

Ester Pérez Adeva

Efficient MIMO Sphere Detection:
Algorithms and Architectures

Beiträge aus der Informationstechnik

Mobile Nachrichtenübertragung

Nr. 73

Ester Pérez Adeva

**Efficient MIMO Sphere Detection:
Algorithms and Architectures**

 VOGT

Dresden 2015

Bibliografische Information der Deutschen Nationalbibliothek
Die Deutsche Nationalbibliothek verzeichnet diese Publikation in der
Deutschen Nationalbibliografie; detaillierte bibliografische Daten sind im
Internet über <http://dnb.dnb.de> abrufbar.

Bibliographic Information published by the Deutsche Nationalbibliothek
The Deutsche Nationalbibliothek lists this publication in the Deutsche
Nationalbibliografie; detailed bibliographic data are available on the
Internet at <http://dnb.dnb.de>.

Zugl.: Dresden, Techn. Univ., Diss., 2015

Die vorliegende Arbeit stimmt mit dem Original der Dissertation
„Efficient MIMO Sphere Detection: Algorithms and Architectures“ von Ester
Pérez Adeva überein.

© Jörg Vogt Verlag 2015
Alle Rechte vorbehalten. All rights reserved.

Gesetzt vom Autor

ISBN 978-3-938860-90-8

Jörg Vogt Verlag
Niederwaldstr. 36
01277 Dresden
Germany

Phone: +49-(0)351-31403921
Telefax: +49-(0)351-31403918
e-mail: info@vogtverlag.de
Internet : www.vogtverlag.de

Technische Universität Dresden

Efficient MIMO Sphere Detection: Algorithms and Architectures

Ester Pérez Adeva

von der Fakultät Elektrotechnik und Informationstechnik
der Technischen Universität Dresden

zur Erlangung des akademischen Grades eines

Doktoringenieurs

(Dr.-Ing.)

genehmigte Dissertation

Vorsitzender: Prof. Dr. rer. nat. habil. Krauthäuser

Gutachter: Herr Prof. Dr.-Ing. Dr. h. c. Fettweis Tag der Einreichung: 04.05.2015
Herr Prof. Dr. Burg Tag der Verteidigung: 26.06.2015

Abstract

Modern wireless communications standards, such as IEEE 802.11 (WLAN), IEEE 802.16 (WiMAX), 3GPP-LTE or LTE-Advanced, define numerous transmission techniques to be supported by receiver architectures while satisfying a vast variety of stringent and most often conflicting requirements. The situation gets even more challenging with the extremely low latencies and high data rates (among other requirements) envisioned for future 5G technologies. In this context, three concepts will play a major role as **key enabling factors: adaptability, efficiency and performance**. In the particular context of multi-antenna spatial-multiplexing transmission, accurate detection/demodulation becomes one of the most computationally intensive processes at the receiver end. Designing efficient MIMO (multiple-input, multiple-output) detector realizations, capable of dynamically adapting to data-rate requirements, battery life, and varying channel conditions constitutes the main focus of this work. Developing adaptive, good-performing, and cost-effective MIMO detectors represents however a fairly challenging task. Ordinary low-complex approaches provide poor detection accuracy, whereas exhaustive search algorithms cannot achieve 4G/5G data rates with reasonable hardware complexity. In this regard, the so-called sphere detector, a tree-search-based detection technique, has arisen as the only approach capable of conveniently trading adaptability, efficiency and performance. Iterative detection-and-decoding enables a dramatic improvement of the communication's reliability by exchanging *soft information* between the detector and the decoder, at the cost of increasing the receiver's complexity. One of the major challenges in this context is represented by the distortive effect that the soft information causes on the search ordering, which represents a critical aspect affecting the detection accuracy and complexity of tree-search algorithms. In this regard, characterizing the influence of the soft information on the detection process and developing strategies which alleviate the inherent complexity and performance loss drawbacks constitute a major objective of this work. Among other strategies, a novel enumeration mechanism which determines the optimal symbol ordering while incurring a much lower computational effort than state-of-the-art approaches is proposed and evaluated. Besides addressing the mentioned challenges from the algorithmic perspective, developing suitable architectures which lead to efficient MIMO detector realizations represents an additional major goal. In order to satisfy the standard-defined high throughput requirements while complying with stringent area and power consumption restrictions (present in

e.g., pico/femtocell base stations, sensor networks or mobile terminals), the focus is laid onto the application-specific instruction-set processor (ASIP) paradigm. To assess the true hardware complexity and efficiency of the proposed VLSI designs, their physical characteristics are analyzed (including on-chip measurements) and compared to the state-of-the-art. Owing to the reported low resource-requirements and the high adaptability in terms of energy and performance, the proposed detector solutions are shown to be applicable to a wide variety of network elements and support numerous communications standards.

Acknowledgements

As it could not be otherwise, my first sign of appreciation and gratitude goes to the man whose academic title has become longer than his own name: Prof. Dr.-Ing. Dr. h. c. Gerhard P. Fettweis. It has been an honour learning from the best during the last five years, while having the opportunity to get involved in a colourful spectrum of activities and projects. I would like to thank him for a most enriching experience at the chair, for the fruitful technical discussions, and for his unconditional support and trust. Thank you, Gerhard, for challenging us so that we grow (scientifically) stronger. The second great contributor to my work is Dr.-Ing. Björn Mennenga, who back in 2008 gave me the chance to join the MNS Chair as a diplomate. I am enormously thankful for his guidance and advice during my first Ph.D. years, as well as for his constructive comments and for the many, many, many, technical discussions we had. I must of course thank our group leader Emil Matus as well as the past and present CATS team, for the productive technical discussions as well as for the after-work leisure. I am also grateful to all students who contributed my work with their respective master theses, student projects and research assistance. Thank you also to Steffen, Kathrin, Sylvia, Rue and Raff, who ensure the chair-engine keeps functioning. Likewise, I would like to thank all the great professionals from industry and academia whom I had the opportunity to meet during the organization of the DATE 2012 and DATE 2014 conferences in Dresden. In this regard, especial thanks to Franziska Kremling and Jörg Herrmann (K.I.T.), for their support, for the very nice cooperation and for the positive feedback they provided. And last but not least, a giant thank you to my family, who support me from the distance and suffer my absence in silence. *Por último, pero no por ello menos importante, gracias a mi familia, quien me apoya desde la distancia y sufre mi larga ausencia en silencio. A mis padres, porque estoy donde estoy gracias a vosotros. A mi hermano y alcahueta, por pelearnos de pequeños pero querernos de mayores.*

To my statistically significant other, because his math is irrefutable.

Contents

Abstract/Kurzfassung	vii
Acknowledgements	ix
List of Figures	xvii
List of Tables	xviii
Acronyms	xix
Notation and Symbols	xxiv
1 Introduction	1
1.1 Motivation and Objectives	1
1.2 Outline	3
2 Fundamentals of MIMO Communications	5
2.1 MIMO Approaches	5
2.2 MIMO Detection for Spatial Multiplexing	7
2.2.1 Maximum Likelihood and <i>a Posteriori</i> Probability Detection	9
2.2.2 (Approximated) Logarithmic MAP Detection	10
2.2.3 Linear Detection	11
2.2.4 Interference Cancellation Detection	13
2.2.5 Tree-Search Detection	15
2.3 Tree Search Detection: Algorithms and Strategies for Complexity Reduction	17
2.3.1 Tree Traversal Strategies	17
2.3.2 Tuning the Error-Rate-Complexity Trade-off at Pre-Processing	20
2.3.3 Tuning the Error-Rate-Complexity Trade-off at Runtime	22
2.4 The Reduced-Complexity Sphere Detector	24
2.4.1 Estimating the Node Enumeration Sequence	25
2.4.2 Parallel Sibling Node Processing	28

2.4.3	Estimating the Metrics	28
2.5	Complexity, Performance and Efficiency Metrics	29
2.5.1	On the Algorithmic Perspective	30
2.5.2	On the Integrated Circuits Perspective	34
3	Soft-Input Soft-Output Sphere Detection	41
3.1	State-of-the-Art and Challenges	41
3.2	<i>A Priori</i> Information: Ally or Enemy?	43
3.2.1	The (not-strictly-Gaussian) Distribution of the <i>A Priori</i> Information	45
3.2.2	Effect of the <i>A Priori</i> Information on the Partial Metrics	49
3.3	Enumeration Strategies	52
3.3.1	The <i>Min-Search</i> Approach	52
3.3.2	<i>The Adaptive-Hypothesis</i> Approach	53
3.3.3	Window-based Enumeration	54
3.3.4	Smart-Sorting Enumeration with Quadrature Metric Computation .	56
4	Performance Analysis of the Sphere Detector	63
4.1	Communications System: Single-Carrier BICM System	63
4.2	SNR-Complexity Trade-off and Search Efficiency	64
4.3	Scalability	67
4.4	Performance in Iterative Detection-and-Decoding Schemes	69
4.4.1	Performance of the SSD Enumeration Approach	70
4.4.2	Performance of the Window-Based Enumeration Approach	70
4.4.3	Performance of the Smart-Sorting Enumeration Approach	74
4.5	Summary	74
5	Architectures for Sphere Detection	79
5.1	Overview: The Architecture Design Space	80
5.2	Towards an Efficient TSD Implementation	81
5.2.1	Fixed-Point Arithmetic	81
5.2.2	Algorithm Partitioning and Regularization	82
5.2.3	Task Scheduling and Parallelization Strategies	85
5.3	The TSD's Architecture	87
5.3.1	Data and Control Path	89
5.3.2	Memory Organization and Control Flow	90
5.3.3	The TSD-SSD-ME MIMO Detection Module	94
5.3.4	The TSD-SSE-QMC MIMO Detection Module	103

6	Implementation Results and Analysis	109
6.1	The Design Toolflow	109
6.2	VLSI Implementation of the TSD-SSD-ME	110
6.2.1	Area	110
6.2.2	Power Consumption	111
6.2.3	Delay and Throughput	112
6.2.4	Energy	113
6.2.5	Summary	114
6.3	VLSI Implementation of the TSD-SSD-MC	115
6.4	VLSI Implementation of the TSD-SSE-QMC	117
6.4.1	Area	117
6.4.2	Delay	118
6.4.3	Throughput and Energy	119
6.4.4	Summary	119
6.5	Architecture Optimization: Pipelining and Retiming	120
6.6	TSD-SSD-ME in Silicon: The “Tommy” and “Tomahawk2” MPSoCs	123
6.6.1	On the Energy Efficiency (I): Voltage and Frequency Scaling	126
6.6.2	On the Energy Efficiency (II): The Multi-Core Approach	127
6.6.3	Comparison to the State-of-the-Art	131
6.7	Summary	139
7	Summary	143
7.1	Conclusions and Contributions	143
7.2	Outlook	145
	Bibliography	163

List of Figures

2.1	Tree search example for a BPSK modulation ($L = 1$) and $N_T = N_R = 4$ antennas. Dashed lines represent pruned paths.	16
2.2	Search sequence determination approaches (64-QAM).	27
2.3	Comparison of sequential and parallel sibling node processing. Dashed lines represent not-yet-examined paths.	28
2.4	Geometrical distances between constellation symbols and the centroid of one of the triangular decision regions.	29
3.1	Example illustrating the interpretation of the <i>a priori</i> contribution to the metrics as a geometrical distance on a 4-QAM constellation.	44
3.2	Distribution of $ L_a $ and estimated probability density functions for different E_b/N_0 and number of detection-and-decoding iterations.	48
3.3	Squared Euclidean distances, sorted in ascending order, for a sample received 64-QAM-symbol.	49
3.4	Partial metrics and contribution of the a priori information, computed for different certainties of the decoded bits. The values are sorted according to the enumeration sequence depicted in Figure 3.3(a).	51
3.5	Examples of the min-search and the adaptive-hypothesis approaches on a 64-QAM constellation.	53
3.6	Symbol ordering resulting from the window-based enumeration approach. The first 22 partial metrics have been sorted using a window size $W = 8$	55
3.7	Sample partial metrics $\lambda_{(i)}$ before and after sorting their quadrature components, corresponding to the symbols $\hat{x}_i^{(k^{\text{st}}, k^{\text{nd}})}$ of a 16-QAM constellation. The gradient color scale varies according to the magnitude of the metric values, to ease visualizing the metrics' ideal ordering.	57

3.8	Smart-sorting enumeration procedure, applied to partial metrics from the example in Figure 3.7. The rightmost tables depict the stack's content. The values in shaded cells correspond to symbols which have been examined during the tree search. The values in black-bordered cells represent the new metrics added to the stack on each step. The symbol selected to be next explored is denoted by bold fonts.	59
4.1	MIMO communications system model with a BICM transmitter and a turbo receiver.	65
4.2	Comparison of $\mathbb{E}[n]$ and ∂ corresponding to $\text{BER} = 10^{-5}$, for different sphere detection approaches. A 64-QAM constellation and a 4×4 MIMO system is considered ($I = 1$). The value n_{min} required by the STS approach is additionally illustrated (dashed line).	66
4.3	Comparison of $\mathbb{E}[n]$ corresponding to $\text{BER} = 10^{-5}$, for different detection approaches, modulation schemes and MIMO antenna configurations ($I = 1$).	68
4.4	Performance comparison of different detection strategies in an iterative receiver with $I = 4$ detection-and-decoding iterations (4×4 MIMO, 64-QAM).	71
4.5	Performance comparison of the TSD approach with window-based enumeration, for different window sizes ($W \in \{3, 5, 7, 9, 12, 15\}$) and number of detection-and-decoding iterations I (4×4 MIMO, 64-QAM).	73
4.6	Node count $\mathbb{E}[n]$ per turbo iteration, corresponding to the TSD-SSE-QMC algorithm with and without internal radius clipping ($T = 16$, 4×4 MIMO, 64-QAM).	75
5.1	Qualitative visualization of the architecture design space trade-offs in terms of area and energy costs, flexibility and performance.	79
5.2	TSD regularized flow diagram.	84
5.3	TSD's task scheduling and pipeline-interleaving schemes.	84
5.4	Synchronous Transfer Architecture (STA) template.	89
5.5	Block diagram of the MIMO detector ASIP architecture.	90
5.6	Extended control flow and pipeline-interleaving schemes, including data memory access.	94
5.7	Block diagram of the TSD-SSD-ME's architecture, including pipeline registers. The depicted modules correspond to the task blocks defined in Figure 5.2.	95
5.8	Block diagram of the TSD-SSD-ME's node enumeration unit (NEU) architecture.	96
5.9	Block diagram of the TSD-SSD-ME's metric computation unit (MCU) architecture.	97

5.10	Block diagram of the TSD-SSD-ME's interference reduction unit (IRU) architecture.	97
5.11	Block diagram of the TSD-SSD-ME's radius administration unit (RAU) (naive) architecture.	98
5.12	Block diagram of the TSD-SSD-ME's radius administration unit (RAU) architecture.	98
5.13	Block diagram of the TSD-SSD-ME's layer determination unit (LDU) architecture.	99
5.14	Block diagram of the TSD-SSD-ME's soft-output generation unit (SOGU) architecture computing the bit-specific metrics.	102
5.15	Block diagram of the TSD-SSD-ME's soft-output generation unit (SOGU) architecture updating the bit-specific metrics.	103
5.16	Block diagram of the TSD-SSE-QMC's architecture, including pipeline registers. The depicted modules correspond to the task blocks defined in Figure 5.2.	104
5.17	Architecture of the SSE-QMC's metric computation and node enumeration approaches.	107
6.1	Area breakdown of the SO and SISO TSD-SSD-ME designs (data path) synthesized for 450 MHz (excluding memories).	111
6.2	Power breakdown of the SO and SISO TSD-SSD-ME designs (data path) at 450 MHz (excluding memories).	112
6.3	Delay of the SISO TSD with search-sequence determination and metric estimation (SSD-ME, dark-coloured), and the SISO TSD with smart-sorting enumeration and quadrature metric computation (SSE-QMC, light coloured). Each of the modules has been synthesized for its maximum frequency (under typical-case operating conditions).	113
6.4	TSD's average throughput τ and energy-per-bit for different tuple sizes T (under typical case operating conditions, $f_{\text{clk}} = 450\text{MHz}$, $I = 1$, 10^{-5} BER).	114
6.5	Area comparison of the MC, ME, SSD-ME and SSE-QMC modules (synthesized under typical case operating conditions). *The area of the SOGU block corresponds to the logic required exclusively for the computation (estimation) of the Euclidean distances according to equation (2.33).	116
6.6	Area-delay diagram of the SISO TSD-SSD-ME design with different number of pipeline stages (TSMC 65-nm libraries, typical case operating conditions).	122
6.7	Top-level diagram and die photo of the Tomahawk2 MPSoC.	124
6.8	Top-level diagram of the TSD module on the Tomahawk2 MPSoC.	125
6.9	Energy efficiency of the TSD core operating on the Tomahawk2 MPSoC.	128

6.10 On-chip measurement results of power consumption and maximum achievable frequency at different supply voltages. 129

6.11 Total power consumption of a multicore TSD design as a function of the processing throughput (applying voltage scaling). 131

6.12 Total power and energy consumption of the multicore-TSD design, applying AVFS to sustain a constant throughput while minimizing power consumption. 132

List of Tables

2.1	General CMOS technology scaling model for short-channel devices (with transistor gate lengths below $1\mu m$) [1].	37
3.1	Comparison of the computational complexity (in number of operations) of the Schnorr-Euchner enumeration (SE) with standard metric computation (MC) and smart-sorting enumeration (SSE) with quadrature metric computation (QMC). To determine the compare and accumulate operation count, the fast sorter architecture described in section 5.3.4 has been assumed. . .	61
5.1	Memory specifications for $N_T = N_R = 4$ and $Q = 64$ ($F = 40, q = 1$) and size dependency on N_T, L, F and q . 8-bit fixed-point precision is used except for the <i>a priori</i> information, represented with 5-bit precision instead.	91
6.1	Area breakdown of the SO and SISO TSD-SSD-ME design synthesized for 450 MHz.	110
6.2	Total power consumption of SO and SISO TSD under typical case operating conditions, at 450 MHz.	111
6.3	Comparison of pre-layout synthesis and power estimation results (under typical-case operating conditions) with respect to on-chip measurements of the TSD-core's top-level.	126
6.4	Comparison of MIMO detectors on CMOS (values in brackets result from technology scaling to 65nm).	134

