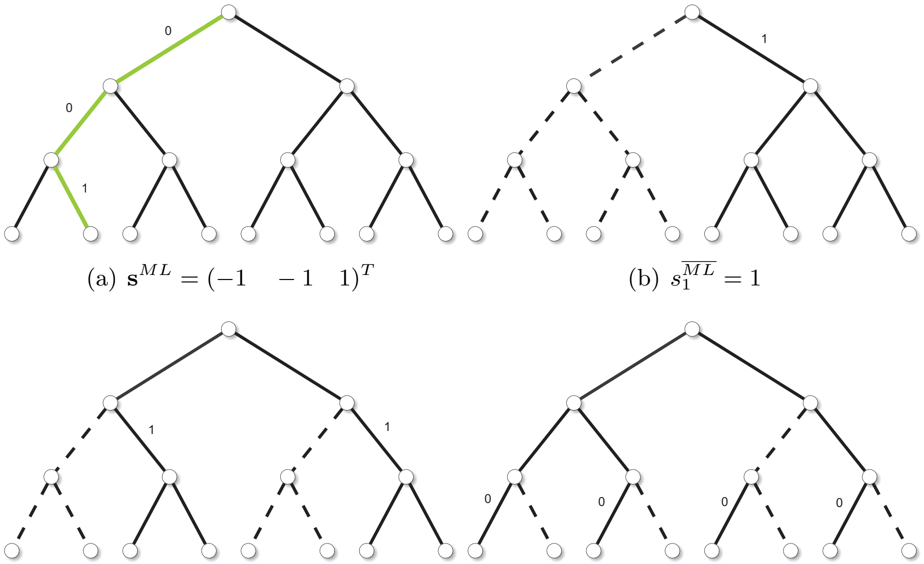
where $\bar{\mathbf{H}} \in \mathbb{C}^{n_R \times n_s}$, $\tilde{\mathbf{H}} \in \mathbb{C}^{n_R \times (n_T - n_s)}$, $\bar{\mathbf{s}} \in \Omega^{n_s}$, $\tilde{\mathbf{s}} \in \Omega^{n_T - n_s}$. The choice of this partitioned model involves a permutation on $\mathbf{H}$ which is performed in a preprocessing stage. How to perform this permutation is not straightforward and depends on the algorithm implemented to deal with (3.5). For example, the hard FSD algorithm described in Subsection 2.2.1 employs this partitioned model and the preprocessing stage of this algorithm attempts to minimize the condition number of the matrix $\tilde{\mathbf{H}}$. There are several methods based on the partitioned model: *Partial Marginalization* [14], SUMIS [64] or soft MMSE [65]. These algorithms carry out different preprocessing stages and their computational complexities and performances depend on the n_s parameter.

3.2 Soft MIMO Detection Algorithms

Some remarkable detectors are going to be presented throughout this section.

3.2.1 Repeated Tree Search

The RTS algorithm [58] starts by computing the hard ML solution ($\mathbf{s}^{\mathrm{ML}}$ and d^{ML}), solving (2.49) and computing (2.50), through a ML SD algorithm (already described in Subsection 2.2.1). The adaptive radius SESD is usually selected for this purpose. Then, the LLRs are obtained by computing the counter-hypothesis distances (2.51). These are obtained by running a ML SESD for each bit in the symbol vector. When the SESD is rerunning, the tree search is prepruned by forcing the detector to exclude all nodes from the search for which $s_{i,b} = s_{i,b}^{ML}$. Figure 3.1 illustrates this prepruning procedure for a BPSK constellation and $n_T = 3$ transmit antennas.



(a) $\mathbf{s}^{ML} = (-1 \quad -1 \quad 1)^T$

(b) $s_1^{\overline{ML}} = 1$

(c) $s_2^{\overline{ML}} = 1$

(d) $s_3^{\overline{ML}} = 0$

Figure 3.1. Prepruning procedure of the RTS algorithm for a BPSK constellation and $n_T = 3$ transmit antennas. Counter-hypothesis distances to the ML solution are found by running SD and discarding the dashed branches.

Therefore, the SESD algorithm must be executed $n_T \cdot k + 1$ times. The drawback of this procedure is clearly the increased complexity, especially for low SNR. However, it must be mentioned that once the hard ML solution has been obtained, the computation of each LLR is independent from the others, so the computation of the LLRs can be straightforwardly parallelized.

Clipping is easily included in the RTS strategy. Since $\mathbf{s}^{\mathrm{ML}}$ and d^{ML} have been computed previously, the SD runs needed for computing each distance $\bar{d}_{j,b}$ are started by using the clipping distance $d^{\mathrm{ML}} + L_{clip}$ as the initial maximum radius. This considerably reduces the number of explored nodes and the computation time [63].

3.2.2 Single Tree Search

The STS algorithm proposed in [59] is a sophisticated method that is designed to compute the hard ML solution and the counter-hypothesis distances at the same time, traversing the tree of possible solutions only once. In [59], the STS algorithm was proven to be more efficient than the RTS algorithm. STS has the standard SESD structure. However, in order to detect d^{ML} and all of the distances $\bar{d}_{i,b}$ simultaneously, the radius must be larger (it must be kept at least as large as $max\left(\bar{d}_{i,b}\right)$), and the radius is recomputed before computing any node or leaf.

Algorithm 1 and Algorithm 2 give a brief overview of the distinguishing features of the STS algorithm, which are the update rules and the method for recalculating the radius. This overview is based on the description given in [59]. Variables $\mathbf{s_{opt}}$ and d_{opt} are used to store the best signal and distance found at the present moment. Before starting the algorithm, the variables d_{opt} and $\bar{d}_{i,b}$ are initialized: $d_{opt} = \bar{d}_{i,b} = \infty,\ \forall i, b$.

Algorithm 1 The update rules to be applied when a feasible leaf is found.

```
If a leaf $\mathbf{s}$ is found such that $d(\mathbf{s}) < d_{opt}$, Then
    $\forall i, b$ such that the bit $s_{i,b} = \overline{s^{\mathrm{ML}}_{i,b}}$
        Set $\bar{d}_{i,b}$ to the value $d_{opt}$.
    Set $d_{opt} = d(\mathbf{s})$.
    Set $\mathbf{s_{opt}} = s$.
End If
If a leaf $\mathbf{s}$ is found such that $d(\mathbf{s}) \geq d_{opt}$, Then
    $\forall i, b$ such that the bit $c_{j,b} = \overline{c^{\mathrm{ML}}_{i,b}}$ and $d(\mathbf{s}) < \bar{d}_{i,b}$,
        Set $\bar{d}_{j,b}$ to the value $d(\mathbf{s})$.
End If
```

When the STS concludes, $\mathbf{s_{opt}} = \mathbf{s}^{\mathrm{ML}}$, $d_{opt} = d^{ML}$, and $\bar{d}_{i,b}$ holds the counter-hypothesis distance. All the nodes (and leaves) with a PED in the interval $[d^{\mathrm{ML}}, \max(\bar{d}_{i,b})]$ will have been visited. If the difference between d^{ML} and $\max(\bar{d}_{i,b})$ is large, then the number of visited nodes can become prohibitively large. Since this happens frequently, clipping is needed for any practical implementation.

Algorithm 2 The method for recalculating the radius (which is applied in every explored node) in order to determine whether the node is expanded or pruned. Let $\mathbf{s}_{k:n_T}$ be a partial transmit vector (node) at level k

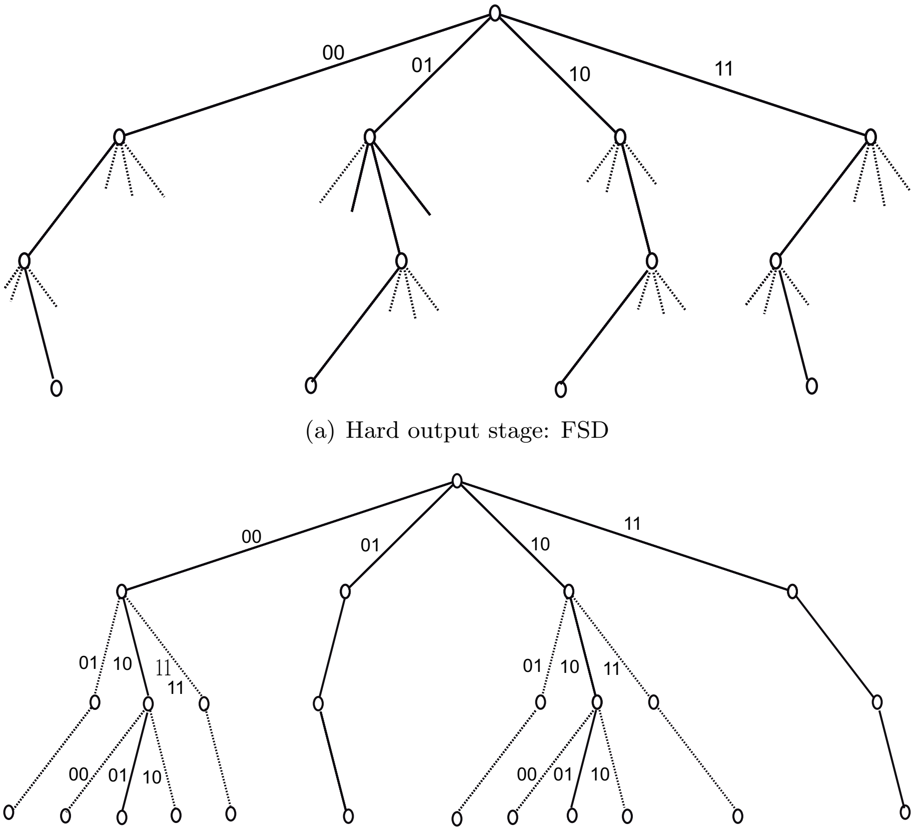
```
1.  Set $d_1 = max\left(\bar{d}_{i,b}\right)$, $\forall b$, for $i = 1 \dots k-1$
2.  Set $d_2 = max\left(\bar{d}_{i,b}\right)$, $\forall b$, for $i = k \dots n_T$
        and $(\mathbf{s}_{k:n_T})_{i,b} = \overline{s_{i,b}^{\mathrm{ML}}}$
3.  If $d(\mathbf{s}_{k:n_T}) > max(d_1, d_2)$ Then, $\mathbf{s}_{k:m}$ is pruned
4.      Else $\mathbf{s}_{k:m}$ is expanded
5.  End If
```

Clipping is included in STS by modifying the updating of Algorithm 1, adding the final update given by $\bar{d}_{i,b} = min\left(\bar{d}_{i,b}, d_{opt} + L_{clip}\right) \forall i, b$ after line 10. When the search concludes, all nodes whose PED is bounded within in the interval $[d^{\mathrm{ML}}, d^{\mathrm{ML}} + L_{clip}]$ have been visited.

3.2.3 Soft Fixed Sphere Decoder

The FSD algorithm is an efficient hard detector that achieves quasi ML hard detection, this algorithm was described in Subsection 2.2.1. The method consists of a preprocessing stage followed by a predetermined tree search with fixed complexity. The soft-output FSD (SFSD) algorithm was proposed in [60] to provide soft information after the FSD search. Thus the SFSD algorithm extends FSD in order to obtain the minimum distances in (2.51) after (2.50) has been solved. The SFSD algorithm is based in two main stages. This stages are depicted in Fig. 3.2 for a $4-$QAM constellation and $n_T = 3$.

In the first stage SFSD runs the FSD algorithm as Subsection 2.2.1 describes, Fig. 3.2(a) illustrates an example of decoding tree for a $n_E = 1$ being n_E the input variable to the FSD algorithm which fixes the computational complexity. In the second stage, the method starts from the list of candidates computed in the first stage and adds new candidates to provide more information about the counter bits. Note that, all the possible values of the constellation for each survivor path in the first n_E levels are assigned



(a) Hard output stage: FSD

(b) Soft extension.

Figure 3.2. Decoding trees of the SFSD algorithm for a $n_T = 3$ MIMO system and $4-$QAM constellation. The values chosen for the n_E and N_{iter} input parameters are 1 and 2 respectively.

to the symbol at the current level (i.e all the necessary values to compute the LLRs of the symbol bits in the first n_E levels are available). Therefore, the list extension must start from the $n_T - n_E$ level, as Fig. 3.2(b) shows. To begin the list extension, the best N_{iter} paths are selected from the initial hard-output FSD list. This is motivated in the heuristics that the lowest-distance paths may be candidates differing from the best paths in only some bits. The symbols belonging to these N_{iter} paths are picked up from the root up to a certain level $i = n_T - n_E$, and, at level $i - 1$, additional $\log_2 M$ branches are explored. Each additional branch has one of the bit of the initial path symbol negated and is completed following the SIC path, as was done in the hard-output FSD algorithm. The same operation is repeated until the lowest level of the tree is reached. In [60] was proved that for a 4×4 MIMO system the values that achieve almost max-log performance are $n_E = 1$ and $N_{iter} = 2, 4, 6$ for $4-$QAM, $16-$QAM and $64-$QAM, respectively.

3.2.4 Subspace Marginalization with Interference Suppression

SUMIS [64] algorithm yields better performance than max-log methods at low and fixed complexity. SUMIS provides a well-defined tradeoff between computational complexity and performance. The algorithm implements an initial sorting stage consisting of selecting channel matrix columns which define the partitioned model on (3.5). Then, two stage are performed based on the permutation matrix. In Stage I, a first approximation to the LLR values is computed and then, in Stage II, new refined LLR values are calculated using these approximate values. Here we give a brief revision of the SUMIS algorithm.

Stage I: The algorithm starts with the partitioned model (3.5) by defining the new model as

$$\overline{\mathbf{y}} = \overline{\mathbf{H}}\overline{\mathbf{s}} + \mathbf{n} \tag{3.6}$$

where $\mathbf{n} = \widetilde{\mathbf{H}}\widetilde{\mathbf{s}} + \mathbf{v}$ is a Gaussian stochastic vector $\mathbf{n} \sim \mathcal{N}(\mathbf{0}, \mathbf{Q})$ with $\mathbf{Q} = \widetilde{\mathbf{H}}\widetilde{\mathbf{H}}^T + \sigma_n^2\mathbf{I}$. Using the following operator $\|\mathbf{x}\|^2_{\mathbf{Q}} \triangleq \mathbf{x}^T\mathbf{Q}^{-1}\mathbf{x}$, we compute the $\lambda_{i,b}$ as

$$\lambda_{i,b} = \log \frac{\sum_{\overline{\mathbf{s}} \epsilon \chi^0_{i,b}} \exp(-\frac{1}{2}\|\mathbf{y} - \overline{\mathbf{H}}\overline{\mathbf{s}}\|^2_Q)}{\sum_{\overline{\mathbf{s}} \epsilon \chi^1_{i,b}} \exp(-\frac{1}{2}\|\mathbf{y} - \overline{\mathbf{H}}\overline{\mathbf{s}}\|^2_Q)}. \tag{3.7}$$

The $\lambda_{i,b}$ values are approximate $L_{i,b}$ LLRs. Thereby, Stage I is performed for all bits $b = 1, \cdots, k$ in all symbols $i = 1, \cdots, n_T$.

Stage II: In the second stage, the LLR values are computed over a new model. In this stage, the interfering vector $\widetilde{\mathbf{s}}$ is suppressed in (3.6) and then the LLR values are computed over a purified model. In this context the new model is given by

$$\mathbf{y}' \triangleq \mathbf{y} - \widetilde{\mathbf{H}}\mathbb{E}\{\widetilde{\mathbf{s}}|\mathbf{y}\} \approx \overline{\mathbf{H}}\overline{\mathbf{s}} + \mathbf{n}', \tag{3.8}$$

where $\mathbb{E}\{\widetilde{\mathbf{s}}|\mathbf{y}\}$ is the conditional expected value of vector $\widetilde{\mathbf{s}}$ and $n' \backsim \mathcal{N}(\mathbf{0}, \mathbf{Q}')$ with $\mathbf{Q}' \triangleq \widetilde{\mathbf{H}}\widetilde{\mathbf{\Upsilon}}\widetilde{\mathbf{H}}^T + \sigma_n^2\mathbf{I}$. $\widetilde{\mathbf{\Upsilon}}$ is the conditional covariance matrix of $\widetilde{\mathbf{s}}$ and can be computed by

$$\widetilde{\mathbf{\Upsilon}} = \mathbb{E}\{\text{diag}(\widetilde{\mathbf{s}})^2|\mathbf{y}\} - \mathbb{E}\{\text{diag}(\widetilde{\mathbf{s}})|\mathbf{y}\}^2 \tag{3.9}$$

where diag(ζ) gives a diagonal matrix with the elements of ζ vector on its diagonal.

The conditional expected value $\mathbb{E}\{s_i|\mathbf{y}\}$ of symbol s_i with $i = 1, \cdots, n_T$ should be computed as

$$\begin{aligned}\mathbb{E}\{s_i|\mathbf{y}\} &\triangleq \sum_{\mathbf{s}\in\Omega} sP(s_i = s|\mathbf{y}) \approx \sum_{\mathbf{s}\in\Omega} sP(s_i = s|\overline{\mathbf{y}})|_{\overline{\mathbf{y}}=\mathbf{y}} \\ &= \sum_{\mathbf{s}\in\Omega} s \prod_{b=1}^{k} \frac{1}{1 + e^{(-2s_{i,b}+1)\lambda_{i,b}}}. \end{aligned} \tag{3.10}$$

Hence, the LLR values are computed as

$$L_{i,b} \approx \log \frac{\sum_{\overline{\mathbf{s}}\epsilon\chi_{i,b}^0} \exp(-\frac{1}{2}\|\mathbf{y}' - \overline{\mathbf{H}}\overline{\mathbf{s}}\|_{Q'}^2)}{\sum_{\overline{\mathbf{s}}\epsilon\chi_{i,b}^1} \exp(-\frac{1}{2}\|\mathbf{y}' - \overline{\mathbf{H}}\overline{\mathbf{s}}\|_{Q'}^2)}. \tag{3.11}$$

The optimal permutation in the preprocessing stage that determines $\overline{\mathbf{H}}$ and $\widetilde{\mathbf{H}}$ allows that the interfering vector $\widetilde{\mathbf{s}}$ in (3.6) has the least influence on the useful signal vector $\overline{\mathbf{s}}$. This permutation is based on $\mathbf{H}^T\mathbf{H}$ [64], which has the next structure

$$\mathbf{H}^T\mathbf{H} = \begin{pmatrix} \sigma_1^2 & \rho_{12} & \cdots \\ \rho_{21} & \sigma_2^2 & \cdots \\ \vdots & \vdots & \ddots \end{pmatrix}. \tag{3.12}$$

Then, when the calculation of the LLR values in the i symbol in $\mathbf{s}$ is performed, we pick the $n_s - 1$ indexes that correspond to the largest $|\rho_{ij}|$ values, being n_s de number of columns in $\overline{\mathbf{H}}$. These indexes, along with the index i, specify the columns from $\mathbf{H}$ that are placed in $\overline{\mathbf{H}}$, the rest columns remain in $\widetilde{\mathbf{H}}$.

Note that the processing per bit can be performed in parallel. This allows for massively parallel algorithmic implementations such as those presented in the last chapter of this dissertation.

3.2.5 Linear Soft-Output MMSE

One of the best low-complexity method to approximate (3.1) is the well-known linear soft-output MMSE detector [65][66]. Let's consider one symbol as the signal of interest and the rest as Gaussian interference. Thus, the MIMO model can be written as

$$\mathbf{y} = \overline{\mathbf{h}}\overline{s} + \tilde{\mathbf{H}}\tilde{\mathbf{s}} + v \approx \overline{\mathbf{h}}\overline{s} + \mathbf{n}. \tag{3.13}$$

Therefore, using partial marginalization, the problem can be seen as the exact LLR computation in (3.13) with n_T single-input multiple-output detection problems that can be independently solved. Furthermore, the soft MMSE detector can be seen as an special case of SUMIS algorithm setting $n_s = 1$ and performing only *Stage I*.

3.2.6 Analysis of Performance and Complexity

In order to assess the performance of the described soft-output algorithms, numerical simulations have been carried out. The simulations were performed for a convolutional code of rate 1/2 and codeword size 4096, generator polynomials $[133_o\ 171_o]$ and constraint length 7. The channel decoding is performed using the sum-product BCJR algorithm with max-log approximation. The number of antennas has been set to $n_T = n_R = 4$ and the constellation order to $4 - QAM$. The performance has been compared in terms of Frame Error Rate (FER) and the results are shown in Fig. 3.3.

The simulation results show that the Linear soft-output MMSE detector gives the worst performance. It is important to note that for the SFSD algorithm the values of the input parameters have been set to $n_E = 1$ and $N_{iter} = 2$, achieving almost max-log performance. The max-log performance represented in Fig. 3.3 can be achieved by the STS or RTS algorithms. On the other hand, the results show clearly that the SUMIS algorithm, setting n_s to 3, outperforms the max-log detection and performs close to the exact solution, as was proved in [64]. The SUMIS algorithm outperforms the max-log detection for different antenna sizes with low values of n_s, even for medium or large number or antennas. That is, SUMIS algorithm outperforms today's state of the art detectors.

It is important to give some details regarding the complexity of the algorithms to carry out a fair comparison,. The linear detector exhibits the lowest complexity, since it can be considered as n_T single-input multiple-

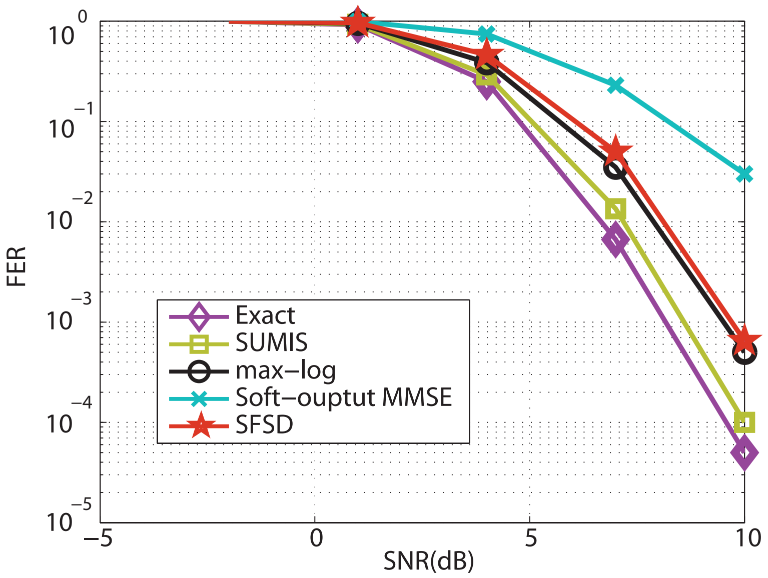


Figure 3.3. FER performance for different soft-output algorithms for a 4 × 4 MIMO system and 4−QAM constellation.

output detection, beside this algorithm presents fixed complexity. The SFSD algorithm achieves almost max-log performance and requires less computational resources than the RTS or STS algorithms. This algorithm also presents fixed complexity since the first n_E levels of the tree are expanded and SIC detection is performed for each expanded node. In summary, a QR decomposition and the SIC expansion are needed. On the other hand, the RTS and STS algorithms achieve max-log performance. However, the computational cost of these algorithms is unknown at the beginning and can not be prefixed. Both algorithms use the SD method to solve equation (3.3). The RTS algorithm runs the tree search one per LLR value to be computed, the main disadvantage is that it may visit the same nodes multiple times. On the other hand, the STS algorithm is more efficient than RTS since it runs only the tree once, providing clear advantages over the RTS algorithm. Even so, STS is not a practical algorithm due to its high computational cost, which increases exponentially with the number of antennas. Finally, the SUMIS algorithm, which outperforms the max-log detection methods, it is a computationally much more efficient algorithm. Employing the SUMIS algorithm M^{n_s} terms are evaluated instead of M^{n_T}, this means that when the number of antennas or constellation order increases, the computational complexity is drastically reduced with respect to the optimal solution and the max-log algorithms. Furthermore, its computational complexity is fixed and known at the beginning.

3.3 Box Optimization for MIMO Detection

Continuous constrained optimization techniques are employed to help SD-based detection algorithms in hard detection [48]. The auxiliary problem to be solved is:

$$\hat{\mathbf{sr}} = \underset{\substack{\mathbf{s}\in\mathbb{C}^m \\ \min(\Re(\Omega))\leq\Re(s_i)\leq\max(\Re(\Omega))\ ,1\leq i\leq n_T \\ \min(\Im(\Omega))\leq\Im(s_i)\leq\max(\Im(\Omega))\ ,1\leq i\leq n_T}}{\arg\min} \|\mathbf{R}\cdot\mathbf{s}-\mathbf{z}\|^2 , \tag{3.14}$$

where $s_i, 1 \leq i \leq n_T$ are the components of the vector $\mathbf{s}$. This problem is derived from (2.27), disregarding the condition that the components of the solution must belong to the constellation Ω. Compared to problem (2.27), this is a continuous problem since the components of the solution vector do not need to belong to Ω. The only restriction is that the search zone be bounded and has the form of a box, hence the name of Box Optimization (BO).

3.3.1 Box Optimization to obtain an initial point and an initial radius for SD

As Subsection 2.2.1 in chapter 2 describes, to start the search, some versions of SD require an initial feasible point, an initial radius, or both. It is quite common to solve the continuous unconstrained least squares problem:

$$\hat{\mathbf{s}} = \arg\min_{\mathbf{s}\in\mathbb{C}^m} \left\|\mathbf{R}\cdot\mathbf{s}-\mathbf{z}\right\|^2 . \tag{3.15}$$

All the components of $\hat{\mathbf{s}}$ are then rounded to the nearest element of the constellation Ω (this process is called quantization). The vector obtained after this process is $\hat{\mathbf{s}}_q$, which is known as the Zero-Forcing (ZF) estimator. This estimator may be a good approximation to $\mathbf{s}^{\mathrm{ML}}$ when the SNR is high, but it is known to give poor results if the SNR decreases.

When one or more of the components of the vector $\hat{\mathbf{s}}$ have real or imaginary parts outside of the interval $[\min\left(\Omega_{\mathbb{R}}\right), \max\left(\Omega_{\mathbb{R}}\right)]$ (it is assumed that there exists a set $\Omega_{\mathbb{R}}$ such that the constellation Ω can be obtained as a cartesian product $\Omega_{\mathbb{R}} \times i\cdot\Omega_{\mathbb{R}}$), we say that the vector $\hat{\mathbf{s}}$ is out of the constellation. Accordingly, we say that $\hat{\mathbf{s}}$ is in the constellation when all its components have their real and imaginary parts inside the interval $[\min\left(\Omega_{\mathbb{R}}\right),\ \max\left(\Omega_{\mathbb{R}}\right)]$.

With large SNR, the estimator $\hat{\mathbf{s}}$ should be in the constellation, or at least very close to it. In this case, the ZF estimator $\hat{\mathbf{s}}_q$ should be reasonably close to the ML solution. However, for small SNR, the estimator $\hat{\mathbf{s}}$ will usually be out of the constellation, and the $\hat{\mathbf{s}}_q$ estimator may no longer be a good approximation to the ML solution. In that case, the estimator $\hat{\mathbf{sr}}_{\mathbf{q}}$, which is computed by quantizing the result of the BO $\hat{\mathbf{sr}}$, will surely be a better approximation to the ML solution $\mathbf{s}^{\mathrm{ML}}$. Therefore, $\hat{\mathbf{sr}}_{\mathbf{q}}$ may be used as an initial point for an sphere decoder or even as a non-ML estimator of $\mathbf{s}^{\mathrm{ML}}$.

Another possibility proposed in [67] is the use of $\hat{\mathbf{sr}}_{\mathbf{q}}$ to compute an initial SD radius as follows,

$$r_{\hat{\mathbf{sr}}_{\mathbf{q}}} = \left\|\mathbf{H}\cdot\hat{\mathbf{sr}}_{\mathbf{q}}-\mathbf{y}\right\| . \tag{3.16}$$

As reported in [67], in large noise situations, the radius estimate $r_{\hat{\mathbf{s}}\hat{\mathbf{r}}_{\mathbf{q}}}$ is usually a closer estimation to d^{ML} than the standard radius estimate computed by the ZF estimator AS

$$r_{\hat{\mathbf{s}}_{\mathbf{q}}} = \left\| \mathbf{H} \cdot \hat{\mathbf{s}}_{\mathbf{q}} - \mathbf{y} \right\| . \tag{3.17}$$

Therefore, as a conclusion for this subsection, BO can be used to obtain a better starting point for the search as well as a initial radius closer to d^{ML}.

3.3.2 Radius Bound for SD Search using Box Optimization

The second technique where BO is involved tries to obtain a tighter radius estimation before the expansion of each node. This technique was first proposed and described in [68]; the proposal was to obtain a bound that is tighter than (2.32). In the level l of the tree ($1 < l < n_T$), a partial transmit vector will have been obtained, which implies that components $l+1, \cdots, n_T$ have already assigned values belonging to the constellation. Components $1, \cdots, l-1$ do not have assigned values yet, and a decision must be taken regarding component l. Therefore, in level l, expression $d(\mathbf{s}) < r^2$ is rewritten as:

$$\begin{gathered} d(\mathbf{s}) = \left\| \mathbf{R} \cdot \mathbf{s} - \mathbf{z} \right\|^2 = \\ \left\| \mathbf{R}_{1:l-1,1:l-1} \cdot \mathbf{s}_{1:l-1} + \mathbf{R}_{1:l-1,l:n_T} \cdot \mathbf{s}_{l:n_T} - \mathbf{z}_{1:l-1} \right\|^2 + \\ \left\| \mathbf{R}_{l:n_T,l:n_T} \cdot \mathbf{s}_{l:n_T} - \mathbf{z}_{l:n_T} \right\|^2 \leq r^2 . \end{gathered} \tag{3.18}$$

Using the remaining term in inequality (3.18), which is given by

$$\left\| \mathbf{R}_{1:l-1,1:l-1} \cdot \mathbf{s}_{1:l-1} + \mathbf{R}_{1:l-1,l:m} \cdot \mathbf{s}_{l:m} - \mathbf{z}_{1:l-1} \right\|^2 , \tag{3.19}$$

a lower bound c can be obtained, so (3.18) can be alternatively written as:

$$d\left(\mathbf{s}^{(l)}\right) = \left\| \mathbf{R}_{l:n_T,l:n_T} \cdot \mathbf{s}_{l:n_T} - \mathbf{z}_{l:n_T} \right\|^2 \leq r^2 - c \, , \tag{3.20}$$

which is a tighter pruning condition than (2.32). This should provide a reduction in the number of feasible values of $\mathrm{s}^{(l)}$, and, consequently, a reduction in the number of visited nodes. If c is indeed a lower bound of (3.19), equation (3.18) holds. Then, if the initial radius is selected so that there is at least a solution fulfilling (3.18), the resulting method will still be max-log.

In [68], several methods to compute lower bounds of (3.19) were proposed, discussed, and evaluated. One of the proposals in [68] was to use BO to compute a lower bound of (3.19). This can be done considering (3.19) as a deflated MIMO detection problem. If the continuous least squares problem is solved:

$$\begin{aligned}\hat{\mathbf{s}}^{(l-1)} = \underset{\mathbf{s}_{1:l-1}\in\mathbb{C}^{l-1}}{\arg\min}\ & \|\mathbf{R}_{1:l-1,1:l-1}\cdot\mathbf{s}_{1:l-1} + \\ & \mathbf{R}_{1:l-1,l:n_T}\cdot\mathbf{s}_{l:n_T} - \mathbf{z}_{1:l-1}\|^2 \ ,\end{aligned} \tag{3.21}$$

the estimator $\hat{\mathbf{s}}^{(l-1)}$ is obtained, which is analogous to the estimator $\hat{\mathbf{s}}$ computed as in (3.15) but for the deflated problem. It must be noted that problem (3.21) is actually a standard triangular system of linear equations, whose solution $\hat{\mathbf{s}}^{l-1}$ is computed exactly and fulfills:

$$\left\|\mathbf{R}_{1:l-1,1:l-1}\cdot\hat{\mathbf{s}}^{l-1} + \mathbf{R}_{1:l-1,l:n_T}\cdot\mathbf{s}_{l:n_T} - \mathbf{z}_{1:l-1}\right\| = 0 \ . \tag{3.22}$$

If $\hat{\mathbf{s}}^{l-1}$ is out of the constellation, then the estimator $\widehat{\mathbf{sr}}^{l-1}$ (which is analogous to $\widehat{\mathbf{sr}}$ for the deflated problem) is computed solving the BO problem for the deflated problem:

$$\widehat{\mathbf{sr}}^{l-1} = \underset{\substack{\mathbf{s}_{1:l-1} \\ \min(\Omega_{\mathbb{R}})\le\Re(\mathrm{s}_i)\le\max(\Omega_{\mathbb{R}});1\le i\le l-1 \\ \min(\Omega_{\mathbb{R}})\le\Im(\mathrm{s}_i)\le\max(\Omega_{\mathbb{R}});1\le i\le l-1\ .}}{\arg\min} \|\mathbf{R}_{1:l-1,1:l-1}\cdot\mathbf{s}_{1:l-1} + \mathbf{R}_{1:l-1,k:n_T}\cdot\mathbf{s}_{l:n_T} - \mathbf{z}_{1:l-1}\|^2 \tag{3.23}$$

Problem (3.23) is analogous to (3.14) and can also be solved using BO techniques. For all $\mathbf{s}_{1:l-1}\in\Omega^{l-1}$, the solution $\widehat{\mathbf{sr}}^{l-1}$ fulfills that:

$$\begin{aligned}&\left\|\mathbf{R}_{1:l-1,1:l-1}\cdot\widehat{\mathbf{sr}}^{l-1} + \mathbf{R}_{1:l-1,l:n_T}\cdot\mathbf{s}_{l:n_T} - \mathbf{z}_{1:l-1}\right\|^2 \\ \le\ &\|\mathbf{R}_{1:l-1,1:l-1}\cdot\mathbf{s}_{1:l-1} + \mathbf{R}_{1:l-1,l:n_T}\cdot\mathbf{s}_{l:n_T} - \mathbf{z}_{1:l-1}\|^2 \ .\end{aligned} \tag{3.24}$$

Hence, the proposal is to use the lower bound c, which is given by

$$c = \left\|\mathbf{R}_{1:l-1,1:l-1}\cdot\widehat{\mathbf{sr}}^{l-1} + \mathbf{R}_{1:l-1,l:n_T}\cdot\mathbf{s}_{l:n_T} - \mathbf{z}_{1:l-1}\right\|^2 , \tag{3.25}$$

in inequality (3.20). Of course, if $\hat{\mathbf{s}}^{l-1}$ is in the constellation, $\hat{\mathbf{s}}^{l-1} = \widehat{\mathbf{sr}}^{l-1}$. Thus, as in (3.22), the bound would be useless since

$$\left\|\mathbf{R}_{1:l-1,1:l-1}\cdot\widehat{\mathbf{sr}}^{l-1} + \mathbf{R}_{1:l-1,l:n_T}\cdot\mathbf{s}_{l:n_T} - \mathbf{z}_{1:l-1}\right\|^2 = 0 \ . \tag{3.26}$$

In this case, it would be better to use other bounding techniques, (e.g., the technique based on the minimum singular value described in [68]) or simply not use any additional bound, since the standard SESD algorithm performs quite well in this case.

Paper [48] presents the implementation of a SESD hard detector including the techniques described above, plus a number of improvements and algorithmic optimizations. We will refer to the hard ML detector described in [48] as the *Box Optimization Hard Detector* (BOHD). The BOHD algorithm is orders of magnitude faster than standard ML SD detectors when it is applied to large problems (large modulation or large number of antennas), especially in the low SNR range. Furthermore, practical results show that the performance of the BOHD algorithm is virtually constant across any SNR range (even for impractical SNRs). The key for the performance of this algorithm is that the BO (proposed in [68] and later improved in [48]) provides an extremely tight bound on the search radius. This causes a drastic decrease both in the number of nodes that must be explored to obtain the ML solution and consequently in the required execution time.

3.3.3 Probability of being out of the constellation

For large SNR, the estimator $\hat{\mathbf{s}}$ should be in the constellation, however, the BOHD algorithm shows improvements for all the SNR values evaluated in [48], not only for low SNR. This situation is due to it depends also on the channel matrix condition number. In this section, the probability that $\hat{\mathbf{s}}$ is out the constellation, P_{out}, is being calculated.

The BO algorithm is used when the ZF estimator is out of the box that is defined by the limits of the constellation. The ZF estimator is calculated in (3.14) as:

$$\hat{\mathbf{s}} = \mathbf{H}^{\dagger}\mathbf{y} = \mathbf{s} + \mathbf{H}^{\dagger}\mathbf{v}, \tag{3.27}$$

where $\mathbf{H} \in \mathbb{C}^{n_R \times n_T}$ with $(n_R \geq n_T)$, $\mathbf{y} \in \mathbb{C}^{n_R \times 1}$, $\mathbf{v} \in \mathbb{C}^{n_R \times 1}$ and $\mathbf{s} \in \mathbb{C}^{n_T \times 1}$. The BO algorithm is applied when any component of $\hat{\mathbf{s}}$ is outside of the box that is delimited by the constellation. Therefore, BO execution probability can be computed as the union of n_T compatible events, since it will be the probability that one or more components of $\hat{\mathbf{s}}$ get outside. In a simple case with only two antennas, we would have the probability of the union of two events:

$$\begin{aligned} P_{out} &= P(s_1 \cup s_2)_{out} \\ &= P(s_1)_{out} + P(s_2)_{out} - P(s_1 \cap s_2)_{out} \\ &= P(s_1)_{out} + P(s_2)_{out} - P(s_1)_{out} \cdot P(s_2)_{out}, \end{aligned} \tag{3.28}$$

where $P(s_1)_{out}$ represents the probability of being out the box of the first component in $\hat{\mathbf{s}}$ and $P(s_2)_{out}$ represents the probability of the second component. On the other hand, $P(s_1 \cap s_2)_{out}$ can be calculated as $P(s_1)_{out} \cdot P(s_2)_{out}$ because the events are independent. Extending this result to the n_T transmit antennas case, we have:

$$\begin{aligned} P_{out} = &\sum_{i=1}^{n_T} P(s_i)_{out} - \sum_{i \neq j} P(s_i \cap s_j)_{out} + \sum_{i \neq j \neq k} P(s_i \cap s_j \cap s_k)_{out} - \\ &- \ldots + (-1)^{(n_T+1)} P \left(\bigcap_{i=1}^{n_T} s_i \right)_{out} , \end{aligned} \tag{3.29}$$

where $P(s_i)_{out}$ represents the probability of being out the box of the s_i component. The next step is to compute the $P(s_i)_{out}$ values.

As can be observed in (3.27), the s_i component may or may not be outside, depending on the noise variance and on the channel matrix pseudoinverse. First of all, when the channel matrix has ones in its diagonal and zeros otherwise, the probability only depends on the noise. Thus, the probability of a component being out ($P(s_i)_{out}$) will be the probability that the real part is out or the probability that the imaginary part is out, or both. In other words:

$$P(s_i)_{out} = 2p(s_i)_{out} - p(s_i)_{out}^2. \tag{3.30}$$

The probability that the real or imaginary part is out is denoted by $p(s_i)_{out}$, so (3.30) can be computed as a union of two independent events.

We will assume that there exists a set $\Omega_{\mathbb{R}}$ such that the constellation Ω can be obtained as $\Omega_{\mathbb{R}} \times \Omega_{\mathbb{R}}$. Therefore, if Ω is an $M-QAM$ constellation, each component (real and imaginary) of the constellation belongs to a L-PAM constellation with L equal to $\sqrt{M}$. Let the sent symbol be the m-th constellation point and let $p(s_{i,m})_{out}$ be the probability of getting outside, which can be calculated as:

$$p(s_{i,m})_{out} = p(n < [1-m]d) + p(n > [L-m]d), \qquad m = 1, \ldots, L. \tag{3.31}$$

where d is the distance between contiguous elements of the constellation and n is a gaussian noise of zero mean and σ_n^2 variance. Since it can be checked by symmetry that

$$p(s_{i,m})_{out} = p(s_{i,L-m+1})_{out}, \tag{3.32}$$

if the symbols are equiprobable, the average probability will be:

$$\begin{aligned} p(s_i)_{out} &= \frac{\sqrt{M}}{M} \cdot 2 \sum_{m=1}^{L/2} p(s_{i,m})_{out} \qquad (3.33) \\ &= \frac{1}{\sqrt{M}} \cdot 2 \sum_{m=1}^{L/2} \left\{ p\left(n < [1-m]d\right) + p\left(n > [L-m]d\right) \right\}. \end{aligned}$$

Given the symmetry of $Q(x)$ we have

$$p(n > z) = p(n < -z) = Q\left(\frac{z}{\sigma_n}\right) \tag{3.34}$$

with

$$Q(x) = \frac{1}{2}\text{erfc}\left(\frac{x}{\sqrt{2}}\right) = \frac{2}{\sqrt{\pi}} \int_x^\infty e^{-t^2} \cdot dt. \tag{3.35}$$

Thus:

$$p\left(n < [1-m]d\right) = p\left(n > [m-1]d\right) = Q\left(\frac{[m-1]d}{\sigma_n}\right) \tag{3.36}$$

$$p\left(n > [L-m]d\right) = Q\left(\frac{[L-m]d}{\sigma_n}\right). \tag{3.37}$$

It is necessary to describe the relationship between the quotient d/σ_n and the SNR. If the constellation is polar, then the coordinates of the $m-th$ symbol (i.e., $E_m = (2m-1)d/2$) will be the opposite of the $M-m+1-th$. Thus, their energies will be equal. In this case, the energy average by symbol can be expressed as:

$$\begin{aligned} E_s &= \frac{1}{L} \sum_{m=1}^{L} E_m = \frac{2}{L} \cdot \sum_{m=1}^{L/2} \left[(2m-1)\frac{d}{2} \right]^2 \qquad (3.38) \\ &= \frac{2}{L} \cdot \sum_{m=1}^{L/2} \left[(2m-1)^2 \frac{d^2}{4} \right] = \frac{d^2}{12}(L^2-1). \end{aligned}$$

Then

$$\mathrm{SNR} = \frac{E_s \cdot n_T}{\sigma^2} = \frac{\frac{1}{12}d^2(L^2-1)}{\sigma^2} \cdot n_T. \tag{3.39}$$

According to the previous equation

$$\frac{d}{\sigma} = \sqrt{\frac{12}{L^2-1} \cdot \frac{\mathrm{SNR}}{n_T}} \tag{3.40}$$

However, a non-ideal channel will affect (3.33). It can be observed in (3.27), how the noise varies depending on the pseudoinverse of the channel, $\breve{\mathbf{n}} = \mathbf{H}^\dagger \mathbf{n}$. Thus, the noise in each receiver antenna is computed as

$$\breve{n}_i = \mathbf{H}^\dagger_{\underline{i}} \mathbf{n}, \tag{3.41}$$

where the subindex $\mathbf{H}^\dagger_{\underline{i}}$ denotes the i-th row of $\mathbf{H}^\dagger$. Therefore, the power of the new noise in each receiver antenna can be obtained as

$$\begin{aligned} E\left\{\breve{n}_i \breve{n}_i{}^H\right\} &= E\left\{\mathbf{H}^\dagger_{\underline{i}} \mathbf{n}\mathbf{n}^H (\mathbf{H}^\dagger_{\underline{i}})^H\right\} = \mathbf{H}^\dagger_{\underline{i}} E\left\{\mathbf{n}\mathbf{n}^H\right\} (\mathbf{H}^\dagger_{\underline{i}})^H & (3.42)\\ &= \mathbf{H}^\dagger_{\underline{i}} \sigma_n^2 (\mathbf{H}^\dagger_{\underline{i}})^H & (3.43)\end{aligned}$$

Then, computing the SVD decomposition of $\mathbf{H}$ as

$$\mathbf{H} = \mathbf{U}\mathbf{S}\mathbf{V}^H = \mathbf{U}\begin{bmatrix}\boldsymbol{\Sigma}\\ \mathbf{0}\end{bmatrix}\mathbf{V}^H, \tag{3.44}$$

we obtain

$$\mathbf{H}^\dagger_{\underline{i}} = \mathbf{V}_{\underline{i}}\begin{bmatrix}\boldsymbol{\Sigma}^{-1} & \mathbf{0}\end{bmatrix}\mathbf{U}^H = \mathbf{V}_{\underline{i}}\begin{bmatrix}\boldsymbol{\Sigma}^{-1} & \mathbf{0}\end{bmatrix}\begin{bmatrix}\mathbf{U}_1^H\\ \mathbf{U}_2^H\end{bmatrix} = \mathbf{V}_{\underline{i}}\boldsymbol{\Sigma}^{-1}\mathbf{U}_1^H. \tag{3.45}$$

Therefore

$$\begin{aligned} \mathbf{H}^\dagger_{\underline{i}}(\mathbf{H}^\dagger_{\underline{i}})^H &= \mathbf{V}_{\underline{i}}\boldsymbol{\Sigma}^{-1}\mathbf{U}_1^H\mathbf{U}_1\boldsymbol{\Sigma}^{-1}\mathbf{V}_{\underline{i}}^H & (3.46)\\ &= \mathbf{V}_{\underline{i}}\boldsymbol{\Sigma}^{-2}\mathbf{V}_{\underline{i}}^H. & (3.47)\end{aligned}$$

Thus, the power of the new noise in each receiver antenna is calculated as

$$\begin{aligned} \sigma_{\breve{n}_i}{}^2 &= \sigma_n^2 \mathbf{H}^\dagger_{\underline{i}}(\mathbf{H}^\dagger_{\underline{i}})^H & (3.48)\\ &= \sigma_n^2 \mathbf{V}_{\underline{i}}(\boldsymbol{\Sigma}^{-2})\mathbf{V}_{\underline{i}}^H. & (3.49)\end{aligned}$$

Therefore, taking into account the channel matrix, the $p(s_i)_{out}$ probability is finally given by

$$p(s_i)_{out} = \frac{1}{\sqrt{M}} \cdot 2 \sum_{m=1}^{L/2} \left\{ Q\left(\frac{[m-1]d}{\sigma_{\breve{n}_i}}\right) + Q\left(\frac{[L-m]d}{\sigma_{\breve{n}_i}}\right) \right\} \tag{3.50}$$

Table 3.1 illustrates the good concordance between P_{out} using (3.29) and experimental probability. These probabilities have been computed as an average of 1000 complex gaussian channel realizations and have been represented for different numbers of antennas, constellation orders, and SNR values.

	4-QAM		16-QAM		64-QAM	
SNR(dB)	4 × 4	8 × 8	4 × 4	8 × 8	4 × 4	8 × 8
0	99,90\99,92	1\1	98,94\98,99	99,99\99,99	97,62\97,57	99,97\99,97
5	99,77\99,81	1\1	96,54\96,70	99,92\99,93	91,85\91,44	99,60\99,55
10	99,68\99,71	1\1	93,49\93,91	99,66\99,73	83,10\82,45	98,00\97,74
15	99,64\99,65	1\1	91,38\91,84	99,35\99,44	74,61\74,64	94,81\94,63
20	99,60\99,62	1\1	90,46\90,72	99,14\99,21	69,35\69,77	91,45\91,73

Table 3.1. Experimental\theoretical probabilities of $\hat{\mathbf{s}}$ be out of the constellation in %.

It is important to no note that for the numbers of antennas, constellation orders, and SNR values evaluated, the estimator $\hat{\mathbf{s}}$ is out the constellation with high probability, this means that the BO algorithm will be executed with very high probability.

3.4 BOHD Soft-Output Algorithm

The BOHD [48] algorithm presented in Section 3.3 implements an efficient max-log hard-output detector. The large reduction in time and visited nodes shown in [48] for hard detection can be employed to implement soft-output detectors. Two algorithms have been proposed, which preserve the max-log property while at the same time show efficiency improvement of soft-output max-log detection. However these algorithms will be carefully described in a later chapter since a priori information has been incorporated and can be employed in an iterative receiver. On the other hand, a proposed non-optimal soft-output algorithm which is based on a repeated strategy to compute the counter-hypothesis distances is detailed in this section. The proposed algorithm is called *Box Optimization Hard Detector with soft-output* (BOHD-SO) algorithm, since provides soft information employing the efficient hard BOHD detector.

First, a tree search is run once, using the BOHD algorithm. Thus, the ML solution in (3.4) is computed in a very efficient way. Then, once $\mathbf{s}^{ML}$ and d^{ML} have been calculated, we need to compute the counter-hypothesis distances ($\bar{d}_{i,b}$) in (3.4). If the optimal counter-hypothesis distances are computed, the computational complexity increases. The distances computed by the BOHD-SO algorithm are not optimal and are obtained using the BO algorithm. Algorithm 3 shows how the BOHD-SO works.

Algorithm 3 BOHD-SO algorithm

```
Input: $\mathbf{z}$, $\mathbf{R}$,$\Omega$,$\mathbf{s}$
Output: $\mathbf{L}_{i,b} \forall i, b$
1:  $\mathbf{s}^{ML}$ and $d^{ML}$ are computed by BOHD
2:  $\hat{\mathbf{s}} = \arg\min_{\mathbf{s}\in\mathbb{C}^{n_T}} \|\mathbf{z} - \mathbf{R}\mathbf{s}\|^2$
3:  if $\hat{\mathbf{s}}$ is in the constellation then
4:      $\mathbf{s}_{bo} = \hat{\mathbf{s}}$
5:  else
6:      BO is applied to obtain $\mathbf{s}^b$ as in (3.14)
7:      $\mathbf{s}_{bo} = \mathbf{s}^b$
8:  end if
9:  for $i = 1$ to $n_T$ do
10:     for $b = 1$ to $k$ do
11:         $\mathbf{s}_{bo}$ is rounded to the nearest element of $\tilde{\Omega}$ obtaining $\bar{d}_{i,b}$
12:         $L_{i,b}$ is computed using (3.4)
13:     end for
14: end for
```

From lines 3 to 8 of Algorithm 3, the general method to perform the BO algorithm is applied but performing a slight variation. That is, if the $\hat{\mathbf{s}}$ estimator is out the constellation, employing the BO algorithm we can obtain a better estimator than $\hat{\mathbf{s}}$. After this step, the rounding operation is performed in line 11, where $\tilde{\Omega}_{i,b}$ is the restricted Ω. By $\tilde{\Omega}_{i,b}$, we mean that the considered constellation points are those that have the corresponding bit complemented. To compute the counter-hypothesis distance associated with the $L_{i,b}$ value when the (i,b) bit of the $\mathbf{s}^{ML}$ solution is equal to zero, we only consider the constellation points with the (i,b) bit equal to one. Figure 3.4 illustrates the restricted constellation in a 2×2 real-valued MIMO system with constellation $\Omega = [-3 \quad -1 \quad 1 \quad 3]$ and $s_{1,2}^{MAP} = 0$.

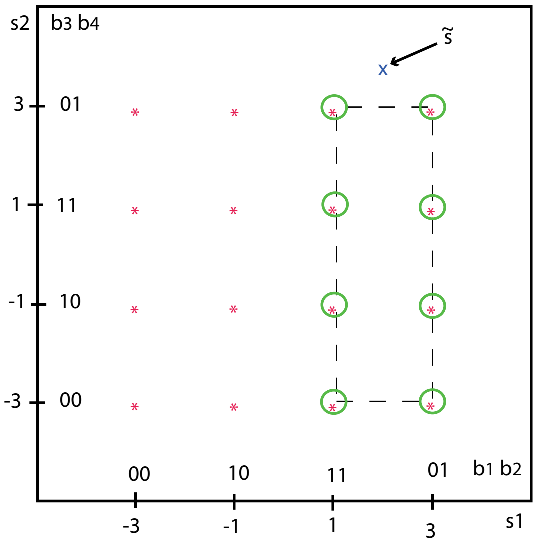


Figure 3.4. Restricted constellation $\tilde{\Omega}_{i,b}$ in a 2×2 real-valued MIMO system with constellation $\Omega = [-3 \quad -1 \quad 1 \quad 3]$ and $s_{1,2}^{MAP} = 0$. The restricted constellation is formed by the symbols belonging to the box delimited by de dashed line.

Thus, we obtain all of the counter-hypothesis distances by repeating the rounding operation strategy for each bit. The BO method could be executed for each counter-hypothesis distance, by applying the BO algorithm and considering $\tilde{\Omega}$ instead of Ω as the input argument when (3.14) is computed. However, this means that the BO method has to be applied for each counter-hypothesis distance, increasing the computational cost.

It is important to note that, from lines 3 to 8 of Algorithm 3, the $\mathbf{s}_{bo}$ vector is computed. However, this vector has already been calculated by the BOHD algorithm in the preprocessing stage, when box optimization is employed to obtain an initial radius for SD (see 3.3.1 or Algorithm 1 in [48]). Thus, the calculation of $\mathbf{s}_{bo}$ can be omitted in Algorithm 3, reducing the computational cost.

It was also verified by simulation that the performance in terms of BER of this algorithm is far from the max-log ML performance, due to the simple way in which the counter-hypothesis distances are obtained.

However, we can get closer to the max-log performance by using a transformed channel matrix with better orthogonality properties than the original, employing lattice reduction techniques (LR). The LR techniques consist of finding another base with better orthogonality properties than the original one. The transformation is performed on the **H** channel matrix finding another base with better orthogonality. Different reduction techniques have been proposed [69], however the algorithm proposed by Lenstra, Lenstra and Lovász known as LLL [70] is the most commonly used because although it is a suboptimal method, it offers a good trade-off between performance and complexity. However, the total number of arithmetic operations required by the LLL algorithm is difficult to predict due to the lack of a bounded worst case complexity. For this reason, the fixed complexity LLL algorithm proposed in [71] is employed. The steps of the fixed complexity LLL are included in Algorithm 4. Then, when LR is carried out before BOHD-SO in a preprocessing stage the result is very close to max-log, as will be illustrated in the Results section.

3.4.1 Analysis and Results

In this section the performance and complexity of the proposed BOHD-SO algorithm is presented. For this purpose, Monte Carlo simulations were carried out in order to evaluate our proposal and compare it with other algorithms. A convolutional encoder of codeword size 4096 and rate 1/2 was used. The generator polynomials $[133_o, 171_o]$, constraint length 7 and max-log BCJR channel decoder with max-log approximation were chosen.

Algorithm 4 Fixed complexity LLL Algorithm [71]

```
Input: $\mathbf{Q}$, $\mathbf{R}$, Y, $\delta$
Output: $\tilde{\mathbf{Q}}$, $\tilde{\mathbf{R}}$, $\mathbf{T}$
$\tilde{\mathbf{Q}} = \mathbf{Q}$, $\tilde{\mathbf{R}} = \mathbf{R}$, $\mathbf{T} = \mathbf{I}_{n_T}$, $m = 2$
for $i = 1, \cdots Y$ do
    while $m \leq n_T$ do
        for $l = m - 1 \cdots, 1$ do
            $\mu = \text{round}(\tilde{R}_{l,m}/\tilde{R}_{l,l})$
            if $\mu \neq 0$ then
                $\tilde{\mathbf{R}}_{i:l,m} = \tilde{\mathbf{R}}_{1:l,m} - \mu\tilde{\mathbf{R}}_{1:l,l}$
                $\mathbf{T}_{:,m} = \mathbf{T}_{:,m} - \mu\mathbf{T}_{:,m}$
            end if
        end for
        if $\delta|\tilde{R}_{m-1,m-1}|^2 > |\tilde{R}_{m,m}|^2 + |\tilde{R}_{m-1,m}|^2$ then
            $\tilde{\mathbf{R}}_{:,m-1} \longleftrightarrow \tilde{\mathbf{R}}_{:,m}$
            $\mathbf{T}_{:,m-1} \longleftrightarrow \mathbf{T}_{:,m}$
            $\boldsymbol{\Theta} = \begin{bmatrix} \tilde{R}_{m-1,m-1} & \tilde{R}_{m,m-1} \\ -\tilde{R}_{m,m-1} & \tilde{R}_{m-1,m-1} \end{bmatrix}$
            $\boldsymbol{\Theta} = \boldsymbol{\Theta}/\|\tilde{\mathbf{R}}_{m-1:m,m-1}\|$
            $\tilde{\mathbf{R}}_{m-1:m,m-1:n_T} = \boldsymbol{\Theta}\tilde{\mathbf{R}}_{m-1:m,m-1:n_T}$
            $\tilde{\mathbf{Q}}_{:,m-1:m} = \tilde{\mathbf{Q}}_{:,m-1:m}\boldsymbol{\Theta}^H$
        end if
        $m = m + 1$
    end while
end for
```

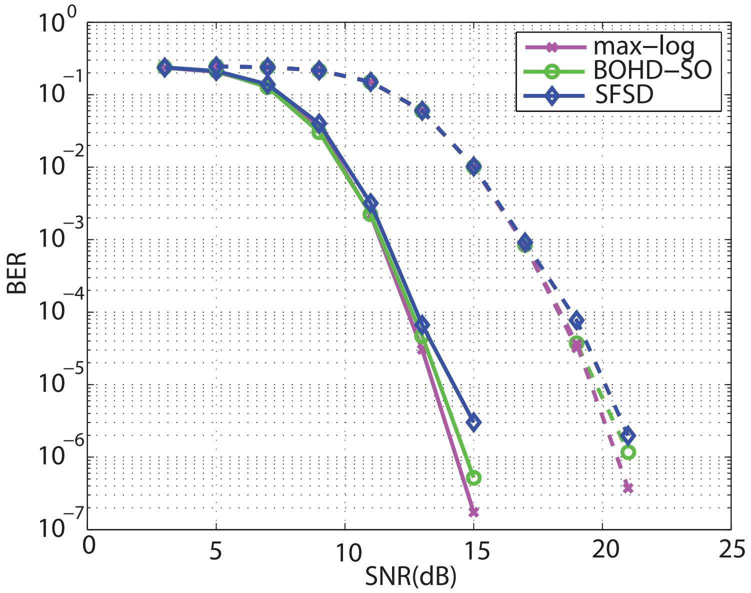


Figure 3.5. BER performance for an optimal max-log method and BOHD-SO and SFSD methods in a 4 × 4 MIMO system with a 16-QAM constellation (continuous lines) and a 64-QAM constellation (discontinuous lines).

The SFSD algorithm was chosen for comparison because, it achieves almost max-log ML performance with low complexity. The performance of the algorithms was compared in terms BER and computational cost. A 4×4 complex MIMO scenario with $16-$QAM and $64-$QAM constellations was chosen. Figure 4.2.3 shows the BER performance for the proposed BOHD-SO method, the SFSD algorithm and the max-log case. The curves illustrate how the BOHD-SO and SFSD achieve almost max-log performance for both constellation sizes, with BOHD-SO being slightly better than SFSD (especially for 64-QAM constellations).

The computational cost in terms of floating point operations (flops) is represented in Fig. 3.6, comparing the efficiency of BOHD-SO and SFSD. Any algorithm which achieves the max-log performance (for example RTS or STS) requires much more computational resources than these algorithms. The curves show how BOHD-SO needs less flops than SFSD, achieving better accuracy in terms of BER.

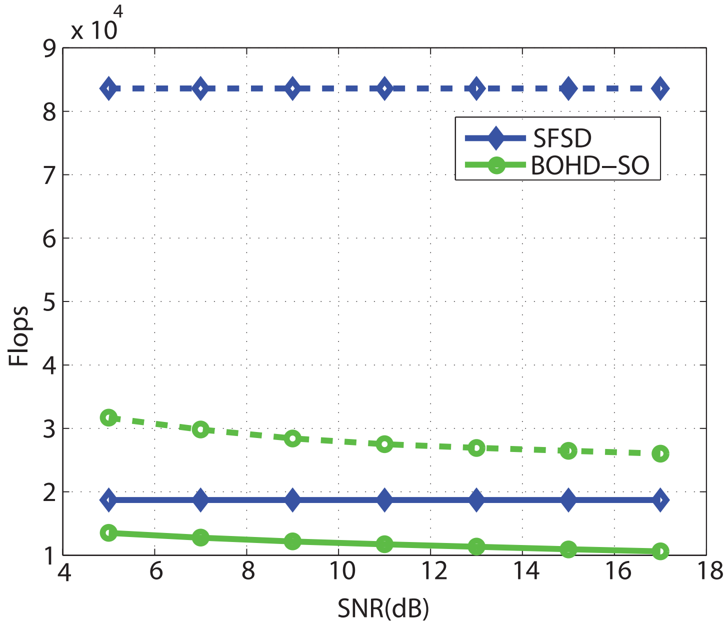


Figure 3.6. Number of flops for the BOHD-SO and SFSD methods in a 4×4 MIMO system with a 16-QAM constellation (continuous lines) and a 64-QAM constellations (discontinuous lines).

Therefore, the BOHD-SO method is a non-optimal soft-output algorithm that achieves almost max-log performance with very low complexity, obtaining less complexity in terms of flops and a more accurate response than the SFSD algorithm when LR is applied in a preprocessing stage. Thus, the BOHD-SO algorithm is a good option when a non-iterative receiver is used, providing very good performance with low complexity.

3.5 SUMIS-BO

The SUMIS algorithm presented in Subsection 3.2.4 provides a clear trade-off between computational complexity and detection performance. The SUMIS algorithm computes the $\lambda_{i,b}$ in *Stage I* using (3.7), being the number of terms in the summation over $\bar{\mathbf{s}}$ is equal to M^{n_s}. This implies that SUMIS algorithm has to compute the term $\exp(-\frac{1}{2}\|\mathbf{y} - \overline{\mathbf{H}}\bar{\mathbf{s}}\|_Q^2)$ a number of times $M^{n_s} \times n_T \times k$ to compute the total $\lambda_{i,b}$ values. The proposed algorithm in this section, called as *Subspace Marginalization with Interference Suppression and Box Optimization* (SUMIS-BO) algorithm, aims to reduce the complexity of this part, computing approximate values of $\lambda_{i,b}$, not the exact ones. These values are not the final LLRs but are employed in an interference suppression mechanism in Stage II.