Acronyms

Acronym	Meaning
3G	<u>3</u> rd <u>G</u> eneration (mobile communications systems)
3GPP	<u>3</u> rd <u>G</u> eneration <u>P</u> artnership <u>P</u> roject
4G	<u>4</u> th <u>G</u> eneration (mobile communications systems)
5G	<u>5</u> th <u>G</u> eneration (mobile communications systems)
ADPLL	<u>A</u> ll- <u>D</u> igital <u>P</u> hase- <u>L</u> ocked <u>L</u> oop
AGU	<u>A</u> ddress <u>G</u> eneration <u>U</u> nit
Alg	<u>A</u> lgorithm
AH	<u>A</u> daptive <u>H</u> ypothesis
APP	<u>A</u> <u>P</u> osteriori <u>P</u> robability
ARQ	<u>A</u> utomatic <u>R</u> epeat <u>r</u> e <u>Q</u> uest
ASIC	<u>A</u> pplication- <u>S</u> pecific <u>I</u> ntegrated <u>C</u> ircuit
ASIP	<u>A</u> pplication- <u>S</u> pecific <u>I</u> nstruction Set <u>P</u> rocessor
AT(E)	<u>A</u> rea- <u>T</u> ime(- <u>E</u> nergy) product
AVS	<u>A</u> daptive <u>V</u> oltage <u>S</u> caling
AVFS	<u>A</u> daptive <u>V</u> FS
AWGN	<u>A</u> dditive <u>W</u> hite <u>G</u> aussian <u>N</u> oise
b	<u>b</u> it
B	<u>B</u> yte
BCJR	<u>B</u> ahl- <u>C</u> ocke- <u>J</u> elinek- <u>R</u> aviv
BER	<u>B</u> it <u>E</u> rror <u>R</u> ate
BF	<u>B</u> readth- <u>F</u> irst (tree search)
BeF	<u>B</u> est- <u>F</u> irst (tree search)
BICM	<u>B</u> it- <u>I</u> nterleaved <u>C</u> oded <u>M</u> odulation
BLER	<u>B</u> lock <u>E</u> rror <u>R</u> ate
bps	<u>b</u> its <u>p</u> er <u>s</u> econd
BPSK	<u>B</u> inary PSK
BW	<u>B</u> and <u>W</u> idth

Acronym	Meaning
CDF	<u>C</u> umulative <u>D</u> istribution <u>F</u> unction
clip	<u>cl</u> ipping
clk	<u>cl</u> ock
CMOS	<u>C</u> omplementary <u>M</u> etal- <u>O</u> xide- <u>S</u> emiconductor
CoMP	<u>C</u> oordinated <u>M</u> ulti <u>P</u> oint
cp	<u>cr</u> itical <u>p</u> ath
dB	<u>de</u> ci <u>B</u> el
Dec	<u>D</u> ecoder
Det	<u>D</u> etector
DF	<u>D</u> ept <u>H</u> - <u>F</u> irst (tree search)
DFE	<u>D</u> ecision <u>F</u> eedback <u>E</u> qualization
DSP	<u>D</u> igital <u>S</u> ignal <u>P</u> rocessor
DVFS	<u>D</u> ynamic <u>V</u> FS
FCU	<u>F</u> low <u>C</u> ontrol <u>U</u> nit
FEC	<u>F</u> orward <u>E</u> rror <u>C</u> orrection
FER	<u>F</u> rame <u>E</u> rror <u>R</u> ate
FFT	<u>F</u> ast <u>F</u> ourier- <u>T</u> ransformation
FIFO	<u>F</u> irst- <u>I</u> n <u>F</u> irst- <u>O</u> ut
FLOPS	<u>F</u> loating-point <u>O</u> perations <u>P</u> er <u>S</u> econd
FP	<u>F</u> incke- <u>P</u> ohst
FPGA	<u>F</u> ield <u>P</u> rogrammable <u>G</u> ate <u>A</u> rray
FSD	<u>F</u> ixed Complexity <u>S</u> phere <u>D</u> etector
FU	<u>F</u> unctional <u>U</u> nit
GE	<u>G</u> ate <u>E</u> quivalent
(GP)GPU	(<u>G</u> eneral <u>P</u> urpose) <u>G</u> raphical <u>P</u> rocessing <u>U</u> nit
GPP	<u>G</u> eneral <u>P</u> urpose <u>P</u> rocessor
HDL	<u>H</u> ardware <u>D</u> escription <u>L</u> anguage
HO	<u>H</u> ard <u>O</u> utput
Hz	<u>H</u> ertz
IC	<u>I</u> ntegrated <u>C</u> ircuit
i.i.d.	<u>i</u> ndependent and <u>i</u> dentically <u>d</u> istributed
IEEE	<u>I</u> nstitute of <u>E</u> lectrical and <u>E</u> lectronics <u>E</u> ngineers, Inc.
IMEM	<u>I</u> nstruction <u>M</u> EMory
IRU	<u>I</u> nterference <u>R</u> eduction <u>U</u> nit
ISS	<u>I</u> nstruction <u>S</u> et <u>S</u> imulator
If-CU	<u>I</u> nterface <u>C</u> ontroller <u>U</u> nit

Acronym	Meaning
IFFT	<u>I</u> nverse <u>F</u> ast <u>F</u> ourier- <u>T</u> ransformation
ISI	<u>I</u> nter <u>S</u> ymbol- <u>I</u> nterference
J	<u>J</u> oule
LD	<u>L</u> inear <u>D</u> etection
LDPC	<u>L</u> ow- <u>D</u> ensity <u>P</u> arity- <u>C</u> heck (code)
LDU	<u>L</u> ayer <u>D</u> etermination <u>U</u> nit
LISS	<u>L</u> ist- <u>S</u> equential Detector
LLL	<u>L</u> enstra- <u>L</u> enstra- <u>L</u> ovasz
LLR	<u>L</u> og- <u>L</u> ikelihood <u>R</u> atio
LR	<u>L</u> attice <u>R</u> eduction
LSD	<u>L</u> ist <u>S</u> phere <u>D</u> etector
LTE(-A)	<u>L</u> ong <u>T</u> erm <u>E</u> volution (<u>A</u> dvanced)
LUT	<u>L</u> ook- <u>U</u> p <u>T</u> able
L-value	<u>LLR</u> <u>value</u>
M2M	<u>M</u> achine <u>t</u> o <u>M</u> achine
MAP	<u>M</u> aximum <u>A</u> <u>P</u> osteriori Probability
maxLogAPP	<u>max-Log</u> approximation of <u>APP</u>
MC	<u>M</u> etric (exact) <u>C</u> omputation
MCU	<u>M</u> etric <u>C</u> omputation <u>U</u> nit
ME	<u>M</u> etric <u>E</u> stimation
MF	<u>M</u> etric- <u>F</u> irst (tree search)
MIMO	<u>M</u> ultiple <u>I</u> nput <u>M</u> ultiple <u>O</u> utput
MIPS	<u>M</u> ega <u>I</u> nstructions <u>P</u> er <u>S</u> econd
MISO	<u>M</u> ultiple <u>I</u> nput <u>S</u> ingle <u>O</u> utput
ML	<u>M</u> aximum <u>L</u> ikelihood
MMSE	<u>M</u> inimum <u>M</u> ean <u>S</u> quare <u>E</u> rror
MOPS	<u>M</u> illion <u>O</u> perations <u>P</u> er <u>S</u> econd
MPSoC	<u>M</u> ulti <u>P</u> rocessor <u>S</u> oC
MS	<u>M</u> in- <u>S</u> earch
MSE	<u>M</u> ean <u>S</u> quare <u>E</u> rror
MUX	<u>M</u> ultiple <u>X</u> er
NEU	<u>N</u> ode <u>E</u> numeration <u>U</u> nit
NoC	<u>N</u> etwork- <u>o</u> n- <u>C</u> hip
NoC-IF	<u>NoC</u> - <u>I</u> nter <u>F</u> ace
OSIC	<u>O</u> rdered <u>S</u> IC
OFDM	<u>O</u> rthogonal <u>F</u> requency <u>D</u> ivision <u>M</u> ultiplexing

Acronym	Meaning
PAM	<u>P</u> ulse- <u>A</u> mplitude <u>M</u> odulation
PCCC	<u>P</u> arallel <u>C</u> oncatenated <u>C</u> onvolutional <u>C</u> ode
PDF	<u>P</u> robability <u>D</u> ensity <u>F</u> unction
PE	<u>P</u> rocessing <u>E</u> lement
PER	<u>P</u> acket <u>E</u> rror <u>R</u> ate
PIC	<u>P</u> arallel <u>I</u> nterference <u>C</u> ancellation
PSA	<u>P</u> ost- <u>S</u> orting <u>A</u> lgorithm
PSK	<u>P</u> hase- <u>S</u> hift <u>K</u> eys
QAM	<u>Q</u> uadrature <u>A</u> mplitude <u>M</u> odulation
QMC	<u>Q</u> uadrature <u>M</u> etric <u>C</u> omputation
QPSK	<u>Q</u> uadrature PSK
QRD	<u>Q</u> R <u>D</u> ecomposition
RAU	<u>R</u> adius <u>A</u> dmistration <u>U</u> nit
RISC	<u>R</u> educed <u>I</u> nstruction- <u>S</u> et <u>C</u> omputing
RTL	<u>R</u> egister- <u>T</u> ransfer <u>L</u> evel
RTS	<u>R</u> epeated <u>T</u> ree <u>S</u> earch Detector
SMEM	<u>S</u> calar <u>M</u> EMory
SD	<u>S</u> phere <u>D</u> etector
SDR	<u>S</u> oftware- <u>D</u> efined <u>R</u> adio
SE	<u>S</u> chnorr- <u>E</u> uchner (enumeration)
SIC	<u>S</u> uccessive <u>I</u> nterference <u>C</u> ancellation
SIMD	<u>S</u> ingle- <u>I</u> nstruction <u>M</u> ultiple- <u>D</u> ata
SIMO	<u>S</u> ingle <u>I</u> nput <u>M</u> ultiple <u>O</u> utput
SISO	<u>S</u> oft- <u>I</u> nput <u>S</u> oft- <u>O</u> utput
SINR	<u>S</u> ignal to <u>I</u> nterference and <u>N</u> oise <u>R</u> atio
SNR	<u>S</u> ignal to <u>N</u> oise <u>R</u> atio
SO	(<u>H</u> ard- <u>I</u> nput) <u>S</u> oft- <u>O</u> utput
SoC	<u>S</u> ystem- <u>o</u> n- <u>C</u> hip
SOGU	<u>S</u> oft- <u>O</u> utput <u>G</u> eneration <u>U</u> nit
(S)RAM	(<u>S</u> tatic) <u>R</u> andom- <u>A</u> ccess <u>M</u> emory
SSD	<u>S</u> earch <u>S</u> equence <u>D</u> etermination
SSE	<u>S</u> mart <u>S</u> orting <u>E</u> numeration
SQRD	<u>S</u> orted <u>Q</u> R <u>D</u>
STA	<u>S</u> ynchronous <u>T</u> ransfer <u>A</u> rchitecture
STS	<u>S</u> ingle <u>T</u> ree <u>S</u> earch Detector
TSD	<u>T</u> uple- <u>S</u> earch <u>S</u> phere <u>D</u> etector

Acronym	Meaning
uB	<u>un</u> Biased
UMTS	<u>U</u> niversal <u>M</u> obile <u>T</u> elecommunications <u>S</u> ystem
V	<u>V</u> olt
V-BLAST	<u>V</u> ertical- <u>B</u> ell <u>L</u> aboratories <u>L</u> Ayered <u>S</u> pace <u>T</u> ime
VFS	<u>V</u> oltage and <u>F</u> requency <u>S</u> caling
VLAMEM	<u>V</u> ector <u>L</u> _a <u>M</u> EMory
VLIW	<u>V</u> ery <u>L</u> ong <u>I</u> nstruction <u>W</u> ord
VLSI	<u>V</u> ery <u>L</u> arge <u>S</u> cale <u>I</u> ntegration
VMEMI	<u>V</u> ector <u>M</u> EMory <u>I</u> nput
VMEMO	<u>V</u> ector <u>M</u> EMory <u>O</u> utput
VLIW	<u>V</u> ery <u>L</u> ong <u>I</u> nstruction <u>W</u> ord
W	<u>W</u> att
WE	<u>W</u> indows-Based <u>E</u> numeration
WiMAX	<u>W</u> orldwide <u>i</u> nteroperability for <u>M</u> icrowave <u>A</u> ccess
WLAN	<u>W</u> ireless <u>L</u> ocal <u>A</u> rea <u>N</u> etwork
ZF	<u>Z</u> ero- <u>F</u> orcing

Notation and Symbols

Operators and Functions

$\mathbb{C}^{n \times m}$	set of complex numbers with dimensions $n \times m$
$\mathbb{N}^{n \times m}$	set of natural numbers with dimensions $n \times m$
$\mathbb{R}^{n \times m}$	set of real numbers with dimensions $n \times m$
\mathbb{Z}	set of integer numbers
\approx	approximately equal
\neq	not equal
\cong	equal or nearly equal
$:=$	definition
\forall	for all
\in	element of
\subseteq	subset
\sum	summation of values
\prod	product of values
Π	interleaving
Π^{-1}	inverse interleaving
$\{\cdot\}$	set of elements
$(\cdot \cdot)$	such that; conditional event
$\lfloor \cdot \rfloor_x$	round (\cdot) to the nearest $x \in \mathcal{X}$
$ \cdot $	absolute value of (\cdot) (absolute-value norm)
$\ \cdot\ _1$	l^1 -norm (Manhattan norm)
$\ \cdot\ _2, \ \cdot\ $	l^2 -norm (Euclidean norm)
$\ \cdot\ _\infty$	l^∞ -norm (maximum norm)
$\ \cdot\ _p$	p -norm
$(\cdot)^T$	transpose of a matrix
$(\cdot)^H$	conjugate transpose of a matrix
$(\cdot)^{-1}$	inverse of a matrix
$(\cdot) \rightarrow x$	the value of (\cdot) tends to x
$\#(\cdot)$	cardinality (i.e. number of elements) of the set (\cdot)

Δ	difference, change in quantity
$\arg \max_{\mathbf{x}} (\cdot)$	argument of the maximum of (\cdot) over the set of points \mathbf{x}
$\arg \min_{\mathbf{x}} (\cdot)$	argument of the minimum of (\cdot) over the set of points \mathbf{x}
$d(\cdot)$	Euclidean distance
$\text{demap}(\mathbf{x})$	Gray-demapping of the constellation symbol vector \mathbf{x}
$\mathbb{E}(\cdot)$	expectation of (\cdot)
$\varepsilon(\cdot)$	error of (\cdot)
$\exp(\cdot)$	natural exponential function (i.e., with base e)
$f(\cdot)$	probability density function (pdf) of (\cdot)
$F(\cdot)$	cumulative distribution function (cdf) of (\cdot)
$\Im(\cdot)$	imaginary part of (\cdot)
$\text{ld}(\cdot)$	logarithm of (\cdot) to base 2
$\text{lg}(\cdot)$	logarithm of (\cdot) to base 10
$\ln(\cdot)$	logarithm of (\cdot) to base e (natural logarithm)
$\text{map}(\mathbf{c})$	Gray-mapping of \mathbf{c} to a constellation symbol vector
$\max_{\mathbf{x}}(\cdot)$	maximum of (\cdot) over the set of points \mathbf{x}
$\min_{\mathbf{x}}(\cdot)$	minimum of (\cdot) over the set of points \mathbf{x}
$P(\cdot)$	probability function of event (\cdot)
$\Re(\cdot)$	real part of (\cdot)
$\text{sign}(\cdot)$	sign of (\cdot)

Symbols

$\mathbf{0}_{n \times m}$	null matrix with dimensions $n \times m$
a	distance between constellation symbols
A	area of an integrated circuit
A_{GE}	area expressed in gate-equivalents
A_{NAND}	area occupied by a single two-input drive-one NAND standard cell
b	bit value ($= \pm 1$)
B	signal bandwidth
$c_{m,l}$	l -th bit sent by the m -th transmit antenna ($c_{m,l} = \pm 1$)
\mathbf{c}'	vector of coded information bits
\mathbf{c}''	vector of bits composing the transmitted symbol vector \mathbf{x} (vector of coded and interleaved information bits, partitioned into blocks of size $N_{\text{T}}L$)
\mathbf{c}	vector of bits corresponding to the estimated symbol vector $\hat{\mathbf{x}}$
\mathbf{c}_t	vector of bits corresponding to the symbol vector $\hat{\mathbf{x}}$ occupying position t within the search tuple \mathcal{T} (tuple-search sphere detector)
C	total effective capacitance of a circuit
C_{gate}	gate capacitance (integrated circuits)
C_{sc}	short-circuit capacitance (integrated circuits)
d_i^p	abbreviation of $d(y_i''', x_i^p)$
$d_i^{p'}$	abbreviation of $d(z^{\text{ref}}, x_i^p)$
$d(y_i''', x_i^p)$	Euclidean distance between y_i''' and a particular constellation symbol x_i occupying position p in the enumeration sequence
$d(z^{\text{ref}}, x_i^p)$	Euclidean distance between a particular reference point z^{ref} within the constellation and the symbol x_i occupying position p in the enumeration sequence (approximation of $d(y_i''', x_i^p)$)
d_a	geometrical representation of the contribution of the <i>a priori</i> information to the metric values as an additive distance
D_j^i	i -th cluster of parallel operations of the j -th detection path
e	Euler's number ($\approx 2, 7182818$)
E	energy dissipated by a circuit
E_b	average received bit-energy (mobile communications)
E_s	average transmitted symbol energy (mobile communications)
$\mathbb{E}[n]$	average number of nodes sequentially examined by a tree-search algorithm
f, f', s_f, s'_f	scaling factors for fixed point arithmetic, required by diverse algorithm variables

f_{clk}	clock frequency of an integrated circuit
$f_{\text{clk}}^{\text{max}}$	maximum clock frequency achieved by an integrated circuit to guarantee error-free operation
f_s	sampling frequency
F	buffer (memory) size (in number of words)
\mathbf{G}	filter matrix
$\mathbf{G}_{\text{MMSE}},$ $\mathbf{G}'_{\text{MMSE}}$	MMSE-filter matrices
$\mathbf{G}_{\text{ZF}}, \mathbf{G}'_{\text{ZF}}$	ZF-filter matrices
\mathbf{H}	channel matrix with complex-valued entries h_{ij}
i	tree layer ($i \in \{1, 2, \dots, N_{\text{R}}\}$)
I	number of iterating detection-decoding runs
I_{leak}	leakage current (integrated circuits)
\mathbf{I}_n	identity matrix with dimensions $n \times n$
I_{sat}	saturation current (integrated circuits)
k^{dyn}	proportionality constant for the dynamic power estimation (integrated circuits)
k^{stat}	proportionality constant for the static power estimation (integrated circuits)
k^{J}	constellation symbol index along the imaginary quadrature component
k^{R}	constellation symbol index along the real quadrature component
K	size of the candidate list (list sphere detector); number of survival paths (K-best detector)
K_i	number of survival paths at each layer i (modified K-best detector)
l_{cp}	loop latency (in clock cycles) of the algorithm's critical path
L	modulation order (number of bits representing each constellation symbol); transistor channel length (integrated circuits)
$L(\cdot)$	<i>a posteriori</i> information
$L^{\text{Dec}}(\cdot)$	<i>a posteriori</i> information from the decoder
$L^{\text{Det}}(\cdot)$	<i>a posteriori</i> information from the detector
$L_{\text{a}}(\cdot)$	<i>a priori</i> information
$L_{\text{a}}^{\text{max}}$	maximum value of the <i>a priori</i> information
$\mathbf{L}_{\text{a}}(\cdot)$	vector of <i>a priori</i> information
$L_{\text{a}}^{\text{Dec}}(\cdot)$	<i>a priori</i> information from decoder
$L_{\text{a}}^{\text{Det}}(\cdot)$	<i>a priori</i> information from detector
L_{a}^{TH}	threshold for the <i>a priori</i> information (bit-pruning approach)
L^{clip}	clipping factor

$L_e(\cdot)$	extrinsic information
$L_e^{\text{Dec}}(\cdot)$	extrinsic information from decoder
$L_e^{\text{Det}}(\cdot)$	extrinsic information from detector
$L_e^{\text{clip}}(\cdot)$	extrinsic information clipped with L^{clip}
L_e^{max}	threshold for the clipping factor
M	number of survival paths (M-algorithm detector)
n	number of nodes sequentially examined by a tree-search algorithm
n_{max}	upper limit for the number of nodes which can be sequentially examined by a tree-search algorithm
n_{min}	lower limit for the number of nodes which are required to generate the LLRs
\mathbf{n}	white gaussian noise (AWGN) vector with complex-valued entries
	n_i
$\tilde{\mathbf{n}}$	vector of residual noise after filtering (e.g., linear filter G , QR-decomposition)
N	number of (sub-)carriers (mobile communications); number of cores (integrated circuits)
N_0	power spectral density
N_T	number of transmit antennas
N_R	number of receive antennas
p	node sequence enumerating index
P	total average power dissipation of an integrated circuit
P_{dyn}	dynamic power (integrated circuits)
P_n	noise power (mobile communications)
P_s	signal power (mobile communications)
P_{sc}	short-circuit component contributing to the dynamic power of a circuit
P_{stat}	static power (integrated circuits)
P_{tran}	capacitive switching component contributing to the dynamic power of a circuit
\mathbf{P}	permutation matrix of the sorted QR-decomposition
q	degree of parallelism
Q	constellation size
\mathbf{Q}	unitary matrix of the QR-decomposition
$\mathbf{Q}_1, \mathbf{Q}_2$	submatrices of \mathbf{Q}
$r_{i,j}$	element at row i and column j of the \mathbf{R} matrix
R	search radius
R_c	code rate

R_{clip}	search radius with internal clipping
\mathbf{R}	upper-triangular matrix of the QR-decomposition
\mathbf{s}	stack employed by the smart-sorting enumeration approach (collection of layer metrics $\lambda_{(i)}$ and symbol indexes $k^{\mathfrak{R}}, k^{\mathfrak{J}}$)
S	technology scaling factor for transistor dimensions and related parameters (transistor scaling theory)
t_c	channel coherence time (mobile communications)
t_s	symbol period (mobile communications)
T	size of the search tuple, $T = \#\mathcal{T}$
T_{clk}	delay of an integrated circuit
T_{cp}	delay of the critical path (integrated circuits)
T_{ox}	thin oxide thickness (integrated circuits)
T_{Q}	flip-flop's clock-to-Q delay (integrated circuits)
T_{su}	preferred flip-flop's setup time (integrated circuits)
T_{uncert}	user-defined clock correction term accounting for imperfections in the clock tree (integrated circuits)
T_{unpipe}	minimum clock period supported by an unpipelined circuit (integrated circuits)
T_Z^{min}	minimum clock period supported by a circuit with Z pipeline stages (integrated circuits)
\mathbf{u}	vector of uncoded i.i.d. information bits
$\hat{\mathbf{u}}$	vector of estimated information bits \mathbf{u}
U	technology scaling factor for voltage magnitudes (transistor scaling theory)
\mathbf{v}	enumeration sequence vector
$\mathbf{v}[p]$	p -th element of the enumeration sequence vector \mathbf{v}
V_0	sub-threshold slope (integrated circuits)
V_{DD}	supply voltage of an integrated circuit
$V_{\text{DD}}^{\text{min}}$	minimum V_{DD} required to achieve a certain clock frequency f_{clk} while guaranteeing error-free operation
V_t	threshold voltage (integrated circuits)
w_k	k -th element within the the window, $w_k \in \{w_0, \dots, w_{W-1}\}$ (window-based enumeration approach)
W	transistor channel width (integrated circuits); window size (window-based enumeration approach)
x_i	constellation symbol sent by the i -th transmit antenna
\hat{x}_i	estimation of the symbol sent by the i -th transmit antenna
\hat{x}_i^p	p -th constellation symbol within the enumeration sequence

$\hat{x}_i^{k^{\Im}}$	imaginary quadrature component of the k^{\Im} -th constellation symbol within the enumeration sequence
$\hat{x}_i^{k^{\Re}}$	real quadrature component of the k^{\Re} -th constellation symbol within the enumeration sequence
\hat{x}_i^{ref}	reference constellation symbol for the search sequence determination approach
\mathbf{x}	vector of transmitted symbols with complex-valued entries x_i
$\hat{\mathbf{x}}$	estimation of the transmitted symbol vector
$\hat{\mathbf{x}}^{\text{MAP}}$	MAP estimation of the transmitted symbol vector
$\hat{\mathbf{x}}^{\text{ML}}$	ML estimation of the transmitted symbol vector
$\tilde{\mathbf{x}}$	received symbol vector \mathbf{y} after application of filter matrix (e.g., linear filter G , QR-decomposition)
y_i	signal captured by the i -th receive antenna
y_i'	signal captured by the i -th receive antenna, after applying the QR-decomposition
y_i''	interference-reduced y_i'
y_i'''	y_i'' after normalizing with r_{ii}
\mathbf{y}	vector of received signals with complex-valued entries y_i
z_i^{ref}	reference geometrical point within the constellation, employed by the metric estimation approach
Z	number of pipeline stages
α	technology-dependent constant accounting for the velocity saturation of CMOS devices (integrated circuits)
η_A	area-throughput efficiency (integrated circuits)
η_{BW}	spectral efficiency (mobile communications)
η_E	energy efficiency (integrated circuits)
κ	shape factor of a gamma distribution (statistics)
λ	metric value (indicator of the “quality”) of a potential detection solution
λ_0	metric of a complete tree path (summation of partial metrics from root to leaves layer)
λ_0^{uB}	metric λ_0 after bias reduction
λ_a	contribution of the <i>a priori</i> information to the partial metrics λ_i
λ_a^{\Im}	imaginary quadrature component of λ_a
λ_a^{\Re}	real quadrature component of λ_a
λ_i	partial metric at layer i (summation of partial metrics from root to layer i)
$\lambda_{(i)}$	partial metric contribution of layer i ($\lambda_{(i)} = \lambda_i - \lambda_{i+1}$)

$\lambda_{(i)}^{\text{I}}$	imaginary quadrature component of $\lambda_{(i)}$
$\lambda_{(i)}^{\text{R}}$	real quadrature component of $\lambda_{(i)}$
μ	mean value (statistics)
π	mathematical constant $\approx 3,14159$
Ψ	inter-layer residual interference matrix
ρ_i	post-equalization SINR on layer i
σ	bias value; standard deviation (statistics)
σ^2	noise-to-signal-energy ratio; variance (statistics)
σ_n^2	noise variance
σ_x^2	average transmit energy per antenna (assuming equal distribution)
$\bar{\tau}_{\text{raw}}$	average (uncoded) detection throughput
$\bar{\tau}$	average effective detection throughput (taking the code rate R_c into account)
θ	scale factor of a gamma distribution (statistics)
Θ_{nom}	nominal peak transmission data rate
Θ_{raw}	uncoded transmission data rate (taking the error-rate into account)
Θ	effective transmission data rate (taking the code rate R_c into account)
∂	overhead ratio of a tree-search algorithm (amount of tree nodes unnecessarily examined)
\mathcal{C}	set of possible transmitted vector of coded bits \mathbf{c}
\mathcal{B}	set of candidate (counter-)hypotheses considered by the tuple-search sphere detector at layer $i = 0$
\mathcal{L}	subset of all possible transmitted symbol vectors considered by tree-search-based detection strategies ($\mathcal{L} \subset \mathcal{V}$)
\mathcal{T}	set of T candidate metrics λ_0 for the determination of the radius (search tuple)
\mathcal{V}	set of possible transmitted symbol vectors \mathbf{x}
\mathcal{X}	set of constellation symbols x
$\mathcal{X}_{m,l}^{\pm 1}$	subset of constellation symbols $x_i \in \mathcal{X}$ with bit $c_{m,l} = \pm 1$

Citations

[·] [†]	reference to the author's works
[·]	reference to the works of others

Introduction

1.1 Motivation and Objectives

Modern wireless communications standards, such as IEEE 802.11 (WLAN), IEEE 802.16 (WiMAX), 3GPP-LTE or LTE-Advanced, define numerous operating modes as well as sophisticated transmission techniques to be supported by receiver architectures, while satisfying a vast variety of stringent (and most often conflicting) requirements. The situation becomes even more challenging with future 5G technologies, provided the envisioned explosion of machine-to-machine (M2M) communications accompanying *the internet of the things* [2, 3], the extremely low latencies required by *the tactile internet* [4], and the unprecedented number of antennas foreseen for *massive distributed MIMO* (multiple-input multiple-output) [5], among other advances [6, 7]. In this regard, three concepts will play a major role as **key enabling factors: adaptability, efficiency, and performance**. Multi-antenna detection belongs to the most computationally intensive constituents of the receiver's baseband signal processing, especially concerning spatial-multiplexing transmission. Practical MIMO detector realizations must comply with restrictive power constraints and satisfy the demanding performance requirements specified in 4G and future 5G standards, while being able to dynamically adapt to e.g., application requirements, battery life or channel conditions. Designing adaptive, good-performing and cost-effective MIMO detectors represents a challenge, especially concerning high-order systems (i.e., $\geq 4 \times 4$ MIMO configurations with ≥ 64 -QAM modulations). Ordinary low-complex detectors such as linear or successive interference cancellation (SIC) approaches provide poor detection accuracy, whereas exhaustive-search algorithms (full max-log-APP detection) cannot achieve 4G/5G data rates with reasonable hardware complexity. Tree-search strategies, on the other hand, are suboptimal approaches which offer a reasonable trade-off between detection accuracy and implementation complexity. In this regard, breadth-first tree-search algorithms such as M-algorithm, K-best and fixed-complexity sphere detectors are

implementation-friendly and highly parallelizable, but they generally sacrifice accuracy and adaptability in favor of fixed data rates. The so-called depth-first sphere detector emerges therefore as the only approach capable of trading adaptability, efficiency and performance. In particular, the reduced-complexity tuple search detector (TSD) proposed in [8, 9] and focused in this work has demonstrated to outperform the error-rate-complexity trade-off of similar depth-first sphere detection strategies like single tree search (STS) [10] or list sphere detection (LSD) [11].

“*In the same way as the compressed air is fed back from the compressor to the main engine, the extrinsic information is fed back to the other decoder*” [12]. With this words, J. Hagenauer drew an analogy between a car’s mechanical turbo engine and the turbo decoding principle in the context of mobile communications [13]. The concept was extrapolated in the late 90’s to the interaction between the detector/demodulator and the channel decoder [14, 15]. The so-called iterative *detection-and-decoding* allows exploiting the full potential of MIMO communications, enabling a dramatic improvement of the communication’s reliability by exchanging *soft-information* between the detector and the decoder. This benefit comes at the cost of increasing the receiver’s complexity: the overall required computing time rises as a consequence of the repeated detection and decoding processes. Additionally, the number of nodes explored by tree-search detection strategies tends to increase in the iterative scenario -fact that translates into a further latency increment in the context of depth-first tree traversal. Since generating accurate soft-information generally requires an unacceptably high computational effort, suboptimum values must be determined to reduce complexity [9]. An additional major challenge concerning these suboptimum, soft-input soft-output (SISO) detectors, is represented by the distortive effect that the soft information from decoder causes on the tree nodes enumeration ordering. The latter is a particularly sensitive aspect of tree-search algorithms, since wrong node selection may compromise the detection’s accuracy. One of the main objectives of this work consists therefore in characterizing the influence of the soft information on the detection’s search process and developing strategies which alleviate the complexity and performance loss drawbacks. Besides introducing and analyzing several MIMO detection approaches, developing suitable architectures which enable efficient VLSI detector realizations constitutes an additional major objective of this work. In the mobile communications domain, the most computationally intensive components of digital transceivers frequently require dedicated hardware to handle real-time and high throughput requirements while complying with stringent area and power consumption restrictions (in e.g., pico/femtocell base stations, sensor networks or mobile terminals). In this regard, application-specific instruction-set processors (ASIPs) offer programmability while including dedicated instructions that accelerate processing beyond the capabilities of ordinary general- and special-purpose processors (such as DSPs), and thus represent the approach followed in this work. In order

to assess the true hardware complexity and efficiency of the proposed VLSI designs, their physical characteristics will be analyzed (including on-chip measurements) and compared to the state-of-the-art.

1.2 Outline

In this manuscript two main areas can be clearly distinguished. The first one, encompassing chapters 2 to 4, covers theoretical fundamentals as well as conceptual work developed from an algorithmic perspective. Note that results presented in this concern are restricted to computer simulations. The second one, comprising chapters 5 and 6, focuses on the VLSI architecture design and implementation of the previously proposed algorithmic concepts. Results presented in this case include, additionally, measurements performed on real hardware systems. Besides this introduction, and the conclusions, contributions and outlook summarized in chapter 7, this dissertation comprises the following sections:

Chapter 2 introduces the fundamentals of MIMO communications required to ease understanding the remainder of this thesis. A general overview of MIMO techniques with special emphasis on detection approaches for spatial multiplexing systems is provided. Additionally, complexity, performance and efficiency metrics setting the basis for equitable comparison against state-of-the-art approaches are defined.

Chapter 3 reviews the state-of-the-art and challenges affecting soft-input soft-output sphere detection and presents a statistical analysis concerning the effect of the *a priori* information on the detection process. In this regard, the most relevant existing node enumeration strategies are revised and two novel approaches are introduced.

Chapter 4 focuses on the performance analysis of the proposed MIMO detection approaches. In particular, the error-rate-complexity trade-off and the search efficiency are evaluated for different MIMO configurations and modulation orders. In addition to this, the performance of the enumeration strategies presented in the previous chapter is investigated in the context of iterative detection-and-decoding.

Chapter 5 provides an overview of the key strategies enabling an efficient VLSI design for MIMO detection. A detailed description of the proposed architectures is provided, covering the control path, memory organization and processing elements, as well as an analysis of the designs' critical paths.

Chapter 6 presents a detailed evaluation of proposed architectural concepts. This chapter is divided in three main sections:

- The first one depicts estimated results on the hardware characteristics (i.e., power, area, delay. . .) of the architectures introduced in chapter 5, which allows to assess the implementation cost of the enumeration strategies proposed in chapter 3.
- The second introduces pipelining and retiming architecture optimizations, based on the design's area-delay-energy trade-off.
- The third and last section concerns the silicon implementation of a MIMO detector, including an analysis and discussion on energy-efficiency techniques and a comparison of measurement results with relevant state-of-the-art CMOS realizations reported in literature.

Fundamentals of MIMO Communications

In this chapter the fundamentals of MIMO communications are briefly summarized, paying special attention to detection approaches for spatial multiplexing systems. An overview of the most relevant detection strategies and mechanisms for complexity reduction is provided, whereas the sphere detector algorithm focused in this work is described in detail. Lastly, a collection of complexity, performance and efficiency metrics is presented, which establishes the basis to enable an equitable comparison of detector solutions.