In the SUMIS-BO, the max-log approximation is employed together with the BO technique to compute the $\lambda_{i,b}$ values. Thereby, the algorithm finds a good approximation of $\min_{\overline{\mathbf{s}} \epsilon \chi^0_{i,b}} \|\mathbf{y} - \overline{\mathbf{H}}\overline{\mathbf{s}}\|^2$ and $\min_{\overline{\mathbf{s}} \epsilon \chi^1_{i,b}} \|\mathbf{y} - \overline{\mathbf{H}}\overline{\mathbf{s}}\|^2$, which are denoted as $\overline{\mathbf{s}}_0$ and $\overline{\mathbf{s}}_1$ respectively. Therefore, the *Stage I* of the original SUMIS algorithm is modified.

Modified Stage I: The main idea of the proposed algorithm is to employ the BO method to compute $\overline{\mathbf{s}}_0$ and $\overline{\mathbf{s}}_1$ in the *Stage I.* To execute the BO algorithm, firstly, the linear detector ZF is computed as in (3.27) obtaining

$$\hat{\overline{\mathbf{s}}} = \overline{\mathbf{H}}^{\dagger}\mathbf{y}. \tag{3.51}$$

This known ZF estimator is a meaningful starting point for the BO algorithm. It is important to note that, ZF estimator requires a matrix inversion which can be computationally very complex for higher antenna dimension. However, the number of columns of $\overline{\mathbf{H}}$ is given by n_s, which it is relatively low. For this reason, we can use the QR decomposition of the $\overline{\mathbf{H}}$ matrix and compute the ZF estimation of the equivalent problem. In this case, the matrix inverse in (3.51) will be of size $n_s \times n_s$, avoiding the problem for high antenna dimensions.

It was demonstrated in [48] that if any component is outside of the box that delimits the constellation, we can improve the accuracy of the ZF estimator by applying the BO method. Then, the new estimated vector is quantized to the nearest element of the constellation. On the other hand, in cases where all components of the ZF estimator are within the constellation, the BO method is not applied and the components of the ZF estimator are rounded to the nearest element of the constellation. It is important to note that the constellation used for quantizing the ZF and BO estimators is restricted, using only the points of the constellations with the corresponding bit equal to one or zero, which depends on whether we calculate $\overline{\mathbf{s}}_1$ or $\overline{\mathbf{s}}_0$ respectively.

Once $\bar{\mathbf{s}}_0$ and $\bar{\mathbf{s}}_1$ have been computed, the $\lambda_{i,b}$ values can be calculated by

$$\lambda_{jb} = \frac{1}{2}(\|\mathbf{y} - \overline{\mathbf{H}}\bar{\mathbf{s}}_0\|_Q^2 - \|\mathbf{y} - \overline{\mathbf{H}}\bar{\mathbf{s}}_1\|_Q^2). \tag{3.52}$$

The *Stage I* of the SUMIS-BO algorithm is summarized in Algorithm 5. The *Stage II* remains equal than the original reviewed in Subsection 3.2.4.

Algorithm 5 *Stage I* SUMIS-BO pseudo-code

```
Input: $H$, $y$, $k$ and $n_s$
for $i = 1$ to $n_T$ do
    Decide a partition in (3.6).
    Calculate $\hat{\bar{\mathbf{s}}}$ in (3.51).
    for $b = 1$ to $k$ do
        Compute $\bar{\mathbf{s}}_1$ and $\bar{\mathbf{s}}_0$ applying BO method.
        Calculate $\lambda_{i,b}$ using (3.52).
    end for
    Calculate $\mathbb{E}\{s_i|\mathbf{y}\}$ using (3.10).
end for
```

Employing the modified first Stage, only two terms are needed for each $\lambda_{i,b}$ value reducing considerably the complexity. Table 3.2 represents the number of terms that we have to compute for SUMIS and SUMIS-BO for a different system parameters. Thus, it is clear that the number of terms to evaluate in the computation of the $\lambda_{i,b}$ values is drastically reduced.

SUMIS\SUMIS-BO	$M = 4$	$M = 16$	$M = 64$
$n_T = 8$	1024\32	131e3\64	125e5\96
$n_T = 16$	2048\64	262e3\128	251e5\192
$n_T = 24$	3072\96	393e3\192	377e5\288

Table 3.2. Number of terms $\frac{1}{2}(\|\mathbf{y} - \overline{\mathbf{H}}\bar{\mathbf{s}}\|_Q^2)$ that SUMIS\SUMIS-BO has to compute for $n_s = 3$.

As we will see in the simulation results, the employed method in this calculation affects very slightly to the final result performance. The key of this technique to keep good performance is the use of the BO method.

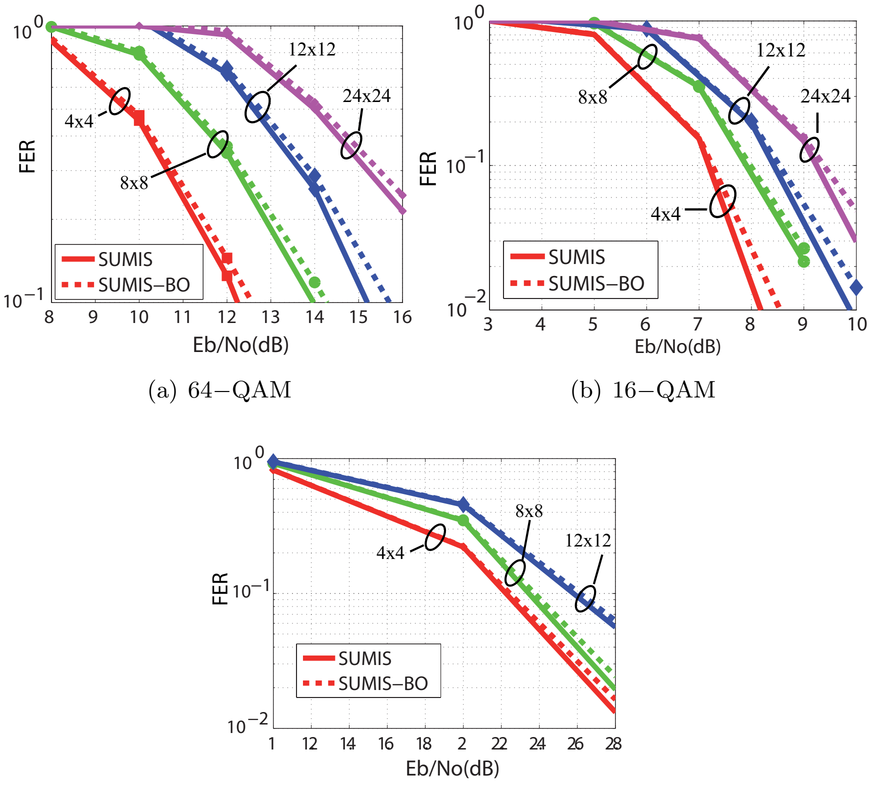
3.5.1 Analysis and Results

In order to evaluate the proposed SUMIS-BO algorithm, it has been compared with the original SUMIS. Since the SUMIS method has yet been compared with other algorithms in a previous section and in [64], we omit plotting this performance and complexity comparisons. The performance has been evaluated by the computation of FER by means of Monte Carlo simulations varying the SNR, defined as E_b/N_0. A rate 1/2 Low-Density Parity-Check (LDPC) code of codeword size 1296 bits is also used. The LDPC encoding and decoding schemes come from the IEEE 802.11n WLAN standard; and some software tools have been download from http://www.csl.cornell.edu/∽studer/software_ldpc.html. There is no iteration between the detector and the decoder and the sum-product algorithm has been chosen as the channel decoding option.

There is no iteration between the detector and the decoder, and the transmitted symbols are assumed to have equal probability. We simulate different complex MIMO systems with different M-QAM constellations: 4×4 with M equal to 16 and 64; 8×8 with M equal to 4, 16 and 64; 12×12 with M equal to 4, 16 and 64; and 24×24 with M equal to 4 and 16. The number n_s over which the partitioning model is done, is equal to 3 since this value offers a well defined performance-complexity tradeoff, see [64].

Fig. 3.7 illustrates the performance comparison between the SUMIS and SUMIS-BO algorithms with different number of antennas and constellation orders. The results show clearly that for $4-$QAM constellation the SUMIS-BO detector performs as SUMIS algorithm for all E_b/N_0 values and all system size. However, note that for higher constellation orders the SUMIS-BO algorithm achieves slightly worse behavior regarding SUMIS algorithm. It is important to note that this performance loss is negligible and in contrast, the improvement in the computational cost of the algorithm is very high.

The computational cost is represented in Fig. 3.8 in terms of flops. Figure 3.8 illustrates that the proposed SUMIS-BO algorithm reduces drastically the computational cost of SUMIS for all the studied problem sizes and retains the fixed complexity of the SUMIS algorithm. Table 3.3 represents the percentage of flops improvement of SUMIS-BO with respect to SUMIS. This table shows that SUMIS-BO algorithm reaches large advances

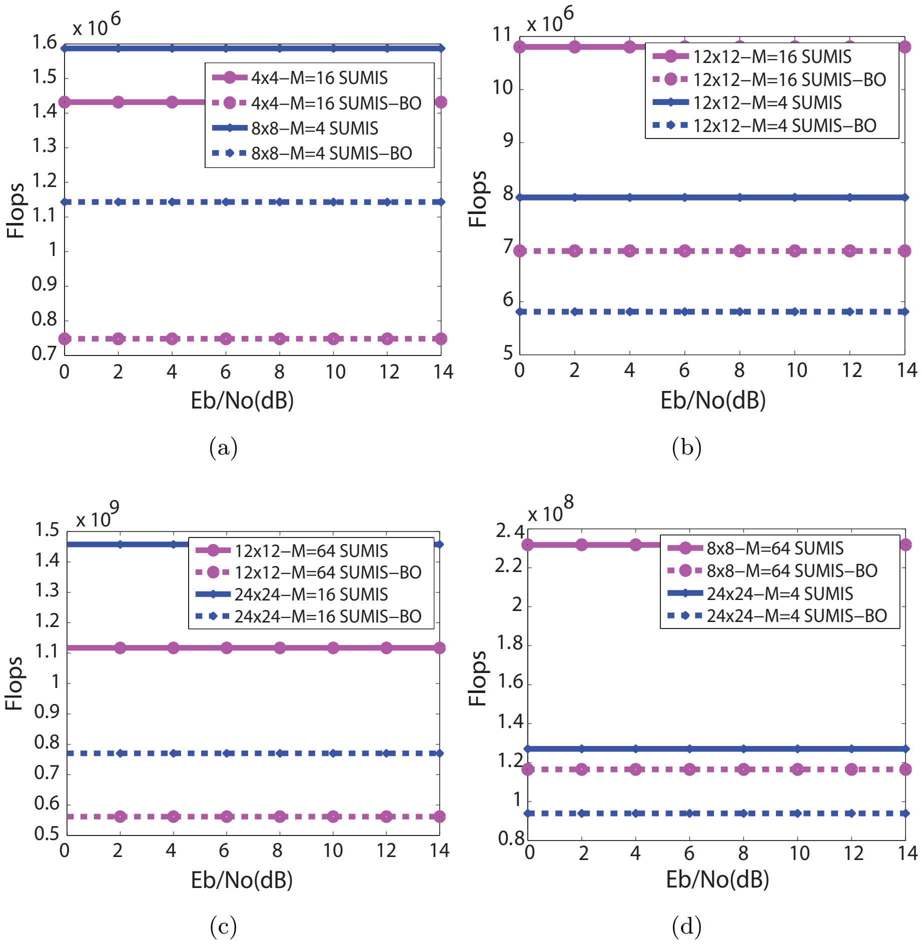


(a) 64−QAM

(b) 16−QAM

(c) 4−QAM

Figure 3.7. FER as function of E_b/N_0 with different system sizes and constellation orders.



(a)

(b)

(c)

(d)

Figure 3.8. Flops as function of E_b/N_0 with different system sizes and constellation orders.

$n_T \backslash M$	4	16	64
8	29.7%	47.7%	49.7%
16	27.9%	47.5%	49.7%
24	27%	47.5%	49.7%
48	26%	47.2%	47.2%

Table 3.3. Percentage of flops improvement of SUMIS-BO with respect to SUMIS.

over the SUMIS algorithm in terms of flops, especially for problems with higher constellation order.

Thus, although the proposed SUMIS-BO algorithm shows a slight performance loss in some of the evaluated cases, it reduces drastically the computational complexity of the original SUMIS. Furthermore, the proposed algorithm retains the fixed complexity of the SUMIS algorithm. The reduction of the computational cost in term of flops is up to 50%, for $64-QAM$ constellation. It can be noted that the reduction becomes greater when the constellation order increases. This is an expected behavior that can be inferred from Table 3.2. The number of terms to be computed in (3.7) is reduced from $M^{n_s} \cdot k \cdot n_T$ to $2 \cdot k \cdot n_T$, i.e., when the constellation order increases the number of terms is drastically reduced, as Table 3.2 shows. However, it is important to note that the total percentage of reduction inferred from Table 3.2 is not represented in 3.3, since the BO algorithm adds some extra computational cost.

3.6 Conclusions

In this chapter we have considered the problem of MIMO detection in a MIMO-BICM system. First, an overview of soft MIMO detection techniques was presented. The performance and complexity comparison between these algorithms have also be given.

Some details of the hard BOHD algorithm have been also presented, this algorithm is much faster than standard hard ML SD detectors, especially for high constellation orders or large number or antennas. A suboptimal soft-output detector has been presented which employs the advantages of the BOHD algorithm and the BO technique. The results show that this algorithm achieves near max-log performance at reduced complexity. Thus the proposed algorithm provides a good trade-off between complexity and performance. Furthermore, it has been calculated the probability of using the BO estimator instead of the ZF. This is, the probability of the ZF estimator is in or out the considered constellation. Results show that for all constellation orders and system sizes evaluated the BO estimator is almost always employed.

In the last part of the chapter, the well-known SUMIS algorithm has been modified to reduce the number of terms to be computed. The results show a very slight performance loss in some of the evaluated cases. On the other hand, the complexity reduction has been verified measuring the number of flops requires by the original SUMIS algorithm and by the proposed scheme. Results show how the computational complexity is drastically reduce, up to 50%, when the proposed algorithm is performed.

Efficient Soft-Input Soft-Output Algorithms

4

Efficient Soft-Input Soft-Output Algorithms 4

This chapter focuses on the improvement of known algorithms for soft-input soft-output (SISO) detection. SISO detection usually has large computational complexity and is needed of optimal and low complexity algorithms. First, taking two well-known optimal SISO detector algorithms as a starting point, a number of modifications are proposed to improve the efficiency of the search. Second, a non-optimal algorithm proposed in the previous chapter for soft-output detection has been extended to perform in an iterative receiver structure. The proposed algorithm has been evaluated and compared in terms of efficiency and computational complexity.

4.1 Soft-Input Soft-Output Detection Problem

Throughout this chapter the structure of an iterative MIMO receiver as such depicted in Fig. 2.4 for an ID-BICM MIMO system is considered. A SISO detector for MIMO systems and a SISO channel decoder are employed in this structure. The iterative receiver is stopped if a maximum number of iterations is reached. The SISO detector estimates soft information in form of LLR values which are iteratively exchanged with the decoder. The LLRs values are computed from the received vector $\mathbf{y}$ and soft-input values in form of LLRs, which are provided by the channel decoder. As Section 2.2.2 in chapter 2 describes, optimum performance in iterative MIMO systems can be achieved by computing exact APPs in form of intrinsic a posteriori LLRs

$$L_{i,b} = \log\left(\frac{p(\mathbf{y}|s_{i,b}=1,\mathbf{H})}{p(\mathbf{y}|s_{i,b}=0,\mathbf{H})}\right). \tag{4.1}$$

It was demonstrated, employing the corresponding PDF and Bayes$'$ theorem the probability in (4.1) can be written as

$$\begin{aligned} L_{i,b} &= \log\left(\sum_{\mathbf{s}\in\chi^1_{i,b}} \exp\left(-\frac{\|\mathbf{y}-\mathbf{H}\mathbf{s}\|^2}{\sigma_n^2}\right) P(\mathbf{s}|s_{i,b})\right) \\ &- \log\left(\sum_{\mathbf{s}\in\chi^0_{i,b}} \exp\left(-\frac{\|\mathbf{y}-\mathbf{H}\mathbf{s}\|^2}{\sigma_n^2}\right) P(\mathbf{s}|s_{i,b})\right). \end{aligned} \tag{4.2}$$

The application of the max-log approximation to (4.2) leads to an equivalent formulation of the intrinsic LLRs as

$$\begin{aligned} L_{i,b} &\approx \min_{\mathbf{s}\in\chi^1_{i,b}} \left(\frac{\|\mathbf{y}-\mathbf{H}\mathbf{s}\|^2}{\sigma_n^2} - \log P(\mathbf{s}|s_{i,b})\right) \\ &- \min_{\mathbf{s}\in\chi^0_{i,b}} \left(\frac{\|\mathbf{y}-\mathbf{H}\mathbf{s}\|^2}{\sigma_n^2} - \log P(\mathbf{s}|s_{i,b})\right). \end{aligned} \tag{4.3}$$

However, iterative detection is based on feeding back extrinsic LLR values, denoted by $L^E_{i,b}\ \forall i,b$, instead of the intrinsic ones computed as in (4.2) and (4.3). The common approach is to compute (4.2) or (4.3) and then subtracting the corresponding a priori LLR value as

$$L^E_{i,b} = L_{i,b} - L^A_{i,b} \qquad \forall i,b. \tag{4.4}$$

Clearly, one of the two minima in (4.3) is the metric (d^{MAP}) given by the MAP transmitted symbol vector, $\mathbf{s}^{MAP}$. The other minimum in (4.3) can be denoted as $\overline{d}_{i,b}$ and has to be calculated for every coded bit. Thus, this minimum is given by the minimum metrics associated to

$$\mathbf{s} \in \mathcal{X}_{i,b}^{(\overline{x_{i,b}^{\mathrm{MAP}}})}, \tag{4.5}$$

where $\overline{x_{i,b}^{\mathrm{MAP}}}$ denotes the complement of bit $x_{i,b}$ in $\mathbf{s}^{\mathrm{MAP}}$. Therefore, the max-log MAP intrinsic LLRs can be computed as

$$L_{i,b}^{I} = \frac{1}{\sigma_n^2}\left(d^{MAP} - \overline{d}_{i,b}\right)\left(1 - 2x_{i,b}^{MAP}\right), \tag{4.6}$$

where the term $(1 - 2s_{i,b}^{MAP})$ adjusts the sign depending on whether d^{MAP} corresponds to the first or the second minimum in (4.3).

SD algorithms can be employed to compute exactly equation (4.3). To this purpose, some considerations have to be considered in the tree search problem.

4.1.1 SISO tree search problem

Using the same procedure as for hard-output tree search described in Section 2.2.1, the ED can be computed through a tree structure. The distance corresponding to a symbol vector $\mathbf{s}$ can be rewritten as

$$d(\mathbf{s}) = \sum_{i=1}^{n_T}\left(\left|z_i - \sum_{j=i}^{n_T} R_{i,j}s_j\right|^2 - \log P\left(s_i\right)\right), \tag{4.7}$$

since the following factorization holds, $P(\mathbf{s}) = \prod_{i=1}^{n_T} P(s_i)$. The expression to compute $\log P(s_i)$ can be derived from (2.42) in Section 2.2.2.

$$\log P(s_i) = -\tilde{B}_i + \sum_{b=1}^{k} s_{i,b}L_{i,b}^{A}, \tag{4.8}$$

with $\tilde{B}_i$ equal to

$$\tilde{B}_i = \sum_{b=1}^{k} \log\left(1 + \exp(L_{i,b}^{A})\right). \tag{4.9}$$

Equation (4.7) can be recursively evaluated by a tree structure (see Section 2.2.1). At each level, the DIs are computed as:

$$|e_i| = \left| z_i - \sum_{j=i}^{n_T} R_{i,j} s_j \right|^2 - \log P\left(s_i\right). \tag{4.10}$$

Note that the DIs in (4.10) need to be non-negative increments to have a proper behavior of the tree search. To this end the prior term $-\log P\left(s_i\right)$ should satisfy to be greater or equal to zero, $-\log P\left(s_i\right) \geq 0$. Nevertheless, equation (4.8) allows negative increments. From (4.9) it follows that the constant term in (4.8) does not depend on $\mathbf{s}$ and cancels out in the computation of the intrinsic LLRs. Therefore, $\tilde{B}_i$ constant could be replaced by [72]

$$B_i = \sum_{b=1}^{k} \frac{1}{2} |L_{i,b}^{A}|, \tag{4.11}$$

guaranteeing non negative increments in (4.10) which is computed as

$$|e_i| = \left| z_i - \sum_{j=i}^{n_T} R_{i,j} s_j \right|^2 + B_i - \sum_{b=1}^{k} s_{i,b} L_{i,b}^{A}. \tag{4.12}$$

Equation (4.12) guarantees non negative increments, furthermore the complexity tree search is reduced since $\tilde{B}_i \geq B_i$ [72], and maintains the max-log MAP LLRs.

Therefore, it is necessary the calculation of the MAP solution and all of the counter-hypothesis distances to compute the max-log MAP LLRs, as this section reports. This implies high computational costs, thus, some strategies like clipping method or non-optimal algorithms can be employed to reduce the computational cost, which leads to a reduction in performance. The RTS and STS algorithms presented in the previous chapter for soft-output detection, are also employed for SISO detection [58][72], following the same strategy than that presented in the previous section.

4.2 Max-log SISO Algorithms

In this section several possibilities for improving the SISO RTS and STS algorithms using the BO technique are investigated. As a result, three alternative methods are proposed: two for the case with clipping and one for the case without clipping.

4.2.1 SISO Box Optimization Repeated Tree Search

A simple and effective proposal for the case without clipping is to perform a straightforward replacement in the RTS algorithm, replacing the standard SESD hard detector by the BOHD algorithm. To this purpose, the hard BOHD algorithm proposed in [48] has to be modified to allow a priori information as input. The large reduction in time and in visited nodes shown in [48] for hard detection is immediately reflected in a large reduction in complexity for the new RTS algorithm, which we will denote as *Box Optimization Repeated Tree Search* (BORTS).

The original BOHD algorithm computes the ML solution, regardless of the prior information provided by the decoder. The a priori information has to be incorporated to compute (4.12) at each evaluated node. Furthermore, to employ (4.12) leads to recompute the pruning criteria evaluated in (3.20). Then, when a priori information is incorporated in the BOHD algorithm, expression $d\left(\mathbf{s}\right) < r^2$ in level l is rewritten as:

$$\begin{gathered} d\left(\mathbf{s}\right) = \|\mathbf{R}\cdot\mathbf{s}-\mathbf{z}\|^2 - \log P(\mathbf{s}) = \\ \|\mathbf{R}_{1:l-1,1:l-1}\cdot\mathbf{s}_{1:l-1} + \mathbf{R}_{1:l-1,l:n_T}\cdot\mathbf{s}_{l:n_T} - \mathbf{z}_{1:l-1}\|^2 - \textstyle\sum_{i=1}^{l-1}\log P(s_i) + \\ \|\mathbf{R}_{l:n_T,l:n_T}\cdot\mathbf{s}_{l:n_T} - \mathbf{z}_{l:n_T}\|^2 - \textstyle\sum_{i=l}^{n_T}\log P(s_i) \le r^2 \ . \end{gathered} \tag{4.13}$$

Therefore, the remaining term in inequality (4.13) becomes

$$\|\mathbf{R}_{1:l-1,1:l-1}\cdot\mathbf{s}_{1:l-1} + \mathbf{R}_{1:l-1,l:m}\cdot\mathbf{s}_{l:m} - \mathbf{z}_{1:l-1}\|^2 - \sum_{i=1}^{l-1}\log P(s_i) \ , \tag{4.14}$$

and a lower bound of this inequality can be obtained. In Section 3.3.2 a lower bound c of $\|\mathbf{R}_{1:l-1,1:l-1}\cdot\mathbf{s}_{1:l-1} + \mathbf{R}_{1:l-1,l:m}\cdot\mathbf{s}_{l:m} - \mathbf{z}_{1:l-1}\|^2$ was proposed. The BO method was employed to this purpose . On the other hand, the lower bound of the contribution caused by the prior term $\sum_{i=1}^{l-1}\log P(s_i)$ can be set to

$$p = \sum_{i=1}^{l-1}\min_{s_i\in\Omega}\left(-logP(s_i)\right) . \tag{4.15}$$

Therefore, (4.13) can be written as:

$$d\left(\mathbf{s}^{(l)}\right) = \|\mathbf{R}_{l:n_T,l:n_T}\cdot\mathbf{s}_{l:n_T} - \mathbf{z}_{l:n_T}\|^2 - \sum_{i=1}^{l-1}\log P(s_i) \le r^2 - c - p \ , \tag{4.16}$$

which is a tighter pruning condition.

The case with clipping is more relevant from a practical point of view, because the complexity of the algorithms without clipping is still to high for practical implementations. The BORTS algorithm described above is easily adapted for the case with clipping, similarly to the RTS algorithm in section 3.2.1. Since the MAP solution is computed previously, when the BOHD reruns to compute each extrinsic metric, the initial maximum radius will be upper bounded by $d^{MAP} + L_{clip}$. Furthermore, it is known that any possible transmitted vector will give us a larger ED than the distance given by $\mathbf{s}^b$, being $\mathbf{s}^b$ the estimated vector computed by BO technique as in (3.14). Therefore, if the following condition is satisfied

$$\left(\|\mathbf{z} - \mathbf{R}\mathbf{s}^b\|^2 + min_{\mathbf{s}\in\Omega^{n_T}} \left(-\log P(\mathbf{s})\right)\right) > \left(d^{MAP} + L_{clip}\right), \qquad (4.17)$$

we can set the corresponding $\overline{d}_{i,b}$ value to $d^{MAP} + L_{clip}$, avoiding the tree search and reducing the complexity.

4.2.2 Double Tree Search

The STS algorithm without clipping does not fit very well with the BO aids, therefore the STS algorithm without clipping cannot be easily combined with the BO technique. The reason is that the BO obtains extremely tight bounds for the radius, while the STS must keep a radius that is large enough to obtain all of the counter-hypothesis distances in a single tree traversal. However, the situation is different when clipping is applied; there are several techniques that can be applied. We have found the following modifications to the STS algorithm to be quite influential.

Precomputation of d^{MAP} and s^{MAP}

This modification is an attempt to take advantage of the BOHD algorithm. Note that when the STS with clipping ends, all of the nodes with a PED within the interval $[d^{MAP}, d^{MAP} + L_{clip}]$ have been visited. The minimum number of nodes to be visited should clearly be the number of nodes with a PED within this interval. However, the STS proceeds like the Schnorr-Euchner detector, i.e., it starts with the initial best distance as $+\infty$ and updates it whenever STS finds a feasible leaf. When the STS finds a leaf with a smaller PED, it updates the best distance. However, these first leaves may have a PED that is larger than d^{MAP}. As long as the STS does not find the ML solution, it must expand nodes with PED that are larger than $d^{\mathrm{MAP}} + L_{clip}$. This means that some (possibly many) extra nodes may have to be expanded.

A simple technique that can be applied to reduce the number of nodes computes firstly $\mathbf{s}^{\mathrm{MAP}}$ and d^{MAP} using BOHD (or any other hard detector). Then the STS is modified so that it does not search $\mathbf{s}^{\mathrm{MAP}}$ since it has already been computed, and after that the maximum value for the counter-hypothesis distances $\bar{d}_{i,b}$ is set to the value $d^{\mathrm{MAP}} + L_{clip}$.

This is quite easy to implement. Since $\mathbf{s}^{\mathrm{MAP}}$ and d^{MAP} have already been computed, the update rules no longer have to consider updates of $\mathbf{s}^{\mathrm{MAP}}$ or d^{MAP}. Therefore the first update rule (Algorithm 1, lines 1-6) is no longer needed and all of the counter-hypothesis distances $\bar{d}_{i,b}$ can be initialized with the value $d^{\mathrm{MAP}} + L_{clip}$. This will avoid the expansion of any node with a PED larger than $d^{\mathrm{MAP}} + L_{clip}$. With this modification, the number of visited nodes should decrease.

Avoiding Radius Recalculations

In our experiments, we have observed that the radius recalculation (Algorithm 2) is quite expensive, especially because it is carried out at every visited node. In terms of computing time, we have found very beneficial to avoid this recalculation. However, if no recalculation is made, the number of visited nodes can be too high. To alleviate this problem, as in the previous proposal, we try to take advantage of the fast BOHD algorithm by computing the hard MAP information in a previous step, then we can use the following as a pruning condition: Given $\mathbf{s}_{l:n_T}$ a partial transmit vector (node) at level l, if $d(\mathbf{s}_{l:n_T}) > d^{\mathrm{MAP}} + L_{clip}$, this node is pruned; otherwise the node is expanded.

The outcome of this modification (compared with the original STS algorithm) should be that the number of visited nodes increases and the average time complexity decreases, because a large number of radius recalculations is avoided.

Since this algorithm uses firstly the BOHD to obtain $\mathbf{s}^{\mathrm{MAP}}$ and d^{MAP} and then carries out a second tree search to obtain the counter-hypothesis distances $\bar{d}_{i,b}$, we will refer to this algorithm (including the two modifications proposed) as the *Double Tree Search* algorithm (DTS).

Actually, these two modifications to the STS algorithm could be applied using any ML hard-output detector for the first search. However, the speed of the BOHD makes the whole method very competitive.

4.2.3 Analysis and Results

In order to evaluate our proposals, we have compared the proposed algorithms with the STS algorithm. The BER performance of two max-log MAP SISO algorithms without clipping is the same. The same occurs when two max-log MAP SISO algorithms with clipping (using the same clipping parameter) are compared. Thereby, since the accuracy of the considered methods is the same, we are comparing the complexity of these algorithms. The computational complexity of MIMO detectors can be evaluated through different metrics: number of nodes expanded, computing time, and number of flops are the metrics that are most commonly used. For most tree search MIMO detectors, the number of expanded nodes would be chosen as the main metric because is independent of the computing platform. However, our experiments show that the algorithms that we are comparing can have large variations in the cost of the expansion of a single node. Therefore, even though the number of expanded nodes is an important factor, it cannot be used alone to evaluate the efficiency of the methods. The number of flops is another metric that is often used, however in this case, it can be somewhat misleading. The reason is that these algorithms perform a large number of comparisons, which in some cases is larger than the number of flops (as will be shown below). Thereby, the number of comparisons have also been recorded. Finally, we have also recorded the computing times. The computing times depend on the computing platform, the implementation, and, in some cases, on the operating system. However, since the final goal is to obtain methods that can be executed faster, the computing times help to identify the actual complexity. We estimated the average number of expanded nodes, flops, comparisons and execution time by means of Monte Carlo simulation. We simulated 4×4 complex MIMO systems with $16-$QAM and $64-$QAM constellations. We also used a rate 1/2 convolutional encoder with generator polynomials $[133_o 171_o]$, constraint length 7, codeword size 4096 and max-log BCJR channel decoder with max-log approximation. The number of iterations of the iterative receiver has been set to 4. The test were carried out using a single core of an Intel Xeon CPU X5680 processor with the Ubuntu operating system. As mentioned above, the BER performance of any MAP algorithm with a given clipping parameter is the same. Therefore, for a given clipping parameter, the algorithms compared will return exactly the same results. Thus, a pure accuracy comparison showing the BER of the proposed algorithms will give exactly the same results and is not of interest. The BER obtained with MAP detectors, 16 and 64-QAM modulations, without clipping, at the first iteration and under the same simulation parameters has been represented in Fig. . Thus, the SNR values represented in Fig. have been chosen for comparing the complexity of the proposed algorithms in this chapter.