2.1 MIMO Approaches

Multiple-input multiple-output (MIMO) refers to the use of multiple antennas at both transmitter and receiver, with MISO (multiple-input, single-output) and SIMO (single-input, multiple-output) systems being particular cases of the first. The use of multiple antennas at the transmitter and/or at the receiver enables significant improvements in terms of link reliability, capacity, and spectral efficiency in comparison to single-antenna systems. During the last century, the benefits of MIMO technology have been intensively investigated. Historically, directive arrays and antenna selection approaches have been used to improve the communications performance [16]. With the development of personal mobile communications systems, multiple antennas have been further employed to increase the reliability of wireless links by exploiting diversity. In the last decades, the demand for ever-increasing data rates together with the limited availability of radio resources has led to the exploitation of the spatial dimension of the MIMO channel in order to enhance spectral efficiency. In general, MIMO strategies can be summarized into the following classification:

Beamforming

Multiple antennas can be employed to provide a power gain, thereby increasing the effective SNR. At the receiver's end the signals from each antenna can be linearly combined with each other (receive beamforming). On the transmitter side, more power can be allocated to the antenna with the better gain and the signals from different antennas can be arranged to be constructively added in-phase at each receive antenna (transmit beamforming). In general, the larger the antenna array is, the sharper can the radiated energy be focused along a desired direction [17].

Antenna Diversity

Antenna diversity (or spatial diversity) is a strategy employed, like other diversity techniques, to reduce the errors occurring in reception due to large channel attenuation, i.e., when the channel is in a deep fade. In its classical definition, it basically consists in supplying to the receiver several replicas of the same information signal transmitted over independently fading channels, thus significantly reducing the probability that all signal components fade simultaneously. For this purpose, a signal is transmitted and/or received through several transmitting and/or receiving antennas, spaced sufficiently far apart so that the signals propagate through independent fading paths [18]. The required antenna separation depends on the channel scattering as well as on the carrier frequency. In a system with N_T transmitting and N_R receiving antennas, the maximum achievable diversity of the channel is $N_T \times N_R$ [17]. Other diversity domains besides the spatial one are time (diversity can be exploited by e.g., coding and interleaving symbols over different coherence time periods) and frequency (in frequency-selective channels diversity can be exploited by using a bandwidth greater than the coherence bandwidth of the channel, e.g., by multipath combining or frequency hopping in spread-spectrum systems).

Spatial Multiplexing

MIMO channels provide additional “degrees of freedom” for communication. In a system with N_T transmitting and N_R receiving antennas, the channel presents $\min\{N_T, N_R\}$ degrees of freedom [17]. In the classical sense, spatial multiplexing is a scheme which enables utilization of all available degrees of freedom of a channel by transmitting independent symbols over different antennas as well as over different symbol times. Independent data streams are hence multiplexed in space. This strategy leads to an increase in the communication capacity which, under the assumption of a rich scattering environment and $N_T = N_R$, is directly proportional to N_R [17]. Spatial multiplexing is the most recent multiple-antenna technology attracting attention, with first practical approaches developed in the mid-90s at Bell Laboratories [19].

Even though according to their classical definitions these MIMO approaches seem to be unrelated, a closer look indicates the contrary. In fact, MIMO channels provide intrinsically a certain degree of spatial diversity as well as a certain number of degrees of freedom. The benefits of MIMO communication are hence two-fold: on the one hand, the data rates can be enhanced by increasing the spatial degrees of freedom (or, equivalently, the multiplexing gain) and on the second, reliability can be improved by exploiting the diversity gain. The diversity-multiplexing relationship reflects the fundamental trade-off to communicate over a fading channel with a certain data rate and error probability. But, is it possible to find a scheme which achieves full diversity and exploits all the channel's degrees of freedom? Generally, fast-fading channels can be assumed to be reliable (since they can benefit from other sources of diversity, like time diversity through coding) and therefore spatial multiplexing can be fully exploited. In contrast, slow-fading channels are more unreliable (since a deep fade may affect many symbols over time) and exploiting antenna diversity is hence required to increase reliability. A simple scheme that repeats the information symbol across the multiple fading paths (repetition coding) achieves full diversity gain but it usually does not take advantage of the degrees of freedom of the channel. More sophisticated schemes, the so called space-time codes ([19–21], among others), can simultaneously increase the data rate and achieve a coding gain along with the diversity gain. In fact, for a given fading channel model, there is an optimal tradeoff between the two types of gains achievable by any space-time coding scheme [17]. In the context of line-of-sight environments beamforming techniques come additionally into play. Several clusters of antennas can be defined within an array in order to spatially multiplex different signals over independent beams (as e.g., specified by the transmission mode 8 -dual layer beamforming- in the 3GPP LTE rel. 9 standard [22]). Additionally, modern wireless communication systems may not only be defined by physically co-located antennas, but also by distributed cooperative base stations (e.g., coordinated multi-point -CoMP) and/or user terminals (e.g., *ad-hoc* communications, multi-user MIMO). In the remainder of this thesis MIMO spatial multiplexing will be focused, whereas other MIMO technologies are regarded as open research topics (listed in chapter 7).

2.2 MIMO Detection for Spatial Multiplexing

In wireless digital communications systems, the binary-represented information $\mathbf{c}'' \in \mathcal{C} = \{+1, -1\}^{N_T \cdot L}$ (with L denoting the number of bits per symbol) to be transmitted is typically mapped onto constellation symbols x_i ($\mathbf{x} = [x_0, \dots, x_{N_T-1}]^T = \text{map}(\mathbf{c}'') \in \mathcal{V}$) from a set $\mathcal{X} \subset \mathcal{V}$ with cardinality $\#\mathcal{X} = 2^L = Q$, corresponding to a particular modulation scheme (e.g., quadrature amplitude modulation -QAM). In a MIMO system with N_T transmitting

and N_R receiving antennas the received signal $\mathbf{y} \in \mathbb{C}^{N_R \times 1}$ is given by:

$$\mathbf{y} = \mathbf{H}\mathbf{x} + \mathbf{n}, \quad (2.1)$$

with $\mathbf{x} \in \mathbb{C}^{N_T \times 1}$ being a complex vector of transmitted symbols, and $\mathbf{H} \in \mathbb{C}^{N_R \times N_T}$ being the matrix of effective complex channel coefficients¹, each representing a point-to-point transmission path between a transmitting and a receiving antenna through the communications channel². The vector $\mathbf{n} \in \mathbb{C}^{N_R \times 1}$ represents additive white Gaussian noise (AWGN) at the receiver, comprised of independent and identically distributed (i.i.d) Gaussian complex random variables with zero mean and variance $\sigma_n^2 = N_0 (\frac{N_0}{2})$ per real dimension), where N_0 represents the power spectral density of the noise. The detection problem consists in determining the most likely transmitted vector of symbols $\hat{\mathbf{x}} \in \mathbb{C}^{N_T \times 1}$ that satisfies equation (2.1). By demapping $\hat{\mathbf{x}} \in \mathcal{V}$, the estimation of the originally transmitted binary stream \mathbf{c}'' is directly obtained $\mathbf{c} = \text{demap}(\hat{\mathbf{x}})$, $\mathbf{c} \in \mathcal{C} = \{+1, -1\}^{N_T \cdot L}$. In addition to the *hard decision* made on the transmitted bits values \mathbf{c}'' , information about the decisions' reliability, i.e., *soft information*, may be provided. By making appropriate use of this information the communications error-rate performance can be considerably improved, whereas the non-negligible computational complexity incurred to generate the soft information must be undertaken. According to the generation and exploitation of soft information, different detection schemes can be distinguished:

Hard-output (HO) detectors make a hard decision on the likely transmitted bits \mathbf{c}'' .

This approach is commonly employed in uncoded systems, where no channel decoder is available to benefit from soft information.

Soft-output (SO) detectors provide, in addition to the estimation of the transmitted bits \mathbf{c}'' , probabilistic information about the decision (i.e., *soft bits*). In coded systems, the channel decoder makes use of this information to boost the communication performance (as shown e.g., in [23]).

Soft-input soft-output (SISO) detectors, in contrast to the previous cases, incorporate soft input information (i.e., *a priori* information) into the detection process. By feeding soft information from the channel decoder back to the detector and allowing both entities to further detect and decode on iterative fashion, the communications performance can be further improved [24], at the cost of additional computational effort. The number of iterations I performed is commonly defined as the number of detection-and-decoding runs, i.e., one iteration ($I = 1$) corresponds to the soft-output case (or *open-loop* system, as referred to in e.g., [25]). The iterative approach

¹Assuming perfect knowledge of channel state information (CSI) or resulting from channel estimation.

²A more detailed description of the transmission and channel model is provided in section 4.1.

($I > 1$, sometimes referred to as *closed-loop* system [25]), is widely known as the *turbo principle* [12, 13].

2.2.1 Maximum Likelihood and *a Posteriori* Probability Detection

In order to determine the most likely transmitted vector of symbols $\hat{\mathbf{x}}$ satisfying (2.1), the *maximum a posteriori* criterium (**MAP** or **max-APP**) is applied [26, 27]. This approach consists of observing the received signal \mathbf{y} and determining the vector $\hat{\mathbf{x}}$ with the highest *a posteriori* probability:

$$\hat{\mathbf{x}} = \arg \max_{\mathbf{x} \in \mathcal{V}} \{P(\mathbf{x}|\mathbf{y})\} = \arg \max_{\mathbf{x} \in \mathcal{V}} \left\{ \frac{P(\mathbf{y}|\mathbf{x})P(\mathbf{x})}{P(\mathbf{y})} \right\} \equiv \arg \max_{\mathbf{x} \in \mathcal{V}} \{P(\mathbf{y}|\mathbf{x})P(\mathbf{x})\}, \quad (2.2)$$

where \mathcal{V} represents the set of 2^{LN_T} possible transmitted symbol vectors. The first equality in (2.2) results from the application of Bayes' theorem, whereas $P(\mathbf{y})$ can be disregarded since it is constant for all hypotheses \mathbf{x} considered in the $\arg \max(\cdot)$ operation [28], resulting in the right-most equivalent expression. Assuming additive Gaussian noise, the N_R -dimensional probability density function of the Gaussian-distributed complex received vector conditioned on the transmitted vector takes the form

$$P(\mathbf{y}|\mathbf{x}) = \frac{1}{(\pi N_0)^{N_R}} e^{-\frac{\|\mathbf{y} - \mathbf{H}\mathbf{x}\|^2}{N_0}}. \quad (2.3)$$

It becomes obvious that maximizing (2.3) is achieved by minimizing the Euclidean distance $\|\mathbf{y} - \mathbf{H}\mathbf{x}\|^2$ [29]. By replacing (2.3) in equation (2.2) and taking its natural logarithm³, the MAP solution is consequently defined as:

$$\hat{\mathbf{x}}^{\text{MAP}} = \arg \min_{\mathbf{x} \in \mathcal{V}} \left\{ \frac{\|\mathbf{y} - \mathbf{H}\mathbf{x}\|^2}{N_0} - \ln P(\mathbf{x}) \right\}, \quad (2.4)$$

where the constant term in (2.3) is again omitted since it does not take effect in the $\arg \max(\cdot)$ operation. An optimal detector thus searches for the solution $\hat{\mathbf{x}}$ which minimizes the metric $\lambda = \|\mathbf{y} - \mathbf{H}\mathbf{x}\|^2 - N_0 \ln P(\mathbf{x})$. Assuming equal *a priori* probability $P(\mathbf{x})$ of all transmit symbol vectors (i.e., in case no *a priori* information is available or it is disregarded), the *maximum-likelihood* (**ML**) solution is obtained:

$$\hat{\mathbf{x}}^{\text{ML}} = \arg \min_{\mathbf{x} \in \mathcal{V}} \left\{ \|\mathbf{y} - \mathbf{H}\mathbf{x}\|^2 \right\}. \quad (2.5)$$

³Since the logarithm function is monotonically increasing with its argument, maximizing the expression in (2.2) is equivalent to maximizing its logarithm.

2.2.2 (Approximated) Logarithmic MAP Detection

A convenient representation of the soft information are the so-called *Log-Likelihood Ratios* (LLRs, also referred to in literature as *L-values*) [13]. Applying standard manipulations to Baye's theorem, the detector's *a posteriori* information $L(c_{m,l}|\mathbf{y})$ corresponding to the bit $c_{m,l}$ (with $m \in \{0, 1, \dots, N_T - 1\}$, $l \in \{0, 1, \dots, L - 1\}$), given the received vector symbol \mathbf{y} can be defined as [30]:

$$\underbrace{\ln \frac{P(c_{m,l} = +1|\mathbf{y})}{P(c_{m,l} = -1|\mathbf{y})}}_{A \text{ Posteriori Information}} = \underbrace{\ln \frac{P(\mathbf{y}|c_{m,l} = +1)}{P(\mathbf{y}|c_{m,l} = -1)}}_{\text{Extrinsic Information}} + \underbrace{\ln \frac{P(c_{m,l} = +1)}{P(c_{m,l} = -1)}}_{A \text{ Priori Information}} \quad (2.6)$$

$$L(c_{m,l}|\mathbf{y}) = L_e(\mathbf{y}|c_{m,l}) + L_a(c_{m,l}),$$

where the detector's *extrinsic* information $L_e(\mathbf{y}|c_{m,l})$ is the "incremental" information that is actually exchanged between the detector and the channel decoder in the turbo receiver, and the *a priori* information $L_a(c_{m,l})$ corresponds to the knowledge fed back by the channel decoder (as further detailed in section 4.1). By applying the law of total probability on $P(\mathbf{y}|c_{m,l} = \pm 1)$ and incorporating (2.3) [28], the **logarithmic APP (log-APP, also log-MAP)** solution is determined by:

$$L(c_{m,l}|\mathbf{y}) = \ln \frac{\sum_{\mathbf{x} \in \mathcal{V}_{m,l}^{+1}} \left[\left(e^{-\frac{\|\mathbf{y} - \mathbf{H}\mathbf{x}\|^2}{N_0}} \right) P(\mathbf{x}|c_{m,l} = +1) \right]}{\sum_{\mathbf{x} \in \mathcal{V}_{m,l}^{-1}} \left[\left(e^{-\frac{\|\mathbf{y} - \mathbf{H}\mathbf{x}\|^2}{N_0}} \right) P(\mathbf{x}|c_{m,l} = -1) \right]} + L_a(c_{m,l}), \quad (2.7)$$

where $P(\mathbf{x}|c_{m,l} = b)$ (with $b = \pm 1$) is derived from the definition of $L_a(c_{m,l})$ in (2.6), assuming statistically independent transmitted bits:

$$P(\mathbf{x}|c_{m,l} = b) = \prod_{i \neq m} \prod_{j \neq l} P(c_{i,j}) = \prod_{i \neq m} \prod_{j \neq l} \frac{e^{(c_{i,j} L_a(c_{i,j}))}}{1 + e^{(c_{i,j} L_a(c_{i,j}))}}.$$

Even though the set \mathcal{V} of possible transmitted symbol vectors is finite and discrete, finding the optimum solution of (2.7) implies a computational complexity of order 2^{LN_T} [24], i.e., growing exponentially with the number of transmit antennas N_T and the number of bits per symbol L . Since this complexity is prohibitive for practical receiver implementations, the so-called *max-log* approximation $\left(\ln \left(\sum_i f(i) \right) \right) \approx \max_i \{ \ln f(i) \}$ of (2.7) is typically preferred, resulting in:

$$L(c_{m,l}|\mathbf{y}) \approx -\frac{1}{N_0} \min_{\mathbf{c}|c_{m,l}=+1} \{ \lambda \} + \frac{1}{N_0} \min_{\mathbf{c}|c_{m,l}=-1} \{ \lambda \}, \quad (2.8)$$

with the *metric* λ defined as:

$$\lambda(\mathbf{y}, \mathbf{c}, \mathbf{L}_a) = \|\mathbf{y} - \mathbf{H}\hat{\mathbf{x}}(\mathbf{c})\|^2 - \frac{N_0}{2} \sum_{i=0}^{N_T-1} \sum_{j=0}^{L-1} c_{i,j} L_a(c_{i,j}). \quad (2.9)$$

At this point it becomes evident that in order to generate soft information, searching for the ML/MAP solution (i.e., the detection *hypothesis*) is not sufficient, since this does not necessarily minimize the two terms in (2.8). Instead, a search for all the LN_R required minima (i.e., the *counter-hypotheses*) must be performed as well. Note that despite the approximation in (2.8), the complexity of the ***max-log-APP*** (also ***max-log-MAP***) detection is still depending exponentially on N_T and L since, in the worst case, an exhaustive search over all likely transmitted vectors is nevertheless performed. This approach enables, however, the application of many detection schemes which approximate (2.8) by exploring only a subset of the 2^{LN_T} vector candidates during the minima search (e.g., the list-based sphere detector proposed in [24]). These algorithms aim at finding the optimal (or at least, a suboptimal) solution with reduced complexity (ideally polynomially growing, instead of exponentially). In the following, the most commonly applied detection approaches are discussed.

2.2.3 Linear Detection

A straightforward approach to estimate the transmitted symbol consists in equalizing the received signal \mathbf{y} with a linear filter $\mathbf{G} \in \mathbb{C}^{N_R \times N_T}$ which attempts to reverse the effect of the channel [26, 27]:

$$\tilde{\mathbf{x}} = \mathbf{G}\mathbf{y} = \mathbf{G}\mathbf{H}\mathbf{x} + \mathbf{G}\mathbf{n} = \mathbf{\Psi}\mathbf{x} + \tilde{\mathbf{n}}. \quad (2.10)$$

The result of this operation $\tilde{\mathbf{x}}$ is simply quantized to the nearest constellation vector $\hat{\mathbf{x}}$ ($\hat{\mathbf{x}} = \lfloor \tilde{\mathbf{x}} \rfloor_x$)⁴. In equation (2.10), $\mathbf{\Psi}$ represents the residual interference among layers, while $\tilde{\mathbf{n}}$ models the correlated noise at the equalizer output. The following two well known approaches are widely applied in order to determine filter matrix \mathbf{G} :

Zero Forcing (ZF) [32, 33]

The ZF criterium defines the filter matrix as the pseudo-inverse of the channel matrix:

$$\mathbf{G}_{ZF} = \mathbf{G}'_{ZF} \mathbf{H}^H = (\mathbf{H}^H \mathbf{H})^{-1} \mathbf{H}^H \quad (2.11)$$

Besides its low complexity, the main advantage of this approach is that, assuming perfect knowledge of the channel state information (CSI) at the receiver, the interference among layers is completely cancelled out ($\mathbf{\Psi} = \mathbf{I}$, being \mathbf{I} the identity matrix). However, this approach suffers frequently from strong noise enhancement: if the channel

⁴This operation is also referred in literature as *slicing*, e.g., in [31]

matrix is *ill-conditioned* the equalized noise can get arbitrarily large [26], leading to significant performance degradation.