The case without clipping

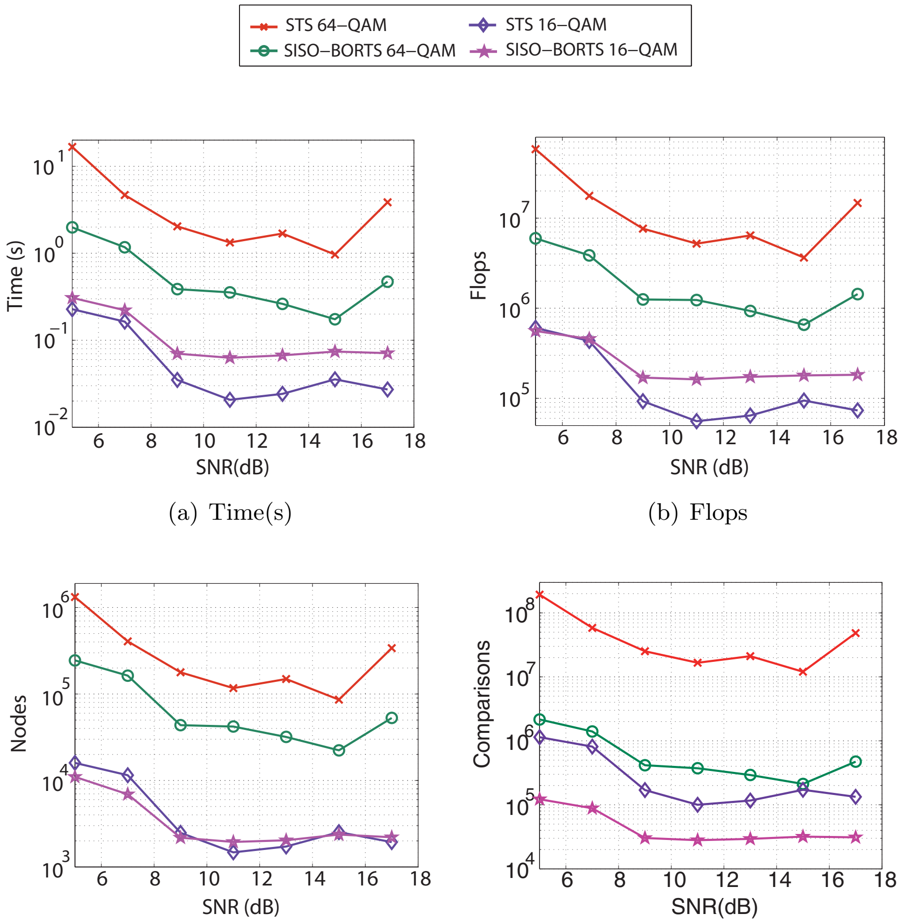
In the case without clipping, we compare the complexity of the BORTS algorithm with the STS algorithm. The numerical results are summarized in Fig. 4.1. The results show that the STS is faster for 16−QAM constellation. However, the BORTS algorithm is almost ten times faster in time, and around five times faster in terms of expanded nodes for 64−QAM. It is also clear that there is a substantial difference in the number of comparisons. In the 64−QAM case the difference is of two orders of magnitude.

The percentage cost reduction between the BORTS and the STS for each evaluated parameter has been represented in Table 4.1, for a 4 × 4 MIMO system employing a $64 - QAM$ constellation without clipping. A higher percentage implies a lower computational cost of the BORTS algorithm with regard to the STS method. Table 4.1 shows the advantage of using SISO BORTS instead of STS method without clipping.

Without clipping

SNR(dB)	5	7	9	11	13	15	17
Time	88.12%	74.87%	80.99%	73.29%	84.41%	81.95%	87.80%
Flops	89.75%	78.14%	83.64%	76.30%	85.46%	82.04%	90.26%
Nodes	81.44%	59.88%	75.56%	64.08%	78.66%	74.02%	84.43%
Comparisons	98.88%	97.59%	98.34%	97.75%	98.61%	98.24%	99.03%

Table 4.1. Percentage reduction of the computational performance parameters between BORTS and STS for a 4×4 MIMO system and a $64 - QAM$ constellation without clipping. A higher percentage reduction indicates a lower computational cost of the BORTS algorithm with regard to the STS method.



(a) Time(s)

(b) Flops

(c) Nodes

(d) Comparisons

Figure 4.1. Average computation performance parameters for SISO detection with the STS and SISO-BORTS methods for a 4 × 4 MIMO system and 16-QAM and 64-QAM without clipping.

The Case with Clipping

In this case, we compare STS (with clipping) with BORTS (with clipping) and DTS. These experiments were repeated with three different clipping parameter values (0.1, 0.2 and 0.4). The results are summarized in different Figures: 4.2, 4.3 and 4.4. In the 16−QAM case, the performance of STS and DTS are similar in terms of computing times and flops , see for example 4.2(a) and 4.2(b). STS carries out more comparisons (due to the radius recalculations in algorithm 2) while it expands fewer nodes. Both methods (STS and DTS) exhibit a better performance when compared with BORTS.

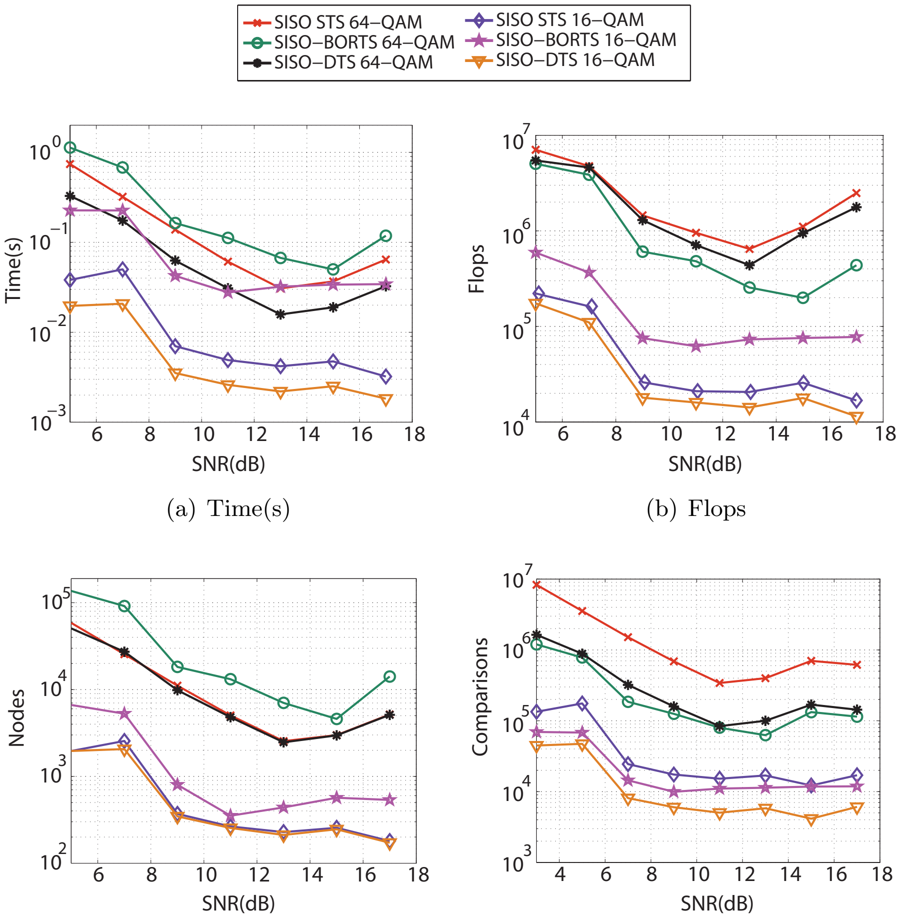
There is a clear change in performance when the order of the modulation changes. In the 64-QAM case, DTS is substantially faster than STS in terms of computing time. However, DTS expands more nodes than STS. Clearly, the computing time per node of DTS is much smaller than for STS. The reason of this behavior has been traced (trough profiling) to the large number of radius recalculations (hence the large number of comparisons) carried out in algorithm 2. This statement is clearly reflected in Table 4.2, which represents the percentage of reduction employing the SISO-DTS method instead of the STS for a $64-QAM$ constellation and $L_{clip}=0.1$.

Clip 0.1

SNR(dB)	5	7	9	11	13	15	17
Time	55.73%	45.56%	54.76%	49.46%	48.75%	48.43%	49.49%
Flops	43.88%	32.77%	44.79%	40.52%	40.32%	38.67%	38.77%
Nodes	13.57%	-6.62%	11.38%	4.52%	2.81%	0.57%	0.91%
Comparisons	80.26%	75.06%	78.79%	76.87%	75.42%	74.92%	75.88%

Table 4.2. Percentage reduction of the computational performance parameters between the SISO-DTS and the STS for a 4×4 MIMO system and a $64-QAM$ constellation with $L_{clip}=0.1$. A higher percentage reduction implies a lower computational cost of the SISO-DTS algorithm with regard to the STS method.

The behavior of BORTS is also worth analyzing. Figures 4.2, 4.3 and 4.4 show that the complexity of BORTS (under any of the metrics considered) has a small or moderate variation when the clipping parameter changes. On the other hand, STS and DTS have comparatively large complexity variations when the clipping parameter varies. Therefore, BORTS becomes comparatively more efficient when the clipping parameter varies. Therefore, BORTS becomes comparatively more efficient when the clipping



(a) Time(s)

(b) Flops

(c) Nodes

(d) Comparisons

Figure 4.2. Average computation performance parameters for SISO detection with the STS and SISO-BORTS methods for a 4 × 4 MIMO system and 16-QAM and 64-QAM with $L_{clip} = 0.1$.

parameter increases. For the largest clipping parameter in the 64−QAM problem, BORTS is faster than STS in all the metrics, see Fig. 4.4, while it is somewhat slower than DTS in computing time. It must be remembered that BORTS can be accelerated further using parallel computing.

Another phenomenon that requires attention is that, in the highest constellation order considered (4×4 64−QAM, $L_{clip} = 0.4$), even though BORTS is slower than DTS, it uses less flops, comparisons and nodes. This phenomenon is due to the algorithmic structure of BORTS. BORTS needs to perform many calls to the subroutine where the BOHD detector is implemented. In turn, the BOHD detector needs to perform calls to other routines (such as the BO subroutine). The algorithmic structure of DTS and STS is quite different from that of BORTS, DTS and STS perform just a few subroutine calls. The extra calls have a significant impact on the computing time of BORTS. In summary, we have verified that the new proposed methods improve their efficiency (compared to STS)

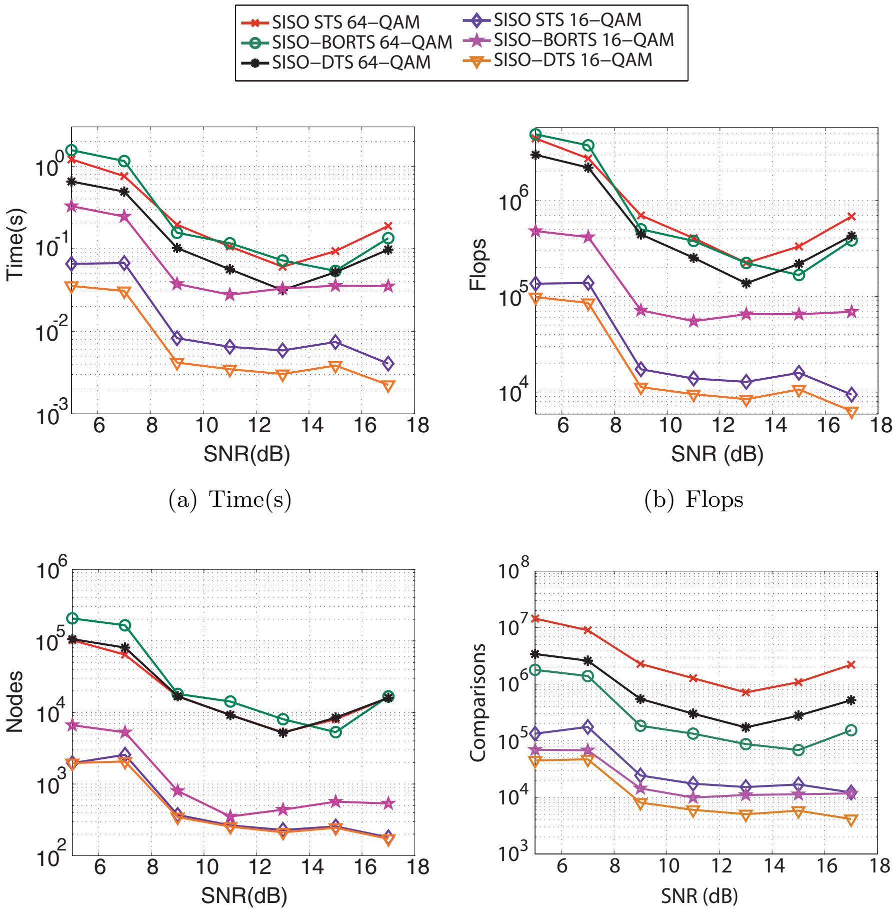
In summary, we have verified that the new proposed methods improve their efficiency (compared to STS) when the size of the problem increases. This is consistent with the increased efficiency of the proposed methods in the 4×4 64−QAM case compared with the 4×4 16−QAM case.

4.3 Non-max-log SISO Algorithms

Equations (4.2) and (4.3) require important computational resources in iterative MIMO systems. In order to reduce their complexity, several algorithms that approximate this equations with low computational complexity have been proposed in the literature [51][53][73].

Minimum Mean square error with Parallel Interference Cancelation

One of the best-known SISO low-complexity approach to approximate (4.1) is referred to as *Minimum Mean square error with Parallel Interference Cancelation* (MMSE-PIC) [73]. In [74] a novel low-complexity variant of the SISO MMSE-PIC was proposed. The main idea underlying MMSE-PIC is to compute estimates of the transmitted symbols based on the a priori LLRs obtained from the SISO channel decoder. These estimates are used to cancel interference in the receiver vector.



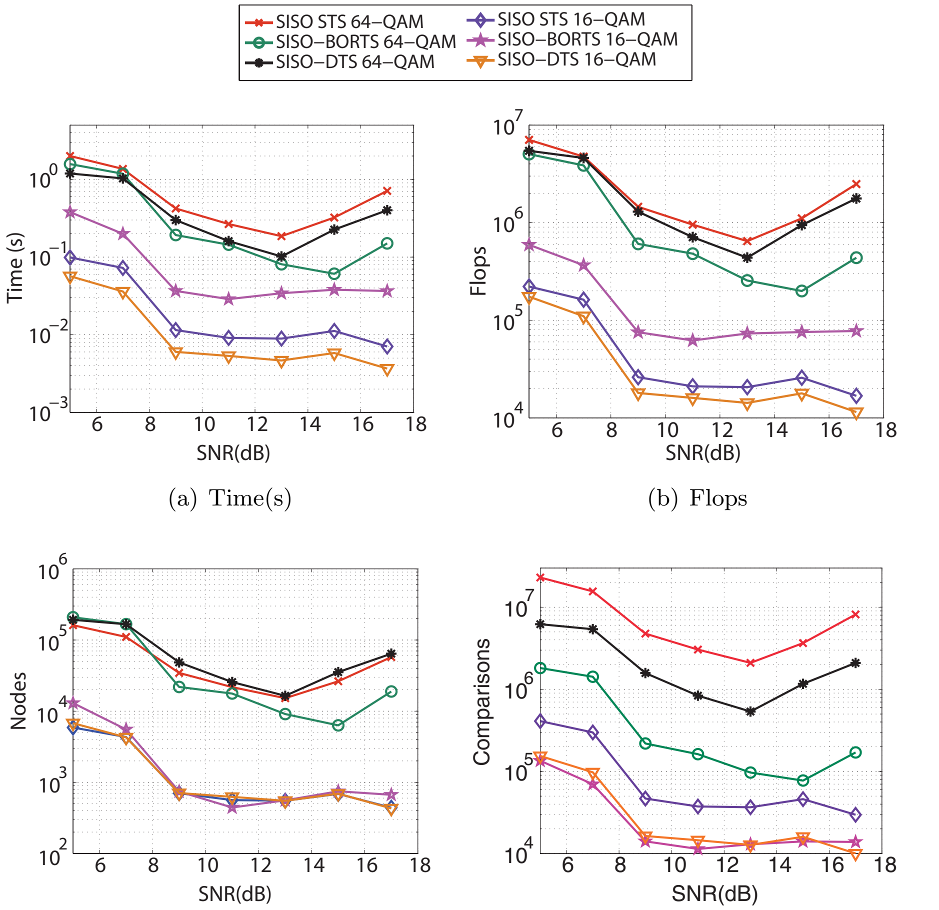
(a) Time(s)

(b) Flops

(c) Nodes

(d) Comparisons

Figure 4.3. Average computation performance parameters for SISO detection with the STS and SISO-BORTS methods for a 4 × 4 MIMO system and 16-QAM and 64-QAM with $L_{clip} = 0.2$.



(a) Time(s)

(b) Flops

(c) Nodes

(d) Comparisons

Figure 4.4. Average computation performance parameters for SISO detection with the STS and SISO-BORTS methods for a 4 × 4 MIMO system and 16-QAM and 64-QAM with $L_{clip} = 0.4$.

In the first step, the soft symbols $\hat{s}_i$ $(i = 1, \cdots, n_T)$are computed according to [73] [75]

$$\hat{s}_i = \mathbb{E}[s_i] = \sum_{c \in \Omega} P(s_i = c) \cdot c. \tag{4.18}$$

The second step cancels the interference in the received vector $\mathbf{y}$ employing the soft-symbols computed in the previous step as

$$\widetilde{\mathbf{y}}_i \triangleq \mathbf{y} - \sum_{j \neq i} \mathbf{h}_j \hat{s}_j = \mathbf{h}_i s_i + \widetilde{\mathbf{n}} \quad \forall i. \tag{4.19}$$

Therefore (4.19) can be considered as the exact LLR computation in n_T single-input multiple-output detection problems. This is the same case as the soft MMSE algorithm in (3.13). The difference with respect to soft MMSE is that now, the elements in $\widetilde{\mathbf{n}}$ have smaller variance than the elements in $\mathbf{n}$ in (3.13).

Figure 4.5 shows that for a given number of iterations the max-log performance outperforms the SISO MMSE PIC performance. However, at low coding rates, for a large number of iterations the performance achieved by the MMSE-PIC algorithm is almost the max-log MAP performance. Nevertheless, at high coding rates the MMSE-PIC behavior worsens and becomes not competitive.

We propose a low-complexity non-optimal SISO detector based on the BOHD-SO algorithm proposed in Section 3.4. The proposed algorithm implements a slight variation with respect to the BOHD-SO algorithm proposed in the previous chapter. The BOHD-SO algorithm runs the hard BOHD algorithm in order to achieve the ML solution. Then, the counter-hypothesis distances are computed using the Algorithm 3. The extension of the soft-output BOHD-SO algorithm proposed in Section 3.4 to iterative detection is easily done. To this purpose it is intuitive to employ the SISO BOHD algorithm described in Section 4.2.1. Once the SISO BOHD achieves the MAP solution, the Algorithm 3 is run, by taking into account the a priori information in the ED. However, it has been proved that this algorithm is not suitable for iterative MIMO detection, i.e. no significant performance improvements can be observed by increasing the number of iterations. This is due to the BO algorithm does not support a priori information and Algorithm 3 achieves the same vector solutions for every iteration.

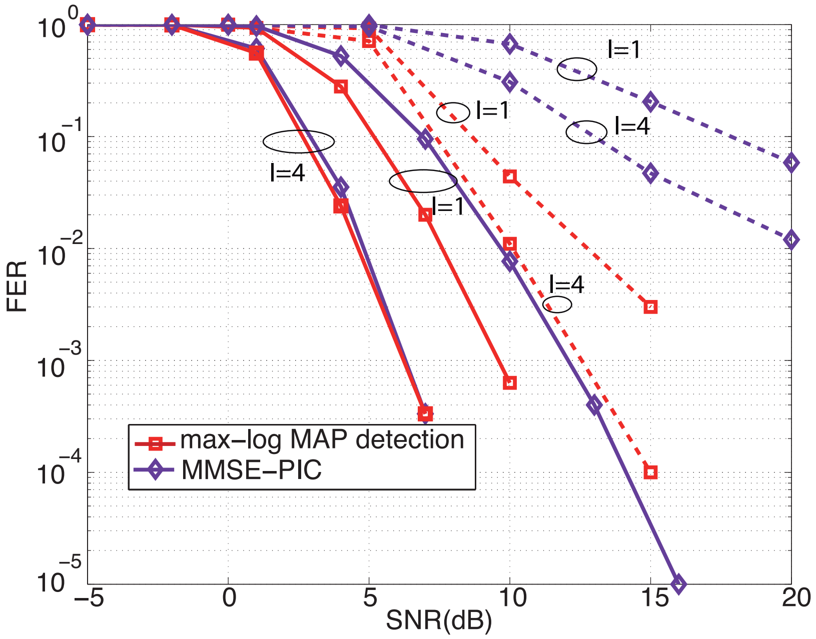


Figure 4.5. FER comparison of MMSE algorithm and max-log ML detection in a 4×4 MIMO systems with $4-$QAM contellation. A convolutional code with different rates ($1/2$ for continuous lines and $5/6$ for dashed lines) has been employed.

However, the results given in Section 3.4 probe that a single execution of the BOHD-SO method is competitive against other non-optimal methods, when no a priori information is provided. Thereby, a variation of the BOHD-SO method is presented in this section, which is employed when the decoder provides a priori information to the detector. Instead of running Algorithm 3, we propose to use Bit-Flipping [17] with SIC to build a candidate list. Then, the candidate list is employed to compute the counter-hypothesis distances. The Bit-Flipping takes the optimal solution provided by the BOHD algorithm as starting point, and at each level it adds $\log 2M$ branches. Each of these new branches has one complemented bit of the hard MAP solution computed by the BOHD algorithm. This method guarantees that all information about the bit values is contained in the constructed candidate list. Afterwards, these new partial paths are

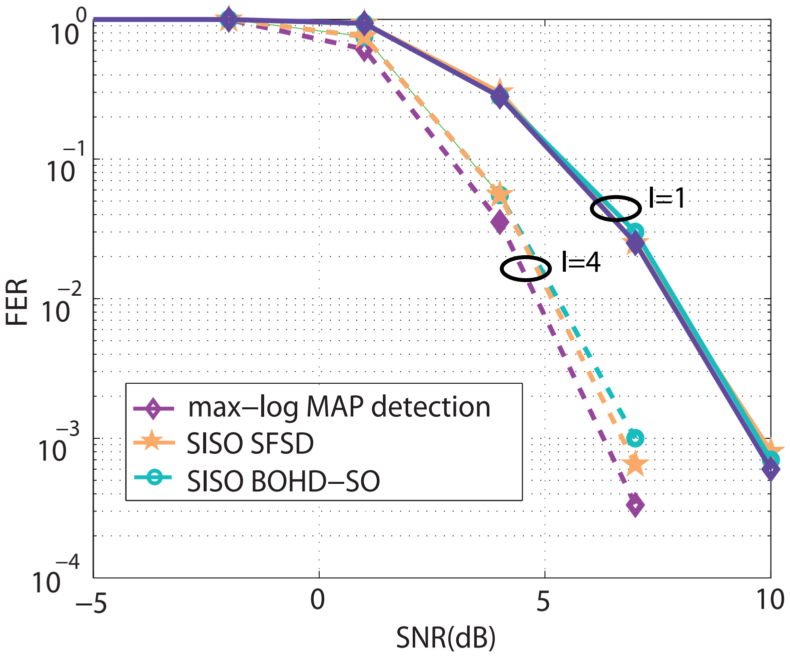


Figure 4.6. FER comparison of different SISO detection algorithms in a 4×4 MIMO systems with $4-$QAM contellation. The first and fourth iteration has been represented in continuous and dashed lines respectively. A convolutional code with rate $1/2$ has been employed.

completed using SIC detection. Then, this operation is repeated at each level, obtaining a candidate list which is employed together with the initial optimal solution to calculate the soft values in (4.6).

The performance of the proposed algorithm has been evaluated for a 4×4 MIMO system and a $4-$QAM constellation. The 2048 information bits are encoded using a convolutional code with generator polynomials $[133_o 171_o]$ and constraint length 7. Figure 4.6 represents the FER performance for a code rate of 1/2 and Figure 4.7 for a code rate of 5/6. The 5/6 code rate is derived by employing puncturing i.e. some of the encoded bits are omitted. The puncturing pattern for each stream of 2048 information bits has been performed according to IEEE 802.11n WLAN standard. The input parameters for the SFSD algorithm are $n_E = 1$ and $N_{iter} = 2$. At the receiver the BCJR algorithm based on the min-sum method with the max-log approximation is employed.

Both figures represent the FER performance for the proposed algorithm together with the max-log MAP performance and the SISO SFSD performance. The max-log MAP performance can be achieved by any of the algorithms described in the previous section. On the other hand, the SISO SFSD algorithm, which was exposed in Section 3.2, has also been chosen for comparison since it provides near max-log performance with less complexity than other algorithms proposed in the literature. It is important to note that the performance of the MMSE-PIC detection has been omitted since it has been represented in Figure 4.5, showing poor results at high coding rates. The computational cost measured in terms of flops for the non-optimal algorithms represented in the Figs. 4.6 and 4.7 has been summarized in Tables 4.3 and 4.4 for 4 and 16 QAM constellations. Results show that the SISO BOHD-SO requires less number of flops to achieve the same performance than the STS algorithm. However, the STS algorithm presents a fixed computational cost which is known at the beginning.

The number of flops of the algorithms that achieves max-log MAP performance has been omitted since the non-optimal algorithms represents huge advances over these algorithms in terms of complexity.

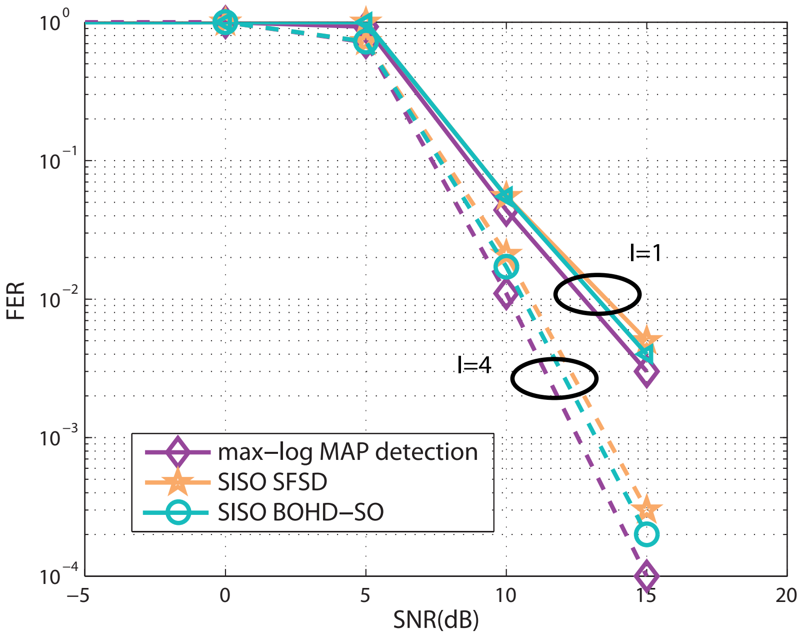


Figure 4.7. FER comparison of different SISO detection algorithms in a 4×4 MIMO systems with $4-$QAM contellation. The first and fourth iteration has been represented in continuous and dashed lines respectively. A convolutional code with different rate $5/6$ has been employed.

SNR	-5	0	5	10
SISO SFSD	$9.68e3$	$9.68e3$	$9.68e3$	$9.68e3$
SISO BOHD-SO	$8.1e3$	$7.78e3$	$7.19e3$	6.23

Table 4.3. Number of flops for non-optimal SISO detection algorithm for a 4×4 MIMO system and $4-$QAM constellation. The input parameters for SFSD have been set to $n_E = 1$ and $N_{iter} = 2$

.

SNR	-5	0	5	10
SISO SFSD	$3.45e4$	$3.45e4$	$3.45e4$	$3.45e4$
SISO BOHD-SO	$3.4e3$	$2.75e4$	$2.39e4$	$1.79e4$

Table 4.4. Number of flops for non-optimal SISO detection algorithm for a 4×4 MIMO system and $16-$QAM constellation. The input parameters for SFSD have been set to $n_E = 1$ and $N_{iter} = 4$

.

4.4 Conclusions

In this section, the SISO MIMO detection problem has been addressed. In this context, two new algorithms for SISO max-log MAP detection have been presented. These algorithms were obtained by combining the RTS and STS algorithms with the hard ML detector BOHD, in different forms. The results are especially good for the 4×4 $64-$ QAM case. In the case with clipping DTS exhibits a smaller computing time, while BORTS expands fewer nodes. Even though there is some uncertainty about the results because the metrics show contradictory trends in some cases, the results still clearly show that the proposed algorithms have an excellent performance with high constellation orders. Furthermore, a non-optimal MIMO detector has been presented. The MMSE-PIC detector almost achieves max-log detection when a certain number of iterations is reached. However, the drawback of this algorithm is that when the code rates are high, the performance becomes poor. The proposed algorithm has been proved to achieve good performance under different code rates. Furthermore, it has been compared with the SISO SFSD algorithm in terms of complexity and it has been illustrated how the proposed algorithm outperforms the complexity of the SFSD algorithm achieving similar performance.

Precoding 5

Precoding 5

This chapter describes several suboptimal precoding techniques to allow multiuser Spatial Multiplexing (SM). The analyzed techniques include ZF, Tomlinson-Harashima precoding (THP) and techniques based on Lattice-Reduction (LR). The performance and the computational cost of the different techniques are shown. Furthermore, several contributions, which involve the use of precoding algorithms, are presented. First, an hybrid precoding based on the MIMO channel matrix condition number is described. The hybrid scheme improves the performance of already proposed algorithms and decreases their expected complexity while keeping an acceptable performance. Second, the combination of Block Diagonalization (BD) with the Lattice-Reduction-Aided Tomlinson-Harashima precoding (LRAP THP) method for Massive MIMO schemes is investigated as an alternative option that outperforms previous proposals.

5.1 Conventional Precoding Schemes

Precoding techniques can mitigate the multiuser interference in MIMO systems [25][76], even when the users are equipped with a single antenna, so the complexity of the receiver can be significantly simplified. Different precoding methods, which differ in performance and computational complexity, have been proposed [77]-[25]. The most employed precoding techniques are briefly described and analyzed throughout this section. It is important to note that the precoding step is carried out assuming perfect knowledge of channel state information (CSIT) at the transmitter. The model employed throughout this section was denoted in Section 2.1.4 by equation (2.14).

5.1.1 Zero-Forcing Precoding

Zero-Forcing (ZF) precoding [24], as the ZF detector, consists in applying the channel inversion, in this case at the transmitter. It can be considered as a particular case of VP where $\mathbf{p} = \mathbf{0}$, so the modulus operation is not required at the receiver. The inversion matrix can be computed as the Moore-Penrose pseudoinverse of the channel matrix, thus the precoded signal can be expressed as:

$$\hat{\mathbf{x}} = \mathbf{H}^{\dagger}\mathbf{s} = \mathbf{H}^{H}(\mathbf{H}\mathbf{H}^{H})^{-1}\mathbf{s}. \tag{5.1}$$

Note that vector $\mathbf{x}$ can be obtained using the QR decomposition of matrix $\mathbf{H}$ to avoid computing $\mathbf{H}^{\dagger}$ explicitly.

5.1.2 Tomlinson-Harashima Precoding

Tomlinson-Harashima precoding (THP) is equivalent to VP (see Section 2.3) where the perturbation vector $\mathbf{p}$ is obtained sequentially and efficiently through feedback filtering and modulus τ operation, as Fig. 5.1 shows. The modulus operator reduces the power of the transmitted signal compared to a linear precoding scheme [25]. In the THP scheme, the channel matrix is decomposed as $\mathbf{H} = \mathbf{L}_0\mathbf{Q}_0$ and the matrices that take part in the precoding algorithm are computed by

$$\begin{aligned} \mathbf{L} &= \mathbf{L}_0\mathbf{G}^{-1} \\ \mathbf{Q} &= \mathbf{G}\mathbf{Q}_0, \end{aligned} \tag{5.2}$$

where $\mathbf{L} \in \mathbb{R}^{K\times K}$ is a lower triangular matrix, $\mathbf{Q} \in \mathbb{R}^{K\times n_T}$ has orthogonal rows and $\mathbf{G}$ is a diagonal matrix containing the diagonal of $\mathbf{L}_0$. Therefore,

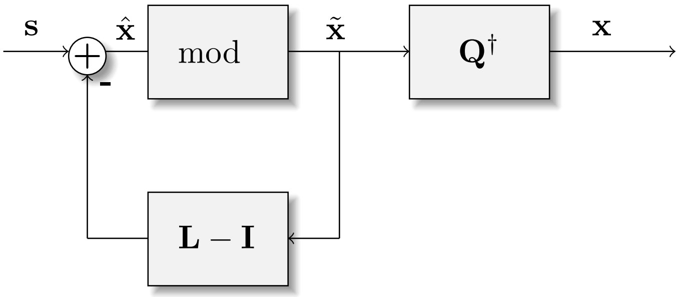


Figure 5.1. Tomlinson-Harashima precoding scheme.

the precoded symbols $\hat{\mathbf{x}}$ can be initially expressed as

$$\hat{x}_j = s_j - \sum_{t=1}^{j-1} l_{j,t}\tilde{x}_t, \tag{5.3}$$

$$\tilde{x}_j = \hat{x}_j \bmod \tau, \tag{5.4}$$

with $j = 1, \ldots, K$. Modulus operation is applied in order to restrict the symbols to the original constellation boundaries. Details about this operation were given in Section 2.3.

Finally, the transmitted signal over the channel is computed as

$$\mathbf{x} = \mathbf{Q}^{\dagger}\tilde{\mathbf{x}} = \mathbf{Q}_0^H \mathbf{G}^{-1}\tilde{\mathbf{x}}. \tag{5.5}$$

5.1.3 Lattice-Reduction-Aided Precoding

The idea of preprocessing the channel matrix to improve the precoding performance has been widely employed [22][78]. Lattice-Reduction-Aided Precoding (LRAP) makes use of Lattice-Reduction (LR) techniques [79] to obtain an efficient approximation of the optimal perturbation given by (2.57). The transformation is performed on a $\mathbf{B}$ matrix, so that other bases can be constructed as $\tilde{\mathbf{B}} = \mathbf{BT}$, where $\mathbf{T}$ is an unimodular transformation matrix with integer elements (i.e. $|\mathbf{T}| = 1$), $\mathbf{B}$ depends on the precoding technique and $\tilde{\mathbf{B}}$ is the lattice-reduced matrix.

Thus, the LR techniques consist of finding another base with better orthogonality properties than the original one. Different reduction techniques have been proposed [69], however as was mentioned in Section 3.4

the LLL [70] method offers fixed complexity and good trade-off between performance and complexity. In [22] two different LRAP precoding techniques were presented; the LRAP Linear and the LRAP VBLAS (LRAP VB). Moreover, [78] proposes a new scheme, the LRAP THP, where THP is applied after performing a LR technique over the channel matrix.

LRAP Linear

In this scheme, the LR stage is applied to the columns of the psudeoinverse matrix $\mathbf{H}^{\dagger}$, giving

$$\mathbf{H}^{\dagger} = \tilde{\mathbf{H}}^{\dagger}\mathbf{T}, \tag{5.6}$$

where $\tilde{\mathbf{H}}^{\dagger}$ is the lattice-reduced channel matrix with better orthogonality properties. Then, the perturbation vector is computed using the rounding off approximation as

$$\mathbf{p} = -\tau\mathbf{T}^{-\mathbf{1}} \left\lceil \frac{\mathbf{Ts}}{\tau} \right\rfloor, \tag{5.7}$$

where $\lceil \cdot \rfloor$ operator maps a real number to the nearest integer of each entry of vector, when a complex value is employed the operator is applied to the real and imaginary parts separately. Finally, as displayed Fig. 5.2 the precoded signal is computed as

$$\mathbf{x} = \mathbf{H}^{\dagger}(\mathbf{s} + \mathbf{p}). \tag{5.8}$$

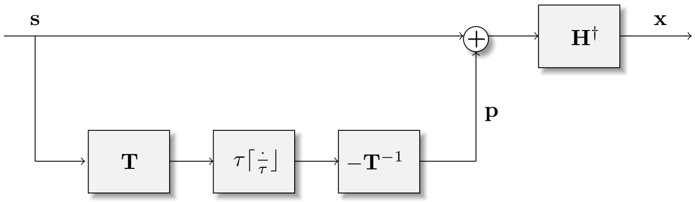


Figure 5.2. Lattice-Reduced-Aided precoding linear scheme.

LRAP VB

The LRAP VB is another variant of this algorithm which is based on SIC, where the VBLAST scheme is applied [22]. In this precoder the $\mathbf{Q}$ and $\mathbf{L}$

matrix can be simply calculated from a QR-type decomposition of $\tilde{\mathbf{H}}^{\dagger}$ such that

$$\mathbf{Q}\tilde{\mathbf{H}}^{\dagger} = \mathbf{L}, \tag{5.9}$$

where $\mathbf{Q}$ contains orthogonal rows and $\mathbf{L}$ is a $K \times K$ lower triangular matrix. Figure 5.3 shows that the first step of the algorithm computes the vector

$$\mathbf{q} = -\mathbf{Q}\mathbf{H}^{\dagger}\mathbf{s}. \tag{5.10}$$

Then, the components of $\tilde{\mathbf{q}}$ are calculated as

$$\tilde{q}_j = \tau \left\lceil \frac{q_j - \sum_{t=1}^{j-1} L_{j,t}\tilde{q}_t}{\tau} \right\rfloor \qquad j = 1, \ldots, K, \tag{5.11}$$

Finally, the perturbation vector $\mathbf{p}$ in Fig. 5.3 can be expressed as

$$\mathbf{p} = \mathbf{T}^{-\mathbf{1}}\tilde{\mathbf{q}}. \tag{5.12}$$

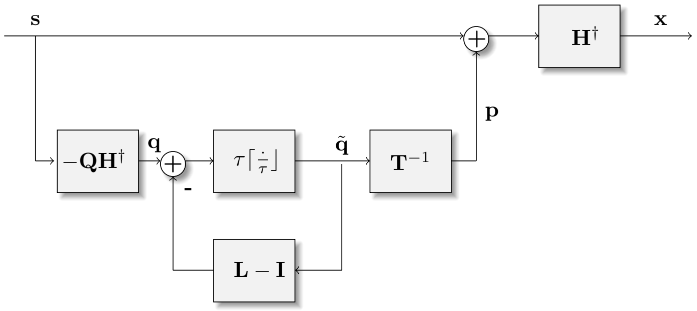


Figure 5.3. Lattice-Reduced-Aided precoding VB scheme.

LRAP THP

The THP strategy can also be performed after an LR transformation over the rows of the channel matrix [78], such that

$$\mathbf{H} = \mathbf{T}\tilde{\mathbf{H}}. \tag{5.13}$$

In this scheme the vector $\mathbf{s}$ is replaced by the new vector $\tilde{\mathbf{s}} = \mathbf{T}^{-1}\mathbf{s}$ (see Fig. 5.4) and the QL decomposition given by (5.2) is performed over the lattice-reduced channel matrix, $\tilde{\mathbf{H}}$

$$\tilde{\mathbf{H}} = \mathbf{L}\mathbf{Q} = \left(\mathbf{L}_0\mathbf{G}^{-1}\right)\left(\mathbf{G}\mathbf{Q}_0\right). \tag{5.14}$$

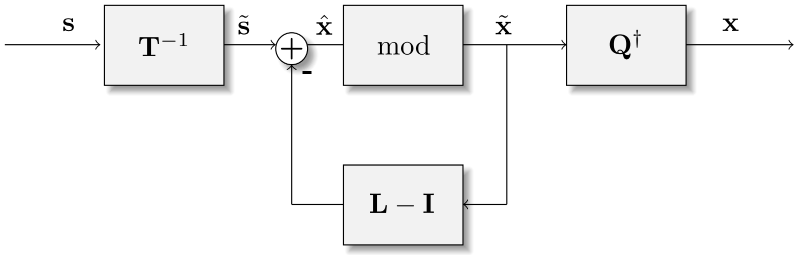


Figure 5.4. Lattice-Reduced-Aided precoding THP scheme.

5.1.4 Performance and Complexity Analisis

In this section a performance and complexity comparison among the conventional precoding algorithms is presented.

The comparison in terms of performance has been carried out in terms of BER within a range of SNR values. A system such as the one presented in (2.14) is considered. A 4-QAM constellation has been used in all the simulations, while different configurations regarding the number of transmit antennas and number of users have been considered: $n_T = K = 4$ and $n_T = K = 8$.

Figure 5.5 shows the BER performance for $n_T = K = 4$. Results show that the THP algorithm outperforms ZF due to the modulus operation. On the other hand the precoding methods which employ LR (LRAP Linear, LRAP VB and LRAP THP) outperform the ZF and THP algorithms. Particularly, the LRAP VB and LRAP THP achieve the best performance in terms of BER. The BER for $n_T = K = 8$ is shown in Fig. 5.6. Results show that the differences between the algorithms have slightly increased: the improvement of THP compared to ZF is more noticeable and the performance of LRAP VB and LRAP THP is better than the LRAP linear.

On the other hand, in practical applications, the computational complexity restrictions can help to select the most suitable algorithm to be used in practice. Thus, it is important to analyze the different precoding schemes from a computational perspective.

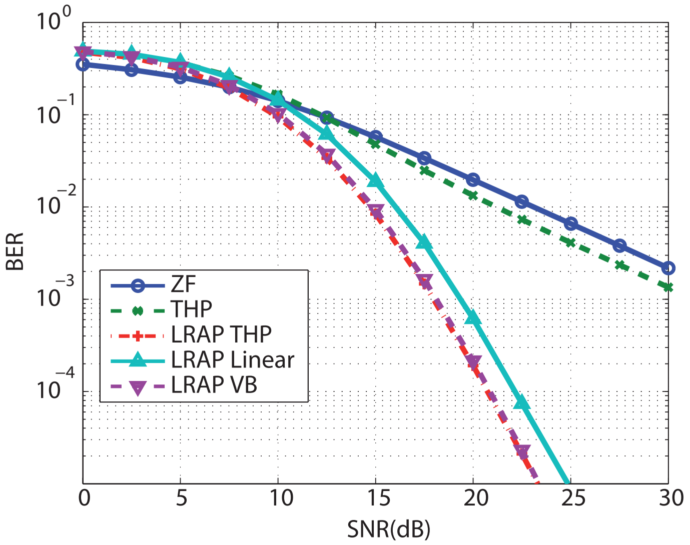


Figure 5.5. BER curves for the conventional precoding techniques for a system with $n_T = 4$ transmitter antennas and $K = 4$ users using 4−QAM constellation.

For the fair comparison of complexity, it is important to consider how often each operation is repeated. For this issue the channel coherence period should be taken into account. There are some operations that are carried out once per coherence period, nevertheless other operations are performed every time the channel is used for transmission. Hence, the average computational cost of the precoding algorithms of a symbol vector can be computed as

$$C_{total} = C_{psv} + C_{pcp}/T_{cp} \tag{5.15}$$

where T_{cp} denotes the number of transmitted symbol vectors per coherence period. C_{psv} and C_{pcp} represent the number of flops performed per symbol vector and per coherence period respectively. In (5.15) the C_{pcp} cost, which we call preprocessing cost, is shared among the T_{cp} transmitted vectors in each coherence period.

The computational cost is measured in term of floating point operations or flops. According to [80][81] the required flops of some operations are:

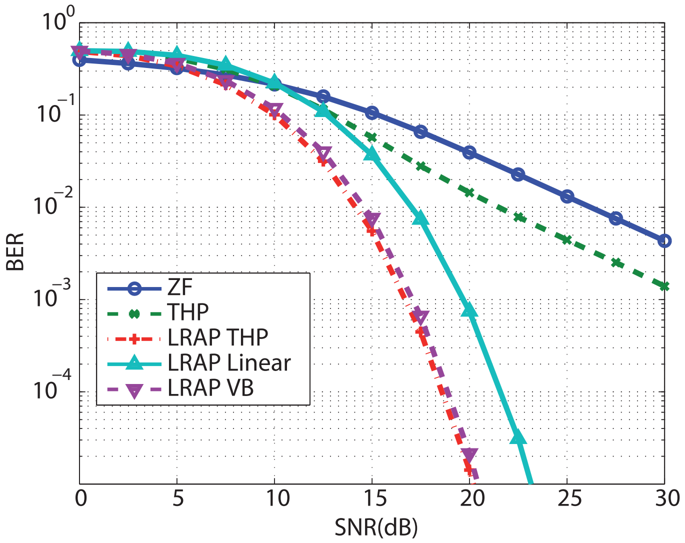


Figure 5.6. BER curves for the conventional precoding techniques for a system with $n_T = 8$ transmitter antennas and $K = 8$ users using $4-$QAM constellation.

- Multiplication of an $m \times n$ complex matrix by an $n \times p$ complex matrix: $8mnp - 2mp$ flops.
- Multiplication of an $m \times n$ complex matrix by its transposed: $3nm^2 - 4nm - m^2 - m$ flops.
- QR decomposition of an $m \times n$ $(m \leq n)$ complex matrix: $16(n^2m - m^2n + (1/3)m^3)$ flops.
- Inversion of an $m \times m$ complex matrix by Gauss-Jordan elimination: $16m^3 - 8m^2 + 2m$ flops.
- LR of an $m \times n$ complex matrix by the fixed complexity LLL algorithm with an L number of LLL loops: $L(44mn - 44n + 28m^2 - 36m + 8)$ flops.

It is important to note that the total number of flops required by the fixed LLL algorithm is difficult to obtain since the LLL conditions (lines 6 and 11 on Algorithm 4) are not always fulfilled. Thus, the number of flops considered is bounded for the worst case.

	QR	LLL	$\left(\mathbf{H}\mathbf{H}^H\right)^{-1}$	$\mathbf{T}^{-1}$
ZF	Yes	No	Yes	No
THP	Yes	No	No	No
LRAP Linear	No	Yes	Yes	Yes
LRAP VB	No	Yes	Yes	Yes
LRAP THP	Yes	Yes	No	Yes

Table 5.1. Main preprocessing stages of precoding algorithms.

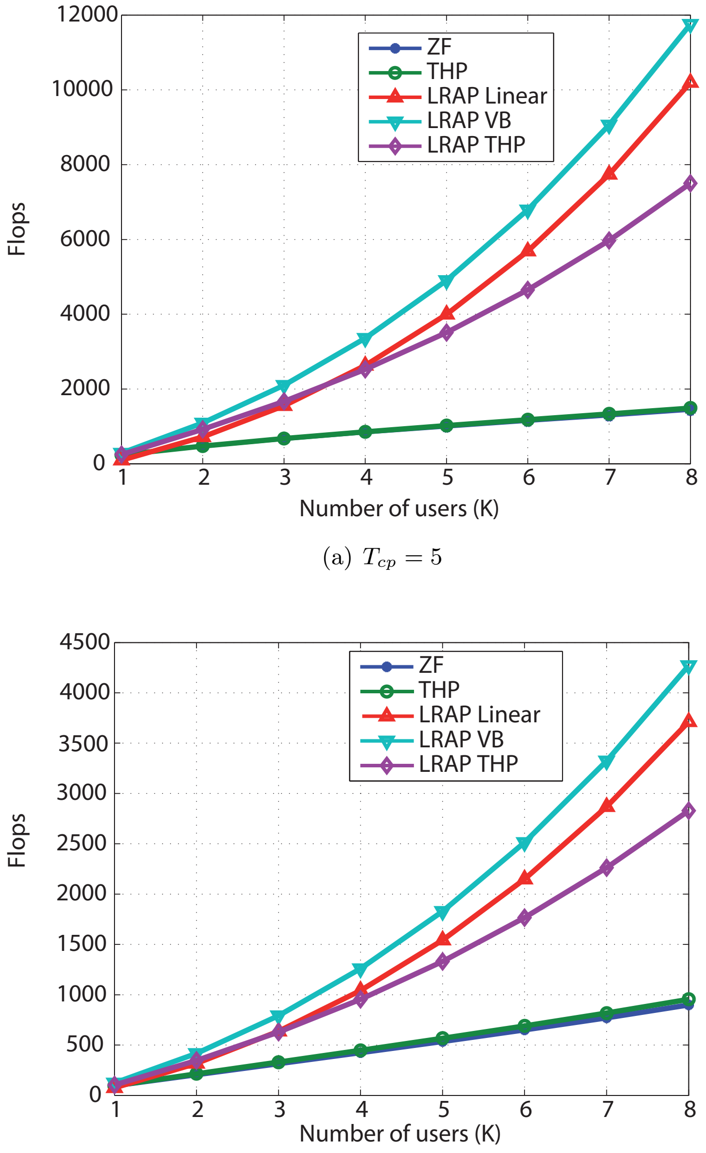
	Per symbol vector	Per coherence Period
ZF	$8Kn_T + 4K^2 - 4K - 2n_T$	$6Kn_T + 6K^2 + 9K$
THP	$8Kn_T + 4K^2 + 4K - 2n_t - 8$	$6Kn_T + 6K^2 + 9K$
LRAP Linear	$8Kn_T + 16K^2 + 4K - 2n_T$	$16K^2n_T - 2Kn_T - 2K^2$
LRAP VB	$8Kn_T + 20K^2 - 2n_T$	$24K^2n_T - 2Kn_T - 4K^2$
LRAP THP	$8Kn_T + 12K^2 + 2K - 2n_T - 8$	$6Kn_T + 6K^2 + 9K$

Table 5.2. Flops of precoding algorithms.

The preprocessing stage is responsible for the calculations related to the channel, not to the signal vector. The typical operations in this stage are shown in Table 5.1 showing which of the precoding algorithms carries out these operations. The number of flops of any other calculations apart from the showed ones in Table 5.1 are included in Table 5.2. This table collects the number of flops per-symbol vector, responsible for calculating the precoded signal vector $\mathbf{x}$ from the original signal vector $\mathbf{s}$, and the number of flops of any other preprocessing calculation apart from the ones discussed in Table 5.1. These operations are carried out once per coherence period.

It is important to note that the THP scheme has been considered for the ZF implementation by suppressing the modulo operation. This is due to the lower computational complexity of the QR decomposition compared to the pseudoinverse.

Taking into account the above results, the total number of arithmetic operations of the precoding algorithms under study has been represented in Fig. 5.7 for $n_T = 8$ and K varying from 1 to 8. It is important to note that the complexity cost depends only on the dimension of the system, not on the modulation order. Figures 5.7(a) and 5.7(b) represent the number of flops for $T_{cp} = 5$ and $T_{cp} = 20$, respectively. The computational cost of the algorithms which perform an LR preprocessing stage is higher that the other algorithms, achieving also the lowest BER. The LRAP Lineal and LRAP VB schemes are the most complex ones. However, the LRAP THP, which exhibits the best performance, requires less computation than the others LRAP algorithms. On the other hand, the ZF and THP algorithms are computationally less expensive and can be employed in case of tighter computational requirements.



(a) $T_{cp} = 5$

(b) $T_{cp} = 20$

Figure 5.7. Total number of flops for the conventional precoding techniques for a system with $n_T = 8$ transmit antennas. The T_{cp} value has been set to 5 and 20.

5.2 Hybrid Precoding scheme

Previous works have shown that the performance of MIMO detectors is highly influenced by the MIMO channel matrix condition number. The same behavior occurs when MIMO precoding is performed. Thus, in this section, a combining scheme in order to reduce the computational cost while maintaining minimal performance loss is proposed. This scheme selects the precoding algorithm applied to the transmitted signal based on the condition number of the channel matrix following a similar strategy to the combined decoder proposed in [82].

The matrix condition number is the ratio of the largest to the smallest singular values. The sensitivity of the solution of a non-singular system of linear equations $\mathbf{Ax} = \mathbf{b}$ with respect to perturbations of the matrix $\mathbf{A}$ is directly related to this parameter [83, 82], which can be computed as $\mathcal{K}(\mathbf{A}) = \|\mathbf{A}\| \left\|\mathbf{A}^{-1}\right\|$. This means that the condition number is a measure of the effect of errors when inverting $\mathbf{A}$. All the precoding methods calculate a pseudoinverse at some point, so the condition number will give us a hint about the conditioning of the matrix and its impact on the BER.

A channel matrix with a low condition number is said to be well-conditioned, which means its inverse could be computed with reasonable accuracy. However, a matrix with a high condition number is said to be ill-conditioned and its inverse is prone to large numerical errors. Furthermore, it has been observed that the condition number generally increases with the size of the channel matrix [84], thus, MIMO systems with a high number of antennas can suffer from a worse precoding performance. Thus, MIMO channel matrix condition number affects precoding performance.

An empirical study of the condition number for the THP and LRAP THP precoders has been realized, Results are shown in Fig. 5.8. The curves represent the BER values obtained for the THP and LRAP THP algorithms when the condition number increases.

Although the LRAP THP method is more complex than the THP algorithm, it has been proved that it works better than THP against a bad conditioned channel matrix due to the LR stage.

The proposed scheme carries out a precoding algorithm that changes depending on the channel condition number. Therefore, when a poor condition number above a selected threshold, is detected, the LRAP THP

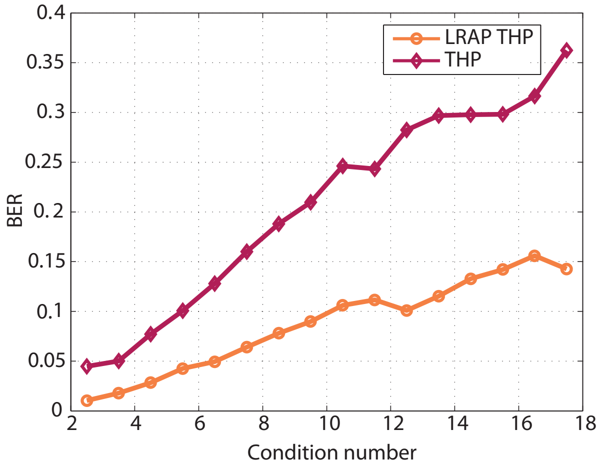


Figure 5.8. BER value as the condition number increases employing THP and LRAP THP methods for a SNR = 5 dB in a 4 × 4 MIMO system and 4−QAM constellation.

method with better performance and more complexity is chosen. When the condition number is lower than a threshold (well-conditioned channel), it switches to a THP method. That means, the LR preprocessing stage is carried out or not depending on the condition number of the channel matrix.

In this context, two different issues have to be solved: a reliable and low cost estimator of the channel condition number and a meaningful way to choose the threshold. For the first issue, a low-complexity method to estimate the condition number proposed in [85] is employed. The condition number ($\mathcal{K}$) of a matrix $\mathbf{A}$ can be equivalently calculated as [83] $\mathcal{K}(\mathbf{A}) = \frac{\sigma_{max}}{\sigma_{min}}$, where σ_{max} and σ_{min} are the maximum and minimum singular values of $\mathbf{A}$.

In our problem, a QR decomposition of the channel matrix is performed (as in Subsection 2.2.1), so the following equality can be used

$$\mathcal{K}(\mathbf{H}) = \mathcal{K}(\mathbf{R}) = \|\mathbf{R}\|\|\mathbf{R}^{-1}\|, \tag{5.16}$$

since $\mathbf{Q}$ is a unitary matrix. $\mathbf{R}$ is a triangular matrix and thereby $\mathcal{K}(\mathbf{R})$ can be computed faster than $\mathcal{K}(\mathbf{H})$.

The condition number estimator proposed in [85] is based on the product of two independent estimators that computes $\|\mathbf{R}\| = \sigma_{max}$ and $\|\mathbf{R}^{-1}\| = 1/\sigma_{min}$. The Power Method (PM) [83] is employed to efficiently compute σ_{max}. Then, a low-complexity method firstly developed in [86] is used to compute $1/\sigma_{min}$.

On the other hand, the selected condition number threshold fixes the complexity of the hybrid precoding algorithm. For a threshold value of $\mathcal{K}_{th}$, the mean complexity of the hybrid precoded can be calculated as

$$C_T = C_{THP} \cdots F_{\mathcal{K}}(\mathcal{K}_{th}) + C_{LTHP} * (1 - F_{\mathcal{K}}(\mathcal{K}_{th})), \tag{5.17}$$

where C_T, C_{THP} and C_{LTHP} represent the complexity for the hybrid, THP and LRAP THP precoders respectively, in this dissertation this complexity is measured in terms of flops. For the Gaussian MIMO channel the cumulative density function (cdf) $F_{\mathcal{K}}(\mathcal{K}_{th})$ on (5.17), can be obtained as [87]

$$F_{\mathcal{K}}(\mathcal{K}_{th}) = \int_1^{\mathcal{K}_{th}} f_{\mathcal{K}}(\mathcal{K}) \cdot d\mathcal{K} \approx \exp\left(-\frac{4n_T^2}{\mathcal{K}_{th}^2}\right). \tag{5.18}$$

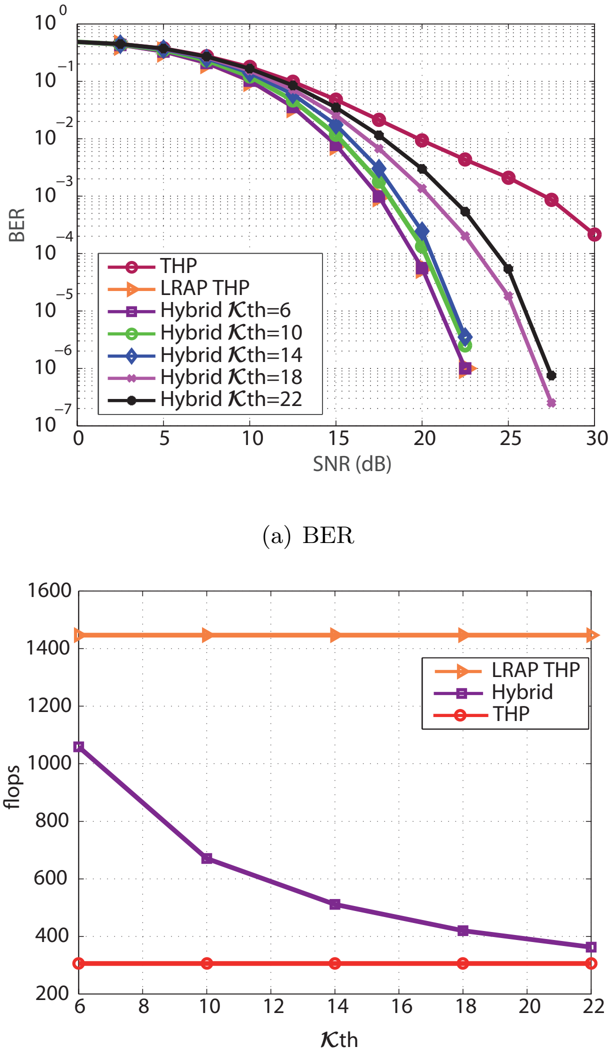
Then, the threshold can be obtained as

$$\mathcal{K}_{th} = 2n_T \log\left(\frac{C_{LTHP} - C_{THP}}{C_T - C_{THP}}\right). \tag{5.19}$$

The selected threshold also determines the BER performance of the hybrid precoding. If a high value of $\mathcal{K}_{th}$ is chosen the complexity of the hybrid precoding will be widely reduced but in contrast the BER performance will be reduced, being similar to the THP performance.

Figure 5.9(a) illustrates the BER performance for the LRAP THP, THP and Hybrid precoding in a system of $n_T = 4$ transmit antennas and $K = 4$ UTs, with a 4−QAM constellation and $T_{cp} = 5$. For the Hybrid precoding different values of $\mathcal{K}th$ have been chosen for comparison. Choosing a high value of $\mathcal{K}th$ the THP will be executed more times than the LRAP THP, consequently the BER performance gets worse. However, if the $\mathcal{K}th$ decreases the LRAP THP precoding is executed more times improving the performance. For a fair comparison, the total number of flops for the different algorithms has been represented in Fig. 5.9(b) showing that the total computational cost of the hybrid precoding depends on the $\mathcal{K}th$ parameter. For example, choosing a $\mathcal{K}th = 10$, the hybrid precoding achieves almost the LRAP THP performance and the computational cost is reduced by more than a half.

Therefore, the hybrid scheme is able to adapt the computational cost and performance on the condition number of the channel matrix in order to reach successfully the desired requirements. Other precoding algorithms can be used within the hybrid scheme that can give different performances.



(a) BER

(b) flops

Figure 5.9. BER performance and number of flops for LRAP THP, THP and Hybrid precoding techniques for a system with $n_T = 4$ transmit antennas and $K = 4$ UTs with a $4-$QAM constellation and $T_{cp} = 5$. The total computational cost of the hybrid precoding scheme depends on the Kth parameter.

5.3 PINV-LRTHP Precoding

Large MU-MIMO system is considered as a potential technique to serve a high number of users simultaneously increasing the achievable data rates. By large we mean a large number of transmit antennas at the BS and a large number of downlink UTs. In large MU-MIMO systems, the complexity of precoding algorithms becomes a bottleneck. In order to reduce the complexity and achieve a good performance a low-complexity precoding scheme based on two steps is proposed in this section. In the proposed scheme the UTs are classified into B blocks of M_I users each group. Then, the total MIMO channel matrix for B blocks is formulated as

$$\mathbf{H} = [\mathbf{H}_1^T \mathbf{H}_2^T \cdots \mathbf{H}_B^T]^T, \tag{5.20}$$

where each $\mathbf{H}_b$ denotes an $M_I \times n_T$ submatrix of $\mathbf{H}$ which includes all elements $h_{i,:}$ of $\mathbf{H}$ with i from $M_I \cdot (b-1) + 1$ to $b \cdot M_I$. Therefore, in the first step, the interference between user′s block is mitigated using a BD method. In the second step, the interference between users belonging to the same block is palliated employing a non-linear complex precoding. In this way, a computationally efficient and good precoding performance which is also versatile and scalable is presented.