Minimum Mean Squared Error (MMSE) [34, 35]

The filter matrix is determined by minimizing the mean squared error between the transmitted signal and the estimated signal $\mathbf{G} = \arg \min_{\mathbf{G}} \varepsilon \{ \|\tilde{\mathbf{x}} - \mathbf{x}\|^2 \}$ and has the form:

$$\mathbf{G}_{\text{MMSE}} = \mathbf{G}'_{\text{MMSE}} \mathbf{H}^H = (\mathbf{H}^H \mathbf{H} + \sigma^2 \mathbf{I}_{N_T})^{-1} \mathbf{H}^H. \quad (2.12)$$

Assuming the transmitting power E_s to be equally distributed among the N_T antennas ($\sigma_x^2 = E_s/N_T$), σ^2 is defined as

$$\sigma^2 = \sigma_n^2 / \sigma_x^2 = \frac{N_T N_0}{E_s}. \quad (2.13)$$

The MMSE criterion thus mitigates the noise amplification problem by taking the receiver noise into account in the design of the filter matrix, at the cost of imperfect interference cancellation among layers ($\mathbf{\Psi} \neq \mathbf{I}$). This approach maximizes the *signal to interference and noise ratio* (SINR) at the output of the filter and generally achieves better error-rate performance than ZF while presenting comparable complexity. It should be noticed that the MMSE estimator is *biased*, which causes the diagonal entries of the effective channel matrix $\mathbf{\Psi}$ to be smaller than one [36]. This results in a marginal degradation of the error-rate performance, which can be avoided e.g., by appropriately scaling \mathbf{G} [36, 37].

Linear detection is therefore a straightforward, low-complex strategy to find $\hat{\mathbf{x}}$ avoiding the exhaustive search required to solve equation (2.5). While this represents an advantage with regard to a practical implementation, the solution provided by this strategy is generally suboptimal, which obviously proves disadvantageous for the sake of reliable communications. In coded systems the performance can be improved through the computation of soft-information, by approximating the *max-log-APP* solution according to [38]:

$$L_e(c_{m,l}) \approx -\rho_i \left(\min_{x_i \in \mathcal{X}_{m,l}^{+1}} \|\hat{x}_i - x_i\|^2 - \min_{x_i \in \mathcal{X}_{m,l}^{-1}} \|\hat{x}_i - x_i\|^2 \right), \quad (2.14)$$

where $x_i \in \mathcal{X}_{m,l}^{\pm 1}$ represents the constellation symbol $x_i \in \mathcal{X}$ with (m, l) -th bit $c_{m,l} = \pm 1$, and ρ_i is the post-equalization SINR on layer i :

$$\rho_i = \frac{1}{\sigma^2} \frac{1}{\left[\mathbf{G}'_{\text{ZF/MMSE}} \right]_{i,i}}.$$

Even though the error-rate performance is significantly improved with regard to the hard-decision linear detection, it is still far away from that of an optimum MAP detector [29].

2.2.4 Interference Cancellation Detection

Since by means of linear equalization the interference is not effectively suppressed and poor overall error-rate performance is achieved [39, 40], several detection techniques based on *interference cancellation* have been developed. The main idea behind this strategy is to detect each of the symbols in the received vector separately, subtracting previously the contribution of the other (interfering) symbols from the received vector. Two major variants of this technique are widely applied:

Parallel Interference Cancellation (PIC)

A tentative estimation of the transmitted signal vector is initially obtained by means of e.g., linear equalization [41] or based on *a priori* information from the channel decoder [42]. Subsequently, N_R parallel filters are applied, each subtracting the contribution of all N_R layers except one from the received signal vector. This process can be performed throughout several stages (e.g., [41]) or iteratively (e.g., [43]) before making a final decision on the transmitted symbol vector.

Successive Interference Cancellation (SIC)

The main idea behind this strategy is to perform a sequential layer-by-layer detection, suppressing the interference corresponding to already detected layers from the received signal before processing subsequent layers. For this purpose, a set of linear filters can be employed, as originally proposed in [19, 44]. Alternatively, the computation of the channel matrix pseudo-inverses can be replaced by a matrix decomposition [45], e.g., the QR Decomposition (QRD).

One of the main advantages of the SIC strategy is the increase in spatial diversity achieved at each layer. While at the first layer the same diversity order is achieved as a linear detector would achieve ($N_R - N_T + 1$), at the i -th layer a spatial diversity order $N_R - N_T + i$ is ideally reached (i.e., assuming perfect interference suppression). The dependence among layers may however be disadvantageous, since a wrong decision at a particular layer will obviously impair the reliability of the decisions at subsequent ones. Consequently, the order in which layers are processed has a direct impact on the detection error-rate performance. The effects of error propagation can be minimized by detecting the most reliable received signals first, i.e., processing the layers in descending order of their post-detection SNR [46]. This approach, commonly referred to in literature as ordered SIC (OSIC), consists in performing the detection on an equivalent system where \mathbf{x} and \mathbf{H} have been appropriately permuted. In the case of QRD-based SIC, the layer ordering can be performed before the matrix decomposition [45] or alternatively, during the matrix decomposition by applying the so-called sorted QRD (SQRD) [47]. It should be noticed that SQRD is a greedy algorithm and consequently, the resulting layer ordering is suboptimal. The ideal layer

succession can be obtained by applying a so-called Post-Sorting-Algorithm (PSA), at the cost of increasing the computational complexity. The error-rate performance loss of SQRD with respect to algorithms implementing the optimal ordering (e.g., V-BLAST [47]) has been nevertheless shown to be marginal in coded systems [48], thus making the application of PSA rather unattractive.

2.2.4.1 MMSE-SQRD-based SIC

The MMSE criterion described in section 2.2.3 can be additionally incorporated into the previously mentioned SQRD's sorting process. Resulting from this, the SINR is maximized for each layer and the computational complexity is significantly reduced [48]. The sorted QR decomposition of the MMSE-extended channel matrix is performed according to:

$$\begin{bmatrix} \mathbf{H} \\ \sigma \mathbf{I}_{N_T} \end{bmatrix} \mathbf{P} = \mathbf{Q} \mathbf{R}, \quad (2.15)$$

where σ is derived from equation (2.13), $\mathbf{P} \in \mathbb{N}^{N_T \times N_T}$ is the corresponding permutation matrix, $\mathbf{Q} \in \mathbb{C}^{(N_R + N_T) \times N_T}$ is a unitary matrix and $\mathbf{R} \in \mathbb{C}^{N_T \times N_T}$ is an upper-triangular matrix with non-negative real-valued entries in its diagonal. By defining $\mathbf{Q} = [\mathbf{Q}_1^T \mathbf{Q}_2^T]^T$ with $\mathbf{Q}_1 \in \mathbb{C}^{N_R \times N_T}$ and $\mathbf{Q}_2 \in \mathbb{C}^{N_T \times N_T}$, the received vector is modified according to:

$$\mathbf{y}' = \mathbf{Q}_1^H \mathbf{y} = \mathbf{R} \tilde{\mathbf{x}} + \tilde{\mathbf{n}}, \quad (2.16)$$

$$y'_i = r_{ii} \hat{x}_i + \sum_{j=i+1}^{N_T-1} r_{ij} \hat{x}_j + \tilde{n}_i, \quad (2.17)$$

with $\tilde{\mathbf{n}}$ representing the effective noise (plus interference) vector, expressed as

$$\tilde{\mathbf{n}} = \mathbf{Q}_1^H \mathbf{n} + \sigma \mathbf{Q}_2^H \mathbf{x}, \quad (2.18)$$

and \mathbf{x} resulting from the application of the permutation $\mathbf{x} = \mathbf{P} \tilde{\mathbf{x}}$. The Euclidean distance can be thus reformulated as $\|\mathbf{y}' - \mathbf{R} \hat{\mathbf{x}}\|^2$, and the estimated symbol at layer i is obtained by suppressing the interference of previously detected layers:

$$y_i''' = \frac{1}{r_{ii}} (y_i'') = \frac{1}{r_{ii}} \left(y_i' - \sum_{j=i+1}^{N_T-1} r_{ij} \hat{x}_j \right) \quad (2.19)$$

and quantizing the result to the nearest constellation symbol $\hat{x}_i = \lfloor y_i''' \rfloor_x$. In this expression y_i''' denotes the (normalized) interference-reduced received signal, and the coefficients r_{ij} represent the entries of the matrix \mathbf{R} . By exploiting the upper-triangular structure of \mathbf{R} , the *max-log*-APP solution can be approximated as:

$$L(c_{m,l}) \approx -\frac{1}{N_0} \min_{x_i \in \mathcal{X}_{m,l}^{+1}} \{\lambda_{(i)}\} + \frac{1}{N_0} \min_{x_i \in \mathcal{X}_{m,l}^{-1}} \{\lambda_{(i)}\}, \quad (2.20)$$

$$\lambda_{(i)} = r_{ii}^2 \|y_i''' - \hat{x}_i\|^2 - \frac{N_0}{2} \sum_{j=0}^{L-1} c_{i,j} L_a(c_{i,j}), \quad (2.21)$$

where $\lambda_{(i)}$ represents the layer metric. It should be noticed that the quality of the LLRs computed by these means strongly depends on the correctness of the hard decisions on \hat{x}_i . In order to minimize the effects of error propagation in turbo receivers, *soft* estimates can be subtracted from the received signal and their variance can be employed to determine the expected residual noise after the cancellation step (*Soft-SIC*), as proposed in e.g., [49].

2.2.5 Tree-Search Detection

While exhaustive ML/MAP detection presents prohibitive computational complexity, the simple LD approach achieves a rather poor error-rate performance and cannot exploit *a priori* information efficiently in order to boost the performance in iterative systems. SIC detection generally outperforms LD in terms of error rate and entails a comparable complexity, however it suffers from error propagation and its performance is still far away from that of optimum ML/MAP detection. Even though the soft-SIC strategy allows benefiting from *a priori* information and alleviates the error propagation problem in iterative systems, determining the required soft symbols and cancellation noise is computationally intensive [50, 51]. The so-called **tree search** algorithms represent a potentially optimum trade-off solution: the ML/MAP/*max-log*-APP estimate can be approximated while avoiding an exhaustive search. The main idea behind these approaches is to represent the set \mathcal{V} of all likely transmitted symbol vectors as a weighted tree structure, as exemplified in Figure 2.1. The number of levels of the tree is defined by the number of MIMO layers, which is equivalent to the number of transmit antennas N_T (assuming spatial multiplexing with one transmitted symbol stream per antenna). Each tree layer i comprises $2^{L(N_T-i)}$ nodes, each representing a constellation symbol $x \in \mathcal{X}$. A set of Q so-called *child* nodes, descends from each *parent* node into the next layer ($i-1$). The tree root is defined by the topmost layer ($i = N_T$), while the so-called *leaf* nodes (or *leaves*) compose the lowest layer ($i = 0$). Each of the tree *paths* (i.e., tree edges connecting parent and child nodes from the root to a leaf node) is weighted by a metric λ_0 .

Instead of searching the set \mathcal{V} of all likely transmitted symbol vectors, tree-search-based detection strategies only consider a subset $\mathcal{L} \subset \mathcal{V}$ of vector candidates, consequently approximating the *max-log*-APP solution in (2.8) as:

$$L(c_{m,l}|\mathbf{y}) \approx -\frac{1}{N_0} \min_{x \in \mathcal{L}|c_{m,l}=+1} \{\lambda_0\} + \frac{1}{N_0} \min_{x \in \mathcal{L}|c_{m,l}=-1} \{\lambda_0\}. \quad (2.22)$$

Note that the size of the subset \mathcal{L} imposes a trade-off between the detection complexity and the quality of the resulting LLRs. On the one hand, it is convenient to shrink the

search space in order to reduce the computational effort with respect to an exhaustive search. On the other hand, the subset \mathcal{L} should be large enough so that the probability that the hypothesis and the required counter-hypotheses are contained is sufficiently high. The latter is a desirable condition in order to approximate the *max-log-APP* solution as closely as possible, consequently minimizing the error-rate degradation with regard to the exhaustive-search detection. To enable mapping all the possible transmit symbol vectors to a tree structure, the channel matrix must be decomposed in such a way that a successive dependency among antennas can be established. This transformation can be performed by means of e.g., the QR decomposition of the channel matrix, as described in section 2.2.4. The triangular structure of the resulting \mathbf{R} matrix allows an ordered layer-wise exploration of the tree from the root to the leaves level, thus enabling the implementation of efficient tree traversal strategies. As in the case of the SIC approach, the interference among layers can be successively suppressed by applying (2.19). The metrics λ_0 required for the LLRs computation in (2.22) are recursively calculated by accumulating the layer metrics of (2.21), resulting in:

$$\lambda_i = \lambda_{i+1} + \lambda_{(i)} = \underbrace{\lambda_{i+1}}_{\substack{\text{metric from} \\ \text{already estimated} \\ \text{symbols}}} + r_{ii}^2 \underbrace{\|y_i''' - \hat{x}_i\|^2}_{\substack{\text{interference} \\ \text{reduced} \\ \text{symbol}}} - \underbrace{\frac{N_0}{2} \sum_{j=0}^{L-1} c_{i,j} L_a(c_{i,j})}_{\substack{\lambda_a(\hat{x}_i) \\ \text{(a priori information)}}, \quad (2.23)$$

where λ_i ($i > 0$) is referred to as a *partial* metric and λ_0 denotes the total path metric, i.e., the metric corresponding to a complete MIMO symbol vector $\hat{\mathbf{x}}$. Note that the contribution $\lambda_a(\hat{x}_i)$ may increase or decrease the value of λ_i , depending on both $c_{i,j}$ and the sign of $L_a(c_{i,j})$. This may lead to the exploration of unfavorable nodes during the tree search as well as to the exclusion of favorable ones. In order to avoid this effect, monotonously

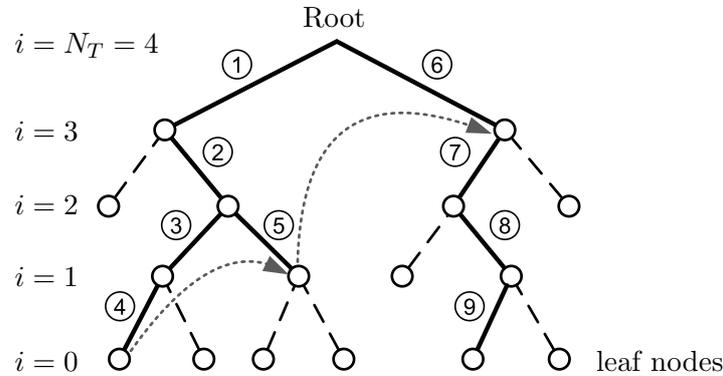


Figure 2.1: Tree search example for a BPSK modulation ($L = 1$) and $N_T = N_R = 4$ antennas. Dashed lines represent pruned paths.

increasing λ_i is considered in this work, by redefining λ_a as [52]:

$$\lambda_a(\hat{x}_i) = - \sum_{j=0}^{L-1} (|L_a(c_{i,j})| - c_{i,j}L_a(c_{i,j})) = -2 \sum_{\substack{j=0 \\ \text{with } c_{i,j} \neq \text{sign}(L_a(c_{i,j}))}}^{L-1} |L_a(c_{i,j})|. \quad (2.24)$$

2.3 Tree Search Detection: Algorithms and Strategies for Complexity Reduction

As previously introduced, tree search detection represents a promising approach to estimate the ML/MAP/max-log-APP solution while avoiding an exhaustive evaluation of all possible transmit symbol vectors. The computational effort incurred by an unrestricted search process can nevertheless be unacceptably high and, for this reason, supplementary mechanisms are required in order to attain a reduction in complexity. In the following, an overview of the most relevant tree search techniques and strategies for complexity reduction is provided.

2.3.1 Tree Traversal Strategies

The aim of tree-search approaches is to explore only a subset of the 2^{LN_T} possible vector candidates during the search for the minima in (2.22), as described in the previous section. For this purpose, several detection algorithms have been proposed in literature, which can be classified according to the underlying tree traversal strategy, as described in the following.

Depth-First Search

Depth-first algorithms descend along a direct path from the root to a leaf, exploring the nodes within the selected path in a sequential manner. As long as possible, the algorithm proceeds downwards through the tree repeatedly (*forward processing*), i.e., until a leaf is reached or until all available nodes at a particular layer have been explored. In this case the algorithm returns to the closest upper layer containing unexplored branches (*back tracking*), selects a different branch and initiates the successive downward traversal behaviour again. This procedure is continuously repeated until no more nodes are available. The sphere detection (SD) approach [24, 53], which represents the quintessential depth-first tree-search algorithm, introduces a certain metric constraint (the *radius*), in order to limit the search space. This kind of algorithms present thus a rather nondeterministic tree traversal behaviour, imposing control-flow and data dependencies which impede straightforward exploitation of parallelization techniques. Another disadvantage with regard to a practical implementation is the fluctuating execution time, which ultimately results in

nonconstant processing throughput -decreasing as the number of visited nodes increases. This is obviously inconvenient for applications requiring constant data rates or the guarantee of a minimum throughput. However, the variable and nondeterministic character of depth-first SD strategies also represents the major advantage of these approaches: the possibility to adapt the search space dynamically, depending on varying channel conditions or on application requirements.

MIMO SD approaches have been intensively investigated during the last decade, leading to a wide spectrum of algorithm variations. They mainly differ in the strategy followed to collect the minima required for the LLRs computation in (2.22) and to define the radius constraint. The original SD algorithm [53], firstly employed for MIMO detection in [54], searches for the hard ML solution and imposes a fixed radius constraint. The so-called Repeated Tree Search (RTS) [55] was one of the first solutions proposed to generate soft information. To this aim, the tree is traversed repeatedly, firstly searching for the ML hypothesis, and subsequently looking for the counter-hypotheses required for the LLRs computation. This is however an inefficient approach, since it obviously entails a large number of redundant computations. In contrast, the Single Tree Search (STS) [56] and the List Sphere Detection (LSD) [24] algorithms search for the ML/MAP hypothesis and the counter-hypotheses concurrently, exploring each node within the tree at most once. Despite the naming distinction, actually both SD algorithms store a set of candidate (counter-)hypotheses. The key difference actually resides in the way the list is handled and the radius is defined. LSD algorithms keep the best K candidates found during the tree search, sorted in ascending order of their metrics. As soon as a new path metric is computed, the list is tentatively updated and re-sorted, regardless of the bit values of the corresponding demapped symbols. The radius may be shrunk during the tree search, adopting the value of the maximum metric in the list as in e.g., [11]. In contrast to this, STS algorithms store the best candidate metric found specifically for each of the LN_T LLRs to be computed, making an explicit distinction of the bit values. The search radius is determined for each (partial) node separately, as the maximum metric from the set of stored counter-hypotheses. As shown in [8], LSD leads to a modestly lower number of explored nodes while providing the same error-rate performance than STS, at a slightly increased computational cost due to the comparison and sorting operations required to update the list. It should be nevertheless noticed that the quality of the approximated LLRs is limited by the list size. LSD may therefore lose potentially valuable candidates which are examined but are finally excluded from the list. Recently, the so-called tuple-search sphere detector (TSD) with bit-specific candidate determination [8], benefiting from both LSD's and STS's assets, has been proposed. On the one hand, a list (or *tuple*) containing the best T candidate metrics found (generally satisfying $T < K$) is stored, with the t -th element defining the radius value. On the other hand, the best metrics found specifically for each bit are kept

separately for the LLR computation, as done by the STS approach. By these means, the complexity-error-rate trade-off is improved with respect to LSD and STS [8].

Breadth-First Search

Breadth-first algorithms compute and sort the metrics corresponding to a certain number of *contending* paths at a particular layer, and subsequently select a set of *surviving* nodes before proceeding to the next layer, starting at the root and finishing at the leaves. The number of tree nodes visited by this procedure is generally predefined, which on the one hand leads to an advantageous predictable runtime and thus constant processing throughput but, on the other, impedes dynamically adapting the size of the search space (and consequently, the computational effort) to varying channel conditions. Besides the fixed runtime property, a major advantage of breadth-first approaches is the proneness to parallelization, since several tree branches can be processed independently at each layer. A typical hazard affecting these algorithms is the possible exclusion of the ML/MAP estimate or relevant contributions for a good approximation of the *max-log*-APP solution. In addition to this, the explored tree paths must be generally stored, which may incur considerable memory costs especially if soft information must be generated.

One of the firstly proposed breadth-first approaches is the M-Algorithm [57], which selects the M nodes with the best (partial) metrics at each layer. The incorporation of a certain metric constraint in order to further shrink the search space led to the so-called T-Algorithm [58]. These approaches evolved later on into one of the most prominent breadth-first strategies nowadays, the K-Best algorithm [23, 59, 60], which explores a maximum of K nodes (per layer) whose metrics comply with a specific radius constraint. While these algorithms impose a unique constraint on the number of nodes to be explored at each layer, recent works (e.g., [61]) employ a different value K_i for each layer i in order to further reduce the complexity. The fixed-complexity sphere detector (FSD) [62, 63] also follows this principle. It differs from K-Best mainly in the fact that the constraint on the number of nodes is imposed to each parent node instead of to each tree layer, leading to a even more deterministic tree traversal behavior. The complexity of these algorithms is generally dominated by the number of metric computations as well as sorting and list maintenance operations involved. Note that most of these approaches incorporate some radius constraint to easily discard tree branches and, consequently, they are frequently regarded in literature also as sphere-detection algorithms.

Best-First Search

Best-first (also referred in literature as *metric-first*) approaches explore firstly the most promising candidates with regard to their partial metrics. In contrast to the previous strategies, the selected nodes may belong to different sub-trees and be located at different tree layers, exchanging the previously described “layer-wise” tree search by a rather

“tree-hopping” traversal strategy. A representative example is the so-called list-sequential (LISS) detector, which collects partial vector candidates in a stack sorted according to their metrics. The search is performed by selecting the best path in the stack which has not yet reached its full length [30]. By these means, the probability of missing the ML/MAP solution is considerably reduced in comparison to the previously discussed approaches. However, the complexity of the algorithm’s control flow is much higher, as so is the computational cost due to the required stack update and sorting operations.

2.3.2 Tuning the Error-Rate-Complexity Trade-off at Pre-Processing

As previously discussed, a triangularization of the channel matrix is performed (e.g., by applying QRD) in order to successively decouple the received signals and to allow an ordered exploration of the tree from the root to the leaves layer, therefore enabling the application of efficient tree traversal approaches. But before the tree search takes place, several additional strategies can be applied to reduce the search space and to additionally improve the error-rate performance. In the following, the most widely applied techniques are briefly described.

Layer Ordering

Tree-search approaches apply the successive interference cancellation principle on each of the explored tree paths and, consequently, these algorithms also suffer from error propagation to some extent. As discussed in section 2.2.4, this problem can be easily mitigated by employing the sorted QRD (SQRD) strategy to allow processing the most reliable signals firstly, thus reducing the probability of wrong decisions at the upper tree layers. For this purpose, the diagonal coefficients of the \mathbf{R} matrix are (ideally) sorted⁵ in ascending order ($r_{ii} < r_{jj}, \forall i < j$). After concluding the tree search, the computed LLRs must be reordered according to the permutation introduced by \mathbf{P} in (2.15). By these means, the error-rate performance is improved in the case of algorithms with fixed complexity (e.g., SIC, FSD), as shown in e.g., [29, 64]. In the case of algorithms with variable search complexity (e.g., depth-first SD), the amount of nodes explored is significantly reduced, as observed in e.g., [10, 65].

MMSE Channel Extension

The improvement in error-rate performance due to the incorporation of the noise variance into LD and SIC processing has been discussed in sections 2.2.3 and 2.2.4, respectively. In the case of (S)QR-based tree-search approaches with variable complexity, introducing

⁵As described in section 2.2.4, SQRD is a greedy approach which approximates the optimal order.

the MMSE criterion (2.15) has rather an impact on the search complexity, as shown in e.g., [65, 66] and detailed in the following. The SNR at each tree layer i depends on the corresponding (squared) diagonal element $|r_{ii}|^2$ of the matrix \mathbf{R} , as shown by (2.17). Consequently, close-to-singular (i.e., ill-conditioned) channels resulting in low effective SNR on one or several of the spatial streams are prone to cause a large number of candidates to be contained within the radius. This effect can be mitigated by “regularizing” the channel matrix as proposed in [48], i.e., taking into account the noise variance at the receiver by applying (2.15). This reduces the condition number⁶ of the effective channel matrix and thereby the search complexity, at the cost of minor error-rate performance degradation. This loss is a consequence of $\tilde{\mathbf{n}}$ not having the same statistics as \mathbf{n} and containing self-interference (i.e., depending on \mathbf{x} , as observed in (2.18)), as well as of \mathbf{Q}_1 not necessarily being unitary (even though \mathbf{Q} do is) [10]. As shown in [67], the MMSE-based extension of the channel matrix incurs a data-dependent contribution (i.e., self-interference) in the Euclidean distance:

$$\|\mathbf{y} - \mathbf{H}\mathbf{x}\|^2 = \|\mathbf{y}' - \mathbf{R}\hat{\mathbf{x}}\|^2 - \sigma^2 \|\hat{\mathbf{x}}\|^2 \quad (2.25)$$

which does not necessarily cancel out in the LLR computation (2.22) and should be therefore removed to avoid degrading the error-rate performance [10, 67]. As proposed in [67] and applied in [8] to the TSD algorithm, in this work the bias term is subtracted from the metric values before computing the LLRs:

$$\lambda_0^{\text{ub}} = \lambda_0 - \sigma^2 \|\hat{\mathbf{x}}\|^2, \quad (2.26)$$

$$L(c_{m,l}|\mathbf{y}) \approx -\frac{1}{N_0} \min_{x \in \mathcal{L}|c_{m,l}=+1} \{\lambda_0^{\text{ub}}\} + \frac{1}{N_0} \min_{x \in \mathcal{L}|c_{m,l}=-1} \{\lambda_0^{\text{ub}}\}. \quad (2.27)$$

Note that by this mean self-interference is compensated but $\tilde{\mathbf{n}}$ is no longer a i.i.d. circularly symmetric complex Gaussian-distributed random variable with variance N_0 , which causes a negligible error-rate performance loss [68]. This strategy is applicable to most list-based tree-search algorithms since the candidate symbols $\hat{\mathbf{x}}$ are collected and stored along with the metric values during the tree search, whereas the STS algorithm is generally an exception. An alternative approach to compensate the self-interference in this case, as well as further corrections to the LLRs computation, are proposed in [68].

Lattice Reduction

The aim of lattice reduction (LR) techniques is to solve the detection problem onto a relaxed infinite lattice with a nearly orthogonal basis. This can be performed by applying the widely known Lenstra-Lenstra-Lovasz (LLL) algorithm [69], at the cost of increasing the receiver’s complexity [29]. In combination with low-complex detection algorithms such

⁶A matrix is said to be well-conditioned, if its condition number is 1 (or close to 1).

as LD and SIC, the detection error-rate performance is considerably improved [70]. In the case of tree-search algorithms, however, relaxing the finite lattice to an infinite one results in a larger search space, hence becoming a rather disadvantageous approach [71]. Even though LR methods can be advantageous for hard-output MIMO detection, as shown in e.g., [72] for a K-Best detector, generating soft information is a major challenge since the original constellation mapping can not be directly applied to the new lattice space [40]. The application of LR techniques will be therefore not considered in this work.

2.3.3 Tuning the Error-Rate-Complexity Trade-off at Runtime

Besides the above described mechanisms, applied at the *pre-processing* stage, additional strategies may be employed in order to reduce the complexity *during* the tree search process, as described in the following.

Radius Reduction

The introduction of a radius constraint R represents the main defining characteristic of the sphere detection approach. In order to avoid exploring the complete tree of possible transmitted MIMO symbols, only those paths satisfying $\lambda_0 < R$ are examined. Subtrees whose partial metrics exceed the radius (i.e., $\lambda_i > R$) may be additionally pruned in advance, since the descending accumulated metrics (i.e., at layers $i - 1, \dots, 0$) have necessarily a value larger than λ_i . This leads to a significant reduction of the node count, especially if layer ordering is applied. The specific value of R has been shown to be a crucial factor affecting the detection performance [36, 40]. In this regard, two strategies can be distinguished, namely fixed radius [54] and adaptive radius [24, 71]. The first strategy can be very easily implemented but presents two major drawbacks. Firstly, a suitable radius value guaranteeing a good error-rate-complexity trade-off has to be determined and, secondly, the search process lacks adaptability to system or channel variations. These disadvantages are mitigated by the adaptive strategy, which initializes the radius to a certain value (e.g., $R = \infty$) and updates (shrinks) it repeatedly, as soon as a better metric λ_0 is found. This strategy achieves the greatest reduction in complexity (in terms of the amount of nodes examined), but some additional conditions are required in order to avoid constraining the search space too much, especially when soft-information needs to be generated. Otherwise, an insufficient number of tree paths and consequently of counter-hypotheses might be examined, hence degrading the error-rate performance [8].

(Radius) Clipping

Tree-search algorithms are suboptimal detection approaches which may not find a counter-hypothesis for each of the bits in the received vector. This causes the resulting extrinsic information to take large values, misleadingly interpreted by the decoder as highly reliable

decisions on the affected bits. In this regard, clipping the extrinsic information $|L_e|^{\text{clip}} = \min\{|L_e|, L^{\text{clip}}\}$ before transferring it to the decoder is a commonly applied method to avoid overflows as well as overestimation of the detected bits' probabilities [11], as further detailed in section 2.4. Choosing a too large value for the clipping factor $L^{\text{clip}} = L_e^{\text{max}}$ may cause the decoder to assume a too high reliability of the bits, whereas setting L^{clip} too low will lead to the opposite effect. In the first case, decision errors might be prevented from being corrected while, in the second, non-existing errors might be indeed induced. The particular limiting value L_e^{max} , crucial for good performance, is chosen so that the mutual information exchange at the output of the detector is maximized [73], as proposed in [40, 74]. In iterative signal processing the maximum *a priori* probability value is additionally incorporated, redefining the clipping factor as

$$L^{\text{clip}} = L_e^{\text{max}} + L_a^{\text{max}} = L_e^{\text{max}} + \max_{i,j}\{|L_a(c_{i,j})|\}. \quad (2.28)$$

The main idea of the *radius clipping* mechanism consists in pruning in advance those subtrees belonging to paths whose extrinsic probabilities will be nonetheless clipped with L^{clip} [52] as previously described. The clipped radius R_{clip} is then determined as the minimum between R and the clipping factor defined as:

$$R_{\text{clip}} = \min\{R, \lambda_0(\mathbf{c}^{\text{ML}}) + N_0 L^{\text{clip}}\}. \quad (2.29)$$

Further Strategies

Further runtime strategies for complexity reduction include other mechanisms affecting the radius (e.g., statistical tree pruning [75]) as well as the introduction of fixed constraints limiting the tree dimensions -such as the maximum number of tree branches, metric computations or node extensions- (e.g., early termination [76]). However, strategies which merely “crop” the search space usually incur a certain error-rate performance degradation, drawback which makes their application rather unattractive. An overview of these techniques and their impact on detection performance and complexity can be found in [40]. The particular node enumeration method employed (i.e., the order in which symbols descending from a certain parent node are examined) also presents a great impact on the search complexity. In this regard, the most popular approach is the so-called Schnorr-Euchner (SE) enumeration [77], which sorts the symbols in ascending order of their partial metrics. Resulting from this, tree paths with favourable metric values are examined first, hence avoiding exploration of subtrees which will be pruned by the radius constraint. Reduced-complexity enumeration approaches approximating the ideal SE ordering are presented in section 3.3. Lastly, different algorithm-specific strategies for complexity reduction may be applied, e.g., varying the size of the underlying list in list-based sphere detection approaches (such as LSD and TSD), or adjusting the clipping factor (as further detailed throughout the next sections).

2.4 The Reduced-Complexity Sphere Detector

From the large variety of tree-search detection strategies previously presented, the tuple search sphere detector (TSD) proposed in [8] has demonstrated to outperform the error-rate-complexity trade-off of contenting sphere detection strategies (such as single tree search (STS) [10], list sphere detection (LSD) [11] or K-best detection [60]) while representing a promising approach towards an efficient practical realization. The TSD detection strategy, which has been the focus of research in this work, will be in the following described along with the most relevant algorithm-specific mechanisms to further reduce the computational effort.

Computing the LLRs in (2.27) requires the determination of a detection hypothesis and LN_T counter-hypotheses, as described in section 2.2.2. Since explicitly searching for all the required minima leads to impractically high computational effort [56], the TSD algorithm restricts the search to a subset containing the T most likely tree paths. The metrics λ_0 of these paths are stored in the so-called search *tuple* $\mathcal{T} := \{\lambda_0(\mathbf{c}_0), \lambda_0(\mathbf{c}_1), \dots, \lambda_0(\mathbf{c}_{T-1})\}$ in ascending order ($\lambda_0(\mathbf{c}_0) < \lambda_0(\mathbf{c}_1) < \dots < \lambda_0(\mathbf{c}_{T-1})$). This list is initialized with $\lambda_0(\mathbf{c}_t) = \infty^7$ and the value at any entry $t \in \{1, \dots, T-1\}$ is replaced by any newly found metric value $\lambda_0(\mathbf{c})$ satisfying $\lambda_0(\mathbf{c}_{t-1}) < \lambda_0(\mathbf{c}) < \lambda_0(\mathbf{c}_t)$ (for $t = 0$ the condition $\lambda_0(\mathbf{c}) < \lambda_0(\mathbf{c}_0)$ is sufficient). Those metric values previously stored in the entry range $[t, T-1]$ are simply shifted one position up in the list, thus occupying the range $[t+1, T]$, whereas the T -th position is not existing and the assigned element will be hence dropped. By these means, the ascending metric order is continually guaranteed. Similarly to the LSD procedure, the sphere radius is defined as the maximum metric in the tuple:

$$R = \max_{\mathbf{c}_t | \mathbf{c}_t \in \mathcal{T}} \{\lambda_0(\mathbf{c}_t)\} = \lambda_0(\mathbf{c}_{T-1}). \quad (2.30)$$

Resulting from this, the radius is initialized with $R = \infty^7$ and scaled down during the tree search, as the metric value $\lambda_0(\mathbf{c}_{T-1})$ decreases. In contrast to the LSD approach, the potential counter-hypotheses are not extracted from the tuple (list). Instead, candidates for the LLRs computation are found during the tree search and stored separately for each of the LN_T bits [8], as likewise done by the STS procedure. It should be noticed that, since the candidates for the LLRs computation are generated from a reduced subset of the paths examined during the tree search, finding a counter-hypothesis for each of the LN_T bits cannot be guaranteed. The non-found counter-hypotheses, originally initialized to ∞^7 , cause misleading extrinsic information for the channel decoder, which interprets these magnitudes as highly reliable decisions on the corresponding bits. Consequently, the overestimated extrinsic bits have to be clipped ($|L_e|^{\text{clip}} = \min\{|L_e|, L_e^{\text{max}}\}$) in order

⁷When using fixed-point arithmetic, the initial value ∞ is replaced by the maximum quantity representable by the corresponding bit-precision.

to avoid degrading the error-rate performance. As shown in [73], defining the clipping limit L_e^{\max} such that the average mutual information at the detector output is maximized leads to good error-rate performance. By combining the tuple-search mechanism for the radius update with the bit-specific candidate determination approach, the resulting TSD algorithm improves the error-rate performance with regard to the LSD strategy, while significantly reducing the average number of examined nodes ($\mathbb{E}[n]$) with respect to STS detection, as shown in [8] and illustrated in section 4.2. Furthermore, varying the trade-off between detection accuracy and complexity (in terms of $\mathbb{E}[n]$) is enabled by the adjustable size of the tuple T .