The proposed *Partial Inverse with Lattice Reduction aided Tomlinson-Harashima* (PINV-LRTHP) precoding computes the $\mathbf{x}$ transmitted signal vector as

$$\mathbf{x} = \mathbf{W_1 a}. \tag{5.21}$$

where $\mathbf{W_1}$ is the precoding matrix computed in the first step and $\mathbf{a}$ is the precoded output vector of the second step.

5.3.1 First Stage: Block Diagonalization

The first step divides users into B blocks of M_I users. the users are divided randomly selected. It was proved by simulation that under the considered parameters (Gaussian channel and time of coherence values) the performance is equivalent to cluster the most correlated users in the same block. Considering another type of channel and different times of coherence, the performance could be improved by clustering the most correlated users in the same block. The $\mathbf{W_1}$ precoding matrix mitigates the interference between the selected blocks as follows:

$$\begin{cases} [\mathbf{H}\mathbf{W}_{\mathbf{1}}]_{ij} \neq 0, & \text{if} \quad i \text{ and } j \text{ belong to the same block,} \\ [\mathbf{H}\mathbf{W}_{\mathbf{1}}]_{ij} = 0, & \text{if} \quad i \text{ and } j \text{ belong to different blocks.} \end{cases} \tag{5.22}$$

BD method is selected for this purpose. Thus a BD method that computes (5.22) efficiently is needed.

BD-SVD

The BD-SVD [88] precoding technique is a well known linear strategy for MU-MIMO sysems. With the use of BD-SVD the MU-MIMO downlink channel can be decomposed into multiple parallel single user MIMO (SU-MIMO) channels [89], since no interference between the users are allowed after the BD application. The precoding matrix is defined as

$$\mathbf{W}_1 = [\tilde{\mathbf{W}}_1, \tilde{\mathbf{W}}_2, \cdots \tilde{\mathbf{W}}_B], \tag{5.23}$$

where $\tilde{\mathbf{W}}_b \in \mathbb{C}^{n_T \times M_I}$ is the b-th user′s block precoding matrix which lies into the null space of the other user's block channel matrices. Without losing generality, excluding the b-th user′s block channel matrix, $\tilde{\mathbf{H}}_b$ is defined as

$$\tilde{\mathbf{H}}_b = [\mathbf{H}_1^T \cdots \mathbf{H}_{b-1}^T \mathbf{H}_{b+1}^T \cdots \mathbf{H}_B^T]^T, \tag{5.24}$$

where $\tilde{\mathbf{H}}_b \in \mathbb{C}^{\tilde{n}_B \times n_T}$ with $\tilde{n}_B = K - M_I$ Then, to eliminate all MUI completely, the following constraint has to be satisfied

$$\tilde{\mathbf{H}}_b \tilde{\mathbf{W}}_b = \mathbf{0}, \quad b = 1, \cdots, B. \tag{5.25}$$

From the Singular Value Decomposition (SVD) of $\tilde{\mathbf{H}}_b$ we obtain

$$\tilde{\mathbf{H}}_b = \tilde{\mathbf{U}} \tilde{\mathbf{\Sigma}} \left[\tilde{\mathbf{V}}_b^{(1)} \tilde{\mathbf{V}}_b^{(0)} \right]^H, \tag{5.26}$$

where $\tilde{\mathbf{U}}$ and $\tilde{\mathbf{\Sigma}}$ denote the left singular vector matrix and the matrix of ordered singular values of $\tilde{\mathbf{H}}_b$, respectively. Matrices $\tilde{\mathbf{V}}_b^{(1)}$ and $\tilde{\mathbf{V}}_b^{(0)}$ denote the right singular vector matrices where each one consists of the singular vectors corresponding to non-zero singular values and zero singular values respectively. Thus, $\mathbf{V}_b^{(0)}$ is an orthogonal basis for the null space of $\tilde{\mathbf{H}}_b$. Therefore, choosing $\tilde{\mathbf{W}}_b = \tilde{\mathbf{V}}_b^{(0)}$ the condition (5.25) is satisfied. In order to maximize the achievable sum rate of the BD, the water filling algorithm can be additionally incorporated [88]. BD precoding achieves good performance but results in high computational complexity overheads. Thus, other alternatives to implement BD have been proposed.

BD-QR precoding algorithm

This approach employed the QR decomposition instead of the SVD decomposition in order to reduce the computational complexity. To find an orthonormal basis for the null space of $\tilde{\mathbf{H}}_b$ the QR decomposition is performed

$$\tilde{\mathbf{H}}_b = [\mathbf{Q}_b^0 \mathbf{Q}_b^1] \begin{bmatrix} \mathbf{R}_b \\ \mathbf{0} \end{bmatrix} = \mathbf{Q}_b^0 \mathbf{R}_b, \tag{5.27}$$

where $\mathbf{Q}_b^1$ is an $\tilde{n}_B \times \tilde{n}_B - n_T$ unitary matrix, $\mathbf{R}_b$ is an $n_T \times n_T$ upper triangular matrix and $\mathbf{Q}_b^0$ and $\tilde{n}_B \times n_T$ matrix. Thus, we can obtain

$$\mathbf{Q}_b^H \tilde{\mathbf{H}}_\mathbf{b} = \begin{bmatrix} \mathbf{Q}_b^0 \\ \mathbf{Q}_b^1 \end{bmatrix}^H \tilde{\mathbf{H}}_\mathbf{b} = \begin{bmatrix} \mathbf{R} \\ \mathbf{0} \end{bmatrix}. \tag{5.28}$$

Thus it follows $(\mathbf{Q}_b^1)^H \tilde{\mathbf{H}}_\mathbf{b} = \mathbf{0}$, and $(\mathbf{Q}_b^1)^H$ can be referred to as the null space of $\tilde{\mathbf{H}}_\mathbf{b}$. Then, the precoded matrix $\tilde{\mathbf{W}}_b$ can be obtained as

$$\tilde{\mathbf{W}}_b = (\mathbf{Q}_b^1)^H. \tag{5.29}$$

As in the BD-SVD precoding water filling algorithm can be applied to achieve the maximum precoding gain for each user.

BD-PINV precoding algorithm

Alternatively to the BD-QR and BD-SVD methods, the pseudo inverse BD (BD-PINV) method has been proposed in [90]. The BD-PINV method implements the BD idea with significant computational reduction, thus it can be considered as a suitable practical option. Thus, the $\mathbf{W}_1$ matrix can be written as

$$\mathbf{W}_1 = \mathbf{H}^H (\mathbf{H}\mathbf{H}^H)^{-1} \mathbf{B} \tag{5.30}$$

where

$$\mathbf{B} = \begin{pmatrix} \mathbf{B}_1 & 0 & \cdots & 0 \\ 0 & \mathbf{B}_2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \mathbf{B}_B \end{pmatrix}, \tag{5.31}$$

each $\mathbf{B}_b$ is a $M_I \times M_I$ corresponding to each user′s block b and is computed as

$$\mathbf{B}_b = \mathbf{U}_b \boldsymbol{\Sigma}_b^{-1/2}, \tag{5.32}$$

where $\mathbf{U}_b$ and $\boldsymbol{\Sigma}_b$ are computed by eigenvalue decomposition as

$$[(\mathbf{H}\mathbf{H}^H)^{-1}]_b = \mathbf{U}_b \boldsymbol{\Sigma}_b \mathbf{U}_b^*. \tag{5.33}$$

It is important to note that $[(\mathbf{H}\mathbf{H}^H)^{-1}]_b$ denotes an $M_I \times M_I$ submatrix which includes all elements $h_{i,j}$ of $(\mathbf{H}\mathbf{H}^H)$ with i and j from $M_I \cdot (b-1)+1$ to $b \cdot M_I$. A detailed description of the BD-PINV method can be found in [90].

Three algorithms to implement the BD method in stage I have been reviewed. The BD-SVD method implements B SVD decompositions of $\tilde{n}_B \times n_T$ size matrix. A QR decomposition is performed for each B user′s block in the BD-QR algorithm. Finally, the BD-PINV method carries out B SVD decompositions of $M_I \times M_I$ size matrices, in addition to an inverse and some matrix multiplications. An SVD decomposition of an $m \times n$ complex matrix by only obtaining U and Σ or V needs $32(nm^2 + 2m^3)$ flops [80]. The number of flops for the rest of operations was detailed on Section 5.1.4.

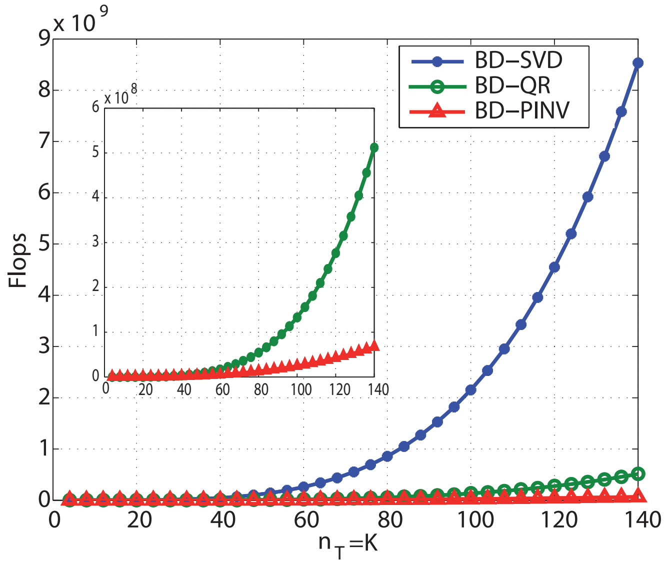


Figure 5.10. Required flops versus the number of n_T, with $n_T = K$ and a user′s block size of 4 ($M_I = 4$).

Figures 5.10 and 5.11 represent the total number of flops required by the BD alternative methods. Figure 5.10 consider the cases that $n_T = K$, fixing M_I to 4 and expressing the computation cost as function of n_T. Figure 5.11 illustrates the computation cost as function of B and fixing $n_T = K = 64$. Figures show that that BD-PINV method obtains a clear advantage in both comparisons, however larger B or n_T makes more significant the advantage.

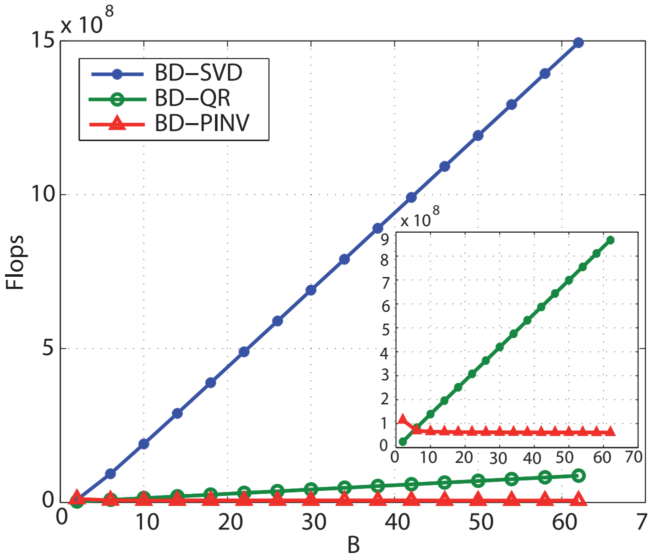


Figure 5.11. Required flops versus the number of the Blocks B for $n_T = K = 64$.

Thus BD-PINV is employed to computed $\mathbf{W}_1$ in the first stage. Therefore, using (5.30) the interference between blocks are mitigated, however the interference between the UTs belonging to the same block remains. After the first stage, these interferences are given by the $\mathbf{B}_b$ matrices, as Figure 5.12 represents. These intra block interferences have to be palliated within the second stage.

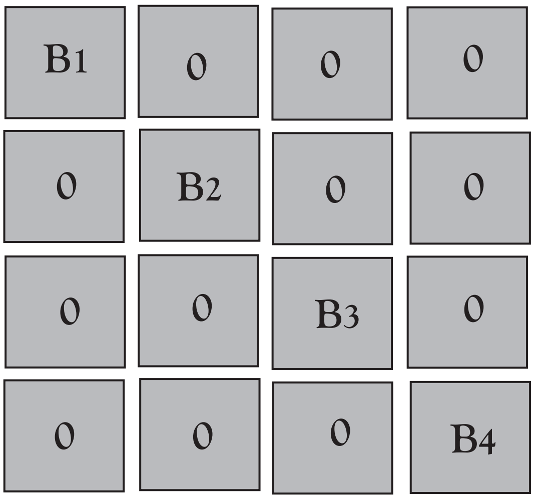


Figure 5.12. Effective channel after applying the first stage of the PINV-LRTHP algorithm.

5.3.2 Second Stage: LRAP THP

To reduce the intra block interference, the LRAP THP method is applied to each block in the second stage. The $\mathbf{s}$ data symbol vector is decomposed in B blocks as $\mathbf{s} = [\mathbf{s}_1^T, \mathbf{s}_2^T, \cdots, \mathbf{s}_B^T]^T$. Therefore each $\mathbf{s}_b$ data symbol vector corresponding to each $\mathbf{B}_b$ block is precoded by the LRAP THP method, giving the $\mathbf{a}_b$ data vector. Thus, at the end of the second stage, the precoded vector can be denoted as

$$\mathbf{a} = [\mathbf{a}_1^T, \mathbf{a}_2^T, \cdots, \mathbf{a}_B^T]^T. \tag{5.34}$$

Therefore in the second stage, the LRAP THP method has to be executed B times, one for each $\mathbf{s}_b$ and $\mathbf{B}_b$. The LLL algorithm [70] is applied to carry out the LR to each $\mathbf{B}_b$. Therefore the transformation is performed on the $\mathbf{B}_b$ matrix finding another base with better orthogonality properties. So that $\mathbf{B}_b = \tilde{\mathbf{B}}_b\mathbf{T}$, where $\mathbf{T}$ is the unimodular transformation matrix with integer elements. Therefore, the $\mathbf{s}_b$ data vector is replaced by the new vector $\tilde{\mathbf{s}}_\mathbf{b} = \mathbf{T}^{-1}\mathbf{s}_\mathbf{b}$ and the $\tilde{\mathbf{B}}_b$ matrix is decomposed as $\tilde{\mathbf{B}}_b = \mathbf{L}_0\mathbf{Q}_0$. Then, the matrices that take part in the precoding process are obtained as

$$\begin{aligned} \mathbf{L} &= \mathbf{L}_0\mathbf{G}^{-1} \\ \mathbf{Q} &= \mathbf{G}\mathbf{Q}_0, \end{aligned} \tag{5.35}$$

where $\mathbf{L} \in \mathbb{C}^{M_I \times M_I}$ is a lower unitary triangular matrix, $\mathbf{Q} \in {}^{M_I \times M_I}$ has orthogonal rows and $\mathbf{G}$ is a diagonal matrix containing the diagonal of $\mathbf{L}_0$. The precoded symbols can be initially expressed as

$$\hat{a}_m = \tilde{s}_m - \sum_{t=1}^{m-1} l_{m,t} \tilde{a}_t, \tag{5.36}$$

$$\tilde{a}_m = \hat{a}_m \bmod M, \tag{5.37}$$

with $m = 1, \ldots, M_I$. Modulus operation is applied in order to restrict the symbols to the original constellation boundaries, being M the constellation size. Finally, the data symbol vector at each user's block will be computed as $\mathbf{a}_b = \mathbf{Q}^{\dagger} \tilde{\mathbf{a}}_b = \mathbf{Q}_0^H \mathbf{G}^{-1} \tilde{\mathbf{a}}_b$.

After the second stage the precoded signal matrix which will be transmitted through the channel will be computed as in (5.21). The pseudocode of the proposed algorithm called as PINV-LRTHP is represented in Algorithm 6.

Algorithm 6 PINV-LRTHP Algorithm

```
Input: $\mathbf{s}$, $\mathbf{H}$, B
Output: $\mathbf{x}$
$\mathbf{B} = \mathbf{0}_{K \times K}$
First Stage:
$\mathbf{L} = (\mathbf{H}\mathbf{H}^H)^{-1}$
for $b = 1, \cdots B$ do
    $\mathbf{L}_b = \mathbf{U}_b \mathbf{\Sigma}_b \mathbf{U}_b^*$
    $\mathbf{B}_b = \mathbf{U}_b \mathbf{\Sigma}_b^{-1/2}$
end for
$\mathbf{W}_1 = \mathbf{H}^H \mathbf{L} \mathbf{B}$
Second Stage:
for $b = 1, \cdots B$ do
    $\mathbf{a}_b = \mathbf{LRAP\ THP}(\mathbf{B}_b, \mathbf{s}_b)$
end for
$\mathbf{x} = \mathbf{W}_1 \mathbf{a}$
```

Complexity and performance analysis

In this subsection, the computational complexity of the PINV-LRTHP is analyzed and compared with the ZF and LRAP THP methods. The ZF has been chosen for comparison since it is the simplest precoding algorithm. On the other hand the LRAP THP method has been chosen because it achieves the best performance and less computational cost among the LRAP methods. The computational cost is measured in terms of flops, similarly by to the case of the conventional precoding algorithms.

As was described in Section 5.1.4 it is important to consider how often each operation is repeated. The computational complexity of the ZF and LRAP THP methods was discussed in Section 5.2. The PINV-LRTHP scheme requires to compute (5.30), (5.32) and (5.33) for each coherence period and (5.21) per symbol vector. Furthermore it carries out the LRAP THP scheme B times, one for each B_b block. However, it is important to note the dimensions of the matrix and vectors that take part each B_b execution of the LRAP THP method. These dimensions are M_I instead of n_T and K.

	ZF	LRTHP	PINV-LRTHP
$\left(\mathbf{H}\mathbf{H}^H\right)^{-1}$	No	No	Yes
QR	Yes	Yes	Yes
LLL	No	Yes	Yes
$\mathbf{T}^{-1}$	No	Yes	Yes
SVD	No	No	Yes

Table 5.3. Main operations of precoding algorithms per channel coherence period.

Table 5.3 summarizes the main operations that the precoding algorithms perform when the channel changes: QR and SVD decompositions, LLL lattice reduction and matrix inversion (either $\left(\mathbf{H}\mathbf{H}^H\right)^{-1}$ or $\mathbf{T}^{-1}$). The number of flops per symbol vector and per coherence period apart from the ones considered in Table 5.3 are represented in Tables 5.4 and 5.5.

The performance in terms of uncoded BER and the computational cost of the precoding algorithms have been simulated and compared for a large MU-MIMO system. In the simulations, two different values of $n_T = K$ has been selected, 64 and 128. For each system size, three different block

	Per symbol vector
ZF	$8Kn_T + 4K^2 - 4K - 2n_T$
LRAP THP	$8Kn_T + 12K^2 + 2K - 2n_T - 8$
PINV-LRTHP	$8Kn_T - 2n_T + 20M_I^2B - 8B$

Table 5.4. Additional flops of precoding algorithms per symbol vector.

	Per symbol vector
ZF	$6Kn_T + 6K^2 + 9K$
LRAP THP	$6Kn_T + 6K^2 + 9K$
PINV-LRTHP	$8K^2n_T - 8n_TKM_I - K^2 - 2n_TK + M_I + 2M_I^2 + 12M_I^2B + 9M_IB$

Table 5.5. Additional flops of precoding algorithms per coherence period.

sizes ($B = \{4, 8, 16\}$) have been evaluated for the PINV-LRTHP algorithm. BER curves have been obtained by Monte Carlo simulations with 1000 independent realizations and 1000 vector transmissions per realization. Inspired from previous works [36][91], the T_{cp} parameter varies from 400 to 900 in the flops curves. This parameter depends on the UTs mobility and propagation environment, e.g. a coherence time of 2 ms and a coherence bandwidth of 200 kHz give a T_{cp} of 400 coherence block.

Figure 5.13 represents the BER performance of the proposed precoder and the reference precoders for a 128×128 MU-MIMO system. The curves show that the LRAP THP method along with the LRAP VB provide the best response in terms of performance. The PINV-LRTHP algorithm has been evaluated for a number of blocks equal to $16, 8$ and 4. When a number of blocks equal to 16 is chosen, the PINV-LRTHP obtains a better efficiency than the ZF or the LRAP Linear methods, however the performance in terms of BER is far from the performance given by the LRAP THP method.

For this configuration, Fig. 5.14 represents the computational cost in terms of flops for the ZF, LRAP THP and PINV-LRTHP methods for a number of blocks equal to 16. Figure 5.14 illustrates that the proposed algorithm significantly reduces the complexity of the LRAP THP method. Besides, it exhibits similar complexity cost than the ZF algorithm. Therefore, we get a much better performance than the ZF with a comparable

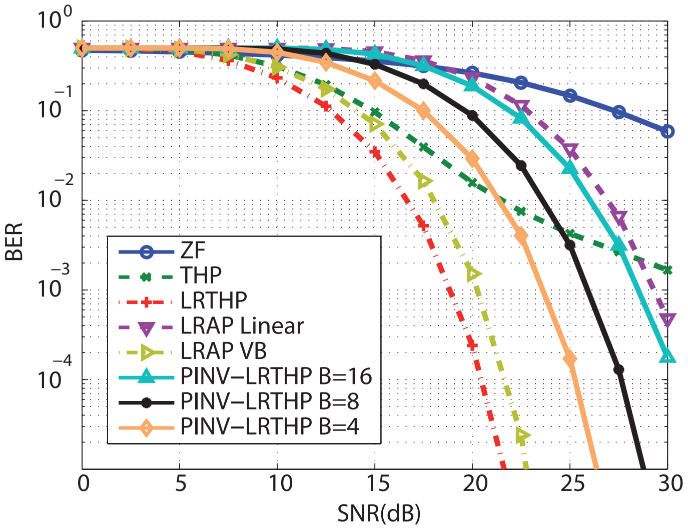


Figure 5.13. Performance of the precoding algorithms for a system with $n_T = 128$ transmitter antennas and $K = 128$ UTs using $4 - QAM$ modulation.

computational cost.

Figure 5.15, represents the saving complexity in flops of the PINV-LRTHP precoding algorithm with respect to the LRAP THP method. This saving complexity has been computed for different size configuration and different number of blocks for the PINV-LRTHP. The saving complexity represents the percentage reduction of the flops between the PINV-LRTHP and the LRAP THP method. A higher percentage reduction (higher saving) implies a lower computational cost of the PINV-LRTHP algorithm with regard to the LRAP THP method.

It is important to remark that the saving on complexity is related to the coherence period. As it can be derived from the Figure 5.15, for all system sizes and B values, longer coherence time implies higher reduction in computational cost. For this reason we can conclude that the proposed method shows a greater cost reduction in slow fading systems.

Figure 5.13 illustrates how the BER performance of the PINV-LRTHP, when the number of blocks is decreased to 8, is closer to the performance of the LRAP THP method. This occurs at the expense of increasing com-

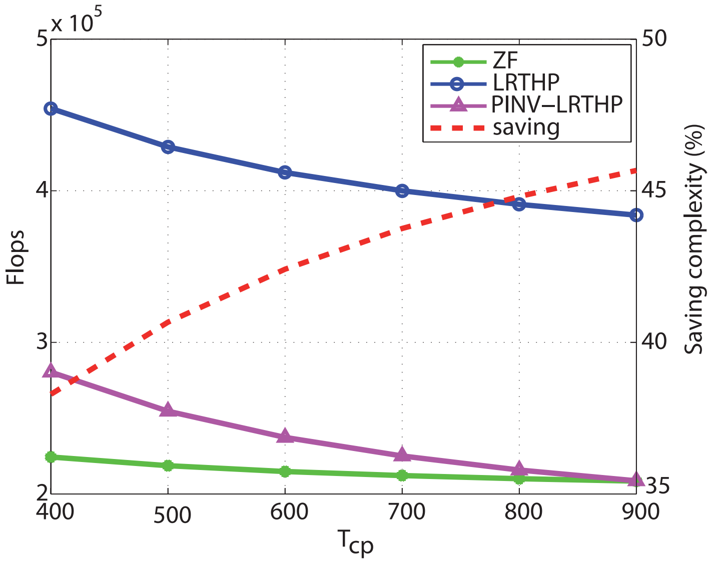


Figure 5.14. Computational cost of the precoding algorithms for a system with $N_t = 128$ transmitter antennas and $K = 128$ UTs using $4 - QAM$ modulation. For the PINV-LRTHP the number of blocks is equal to 16.

plexity. Even so, the saving cost of the PINV-LRTHP with respect to the LRAP THP method is up to 40%, as Figure 5.15 shows.

When the number of block employed is decreased to 4 the performance of the PINV-LRTHP improves with respect to chose a number of blocks equal to 8. Thereby, we can conclude that reducing the number of blocks the PINV-LRTHP algorithm improves, approaching the LRTHP performance. This is an expected behavior because if the PINV-LRTHP is executed with only one block, the LRAP THP performance is achieved. Therefore decreasing the number of blocks to get a better performance in terms of BER is at the expense of an increase in complexity, reducing the saving complexity in Figure 5.15. Thus, the number of blocks represents a trade-off parameter between complexity and performance.

Figure 5.16 represents the BER performance for a B equal to 16, 8 and 4 in a 64×64 MU-MIMO system. Understandably, the behavior in performance by decreasing the B value is similar to the behavior for the 128 antennas configuration. However, the saving cost for each possible B

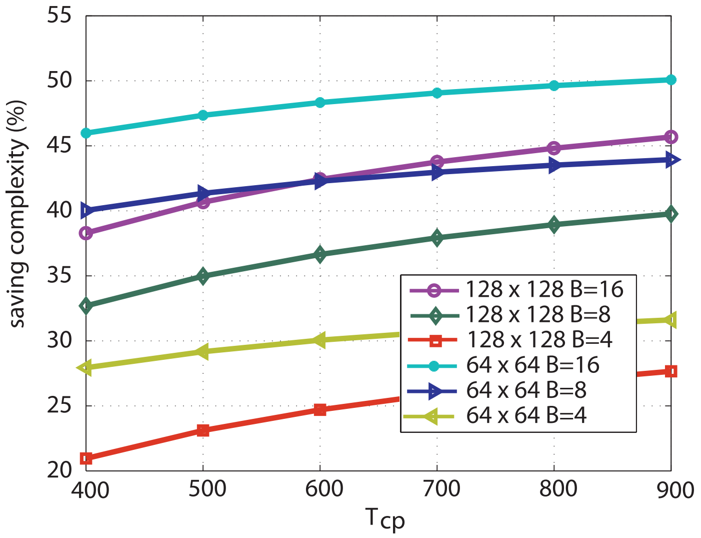


Figure 5.15. Saving complexity of the PINV-LRTHP precoding algorithm with respect to the LRAP THP method for different configuration systems and number of blocks.

values is reduced by increasing the system size as Figure 5.15 shows.

5.4 Conclusions

This chapter has described different precoding algorithms and has presented comparison among them in terms of BER and computational complexity. In the first section, the performance and computational cost of widely employed precoding algorithms were compared. Regarding the computational cost, it was shown that the overall cost depends on the coherence period. Thus, this parameter should be taken into account to direct our efforts towards reducing the term that is predominant in the overall cost. The results show how the LRAP THP and LRAP VB methods achieve the best performance, whereas, regarding the computational complexity, the LRAP THP is significantly lower than the LRAP-VB, doing the LRAP THP the best option of the LRAP type precoders. However, if tighter computational requirements are required the ZF or THP should be employed instead.

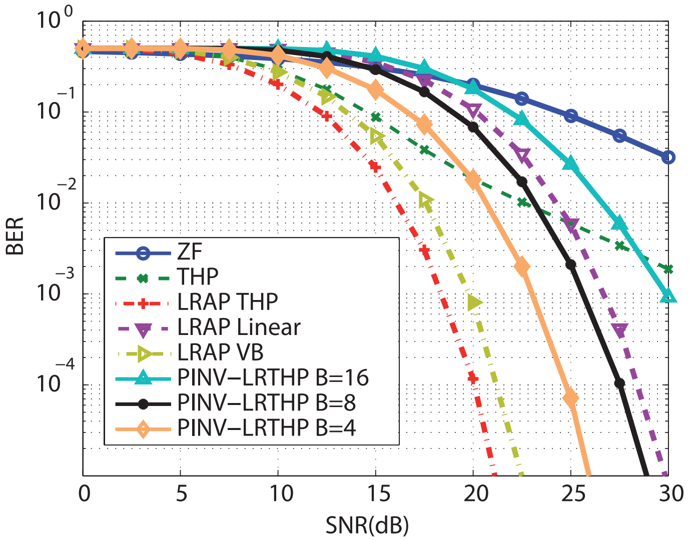


Figure 5.16. Performance of the precoding algorithms for a system with $N_t = 64$ transmitter antennas and $K = 64$ UTs using $4 - QAM$ modulation. For the PINV-LRTHP the number of blocks is equal to 16.

Throughout the second section of this chapter an hybrid precoding scheme was developed, which performs LR in a preprocesing stage depending on the condition number of the MIMO channel matrix. The performance and complexity results showed that the hybrid precoding scheme achieves almost the same performance as LRAP THP precoding but with the advantage of having lower average complexity.

The last part of the chapter presents a precoding algorithm named PINV-LRTHP is presented for a large MU-MIMO systems. This precoding method has been compared in terms of performance and computational complexity with the conventional LRAP THP and the ZF algorithms. The proposed precoding algorithm introduces a BD preprocessing to reduce the computational complexity of LRAP THP. The performance comparison reveals that the PINV-LRTHP algorithm suffers from performance loss, which depends on the number of UT blocks. Analytical and simulation results show that the proposed precoder requires much lower computational complexity than the LRAP THP, saving up to 50%. The saving also depends on the number of blocks, being this parameter a trade-off between complexity and performance. In some cases the computational cost achieves the same values as the ZF algorithm, obtaining a much better performance in terms of BER.

6 Parallel implementations

Parallel implementations 6

MIMO systems with large number of antennas and high constellation orders are currently generating considerable interest since they provide significantly higher spectral efficiency. Unfortunately, increased complexity of the signal detection stage is the price to pay for large MIMO systems. The Subspace Marginalization with Interference Suppression (SUMIS) detector exhibits good performance with reduced complexity and its design allows massively parallel algorithmic implementations. This section presents two practical parallel approaches of the SUMIS detector for large MIMO systems: 1) *using multicore processors and* 2) *using graphics processing units (GPU). Both approaches have been evaluated and compared in terms of performance and complexity with other detectors for different system parameters. Results show how these parallel versions allow to accelerate dramatically the detection, especially if very large systems and high order modulations are considered.*

6.1 MIMO Detection in Massive MIMO

As was mentioned in previous sections, an emerging research area called Large MIMO systems uses very large number of antennas, e.g., one hundred ore more [21], in contrast to conventional MIMO systems, which employ up to about ten antennas. The price to pay is an increased complexity and energy consumption at both ends. Particularly the MIMO detection problem is in general very expensive computationally to deal with. Thus, an adequate balance between efficiency and complexity is critical, especially in large MIMO systems [33] and large constellation sizes.

The optimal detector, which solves the MIMO detection problem optimally, computes the log-likelihood ratios (LLRs) values exactly and holds prohibitively high computational complexity, which grows with the size of the signal constellation and the number of antennas. In the above context, several detectors; which exhibit different trade-offs between complexity and performance, have been recently proposed. *Single Tree Search* (STS) [59] and *Repeated Tree Search* (RTS) [57] algorithms are the most common detectors which achieve the max-log approximation exactly. Both algorithms are based on the *Sphere Decoder* (SD) method and their computational complexity varies depending on the channel and noise realization. Suboptimal max-log algorithms reduce the complexity at the expense of a certain performance loss. Examples of such detectors are: *Soft Fixed Sphere Decoder* (SFSD) [13], *List Sphere Decoder* (LSD) [51] or *Soft Output k-Best* [62], among others. On the other hand, *Partial Marginalization* (PM) [14] and SUMIS [64] algorithms represent an intermediate approach between the optimal detector and its max-log approximation. The SUMIS algorithm offers a good trade-off between exact and approximate computation of the LLR values and a given complexity. This algorithm has been compared in this dissertation and [64] with some optimal and suboptimal detectors for a system of moderated size. In those cases, the SUMIS algorithm was proven to outperform the max-log approximation.

The performance and complexity of SUMIS for higher number of antennas and constellation order is assessed in this chapter. The use of exact and max-log algorithms, such as RTS or STS, become prohibitive for this number of antennas. On the other hand, SFSD for moderate number of antennas achieves almost max-log behavior at reasonable computing cost, for this reason it has been chosen for comparison. An efficient version called

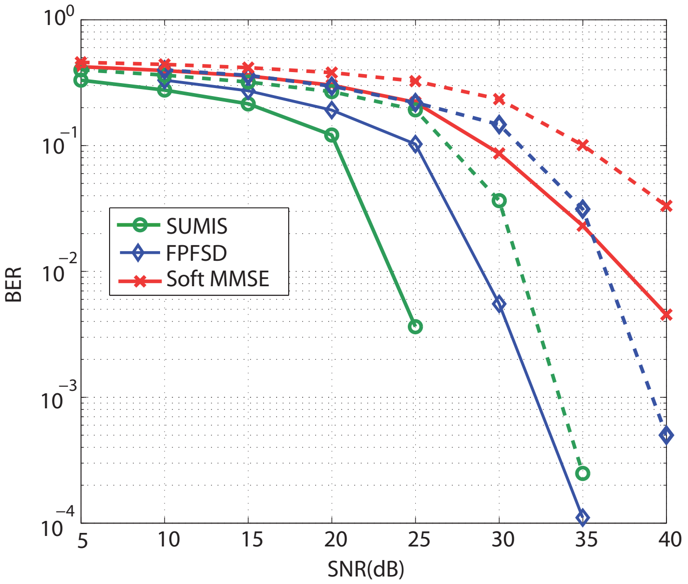


Figure 6.1. BER as a function of SNR for the 100 × 100 MIMO system with the LDPC code of rate 1/2. The dashed curves show the performance for a 64-QAM constellation and the solid curves show the performance for a 256-QAM constellation.

Fully Parallel Fixed Sphere Decoder (FPFSD) [92] has been chosen. A linear detector with reduced complexity such as Soft MMSE has also been chosen.

Using Monte Carlo simulations, we evaluate the performance of SUMIS and the algorithms chosen for comparison in terms of BER and Mutual Information (MI). The transmitted symbols are assumed to be independent and with equal probability. The transmitted bits are encoded using a 1/2 LDPC code of codeword size 648 bits, which is available from http://www.csl.cornell.edu/vstuder/software ldpc.html and implements a LDPC code from the IEEE 802.11n wireless LAN standard. There is no iteration between the detector and the decoder and the sum-product algorithm has been chosen as the channel decoding option.

BER performance, for a 100 × 100 MIMO system, is represented for SUMIS, Soft MMSE and SFSD algorithms in Fig. 6.1. The curves show

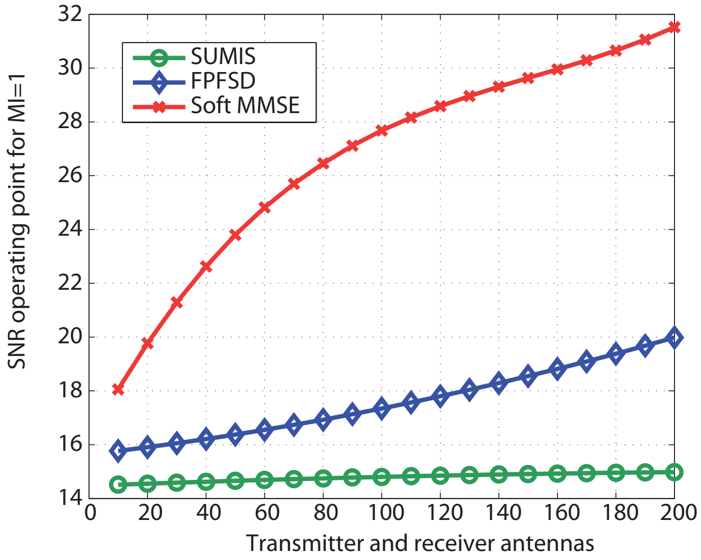


Figure 6.2. Minimum SNR required to achieve a value of 1 in the MI parameter. The MI is calculated between the LLR values obtained after decoding and the transmitted coded bits. The required SNR is represented as a function of the number of antennas (with $n_T = n_R$). The simulation have been carried out for a 16-QAM constellation and the LDPC code of rate 1/2.

clearly that the SUMIS detector performs much better than the other algorithms for 64-QAM and 256-QAM constellations. It is noteworthy that SUMIS algorithm exhibits an improvement or gain up to 5 dB in SNR with respect to the FPFSD.

In addition to the performance comparison in terms of BER, we also provide a detector comparison in terms of MI [93][94]. In Fig. 6.2 the minimum SNR required to achieve a given MI is referred as the “minimum SNR” for that MI. This Figure represents that minimum depending on the number of antennas. The results in Fig. 6.2 clearly show that the SUMIS detector achieves a MI equal to 1 at lower SNR. It’s worth noting that the “minimum SNR” scales more linearly with the increase in overall antennas for the SUMIS algorithm than for the other algorithms. The behavior is the same for higher constellation orders.

The performance for the three different algorithms in Large MIMO systems has been illustrated by Figs. 6.1-6.2. Aditionally, the computational cost is represented in Fig. 6.3 in terms of flops. Fig. 6.3 curves represent the number of flops for different algorithms depending on the number of antennas for a 256-QAM system. The bottleneck of the computational cost in the three analyzed algorithms is due to the number of antennas of the system ($\mathcal{O}(n_T^3)$ for the three algorithms). A variation in the constellation order affects only slightly the results. Fig. 6.3 illustrates that, for Large MIMO systems, the computational cost of SUMIS is in the same order of magnitude as the linear method.

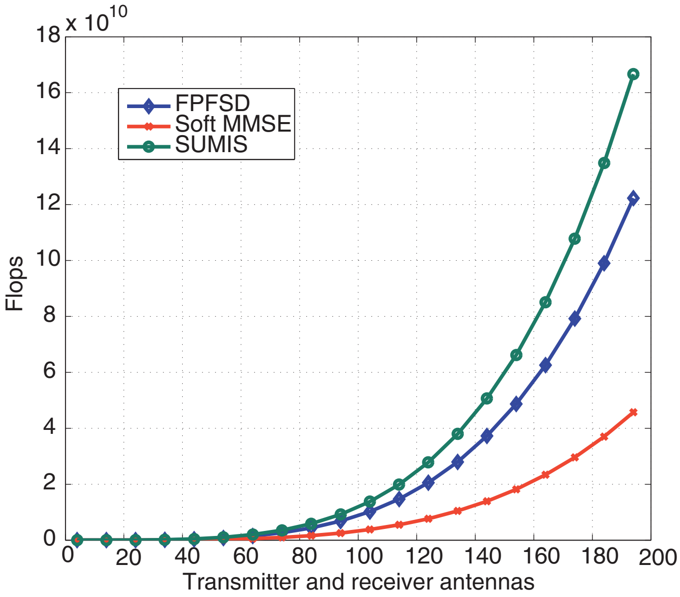


Figure 6.3. Flops as a function of the number of antennas (with $n_T = n_R$) with the LDPC code of rate 1/2 and a 256-QAM constellation.

Even so, SUMIS detector can be the bottleneck for the overall system performance if large number of antennas or high constellation orders are used. Thus, there exits a needed of MIMO detector algorithms with suitable performance that can be evaluated for MIMO systems. In the above context, this chapter aims to reduce the computational cost of the SUMIS method, not only from a theoretical point of view, but through its scalable and versatile implementation for efficient processing thereof e.g. multicore processors and GPUs. This allows to guarantee the SUMIS detection performance in large MIMO systems with higher throughput.

6.1.1 Proposed Paralelization

The use of the last generation of High Performance Computing (HPC) systems such as multi-core CPUs and GPUs has become attractive for the efficient implementation of parallel signal processing algorithms with high computational requirements, such as high throughput MIMO detectors [95][96] and fast LDPC or Turbo decoders [97][98]. The implementation of advanced algorithms able to use both architectures is crucial in MIMO research, since it allows to fully exploit the capabilities of the modern computer architectures and to reduce the response time of computationally expensive problems. The programming challenge involves a deep knowledge of different programming languages and the architecture features. In this sense, high performance libraries become valuable tools for specialists of a particular field, since they ease the development of scientific codes.

The SUMIS algorithm computes the $\lambda_{i,b}$ and $L_{i,b}$ values using (3.7) and (3.11) respectively, where the number of elements in the two summations over $\bar{\mathbf{s}}$ is M^{n_s}. Therefore, SUMIS algorithm has to calculate the expression $\exp(-\frac{1}{2}\|\mathbf{y}-\overline{\mathbf{H}}\bar{\mathbf{s}}\|^2_Q)$ a number of times M^{n_s} to achieve the total $\lambda_{j,b}$ or $L_{j,b}$ values in each stage. For this reason almost the 98% of the total signal detection time is consumed to compute these terms. Therefore, the parallelization is focused on reducing the computational cost of the $\exp(-\frac{1}{2}\|\mathbf{y}-\overline{\mathbf{H}}\bar{\mathbf{s}}\|^2_Q)$ terms in (3.7) and (3.11).

In [64], Appendix A, an optimization of the SUMIS algorithm is explained. Following this approach the next expression can be derived by simple matrix manipulations:

$$\begin{aligned}\|\mathbf{y}-\overline{\mathbf{H}}\bar{\mathbf{s}}\|^2_Q =&((\overline{\mathbf{H}}^T\mathbf{Q}^{-1}\overline{\mathbf{H}})^{-1}\overline{\mathbf{H}}^T\mathbf{Q}^{-1}\mathbf{y}-\bar{\mathbf{s}})^T\overline{\mathbf{H}}^T\mathbf{Q}^{-1}\overline{\mathbf{H}}\\ &\cdot((\overline{\mathbf{H}}^T\mathbf{Q}^{-1}\overline{\mathbf{H}})^{-1}\overline{\mathbf{H}}^T\mathbf{Q}^{-1}\mathbf{y}-\bar{\mathbf{s}})+\mathbf{y}^T\mathbf{Q}^{-1}\mathbf{y}\\ &-\mathbf{y}^T\mathbf{Q}^{-1}\overline{\mathbf{H}}(\overline{\mathbf{H}}^T\mathbf{Q}^{-1}\overline{\mathbf{H}})^{-1}\overline{\mathbf{H}}^T\mathbf{Q}^{-1}\mathbf{y}.\end{aligned} \tag{6.1}$$

Renaming: $\mathbf{C}=\mathbf{Q}^{-1}\overline{\mathbf{H}}$, $\mathbf{D}=\overline{\mathbf{H}}^T\mathbf{Q}^{-1}$, $\mathbf{B}=\mathbf{D}\overline{\mathbf{H}}$, $\mathbf{A}=\mathbf{B}^{-1}\mathbf{D}$, $\delta_1 =$

$\mathbf{y}^T\mathbf{Q}^{-1}\mathbf{y}$ and $\delta_2 = \mathbf{y}^T\mathbf{C}\mathbf{A}\mathbf{y}$, equation (6.1) can be finally expressed as:

$$\|\mathbf{y} - \overline{\mathbf{H}}\overline{\mathbf{s}}\|_Q^2 = (\mathbf{A}\mathbf{y} - \overline{\mathbf{s}})^T\mathbf{B}(\mathbf{A}\mathbf{y} - \overline{\mathbf{s}}) + \delta_1 - \delta_2, \tag{6.2}$$

where the terms δ_1 and δ_2 in (6.2) do not depend on $\overline{\mathbf{s}}$. Equation (6.2) can be computed using the Linear Algebra package (LAPACK) [99]. This library is composed of several optimized math routines including basic vector matrix operations (BLAS) [100] and routines for solving systems of simultaneous linear equations among others. Specifically, we are using the following BLAS and LAPACK functions:

- Matrix-Matrix multiplication: *dgemm*
- Matrix-Vector multiplication: *dgemv*
- LU factorization: *dgetrf*
- Inverse of a matrix using the LU factorization: *dgetri*

For the multicore implementation we are using the Intel Math Kernel Library (MKL) [101], which is composed of several optimized math routines and it is optimized specifically for Intel processors. For the GPU implementation we use the cuBLAS Library [102], which is a GPU-accelerated version of the complete standard BLAS library.

6.1.2 Multi-threaded Implementation

During the last years, the main microprocessors manufacturers such as Intel or AMD have been focused on developing faster and smarter chips. Their purpose is to get maximum performance with minimal consumption by integrating multiple processing units (called cores) onto a single processor. Thus, these cores can process simultaneously multiple tasks although at a lower clock rate.

OpenMP is an Application Programming Interface (API) [103] for programming shared-memory parallel computers. It consists of a set of compiler directives, callable library routines and environment variables that may be embedded in a code written in a programming language such as Fortran or C/C++ and operating systems such as GNU/Linus, Max OS X, and Windows. A master thread launches a number of slave threads and divides the workload among them. The runtime will attempt to allocate the threads to different processors and the threads will run concurrently.

For the parallelization of the SUMIS algorithm we assume a MIMD computer (i.e. multiple instruction, multiple data) with shared memory. This system has p processors which share a common central memory. The MKL multi-threaded version uses the OpenMP application programming interface [103] to distribute the computation among the different cores. We can parallelize each MKL function by selecting the desired number of threads before calling the function. However, the performance of these functions greatly depends on the size of the channel matrix. Even though we are considering channel matrix sizes larger than usual used in MIMO communication systems, they are still too small to fully exploit the MKL performance. Previous results show how increasing the number of threads barely manage to accelerate the sequential version. For this reason we have chosen to parallelize at a higher level. To clarify how the distribution of tasks has been carried out, we have presented the complete pseudo-code related to the SUMIS detector in Algorithm 7. The OpenMP pragma ***omp parallel for*** in line 7 distributes the for loop iterations among the threads and therefore the SUMIS processing is performed per ith-symbol in parallel.

6.1.3 CUDA Implementation

A GPU is a coprocessor originally designed for accelerating the computation of computer graphics. However in recent years a trend in the scientific computing community has appeared based on the use of the GPU to handle general purpose problems traditionally solved by the Central Processing Unit (CPU). GPUs are present in almost all computing systems, laptops, PCs and supercomputers [104]. The fast advance in programmability has allowed its use to deal with many problems with an insatiable appetite for computing power. Compute Unified Device Architecture (CUDA) [105] is a software programming model that exploits the massive computation potential offered by GPUs. A GPU can have multiple stream multiprocessors with a certain number of pipelined cores each.

In contrast to multicores, the GPUs follow a single-instruction multiple-threads (SIMT) programming model, i.e., a single set of instructions is executed on different data sets. In this model, the programmer defines the kernel function that contains a set of common operations. At runtime, the kernel is called from the main CPU and spawns a large number of

Algorithm 7 SUMIS pseudo-code OpenMP implementation.

```
Input: $\mathbf{H}$, $\mathbf{y}$, $n_s \in \{1, \ldots, n_T\}$
Output: Log Likelihood Ratios ($\mathbf{L}$)
 1: Calculate $\mathbf{H}^T\mathbf{H}$
 2: /* Stage I */
 3: for $i = 1, \ldots n_T$ do
 4:     Decide upon a partitioning in 3.6, $\mathbf{H} = [\overline{\mathbf{H}} \quad \widetilde{\mathbf{H}}]$ based on $\mathbf{H}^T\mathbf{H}$
 5: end for
 6: Set $\widetilde{\boldsymbol{\Upsilon}} = \mathbf{I}$
 7: # pragma omp parallel for
 8: for $j = 1, \ldots, n_T$ do
 9:     Variance matrix $\mathbf{Q} = \widetilde{\mathbf{H}}\widetilde{\boldsymbol{\Upsilon}}\widetilde{\mathbf{H}}^T + \sigma_n^2\mathbf{I}$
10:     for $p = 1, \ldots, M^{n_s}$ do
11:         $\omega[j]_p = \exp(-\frac{1}{2}\|\mathbf{y} - \overline{\mathbf{H}}\overline{\mathbf{s}}_{:,p}\|_Q^2)$ such as (6.2)
12:     end for
13:     for $b = 1, \ldots, k$ do
14:         $\lambda_{i,b} = \log \frac{\sum_{p:\overline{\mathbf{s}}_{:,p} \epsilon \chi_{j,b}^0} w[j]_p}{\sum_{p:\overline{\mathbf{s}}_{:,p} \epsilon \chi_{i,b}^1} w[j]_p}$
15:     end for
16: end for
17: /* Stage II */
18: for $i = 1, \ldots, n_T$ do
19:     $\mathbb{E}\{s_j|\mathbf{y}\} \triangleq \sum_{\mathbf{s}\in\Omega} s \prod_{b=1}^{k} \frac{1}{1+e^{(-2s_{i,b}+1)\lambda_{i,b}}}$
20:     Suppress the interfering vector
21:     $\mathbf{y}' \triangleq \mathbf{y} - \widetilde{\mathbf{H}}\mathbb{E}\{\widetilde{\mathbf{s}}|\mathbf{y}\}$
22:     Calculate $\widetilde{\boldsymbol{\Upsilon}} = \mathbb{E}\{\text{diag}(\widetilde{\mathbf{s}})^2|\mathbf{y}\} - \mathbb{E}\{\text{diag}(\widetilde{\mathbf{s}})|\mathbf{y}\}^2$
23: end for
24: Repeat steps 7 to 15 with [$\mathbf{y}'$, $\mathbf{Q}'$, $L_{i,b}$] instead of [$\mathbf{y}$, $\mathbf{Q}$, $\lambda_{i,b}$]
```

threads blocks, which is called grid. Each thread block contains multiple threads, usually up to 1024, and all of the blocks within a grid must share the same size. Blocks and grids can be one-dimensional, two-dimensional or tri-dimensional but they must not exceed a certain size stated in the GPU's specifications. Each thread can select a set of data using its own unique ID and executes independently the kernel function on the selected set of data. Threads of the same block can share data between them by using the shared memory. However, threads of different blocks are independent and should use global memory to share data once all threads have finished running the full kernel. The equivalent cuBLAS [102] functions of each selected LAPACK functions are:

- Matrix-Matrix multiplication: *cublasDgemmBatched*
- LU factorization: *cublasDgetrfBatched*
- Inverse of a matrix using the LU facorization: *cublasDgetriBatched*

Batched functions are intended to be used for matrices of small sizes where the launch overhead is a significant factor. For small sizes, this kind of functions improve significantly the performance compared to making calls to its corresponding cublas routine. However, matrix-vector and vector-vector functions are not yet available for batched computation. For this reason we have implemented three specific functions in CUDA. These functions are described in Algorithms 8, 9 and 10 and explained bellow.

- cudaDgemvBatched: The kernel in Alg. 8 performs the matrix-vector multiplications of an array of matrices and vectors $z[n] = A[n] * y[n]$ for $n = 0, ..., n\_batch - 1$. Each A variable is a $(n_{rows} \times n_{cols})$ matrix, z is a $(1 \times n_{rows})$ vector and y is a $(n_{cols} \times 1)$ vector. Each thread determines the element to process by using its unique ID using: its block index (blockIdx.x), its thread index (threadIdx.x) and the number of threads per block (blockDim.x). A bidimensional grid configuration with $N_{Bx} = N_B$, $N_{By} = N_B$ blocks per dimension has been considered for kernels. The number of blocks N_B depends on the number of threads per dimension, which are denoted by N_{tx} and N_{ty}, respectively. Then the value of NB is obtained as

$$N_B = \left\lceil \sqrt{\frac{n_{th}}{N_{tx} \times N_{ty}}} \right\rceil \tag{6.3}$$

  where n_{th} is the number of CUDA threads, in this case $n_{th} = n\_batch = n_T$. Each thread calculates its unique identifier (n) and executes the kernel on the selected set of data independently.

- cudaDgevmvBatched: The kernel in Alg. 9 performs the vector-matrix-vector multiplications of an array of matrices and vectors $\delta[n] = z[n] * A[n] * y[n]$ for $n = 0, ..., n\_batch - 1$. Each A variable is a $(n_{rows} \times n_{cols})$ matrix, z is a $(1 \times n_{rows})$ vector, y is a $(n_{cols} \times 1)$ vector and *delta* is a $(1 \times n\_batch)$ vector (see Alg. 9). This function is used to compute δ_1 and δ_2 values in (6.2). In this case the number of threads is $n_{th} = n\_batch = n_T$.

- cudaDgevmvBatched_v2: This function is similar to the previous *cudaDgevmvBatched* kernel. In these case the number of threads is $n_{th} = n_T \times M^{n_s}$ since we need to compute $(\mathbf{Ay} - \bar{\mathbf{s}})^T \mathbf{B}(\mathbf{Ay} - \bar{\mathbf{s}})$ for each $\bar{\mathbf{s}} \in \Omega^{n_s}$. All vectors $\mathbf{v} = \mathbf{Ay} - \bar{\mathbf{s}}$ are previously computed. This function performs the vector-matrix-vector multiplications of an array of matrices and vectors $\omega[n]_p = \exp(-\frac{1}{2}(v[n]'_{:,p} * B[n] * v[n]_{:,p} + \delta_1[n] - \delta_2[n]))$ for $n = 0, ..., n_T - 1$ and $p = 0, ..., M^{n_s} - 1$.

Algorithm 8 Kernel of the cudaDgemvBatched function. The nth thread computes a matrix-vector multiplication.

```
Input: $\mathbf{A}$, $\mathbf{y}$, $n_{rows}$, $n_{cols}$
Output: $\mathbf{z}$
n ← ID
for $i = 1, \ldots, n_{rows}$ do
    $z[n]_i = 0$
    for $j = 1, \ldots, n_{cols}$ do
        $z[n]_i = z[n]_i + A[n]_{i,j} * y[n]_j$
    end for
end for
```

As said above, the $\exp(-\frac{1}{2}\|\mathbf{y} - \overline{\mathbf{H}}\bar{\mathbf{s}}\|^2_Q)$ terms have been parallelized for the LLR calculation. Other operations in (3.7) and (3.11) show low complexity compared to the previous terms and does not have a parallel pattern (see lines 13-15 Alg.1). Moreover, the so-called *warp divergence* [105] plays an important role in the performance. In CUDA, threads are executed in warps of 32 threads, with all threads in the warp executing the same instruction at the same time. However, in this part different threads in a warp need to cope with different tasks. CUDA serializes the different execution paths to generate correct code. For this reason, these operations have not been parallelized.

Algorithm 9 Kernel of the cudaDgevmvBatched function. The nth thread computes a vector-matrix-vector multiplication.

```
Input: $\mathbf{A}$, $\mathbf{y}$, $\mathbf{z}$, $n_{rows}$, $n_{cols}$
Output: $\boldsymbol{\delta}$
$n \leftarrow ID$
for $i = 1, \ldots, n_{rows}$ do
    $aux = 0$
    for $j = 1, \ldots, n_{cols}$ do
        $aux = aux + A[n]_{i,j} * y[n]_j$
    end for
    $\delta[n] = \delta[n] + z[n]_i * aux$
end for
```

Algorithm 10 Kernel of the cudaDgevmvBatched_v2 function.

```
Input: $\mathbf{B}$, $\mathbf{v}$, $\boldsymbol{\delta_1}$, $\boldsymbol{\delta_2}$, $n_{rows}$, $n_{cols}$
Output: $\boldsymbol{\omega}$
$n \leftarrow ID/M^{n_s}$
$p \leftarrow ID\%M^{n_s}$
$aux_2 = 0$
for $i = 1, \ldots, n_{rows}$ do
    $aux = 0$
    for $j = 1, \ldots, n_{cols}$ do
        $aux = aux + B[n]_{i,j} * v[n]_{j,p}$
    end for
    $aux_2 = aux_2 + v[n]_{j,p} * aux$
end for
$\omega[n]_p = \exp(-\frac{1}{2}(aux_2 + \delta_1[n] - \delta_2[n]))$
```

6.2 Results

We measure the execution time of the detection, with different number of antennas and constellation sizes to evaluate the SUMIS efficiency of the proposed parallel prototypes. For this purpose, a machine with one Nvidia Tesla K40C GPUs and two multicore Intel processors has been employed. Each multicore is an Intel Xeon CPU E5-2697 at 2.70 GHz with 12 cores per CPU. On the other hand, the most relevant features and specifications of the GPU are listed in Table 6.1.

Number of stream multiprocessors	15
Number of cores	2880
Clock Rate	0.75GHz
Global Memory	12 GB
Memory Clock Speed	6GHz

Table 6.1. Nvidia Tesla K40c especifications.

The speedup (S_P) has been computed as the ratio between the execution time of the *"reference SUMIS algorithm"* (T_S)(see Table 6.2) , and the execution time of the OpenMP and the GPU versions (T_P). The MKL implementation with the *-mkl=sequential* compiler option was selected for the *"reference SUMIS algorithm"*.

T_S	M = 16	M=64	M=256	M=1024
$n_T = 8$	0.06	0.12	0.56	4.30
$n_T = 24$	0.24	0.38	1.70	13.14
$n_T = 48$	1.24	1.56	4.20	27.18
$n_T = 100$	12.82	13.34	19.28	67.12
$n_T = 200$	158.66	160.34	164.58	260.46

Table 6.2. Sequential Execution Times in miliseconds (T_S) for the SUMIS algorithm with different number of antennas and different QAM constellation.

Tables 6.3 and 6.4 illustrate the execution time and speedup of the different configurations. It can be observed that, generally for the OpenMP version, higher system sizes achieve higher speedup. The soft detection can be accelerated up to 9 times.

The experimental GPU measurements in Tab. 6.4 show how CUDA version fails to accelerate the sequential version when the number of antennas and the constellation order is small. This is due to the lower complexity of the detector with these parameters. This problem gradually disappears when the complexity of the detection stage increases, for example when the number of transmitter antennas n_T is 200 and M is 1024. In this scenario, where very large MIMO arrays are considered, the CUDA detector is up to 4 times faster than the sequential version.

$T_P\backslash S_P$	M=16	M=64	M=256	M=1024
$n_T = 8$	0.04\1.50	0.08\1.50	0.21\2.67	1.35\3.19
$n_T = 24$	0.18\1.33	\0.25\1.52	0.632.70	2.67\4.92
$n_T = 48$	0.67\1.85	1.06\1.47	1.79\2.35	5.57\4.88
$n_T = 100$	2.19\5.85	3.06\4.36	4.24\4.55	13.08\5.13
$n_T = 200$	17.00\9.33	18.80\8.53	21.86\7.53	33.26\7.83

Table 6.3. Multicore Execution Times in milliseconds (T_P) and Speedup (S_P) for the OpenMP SUMIS algorithm with different number of antennas and different QAM constellation.

$T_P\backslash S_p$	M=16	M=64	M=256	M=1024
$n_T = 8$	0.27\0.22	0.25\0.48	0.51\1.10	1.52\2.82
$n_T = 24$	0.18\1.33	0.46\0.82	0.63\2.70	1.52\8.64
$n_T = 48$	1.29\0.96	1.49\1.05	2.36\1.77	3.50\7.76
$n_T = 100$	5.85\2.19	6.26\2.13	7.30\2.64	21.10\3.18
$n_T = 200$	39.30\4.04	40.30\3.97	42.31\3.89	70.80\3.68

Table 6.4. GPU Execution Times in milliseconds (T_P) and Speedup (S_P) for the CUDA SUMIS algorithm with different number of antennas and different QAM constellation.

It is interesting to note how both parallel versions allow to boost the performance of the system with a speed comparable to a low-complexity linear detector such as MMSE. For example, with $n_T = 200$ and 256-QAM, the complexity of SUMIS ($\approx 16 \cdot 10^{10}$) is 4 times higher than the MMSE detector ($\approx 4 \cdot 10^{10}$) (see Fig. 6.3). However, by using the parallel implementations, the complexity of the SUMIS detector is reduced to $\approx 2 \cdot 10^{10}$ for the OpenMP version and $\approx 4 \cdot 10^{10}$ for the GPU version. Thus, we can detect signals with a similar and even higher throughput than the MMSE detector with much better BER.

6.3 Conclusions

This Chapter has provided details about the parallelization trough GPU and multi-core CPUs, of the SUMIS algorithm presented in Chapter 3. Furthemore, the SUMIS algorithm has been evaluated together with the SFSD and soft MMSE detectors for a very lar MIMO systems reaching up to 200 transmitter/receiver antennas and up to 1024-QAM constellations. This evaluation has been done in terms of performance and complexity. The comparison between the algorithms shows the robust, versatile and scalable behavior of the SUMIS algorithm. This detector behaves much better than the other detectors in terms of BER, achieving up to 5 dB improvement in SNR compared to SFSD.

Currently, we are still far from reaching the speeds required by the IEEE 802.11n wireless LAN standard, which makes the implementation unfeasible for on line use. However, these approaches allow to reduce considerably the complexity of the simulation of large MIMO systems with scalable quasi optimal soft detector, opening the door to foresee new technology performance faster than by conventional simulation.

Conclusions 7

Conclusions 7

This chapter summarizes the findings of this research work, revisiting the research objectives given in the introductory chapter. At the beginning, the first Section reviews the contents of this study, outlining the main conclusions that were extracted from each chapter. Next section presents different recommendations for future research. The final part contains a list of works published during the course of candidature for the PhD degree. Additionally, institutional acknowledgements are given.

7.1 Summary

MIMO communication system have emerged as one of the most promising technologies that allow exploiting the spatial dimension in addition to the time and frequency dimensions. MIMO techniques can also be used in a multiuser scenario. However the benefits come at the expense of more complex system. For this reason it is necessary the design of algorithm with low complexity and good performance. This thesis focuses mainly on improving the efficiency of precoding and detection implementations.

The first part of this dissertation presents a performance and a complexity study of different soft-output algorithms. Furthermore two soft MIMO detection algorithms have been presented. The first algorithm extends the efficient hard ML detector BOHD to the soft-output detection. The results demonstrated that the BOHD-SO algorithm requires less number of flops than other known low-complexity algorithms and the performance achieved is almost the max-log performance. The second proposed algorithm reduces the complexity of the SUMIS algorithm and the improvements are more noticeable when the number of antennas or constellation order increase.

Chapter 4 reviews the SISO detection in an ID-BICM MIMO system. The computational cost of the STS optimal algorithm has been reduced throughout two different algorithms and maintaining the max-log performance. In the case with clipping (which is the most relevant from a practical point of view), DTS exhibits a smaller computing time, while BORTS expands fewer nodes. Even though there is some uncertainty about the results because the metrics show contradictory trends in some cases, the results still clearly show that the proposed algorithm has an excellent performance in large MIMO detection problems, since the proposed method can perform comparatively even better for larger constellations or for systems with many antennas. Furthermore the non-optimal BOHD-SO algorithm proposed in the previous section has been extended to allow a priori information.