In order to minimize the search complexity (i.e., $\mathbb{E}[n]$), the widely known preprocessing strategies described in section 2.3.2 may be applied. In particular, noise-aware layer ordering is implemented by employing the MMSE-SQRD mechanism described in section 2.2.4.1. Note that the separate storage of counter-hypotheses (including not only the metric values λ_0 but also the corresponding constellation symbol vectors $\hat{\mathbf{x}}$), permits eliminating the bias from the metrics (2.26) prior to the LLRs computation according to (2.27), as detailed in section 2.3.2. Several of the runtime approaches for complexity reduction described in section 2.3.3 are additionally considered. Besides adjusting $\mathbb{E}[n]$ through the tuple size T , as previously mentioned, the radius clipping and the early tree-search termination mechanisms are supported. The latter is performed by constraining $\mathbb{E}[n]$ to a maximum value n_{\max} , which allows guaranteeing a minimum detection throughput. Furthermore, by setting $n_{\max} = N_T$, SIC (section 2.2.4) detection performance is obtained. Tree-pruning mechanisms have been shown in [78] to provide only a marginal benefit in complexity reduction which does not compensate the consequent loss in error-rate performance, and are therefore not considered in this work. Even though the TSD algorithm, in combination with these strategies, already represents an effective approach to drastically decrease $\mathbb{E}[n]$, it is possible to further reduce the detection's computational effort by additionally lowering the amount of computations performed and simplifying (e.g., approximating) the required operations. Relevant techniques to achieve this are described in the following. The resulting reduction of the computational effort has a direct impact on the complexity of the resulting VSLI implementation, especially concerning area, throughput and power consumption, as further detailed in chapter 6.

2.4.1 Estimating the Node Enumeration Sequence

The complexity of the tree search process is strongly influenced by the order in which the constellation symbols descending from a parent node are explored during the tree search. Tree paths which are advantageous for (2.27) (i.e. presenting small metrics) should be explored firstly in order to avoid spending computational effort on subtrees

which will be eventually pruned by the radius constraint. The Schnorr-Euchner (SE) enumeration [77] is a widely known strategy defining the ideal node sequence, which requires computing the metrics of all constellation symbols and sorting them in ascending order. The resulting enumeration sequence \mathbf{v} is defined as an ordered set of constellation symbols $\mathbf{v} := [\hat{x}_i^0, \hat{x}_i^1, \dots, \hat{x}_i^p, \dots, \hat{x}_i^{(Q-1)}]$ with metrics satisfying $\lambda_i(\hat{x}_i^0) \leq \lambda_i(\hat{x}_i^1) \leq \dots \leq \lambda_i(\hat{x}_i^p) \leq \dots \leq \lambda_i(\hat{x}_i^{(Q-1)})$. The complexity of the SE method is nevertheless outrageous since the described procedure must be reiterated for each parent node in the tree. Moreover, all symbols whose metrics violate the radius constraint ($\lambda_i(\hat{x}_i^p) > R$) will not be explored and hence enumerating them should be avoided.

Assuming equal bit probability (as in the case of non-iterative detection-and-decoding, where *a priori* information is not available) the contribution $\lambda_a(\hat{x}_i)$ to the metric computation in (2.23) has no effect on the resulting enumeration, which is thus uniquely depending on the (squared) Euclidean distances between the (normalized) interference-reduced received signal and the constellation symbols:

$$d(y_i''', x_i^p) = d_i^p = \|y_i''' - \hat{x}_i^p\|^2, \quad \text{with } p = \{0, \dots, Q-1\}. \quad (2.31)$$

On the basis of (2.31) the enumeration can be directly visualized on the constellation plane, where geometrical properties can be exploited in order to approximate the ideal SE ordering with a significantly lower computational cost. The circular or column-wise zig-zag enumerations employed in [31] and [79], respectively, as well as the sector-based approach proposed in [9] are some examples. The latter, so-called search sequence determination (SSD) is a heuristic method which divides the constellation space into geometrical decision regions, each associated to a sequence of symbols approximating the ideal SE enumeration. The SSD strategy replaces the numerous metric and sorting operations required by the SE enumeration by a few inexpensive basic operations (such as sign checks, comparisons and bit shift operations). The symbols are enumerated according to a geometrical position analysis relative to reference nodes \hat{x}_i^{ref}

$$\hat{x}_i^{\text{ref}} = \lfloor y_i''' \rfloor = \lfloor \frac{y_i''}{r_{ii}} \rfloor_x \quad (2.32)$$

(with $\lfloor \cdot \rfloor_x$ representing a rounding operation to the closest constellation symbol). The corresponding enumeration is determined by a fixed node sequence associated to the decision region where y_i''' lays. As proposed in [9], two different enumeration approaches may be applied depending on the tree layer i :

1. At layers $i \in \{1, \dots, N_{T-1}\}$, the node enumeration is based on the relative position between y_i''' and \hat{x}_i^{ref} , which is quantized to a certain number of geometric regions, as exemplified in Figure 2.2(a). The accuracy of this approach is variable, depending

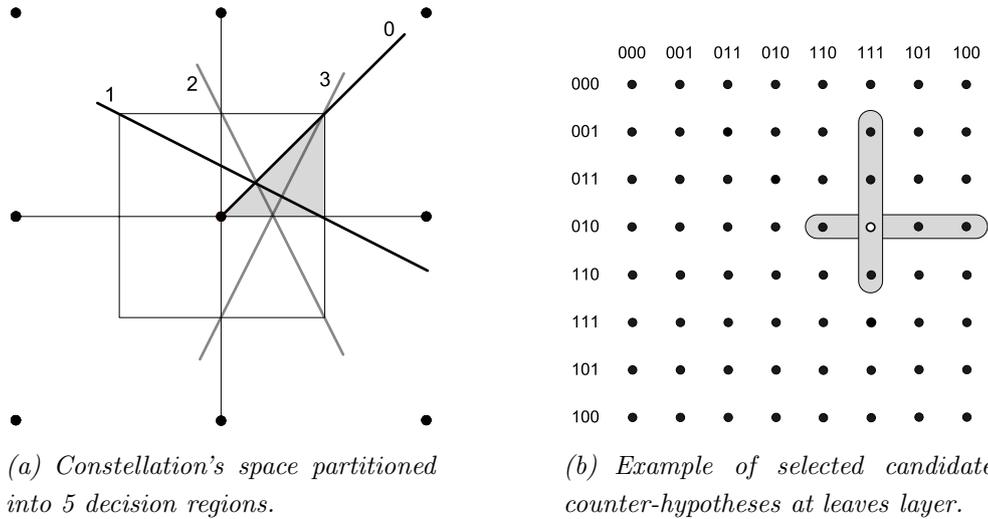


Figure 2.2: Search sequence determination approaches (64-QAM).

on how many regions, and thus enumeration sequences, are considered. As demonstrated in [9], a maximum of 5 decision regions shall be employed, since no significant benefit is obtained by further dividing the constellation space. By means of this approach, the costly node enumeration task is simplified to an elementary geometrical region differentiation. The computational complexity of this strategy can be further decreased by reducing the amount and length of the considered sequences.

- At layer $i = 0$, sequences explicitly intended to find counter-hypotheses by exploiting the constellation mapping properties are rather employed (Figure 2.2(b)). Due to the convenient Gray-code mapping, those constellation symbols laying on the same real and imaginary axes as \hat{x}_i^{ref} (i.e., the potential hypothesis) represent the best (in terms of Euclidean distance) candidate counter-hypotheses [9]. The candidate set \mathcal{B} is hence comprised by one hypothesis (\hat{x}_i^{ref}) and L counter-hypotheses, which are enumerated in ascending order of their geometrical distances to y_i^{ref} .

In the presence of *a priori* information, however, the metric values do not depend uniquely on the Euclidean distances but also on the contribution $\lambda_a(\hat{x}_i)$ and, consequently, a sorted sequence of symbols can not be predicted by solely examining d_i^2 . Such Euclidean-distance-based enumeration strategies are thus suboptimal and may lead to considerable error-rate performance degradation [80][†], as shown in section 4.4.1. A more detailed insight on this problem as well as an overview of existing and proposed solutions are presented in chapter 3.

2.4.2 Parallel Sibling Node Processing

The SSD strategy reduces the computational complexity considerably, as previously discussed, but it introduces a penalty on the number of internal cycles required by the tree search [9]. This drawback is a direct consequence of the enumeration process providing nodes in a sequential and on-demand manner, assuming a so-called one-node-per-cycle architecture [31]. Tree nodes are not enumerated until they are required by the search algorithm and, consequently, a direct selection of the next sibling node to be examined is not immediately possible. Instead, an additional cycle is required in order to firstly determine this node and its corresponding partial metric. In order to mitigate the increased cycle count problem, parallel processing of parent nodes has been proposed in [9]. According to this principle, a next favourable sibling parent node must be already available whenever the decision to extend this node is made, as exemplified in Figure 2.3. For this purpose, within each tree search cycle a sibling node is computed ahead, in parallel to the currently examined one. In addition to this, parallel processing of leaf nodes is proposed in order to reduce the cycle count further [9].

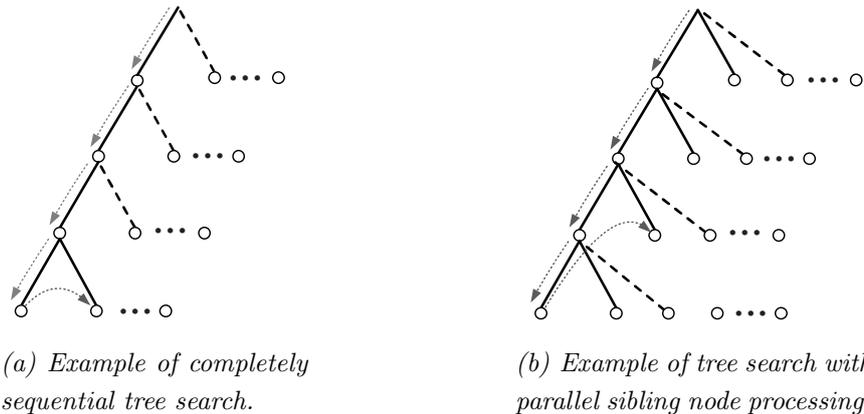


Figure 2.3: Comparison of sequential and parallel sibling node processing. Dashed lines represent not-yet-examined paths.

2.4.3 Estimating the Metrics

The computationally intensive operations required for the metric calculation in (2.23) can be considerably simplified by an estimation based on the sector-aided enumeration approach described in section 2.4.1. The distances d_i^p between the constellation symbols and the (normalized) interference-reduced received signal can be replaced by predefined geometrical distances $d_i^{p'} = d(z_i^{\text{ref}}, x_i^p)$ [28], [81][†] between the constellation symbols and fixed reference points z_i^{ref} (such as the geometric centers of the defined decision regions):

$$r_{ii}^2 (d_i^p)^2 = r_{ii}^2 \|y_i''' - \hat{x}_i\|^2 \approx r_{ii}^2 \|z_i^{\text{ref}} - \hat{x}_i\|^2 = r_{ii}^2 (d_i^{p'})^2, \quad (2.33)$$

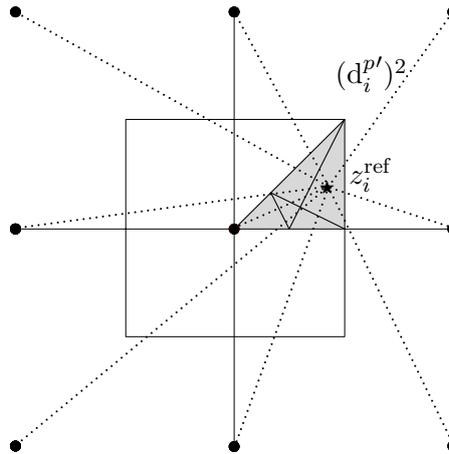


Figure 2.4: Geometrical distances between constellation symbols and the centroid of one of the triangular decision regions.

as illustrated in Figure 2.4 for one of the depicted sample decision regions. It is additionally possible to precalculate r_{ii}^2 as well as $r_{ii}^2(d_i^{p'})^2$, consequently simplifying the complex-value products in (2.23) to a single real-value multiplication⁸ or even causing this operations to vanish completely. The low computational complexity of this approach makes it seemingly attractive for practical hardware implementations. However, it should be noticed that the apparent simplification comes at the cost of degrading the detection error-rate performance (due to the loss of accuracy), and increasing the memory (to store the precomputed (squared) distances $(d_i^{p'})^2$ or products $r_{ii}^2(d_i^{p'})^2$), as shown in sections 4.2 and 6.3, respectively. Note that the additive contribution of the *a priori* information $\lambda_a(\hat{x}_i)$ is not part of the estimation in (2.33) and is nevertheless computed according to (2.23) and (2.24).

2.5 Complexity, Performance and Efficiency Metrics

As already emphasized in previous works from literature (e.g., [36, 82, 83]), comparing different baseband digital signal processing solutions is a non-trivial, burdensome challenge. The major difficulty in this regard is introduced by the great diversity of wireless communications standards, which define a large variety of scenarios and configuration parameters to be considered. In addition to this, the numerous constraints imposed not only by the standards, but also by the applications' and users'/market demands, lead to diverse optimization trade-offs to be pursued as well as to a broad collection of metrics to be examined to evaluate the resulting solutions. In order to perform an unbiased, equitable comparison, different metric normalization methods may be applied. In the following, the metrics considered in this work and their dependance on the transmission and system setup are

⁸For a conveniently chosen QRD, r_{ii} only contains positive real values.

described.

2.5.1 On the Algorithmic Perspective

Signal to Noise Ratio (SNR)

SNR is defined as the ratio of signal power P_s in a specific bandwidth over the locally present noise power P_n in the same bandwidth:

$$\text{SNR} = \frac{P_s}{P_n}$$

with $\text{SNR}^{\text{dB}} = 10 \log_{10}(\text{SNR})$. The SNR averages therefore the communication signal over possibly infinite time. In digital wireless communications systems, however, a normalized measure of communications performance which is quantized to finite time intervals and is independent of the bandwidth is generally more practical. A widely applied example of such a metric is the information **symbol-energy-to-noise-density ratio**:

$$\frac{E_s}{N_0} = \frac{P_s t_s}{P_n / B} = \frac{\text{SNR}}{f_s / B} \equiv \frac{E_b}{N_0} \frac{N_T L R_c}{N_R}, \quad (2.34)$$

with $f_s = 1/t_s$ and t_s representing the sampling frequency and the symbol period, respectively, and B representing the (noise equivalent) bandwidth. Note that the E_s/N_0 is thus related to the SNR by the bandwidth efficiency f_s/B . For a coded system with coding rate R_c and a modulation scheme employing L bits to represent each symbol, the E_s/N_0 can be also expressed as a function of the **bit-energy-to-noise-density ratio** E_b/N_0 (right-most expression in equation (2.34)). Assuming that the system is operating at the Nyquist bandwidth $B = f_s/2$ and that the dominant source of noise is AWGN with noise density $N_0/2$ per real dimension, the SNR equals the information symbol-energy-to-noise-density ratio ($\text{SNR} = E_s/N_0$) [84].

Error Rate

The error rate is an indicator of the communications reliability, typically expressed in terms of bit error rate (BER) or frame error rate (FER). The BER and FER metrics represent the ratio of incorrectly decoded information bits and frames, respectively, over the corresponding total transmitted information. BER and FER are typically represented as a function of SNR (or, alternatively, of E_s/N_0 or E_b/N_0). For this reason, error-rate performance degradation/gain is frequently expressed in terms of the required increase/decrease in SNR (or E_s/N_0 or E_b/N_0) to reach the same BER/FER. Note that, while the BER is independent of the frame structure, the FER is strongly influenced by the frame size, which should be therefore taken into account when comparing algorithms applied to different transmission schemes. Analogous to FER, the block error rate (BLER) and the packet

error rate (PER) can be defined. The notation and definition of “frame/block/packet” varies with the wireless standard. BLER is considered in e.g., 3GPP LTE(-Advanced) standards, whereas PER is commonly examined in e.g., IEEE 802.11/16 WLAN/WiMAX standards. In 3GPP UMTS standards FER is rather considered. In this work, the FER notation will be indistinctively employed (the definition of “frame” will be presented later on). The numerous communications standards also define diverse acceptable error-rate thresholds, such as 10% PER for IEEE 802.11n WLAN [85], 10^{-6} BER for IEEE 802.16 WiMAX [86] and 10% BLER for 3GPP LTE/LTE-Advanced [22]. Whenever the link quality degrades beyond the specified threshold different strategies are implemented, varying with the transmission mode and the standard. Decreasing the modulation order and/or the code rate, choosing a different transmission channel or handing the communication over to a different base station are some commonly applied examples.

Complexity

Measuring and comparing the complexity of different digital baseband approaches from the algorithmic perspective is a burdensome task. The different arithmetic precisions employed, the varying complexity of the performed computations and the undetermined memory and control logic requirements permit only a rough estimation of the conceivable implementation complexity and hinder equitable comparison of algorithms. The most commonly examined metric in this regard is the computational complexity, expressed in terms of the number of “costly” operations performed with a particular precision. The kind of operations to be considered differs, however, depending on the underlying implementation technology. For floating-point DSPs, arithmetic operations generally entail comparable execution times and therefore the number of floating-point operations is a suitable indicator of an algorithm’s computational complexity. In the case of dedicated fixed-point solutions, in contrast, a differentiation is required since the various operation types usually entail different costs which moreover vary depending on the employed arithmetic precision. Hardware accelerators further worsen the problem since sequences of basic computations are merged into faster single-cycle custom operations. The arithmetic operations count may be consequently misleading and it generally represents a poor estimate of the integrated circuit complexity, provided that additional hardware costs incurred by parallelization, control logic and memory are not accounted for. An additional cause of unequitable comparisons between different detector solutions is represented by the fact that the incurred complexity may vary depending on given latency and/or energy-consumption constraints. In this regard, a combination of complexity and performance metrics (such as the cost and efficiency factors described in section 2.5.2) is required.

Besides technology-dependent complexity metrics, the different functional nature of the considered detection approaches also calls for algorithm-specific metrics, which usu-

ally cannot be directly translated into other magnitudes for comparison. In the particular case of tree-search detection approaches, a representative metric of the algorithm's computational complexity is provided by the number of nodes $\mathbb{E}[n]$ examined within the tree. Even though this metric is applicable to all tree-search based detectors, its use presents three main disadvantages: i) the computational complexity entailed by a node extension operation may vary depending on the tree-search approach, ii) this complexity cannot be compared to or translated into other complexity metrics (such as e.g., matrix inversions) in a straightforward way, and iii) it has different implications in the posterior hardware implementation, varying with the employed tree-search scheme. For instance, a higher number of explored nodes typically leads to larger area in the case of breadth-first algorithms (exploring several branches in parallel), while in the case of depth-first algorithms it rather has a direct impact on the detection runtime and thereby on the processing throughput. For these reasons, in this work the number of nodes is only considered in order to compare several optimized variants of the same algorithm. For comparison to other works the actual integrated circuit complexity will be examined, as detailed below.

Search Efficiency

In addition to evaluating the error-rate performance and $\mathbb{E}[n]$, in this work the novel efficiency analysis introduced in [28], [81][†] is applied. For a given BER performance, the LLRs will be approximated on the basis of the potential hypothesis and set of counter-hypotheses found during the tree search, as described in section 2.3. This obviously implies that only a subset of the overall searched nodes belong to tree paths actually contributing to the final LLRs computation. The amount of nodes belonging to these paths, excluding the leaves, represents therefore the minimum number of nodes n_{min} which have to be necessarily examined in order to build the required set of counter-hypotheses. The efficiency of the search algorithm can be thus measured as the inverse of the amount of information unnecessarily processed, i.e., the overhead ratio

$$\partial = \frac{\mathbb{E}[n]}{n_{min}^{STS}}, \quad (2.35)$$

defined as the ratio between $\mathbb{E}[n]$ and n_{min} of the STS approach [8]. STS is taken as reference since it finds all minima $\arg \min_{\hat{\mathbf{x}}(\mathbf{c}) | \mathbf{c} \in \mathcal{C}, c_{m,l} \neq c_{m,l}^{ML}} \{\lambda_0\}$ within the search sphere for the given target error-rate performance.

Transmission and Processing Throughput

The term ‘‘throughput’’ can be generally defined as the ratio between the amount of data produced and the time required to produce this data. In the context of digital wireless communications, **transmission throughput** (also *goodput*) refers to the number of bits successfully delivered over the communication channel during a given symbol period t_s . In

spatial-multiplexing multi-carrier systems with LN_T bits representing each MIMO symbol, the nominal peak transmission data rate can be generally expressed as [87]:

$$\Theta_{\text{nom}} = \frac{LN_T N}{t_s} \quad [\text{bps}], \quad (2.36)$$

where N corresponds to the number of effective (sub-)carriers (for single-carrier systems $N = 1$). Assuming that incorrectly decoded frames are discarded at the receiver and that retransmission of this information is consequently required (as is the case of e.g., simplistic ARQ -automatic repeat request- techniques), the transmission data rate is affected by the reliability of the communication channel, i.e., by the frame error-rate probability:

$$\Theta_{\text{raw}} = (1 - \text{FER})\Theta_{\text{nom}} \quad [\text{bps}]. \quad (2.37)$$

Note that this definition corresponds to the *raw* or *uncoded* transmission throughput. In coded systems, where only a fraction R_c of the bits actually contains information, the *effective* information throughput is hence reduced by the coding rate to $\Theta = \Theta_{\text{raw}}R_c$. So far only the transmission data rate has been considered. However, the signal processing at the receiver side also incurs a certain delay. In this context, the **processing throughput** is defined as the ratio between the amount of detected/decoded bits and the detection/decoding time. For the particular case of the depth-first tree-search detection algorithms focused in this work, the *average* detection throughput is determined as:

$$\bar{\tau}_{\text{raw}} = \frac{LN_T}{l_{\text{cp}}\mathbb{E}[n]} \quad [\text{bits/clock cycles}], \quad (2.38)$$

where $l_{\text{cp}}\mathbb{E}[n]$ denotes the average (over sufficient channel and noise realizations at constant SNR) number of clock cycles required to detect a MIMO symbol vector in the considered scenario (with l_{cp} representing the cycle count per node). Again, the effective information processing throughput is reduced by the coding rate ($\bar{\tau} = \bar{\tau}_{\text{raw}}R_c$). The processing throughput therefore refers to the nominal rate at which the algorithm can make a decision on the transmitted bits and, in contrast to the transmission throughput, it provides no information about the reliability of the decision. The resulting error-free data rate (goodput) experienced by a user will be hence limited by the minimum between the transmission rate and the receiver's processing throughput ($\min\{\bar{\tau}, \Theta\}$).

Spectral Efficiency

The spectral efficiency (also referred in literature as bandwidth efficiency [84]) is a measure of the transmission data rate normalized to the (occupied) spectrum bandwidth:

$$\eta_{\text{BW}} = \frac{\Theta}{B} \quad [\text{bps/Hz}], \quad (2.39)$$

where Θ is here generalized to any of the previously defined (i.e., peak nominal, raw or effective) transmission data rates and B represents the channel bandwidth. It should be noticed that the peak nominal spectral efficiency specified by the wireless communications standards is only achieved under ideal transmission conditions. In practice, as previously discussed, only a fraction R_c of the processed bits actually contain information data and the link unreliability may lead to a high number of retransmissions, thereby reducing the communications data rate. Other factors limiting the throughput are the overhead introduced by the pilot and control channels, the interference caused by transmissions in neighboring cells on the same frequency band (inter-cell interference) and the distribution of cell capacity among users. In addition to this, transmission rates may also be constrained by the resources availability (e.g., large spectrum bands are frequently unavailable), and link adaptation mechanisms may automatically reduce the modulation order and/or the coding rate in order to comply with a certain quality or reliability constraint. According to [84], the actual throughput per cell is in practice about only 30-50% of the theoretical peak values specified by the standards.

2.5.2 On the Integrated Circuits Perspective

Area

While the computation count only permits an inconclusive assessment of an algorithm's complexity, as previously discussed, the silicon area occupied by the algorithm's realization is an indicator of the *true* implementation complexity. Since this metric depends on the transistor feature size, technology scaling must be applied to enable equitable comparison of ICs manufactured on different processes (as further detailed in section 2.5.2.1). The gate equivalent (GE) count is an additional, widely employed, technology-independent indicator of chip complexity. It is determined as the total occupied silicon area A normalized to the area A_{NAND} occupied by a single two-input drive-one NAND standard cell on the corresponding technology:

$$A_{\text{GE}} = \frac{A}{A_{\text{NAND}}} \quad (2.40)$$

In order to compare different algorithm realizations, the area corresponding to logic as well as to memory blocks should be distinguished and taken into account appropriately.

Maximum Clock Frequency

The maximum clock frequency of a synchronous digital circuit is constrained by the combinatorial path (from flip-flop output to flip-flop input) with the largest propagation delay, i.e., by the design's critical path. Specifically, the frequency $f_{\text{clk}}^{\text{max}}$ is inversely proportional to the total delay T_{cp} of the critical path ($f_{\text{clk}}^{\text{max}} = 1/T_{\text{cp}}$). In general, increasing

the clock frequency of a circuit is possible by retiming and inserting *temporal* parallelism, i.e., pipelining the design. Pipelining techniques reduce the propagation time by inserting delay elements along the critical path, whereas the overall processing delay is increased, as further detailed in section 5.2.3.2. Appropriate scaling rules must be applied when comparing $f_{\text{clk}}^{\text{max}}$ on different technologies and/or at different supply voltages, as discussed in section 2.5.2.1.

Circuit Performance

Assessing a design's performance is strongly dependent on the underlying implementation technology. Designers of programmable architectures usually refer to (mega) instructions per second (MIPS), million operations per second (MOPS) or floating-point operations per second (FLOPS). However, the definition and complexity of the considered instructions or operations, especially the application-specific ones, may differ considerably. Furthermore, in designs where the performance is constrained by the communication plane (e.g., memory access) rather than by the computations, the computing capacity does not represent the actual system performance. In the context of digital wireless communications, performance is generally related to the amount of information data processed within a given time period. The processing throughput definition provided in section 2.5.1 for depth-first tree-search detection algorithms will be hence focused in the following. From the integrated circuits perspective, the ratio between the amount of detected bits and the detection time in equation (2.38) is simply extended to include the system clock frequency:

$$\bar{\tau}_{\text{raw}} = \frac{LN_{\text{T}}}{l_{\text{cp}}\mathbb{E}[n]}f_{\text{clk}} \quad [\text{bps}]. \quad (2.41)$$

Due to the dependency on f_{clk} , appropriate technology scaling (2.5.2.1) must be applied when comparing realizations on different processes and/or at different supply voltages. Note that the effective information processing throughput is reduced by the coding rate ($\bar{\tau} = \bar{\tau}_{\text{raw}}R_{\text{c}}$). In the following, uncoded throughput will be considered to compare different MIMO detection algorithms, unless otherwise stated.

Power and Energy

The average power consumption of a design measures how much energy is consumed per unit of time and is an indicator of how much heat the circuit dissipates. These factors are closely related to physical constraints such as power-supply, battery lifetime, packaging and cooling requirements. In digital integrated circuits the total power dissipation P is contributed by a static and a dynamic component ($P = P_{\text{stat}} + P_{\text{dyn}}$). The latter results from the gates' switching activity and is therefore proportional to the circuit's clock frequency. It is mainly caused by a dominant capacitive switching component (P_{tran}) and by a marginal direct-path contribution generated by short-circuit currents during transients (P_{sc}). The

effect of the short-circuit factor is nevertheless generally small, can be easily kept within bounds by careful design and it becomes of even lesser importance as the supply voltage gets closer to the threshold voltage in deep-submicron technologies [1]. The static component is caused by static conductive paths between the supply rails or by leakage currents (I_{leak}) and is consequently present even when no switching occurs. This has become an issue of concern in processes below 28 nm, as in this regime leakage rises considerably. The total power consumption is thus expressed as the sum of the mentioned components:

$$P = \underbrace{CV_{\text{DD}}^2 f_{\text{clk}}}_{P_{\text{tran}}} + \underbrace{C_{\text{sc}}V_{\text{DD}}^2 f_{\text{clk}}}_{P_{\text{sc}}} + \underbrace{I_{\text{leak}}V_{\text{DD}}}_{P_{\text{stat}}}, \quad (2.42)$$

$\underbrace{\hspace{10em}}_{P_{\text{dyn}}}$

with C and C_{sc} representing the total effective and the short-circuit capacitances, respectively. Even though power dissipation is a relevant design characteristic affecting the feasibility, cost, and reliability of an integrated circuit, for battery-powered devices the energy E is usually a more interesting metric for comparison, since it involves not only the power consumption but also the time T_{clk} required by the particular signal processing task ($E = P \cdot T_{\text{clk}}$). In the context of digital wireless communications, the energy normalized to the amount of processed data is commonly considered:

$$E/\text{bit} = \frac{P}{\bar{\tau}} \quad [\text{J/b}]. \quad (2.43)$$

Cost and Efficiency Metrics

Due to the numerous performance and complexity metrics characterizing a circuit, additional metrics combining several of the circuit's properties at once are required in order to ease performing an equitable comparison of different architectures:

- **AT- and ATE-Product:** Defined as the product of a circuit's silicon area (or, alternatively, its gate-equivalent count) and the time required to process a data unit, the AT-product represents the amount of resources required to achieve a certain throughput. This metric can be extended to incorporate the energy cost, resulting in a combined indication of the area and energy required per processed data unit:

$$\text{ATE-product} = E \cdot \text{AT-product} = E \cdot \frac{A_{\text{GE}}}{\bar{\tau}} \quad [\text{GE} \cdot \text{J/bps}]. \quad (2.44)$$

- **Energy efficiency:** The energy efficiency metric represents the amount of information that can be processed at the energetic cost of 1J, being hence defined as the inverse of the energy-per-bit previously introduced (2.43):

$$\eta_{\text{E}} = \frac{1}{E/\text{bit}} = \frac{\bar{\tau}}{P} \quad [\text{b/J}]. \quad (2.45)$$

Parameter	Symbol	Dependencies	Scaling factor
Transistor dimensions (width, length, oxide thickness)	W, L, T_{ox}	-	$1/S$
Nominal supply, threshold voltage	V_{DD}, V_t	-	$1/U$
Area	A	WL	$1/S^2$
Gate count	A_{GE}	-	1
Saturation current	I_{sat}	$WV_{\text{DD}}/T_{\text{ox}}$	$1/U$
Capacitance	C_{gate}	WL/T_{ox}	$1/S$
Power ^a	P	$C_{\text{gate}}V_{\text{DD}}^2f_{\text{clk}}$ $+I_{\text{sat}}V_{\text{DD}}$	$1/U^2$
Max. clock frequency	$f_{\text{clk}}^{\text{max}}$	$\frac{I_{\text{sat}}}{C_{\text{gate}}V_{\text{DD}}}$	S
Energy	E	P/f_{clk}	$1/(SU^2)$

^aThe power contribution due to short-circuit currents is marginal in this scenario and has been thus disregarded.

Table 2.1: General CMOS technology scaling model for short-channel devices (with transistor gate lengths below $1\mu\text{m}$) [1].

- **Area-throughput efficiency:** The area-throughput efficiency provides an indication of the throughput achieved per resource unit. It is thus inversely related to the AT-product:

$$\eta_{A,\tau} = \frac{1}{\text{AT-product}} = \frac{\bar{\tau}}{A_{\text{GE}}} \quad [\text{bps/GE}]. \quad (2.46)$$

2.5.2.1 CMOS Technology Scaling

Besides the algorithmic and architectural divergences previously discussed, the different CMOS processes employed represent another factor impeding direct comparison of integrated circuit designs. The transistor scaling theory provides a solution in this regard, as it allows estimating the circuit characteristics corresponding to a specific process for a different one. The general technology scaling model employed for short-channel devices (typically applied for transistors with gate lengths below $1\mu\text{m}$) [1] is shown in Table 2.1. In the general model, the device dimensions and related parameters are scaled by a factor $S > 1$, representing the ratio between the feature sizes of the technologies under consideration. In contrast to the classical fixed-field scaling regime (or Dennard scaling [88]), a scaling factor U (with $1 < U < S$) is applied to the voltage magnitudes instead of S . The main reason for this is that, while transistor densities and speeds continue to increase according to Moore's law [89], Dennard scaling [88] ruling the downsizing of transistor power has broken down due to the dramatic rise of leakage occurring at processes below 90 nm.

Leakage currents increase exponentially as the threshold voltage is reduced, thereby augmenting the static power dissipation. As a consequence, the chip temperature rises, which at the same time strengthens the leakage [1]. In other words, with the continuously increasing transistor densities in current and future multiprocessor system-on-chips (MPSoCs), the speed at which transistors can be switched is significantly exceeding the capacity to dissipate the heat consequently generated. The result is a technology-imposed “utilization wall” caused by the limitations of the underlying power and cooling delivery medium [90], restricting the fraction of the chip that can operate simultaneously at full speed. In fact, switching off several components during certain periods is generally required, an effect that has been recently branded as *dark silicon* [91–93]. In this leakage-limited regime, constant-voltage scaling ($U = 1$) is rather applied, leading to an overall chip power consumption scaling with S^2 (at maximum frequency). For a given fixed power budget, the utilization wall is therefore dramatically worsening with each process generation and is expected to further aggravate with the introduction of 3D CMOS technology [94]. Several research efforts have addressed these issues from different fronts [92]. Introducing new materials and device structures, developing heterogenous multi-core architectures with application-specific hardware dedicated to reduce the energy consumption of computationally-intensive applications (so-called *conservation cores* [94,95]), or integrating active liquid cooling microchannels directly on the MPSoC die [90] are some examples of recently investigated solutions.

Selecting the most suitable technology for the circuits intended in this work is hence not a trivial task. 130nm-and-above technologies are not well suited if small area, low power consumption, and high performance are targeted. Very high speed and low (dynamic) power consumption can be achieved by employing smaller feature sizes, such as 28nm-and-below. In the latter case, however, leakage is expected to be a concern rising the static power consumption, as previously discussed. Lying nearly halfway between these extremes, 65nm technology represents a reasonable choice to design fast and efficient circuits while keeping the static power at acceptable levels.

Related Own Publications

- [81][†] E. P. Adeva, B. Mennenga and G. Fettweis: *Survey on an Efficient, Low-complex Tuple Search Based Sphere Detector*. In: *Proceedings of the IEEE 34th Sarnoff Symposium (Sarnoff'11)*, Princeton, USA, 2011.
- [80][†] E. P. Adeva, T. Seifert and G. Fettweis: *VLSI Architecture for MIMO Soft-Input Soft-Output Sphere Detection*. In: *Journal of Signal Processing Systems (JSPS)*, vol. 70, no. 2, February 2013.