Chapter 5 describes different precoding algorithms and presents a comparison among them in terms of BER and computational complexity. Furthermore it was proved that the channel condition number has influence on the precoding performance. Thus, taking into account the performance and computational complexity, an hybrid precoding scheme based on the channel condition number has been proposed. Results show that with the hybrid precoding scheme we can achieve the performance of the LRAP THP method at less complexity. On the other hand, a precoding method called PINV-LRTHP also has been proposed. The PINV-LRTHP has been proposed to employ in Large MIMO systems reducing the complexity of the conventional algorithms, such us LRAP THP, achieving suitable performance. Furthermore, a variable parameter B allows to change the tradeoff between performance and complexity depending on the system requirements.

Chapter 6 provides some details on the implementation of the SUMIS algorithm on a multicore and GPU platforms. The implementations was compared with a single CPU, showing that the parallel implementations were able to considerably accelerate the computation by simultaneously processing. Results show that the OpenMP version outperforms the sequential version for all constellation orders and system sizes evaluated. It can be observed that for higher system sizes the speedup increases. However, the GPU version does not produces any improvement when the number of antennas and constellation order is small. Nevertheless, when these parameters increase the situation is really different and the CUDA implementation is up to 4 times faster than the sequential version. Results show that we are still far form reaching the speeds required by the communication standards as IEEE 802.11n. However, the parallel implementations allow to reduce considerably the complexity of the simulation of large MIMO systems opening the door for researchers to assess this kind of technologies faster than by conventional simulation.

7.2 Further Work

Following the investigations described in this thesis, the main lines of research that remain open are listed below:

- This thesis was focused on the design of MIMO detectors and precoding assuming perfect knowledge about the channel matrix (perfect CSIT). In practice, however, it is common to only have information about a version of the channel matrix which includes estimation errors. Hence, it would be interesting to evaluate and redesign the detection approaches developed in this thesis taking into account channel estimation errors or partial knowledge of the channel matrix.

- The MIMO algorithms developed in this thesis were mainly designed and evaluated assuming the transmission through flat fading channels with zero-mean Gaussian entries. Either using more realistic channel models or directly real channel measurements would be interesting topics for future research and develope taylorized MIMO detectors.

- In MIMO detection, it is very common to sort the columns of the channel matrix to improve the efficiency of the tree search. There are many possible reorderings; those described in [13] [46][106] are just some of them. As usual, there is a trade-off between the complexity of the reordering and the benefits obtained (reduction of number of expanded nodes). Any of these reordering can be similarly applied to the proposed BORTS and DTS methods. Since the techniques described are not linked to any particular reordering, we have chosen to use no reordering to obtain the results shown in the corresponding chapter. However, some preliminary experiments show similar benefits from applying a given reordering to any of the proposed methods. In other words, the improvement obtained from applying the same reordering to BORTS and DTS. However, there are many reorderings not yet tested, so that this matter must be explored further.

- In chapter 5 the PINV-BD algorithm has been proposed. This algorithm divides the users into several groups and cancel the interference among the different groups with a BD algorithm. Then the interference among the users within the same group is canceled through a non-linear precoding algorithm such us LRAP THP. It was proved by simulation that under the considered channel, by clustering the users in aleatory way the same performance than clustering by correlation is achieved. However, a better performance could be obtaining analyzing other clustering algorithms under other channels.
- Chapter 6 presents a parallel implementation of the SUMIS algorithm. However other of the proposed algorithm through this dissertation are allowed to implement in parallel architectures. This is the case of BORTS proposed in chapter 4, since every run of the BOHD algorithm can be performed in parallel. The PINV-BD precoding algorithm proposed in chapter 5 is also suitable to be parallelized. In this algorithm, after the first stage, the interference between users of the same group is mitigated in the second step, and thus operation can be realized for each group at the same time.

7.3 List of Publications

A list of published work produced during the course of candidature for the degree is presented in what follows.

Refereed ISI Journals

- **M. Simarro**, V. Garcia, A.M.Vidal, F. J. Martínez-Zaldívar, A. Gonzalez "Soft MIMO Detection Through Sphere Decoding and Box Optimization ". *Signal Processing.* Submitted for review.
- **M. Simarro**, F. J. Martínez-Zaldívar, A. Gonzalez, "Combined precoding for Multiuser MIMO Satellite Communications". *Computers and Electrical Engineering, Elsevier.* Submitted for review.
- V. Garcia, **M. Simarro**, F. J. Martínez-Zaldívar, A.M.Vidal, A. Gonzalez, "Maximum linkelihood soft-output detection through Sphere Decoding combined with box optimization". *Signal Processing*, vol. 125, no. 0, pp. 249-260, 2016.
- C. Ramiro, **M. Simarro**, F. J. Martínez-Zaldívar, A.M.Vidal, A. Gonzalez, "A GPU implementation of an iterative receiver for energy saving MIMO ID-BICM systems". *Journal of Supercomputing*, vol. 70, no. 2, pp. 541-551, January 2014.

Peer-reviewed non-ISI Journals

- **M. Simarro**, C. Ramiro, F. J. Martínez-Zaldívar, A.M. Vidal, A. Gonzalez, G. Piñero, V.M. Garcia, "Parallel Implementation Strategies for MIMO ID-BICM Systems", in *Waves*, vol. 5, pp.1889-8297, 2013.

Papers in International Conferences

- C. Ramiro, **M. Simarro**, A. Gonzalez, A.M.Vidal, "Parallel SUMIS Soft Detector for Large MIMO Systems on Multicore and GPU". *Proceedings of the 17th International Conference on Computational and Mathematical Methods in Science and Engineering (CMMSE)*, Rota, Spain, July 2017. Accepted.
- **M. Simarro**, F. Domene, F. J. Martínez-Zaldívar, A. Gonzalez, "Block Diagonalization aided Precoding algorithm for Large MU-MIMO Systems". *Proceedings of the 13th International Wireless*

Communications and Mobile Computing Conference, Valencia, Spain, June 2017. Accepted.

- **M. Simarro**, V. Garcia, F. J. Martínez-Zaldívar, A. Gonzalez, A.M. Vidal, "Complexity reduction of SUMIS MIMO detection based on Box Optimization for Large Systems". *Proceedings of the 41th IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP 2016)*, Shanghai, China, , March 2016.

- **M. Simarro**, F. J. Martínez-Zaldívar, A. Gonzalez, V. Garcia, A.M. Vidal, "Low Complexity Soft-Input Soft-Output detector based on repeated tree search strategy". *Proceedings of the 81th Vehicular Technology Conference (VTC2015-Spring)*, Glasgow, Scotland, May 2015.

- **M. Simarro**, C. Ramiro, F. J. Martínez-Zaldívar, A.M. Vidal, A. Gonzalez, G. "A parallel iterative MIMO receiver with variable complexity detectors". *Proceedings of the 13th International Conference on Computational and Mathematical Methods in Science and Engineering (CMMSE)*, Cabo de Gata, Almería, Spain, June 2013.

Papers in National Conferences

- **M. Simarro**, F. J. Martínez-Zaldívar, A. Gonzalez, V. Garcia, "Receptor Iterativo para un sistema MIMO con detección soft mediante la combinación de la estrategia de búsqueda repetitiva con ZF". *Proceedings of the XXIX Simposium Nacional de la Unión Científica Internacional de Radio (URSI)*, Valencia, Spain, September 2014.

7.4 Institutional Acknowledgements

This work has received financial support of the following projects:

- PROMETEO II: High Performance Computation and Communications and Applications in Engineering (PROMETEOII/2014/003.

Generalitat Valenciana. Spain.)

- DISCOSOUND: Distributed and Collaborative Processing of Sound Signals: Algorithms, Tools and Applications (TEC2012-38142-CO4-01).

- PROMETEO: High Performance Computing Tools for solving Signal Processing Problems on Parallel Architectures (PROMETEO/2009/ 013. Generalitat Valenciana. Spain.)

- FPU: Ayudas para la Formación de Profesorado Universitario. (FPU) (FPU 12/01274 . Ministerio de Educación, Cultura y Deporte. Spain).

- RACHEL: Técnicas de acceso radio para redes inalámbricas heterogéneas (TEC2013-47141-C4-4-R. Ministerio de Economia, Industria y Competitividad. Spain).

Bibliography

[1] M. Stanley, “The mobile Internet report,” *Morgan Stanley research*, 2009.

[2] C. V. N. Index, “Global mobile data traffic forecast update, 2016-2021,” Cisco, Tech. Rep., Febrery 2017.

[3] Ericsson, “Ericsson mobility report, on the pulse of the networked society,” Ericsson, Tech. Rep., June 2016.

[4] A. J. Paulraj, D. A. Gore, R. U. Nabar, and H. Bolcskei, “An overview of MIMO communications-a key to Gigabit wireless,” *Proceedings of the IEEE*, vol. 92, no. 2, pp. 198–218, 2004.

[5] D. Gesbert, M. Shafi, D.-s. Shiu, P. J. Smith, and A. Naguib, “From theory to practice: An overview of MIMO space-time coded wireless systems,” *IEEE Journal on selected areas in Communications*, vol. 21, no. 3, pp. 281–302, 2003.

[6] G. J. Foschini and M. J. Gans, “On limits of wireless communications in a fading environment when using multiple antennas,” *Wireless personal communications*, vol. 6, no. 3, pp. 311–335, 1998.

[7] J. G. Andrews, S. Buzzi, W. Choi, S. V. Hanly, A. Lozano, A. C. Soong, and J. C. Zhang, “What will 5G be?” *IEEE Journal on selected areas in communications*, vol. 32, no. 6, pp. 1065–1082, 2014.

[8] E. Perahia, "IEEE 802.11n development: History, process, and technology," *IEEE Communications Magazine*, vol. 46, no. 7, 2008.

[9] I. . W. Group *et al.*, "IEEE standard for local and metropolitan area networks. part 16: Air interface for fixed broadband wireless access systems," *IEEE Std*, vol. 802, pp. 16–2004, 2004.

[10] S. Sesia, M. Baker, and I. Toufik, *LTE-the UMTS long term evolution: from theory to practice.* John Wiley & Sons, 2011.

[11] E. Biglieri, R. Calderbank, A. Constantinides, A. Goldsmith, A. Paulraj, and H. V. Poor, *MIMO wireless communications.* Cambridge university press, 2007.

[12] B. Hassibi and H. Vikalo, "On the sphere-decoding algorithm I. expected complexity," *IEEE transactions on signal processing*, vol. 53, no. 8, pp. 2806–2818, 2005.

[13] L. G. Barbero and J. S. Thompson, "Extending a fixed-complexity sphere decoder to obtain likelihood information for turbo-MIMO systems," *IEEE Transactions on Vehicular Technology*, vol. 57, no. 5, pp. 2804–2814, 2008.

[14] E. G. Larsson and J. Jalden, "Fixed-complexity soft MIMO detection via partial marginalization," *IEEE transactions on Signal Processing*, vol. 56, no. 8, pp. 3397–3407, 2008.

[15] D. Wübben, D. Seethaler, J. Jaldèn, and G. Matz, "Lattice reduction: A survey with applications in wireless communications," vol. 28, no. 3, pp. 70–91, May 2011.

[16] J. Choi and H. Nguyen, "SIC-based detection with list and lattice reduction for MIMO channels," *IEEE Transactions on Vehicular Technology*, vol. 58, no. 7, pp. 3786–3790, 2009.

[17] D. L. Milliner, E. Zimmermann, J. R. Barry, and G. Fettweis, "A fixed-complexity smart candidate adding algorithm for soft-output MIMO detection," *IEEE Journal of Selected Topics in Signal Processing*, vol. 3, no. 6, pp. 1016–1025, 2009.

[18] L. Bai and J. Choi, "Partial MAP-based list detection for MIMO systems," *IEEE Transactions on Vehicular Technology*, vol. 58, no. 5, pp. 2544–2548, 2009.

[19] E. G. Larsson, “MIMO detection methods: How they work [lecture notes],” *IEEE Signal Processing Magazine*, vol. 26, no. 3, 2009.

[20] D. Gesbert, M. Kountouris, R. W. Heath Jr, C.-B. Chae, and T. Salzer, “Shifting the MIMO paradigm,” *IEEE signal processing magazine*, vol. 24, no. 5, pp. 36–46, 2007.

[21] E. G. Larsson, O. Edfors, F. Tufvesson, and T. L. Marzetta, “Massive MIMO for next generation wireless systems,” *IEEE Communications Magazine*, vol. 52, no. 2, pp. 186–195, 2014.

[22] C. Windpassinger, R. F. Fischer, and J. B. Huber, “Lattice-reduction-aided broadcast precoding,” *IEEE Transactions on Communications*, vol. 52, no. 12, pp. 2057–2060, 2004.

[23] C. B. Peel, B. M. Hochwald, and A. L. Swindlehurst, “A vector-perturbation technique for near-capacity multiantenna multiuser communication-part I: channel inversion and regularization,” *IEEE Transactions on Communications*, vol. 53, no. 1, pp. 195–202, 2005.

[24] B. M. Hochwald, C. B. Peel, and A. L. Swindlehurst, “A vector-perturbation technique for near-capacity multiantenna multiuser communication-part II: Perturbation,” *IEEE Transactions on Communications*, vol. 53, no. 3, pp. 537–544, 2005.

[25] C. Windpassinger, R. F. Fischer, T. Vencel, and J. B. Huber, “Precoding in multiantenna and multiuser communications,” *IEEE Transactions on Wireless Communications*, vol. 3, no. 4, pp. 1305–1316, 2004.

[26] K. Tan, H. Liu, J. Zhang, Y. Zhang, J. Fang, and G. M. Voelker, “Sora: high-performance software radio using general-purpose multi-core processors,” *Communications of the ACM*, vol. 54, no. 1, pp. 99–107, 2011.

[27] G. J. Foschini, “Layered space-time architecture for wireless communication in a fading environment when using multi-element antennas,” *Bell Labs Technical Journal*, vol. 1, no. 2, pp. 41–59, 1996.

[28] A. Paulraj, R. Nabar, and D. Gore, *Introduction to space-time wireless communications.* Cambridge university press, 2003.

[29] G. Caire, G. Taricco, and E. Biglieri, "Bit-interleaved coded modulation," *IEEE transactions on information theory*, vol. 44, no. 3, pp. 927–946, 1998.

[30] S. H. Muller-Weinfurtner, "Coding approaches for multiple antenna transmission in fast fading and OFDM," *IEEE transactions on signal processing*, vol. 50, no. 10, pp. 2442–2450, 2002.

[31] X. Li and J. A. Ritcey, "Bit-interleaved coded modulation with iterative decoding," *IEEE Communications Letters*, vol. 1, no. 6, pp. 169–171, 1997.

[32] "Mitsubishi Electric's New Multibeam Multiplexing 5G technology achieves 20Gbps throughput," [Online]. Available: http://www.mitsubishielectric.com/news/2016/0121.html., January 2016.

[33] F. Rusek, D. Persson, B. K. Lau, E. G. Larsson, T. L. Marzetta, O. Edfors, and F. Tufvesson, "Scaling up MIMO: Opportunities and challenges with very large arrays," *IEEE Signal Processing Magazine*, vol. 30, no. 1, pp. 40–60, 2013.

[34] L. Lu, G. Y. Li, A. L. Swindlehurst, A. Ashikhmin, and R. Zhang, "An overview of massive MIMO: Benefits and challenges," *IEEE Journal of Selected Topics in Signal Processing*, vol. 8, no. 5, pp. 742–758, 2014.

[35] T. L. Marzetta, "Massive MIMO: an introduction," *Bell Labs Technical Journal*, vol. 20, pp. 11–22, 2015.

[36] E. Björnson, E. G. Larsson, and T. L. Marzetta, "Massive MIMO: Ten myths and one critical question," *IEEE Communications Magazine*, vol. 54, no. 2, pp. 114–123, 2016.

[37] L. Bahl, J. Cocke, F. Jelinek, and J. Raviv, "Optimal decoding of linear codes for minimizing symbol error rate (corresp.)," *IEEE Transactions on information theory*, vol. 20, no. 2, pp. 284–287, 1974.

[38] L. Bai and J. Choi, *Low Complexity MIMO Detection.* Bostn,MA,USA: Springer, 2012.

[39] B. Sklar, *Digital communications.* Prentice Hall Upper Saddle River, 2001, vol. 2.

[40] A. Zanella, M. Chiani, and M. Z. Win, "MMSE reception and successive interference cancellation for MIMO systems with high spectral efficiency," *IEEE Transactions on Wireless Communications*, vol. 4, no. 3, pp. 1244–1253, 2005.

[41] T. Kailath, H. Vikalo, and B. Hassibi, "MIMO receive algorithms," *Space-Time Wireless Systems: From Array Processing to MIMO Communications*, 2005.

[42] M. Pohst, "On the computation of lattice vectors of minimal length, successive minima and reduced bases with applications," *ACM Sigsam Bulletin*, vol. 15, no. 1, pp. 37–44, 1981.

[43] U. Fincke and M. Pohst, "Improved methods for calculating vectors of short length in a lattice, including a complexity analysis," *Mathematics of computation*, vol. 44, no. 170, pp. 463–471, 1985.

[44] C.-P. Schnorr and M. Euchner, "Lattice basis reduction: Improved practical algorithms and solving subset sum problems," *Mathematical programming*, vol. 66, no. 1-3, pp. 181–199, 1994.

[45] O. Damen, A. Chkeif, and J.-C. Belfiore, "Lattice code decoder for space-time codes," *IEEE Communications letters*, vol. 4, no. 5, pp. 161–163, 2000.

[46] K. Su and I. J. Wassell, "A new ordering for efficient sphere decoding," in *Communications, 2005. ICC 2005. 2005 IEEE International Conference on*, vol. 3. IEEE, 2005, pp. 1906–1910.

[47] K. Su, "Efficient maximum likelihood detection for communication over multiple input multiple output channels," *Department of Engineering, University of Cambridge*, 2005.

[48] V. M. Garcia-Molla, A. M. Vidal, A. Gonzalez, and S. Roger, "Improved maximum likelihood detection through sphere decoding combined with box optimization," *Signal Processing*, vol. 98, pp. 284–294, 2014.

[49] Z. Guo and P. Nilsson, "Algorithm and implementation of the k-best sphere decoding for MIMO detection," *IEEE Journal on selected areas in communications*, vol. 24, no. 3, pp. 491–503, 2006.

[50] L. G. Barbero and J. S. Thompson, "Fixing the complexity of the sphere decoder for MIMO detection," *IEEE Transactions on Wireless Communications*, vol. 7, no. 6, 2008.

[51] B. M. Hochwald and S. Ten Brink, "Achieving near-capacity on a multiple-antenna channel," *IEEE transactions on communications*, vol. 51, no. 3, pp. 389–399, 2003.

[52] J. Hagenauer, E. Offer, and L. Papke, "Iterative decoding of binary block and convolutional codes," *IEEE Transactions on information theory*, vol. 42, no. 2, pp. 429–445, 1996.

[53] S. Baro, J. Hagenauer, and M. Witzke, "Iterative detection of MIMO transmission using a list-sequential (LISS) detector," in *Communications, 2003. ICC'03. IEEE International Conference on*, vol. 4. IEEE, 2003, pp. 2653–2657.

[54] P. Robertson, E. Villebrun, and P. Hoeher, "A comparison of optimal and sub-optimal MAP decoding algorithms operating in the log domain," in *Communications, 1995. ICC'95 Seattle,'Gateway to Globalization', 1995 IEEE International Conference on*, vol. 2. IEEE, 1995, pp. 1009–1013.

[55] G. Caire and S. Shamai, "On the achievable throughput of a multiantenna gaussian broadcast channel," *IEEE Transactions on Information Theory*, vol. 49, no. 7, pp. 1691–1706, 2003.

[56] M. Costa, "Writing on dirty paper (corresp.)," *IEEE transactions on information theory*, vol. 29, no. 3, pp. 439–441, 1983.

[57] R. Wang and G. B. Giannakis, "Approaching MIMO channel capacity with reduced-complexity soft sphere decoding," in *Wireless Communications and Networking Conference, 2004. WCNC. 2004 IEEE*, vol. 3. IEEE, 2004, pp. 1620–1625.

[58] S.-L. Shieh, R.-D. Chiu, S.-L. Feng, and P.-N. Chen, "Low-complexity soft-output sphere decoding with modified repeated tree search strategy," *IEEE Communications Letters*, vol. 17, no. 1, pp. 51–54, 2013.

[59] C. Studer, A. Burg, and H. Bolcskei, "Soft-output sphere decoding: Algorithms and VLSI implementation," *IEEE Journal on Selected Areas in Communications*, vol. 26, no. 2, 2008.

[60] L. G. Barbero, T. Ratnarajah, and C. Cowan, “A low-complexity soft-MIMO detector based on the fixed-complexity sphere decoder,” in *Acoustics, Speech and Signal Processing, 2008. ICASSP 2008. IEEE International Conference on.* IEEE, 2008, pp. 2669–2672.

[61] S. Chen, T. Zhang, and Y. Xin, “Relaxed k-best MIMO signal detector design and vlsi implementation,” *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 15, no. 3, pp. 328–337, 2007.

[62] X. Wu and J. S. Thompson, “A fixed-complexity soft-MIMO detector via parallel candidate adding scheme and its FPGA implementation,” *IEEE Communications Letters*, vol. 15, no. 2, pp. 241–243, 2011.

[63] M. S. Yee, “Max-log MAP sphere decoder,” in *Acoustics, Speech, and Signal Processing, 2005. Proceedings.(ICASSP’05). IEEE International Conference on*, vol. 3. IEEE, 2005, pp. iii–1013.

[64] M. Cirkic and E. G. Larsson, “SUMIS: Near-optimal soft-in soft-out MIMO detection with low and fixed complexity,” *IEEE Transactions on Signal Processing*, vol. 62, no. 12, pp. 3084–3097, 2014.

[65] M. R. Butler and I. B. Collings, “A zero-forcing approximate log-likelihood receiver for MIMO bit-interleaved coded modulation,” *IEEE Communications Letters*, vol. 8, no. 2, pp. 105–107, 2004.

[66] D. Seethaler, G. Matz, and F. Hlawatsch, “An efficient MMSE-based demodulator for MIMO bit-interleaved coded modulation,” in *Global Telecommunications Conference, 2004. GLOBECOM’04. IEEE*, vol. 4. IEEE, 2004, pp. 2455–2459.

[67] X.-W. Chang and Q. Han, “Solving box-constrained integer least squares problems,” *IEEE Transactions on wireless communications*, vol. 7, no. 1, 2008.

[68] M. Stojnic, H. Vikalo, and B. Hassibi, “Speeding up the Sphere Decoder with H^{∞} and SDP inspired lower bounds,” *IEEE Transactions on Signal Processing*, vol. 56, no. 2, pp. 712–726, 2008.

[69] D. Wubben, D. Seethaler, J. Jaldén, and G. Matz, “Lattice reduction,” *IEEE Signal Processing Magazine*, vol. 28, no. 3, pp. 70–91, 2011.

[70] A. K. Lenstra, H. W. Lenstra, and L. Lovász, "Factoring polynomials with rational coefficients," *Mathematische Annalen*, vol. 261, no. 4, pp. 515–534, 1982.

[71] H. Vetter, V. Ponnampalam, M. Sandell, and P. A. Hoeher, "Fixed complexity LLL algorithm," *IEEE Transactions on Signal Processing*, vol. 57, no. 4, pp. 1634–1637, 2009.

[72] C. Studer and H. Bolcskei, "Soft-input soft-output single tree-search sphere decoding," *IEEE Transactions on Information Theory*, vol. 56, no. 10, pp. 4827–4842, 2010.

[73] X. Wang and H. V. Poor, "Iterative (turbo) soft interference cancellation and decoding for coded CDMA," *IEEE Transactions on communications*, vol. 47, no. 7, pp. 1046–1061, 1999.

[74] C. Studer, S. Fateh, and D. Seethaler, "Asic implementation of soft-input soft-output MIMO detection using MMSE parallel interference cancellation," *IEEE Journal of Solid-State Circuits*, vol. 46, no. 7, pp. 1754–1765, 2011.

[75] M. Tuchler, A. C. Singer, and R. Koetter, "Minimum mean squared error equalization using a priori information," *IEEE Transactions on Signal processing*, vol. 50, no. 3, pp. 673–683, 2002.

[76] F. Domene, "Evaluation of precoding and feedback quatization schemes for multiuser communications systems." Ph.D. dissertation, Universitat Politécnica de Valéncia, 2015.

[77] Q. H. Spencer, A. L. Swindlehurst, and M. Haardt, "Zero-forcing methods for downlink spatial multiplexing in multiuser MIMO channels," *IEEE Transactions on Signal Processing*, vol. 52, no. 2, pp. 461–471, 2004.

[78] D. Xu, Y. Huang, and L. Yang, "Improved nonlinear multiuser precoding using lattice reduction," *Signal, image and video processing*, vol. 3, no. 1, pp. 47–52, 2009.

[79] S. Roger, A. Gonzalez, V. Almenar, and A. M. Vidal, "Extended LLL algorithm for efficient signal precoding in multiuser communication systems," *Communications Letters, IEEE*, vol. 14, no. 3, pp. 220–222, 2010.

[80] K. Zu, R. C. de Lamare, and M. Haardt, "Generalized design of low-complexity block diagonalization type precoding algorithms for multiuser MIMO systems," *IEEE Transactions on Communications*, vol. 61, no. 10, pp. 4232–4242, 2013.

[81] A. Jennings and J. J. McKeown, *Matrix computation.* John Wiley & Sons Inc, 1992.

[82] S. Roger, A. Gonzalez, V. Almenar, and A. Vidal, "Combined K-best sphere decoder based on the channel matrix condition number," in *Communications, Control and Signal Processing, 2008. ISCCSP 2008. 3rd International Symposium on.* IEEE, 2008, pp. 1058–1061.

[83] G. H. Golub and C. F. Van Loan, *Matrix computations.* JHU Press, 2012, vol. 3.

[84] J. E. Gentle, *Numerical linear algebra for applications in statistics.* Springer Science & Business Media, 2012.

[85] S. Roger, A. Gonzalez, V. Almenar, and A. M. Vidal, "MIMO channel matrix condition number estimation and threshold selection for combined K-best sphere decoders," *IEICE transactions on communications*, vol. 92, no. 4, pp. 1380–1383, 2009.

[86] A. K. Cline, C. B. Moler, G. W. Stewart, and J. H. Wilkinson, "An estimate for the condition number of a matrix," *SIAM Journal on Numerical Analysis*, vol. 16, no. 2, pp. 368–375, 1979.

[87] A. Edelman, "Eigenvalues and condition numbers of random matrices," *SIAM Journal on Matrix Analysis and Applications*, vol. 9, no. 4, pp. 543–560, 1988.

[88] Q. H. Spencer, A. L. Swindlehurst, and M. Haardt, "Zero-forcing methods for downlink spatial multiplexing in multiuser MIMO channels," *IEEE Transactions on Signal Processing*, vol. 52, no. 2, pp. 461–471, 2004.

[89] Q. H. Spencer and M. Haardt, "Capacity and downlink transmission algorithms for a multi-user MIMO channel," in *Signals, Systems and Computers, 2002. Conference Record of the Thirty-Sixth Asilomar Conference on*, vol. 2. IEEE, 2002, pp. 1384–1388.

[90] W. Li and M. Latva-aho, “An efficient channel block diagonalization method for generalized zero forcing assisted MIMO broadcasting systems,” *IEEE Transactions on Wireless Communications*, vol. 10, no. 3, pp. 739–744, 2011.

[91] E. Björnson, E. G. Larsson, and M. Debbah, “Massive MIMO for maximal spectral efficiency: How many users and pilots should be allocated?” *IEEE Transactions on Wireless Communications*, vol. 15, no. 2, pp. 1293–1308, 2016.

[92] S. Roger, C. Ramiro, A. Gonzalez, V. Almenar, and A. M. Vidal, “Fully parallel gpu implementation of a fixed-complexity soft-output mimo detector,” *IEEE Transactions on Vehicular Technology*, vol. 61, no. 8, pp. 3796–3800, 2012.

[93] M. Stojnic, H. Vikalo, and B. Hassibi, “Speeding up the Sphere Decoder with H^{∞} and SDP inspired lower bounds,” *IEEE Transactions on Signal Processing*, vol. 56, no. 2, pp. 712–726, 2008.

[94] M. Senst, G. Ascheid, and H. Lüders, “Performance evaluation of the markov chain monte carlo MIMO detector based on mutual information,” *Communications (ICC), 2010 IEEE International Conference on*, pp. 1–6, 2010.

[95] D. Wu, J. Eilert, and D. Liu, “Implementation of a high-speed MIMO soft-output symbol detector for software defined radio,” *Journal of Signal Processing Systems*, vol. 63, no. 1, pp. 27–37, 2011.

[96] K. Li, B. Yin, M. Wu, J. R. Cavallaro, and C. Studer, “Accelerating massive MIMO uplink detection on GPU for SDR systems,” *Circuits and Systems Conference (DCAS), 2015 IEEE Dallas*, pp. 1–4, 2015.

[97] G. Falcão, V. Silva, and L. Sousa, “How GPUs can outperform ASICs for fast LDPC decoding,” *Proceedings of the 23rd international conference on Supercomputing*, pp. 390–399, 2009.

[98] A. Li, R. G. Maunder, B. M. Al-Hashimi, and L. Hanzo, “Implementation of a fully-parallel turbo decoder on a general-purpose graphics processing unit,” *IEEE Access*, vol. 4, pp. 5624–5639, 2016.

[99] E. Anderson, Z. Bai, C. Bischof, L. S. Blackford, J. Demmel, J. Dongarra, J. Du Croz, A. Greenbaum, S. Hammarling, A. McKenney *et al.*, *LAPACK Users’ guide.* SIAM, 1999.

[100] C. L. Lawson, R. J. Hanson, D. R. Kincaid, and F. T. Krogh, “Basic linear algebra subprograms for fortran usage,” *ACM Transactions on Mathematical Software (TOMS)*, vol. 5, no. 3, pp. 308–323, 1979.

[101] *Intel MKL Reference Manual (2015) [Online]*, online at: https://software.intel.com/en-us/articles/mkl-reference-manual.

[102] *cuBLAS Documentation (2015) [Online]*, online at: http://docs.nvidia.com/cuda/cublas.

[103] L. Dagum and R. Enon, “OpenMP: an industry standard API for shared-memory programming,” *Computational Science & Engineering, IEEE*, vol. 5, no. 1, pp. 46–55, 1998.

[104] The 500 most powerful commercially available computer systems, “Top500.org,” online at: http://www.top500.org/lists/2014/11/, November 2014.

[105] *CUDA Toolkit Documentation, Version 7.5 (2015) [Online]*, online at: https://developer.nvidia.com/cuda-toolkit.

[106] P. W. Wolniansky, G. J. Foschini, G. Golden, and R. A. Valenzuela, “V-BLAST: An architecture for realizing very high data rates over the rich-scattering wireless channel,” in *Signals, Systems, and Electronics, 1998. ISSSE 98. 1998 URSI International Symposium on.* IEEE, 1998, pp. 295–300.