

DISSERTATION

Efficient Near-Optimum Detection Algorithms for MIMO Communication Systems

ausgeführt zum Zwecke der Erlangung des akademischen Grades eines
Doktors der technischen Wissenschaften

unter der Leitung von
Ao. Univ.-Prof. Dr. Franz Hlawatsch
Institut für Nachrichtentechnik und Hochfrequenztechnik

eingereicht an der Technischen Universität Wien
Fakultät für Elektrotechnik

von
Dominik Seethaler
Goldeggasse 34/8
1040 Wien

Wien, im September 2006

Die Begutachtung dieser Arbeit erfolgte durch:

1. Ao. Univ.-Prof. Dipl.-Ing. Dr. F. Hlawatsch

Institut für Nachrichtentechnik und Hochfrequenztechnik
Technische Universität Wien

2. Prof. Dr. A. Gershman

Institut für Nachrichtentechnik
Technische Universität Darmstadt

Abstract

Wireless communications continue to strive for higher data rates and a better link reliability in order to provide more advanced services. The use of multiple antennas at both the transmitter and receiver side, i.e., *multiple-input multiple-output* (MIMO) communications, is one of the most promising technologies to satisfy these demands. Indeed, MIMO systems are capable of achieving increased data rates and an improved link reliability compared to single-antenna systems without requiring additional bandwidth or transmit power. These improvements, however, necessitate the use of more computationally intensive data detection algorithms at the receiver side. In particular, optimum data detection can easily become prohibitively complex. Conventional suboptimum detection techniques have a low computational cost but their performance is in general significantly inferior to that of optimum data detection. Thus, there is a strong demand for computationally efficient data detection algorithms that are able to reduce this performance gap.

In this thesis, novel algorithms for efficient near-optimum data detection in MIMO systems are proposed and investigated. First, we show that specific “bad” realizations of the MIMO channel are to a great extent responsible for the inferior performance of conventional suboptimum data detection algorithms. Motivated by this insight, we then introduce an idealized model for bad channels that enables a simplified implementation of the optimum detector. With some modifications, we then obtain efficient detection algorithms that are robust to bad channels and that can achieve near-optimum performance. Secondly, we focus on a popular suboptimum data detection technique based on nulling and cancelling. The order in which the nulling and cancelling operations are performed has a strong influence on the performance of this scheme. The conventional ordering approach merely takes the current (static) channel realization into account. However, we propose to base the ordering also on the current received signal vector in addition to the channel realization. The resulting “dynamic” ordering method yields a significant performance advantage over the conventional scheme, such that near-optimum performance can be achieved at low computational cost. Finally, we extend the proposed concepts and algorithms to soft-output detection, which yields an additional performance improvement in coded MIMO wireless systems.

Kurzfassung

Drahtlose Kommunikationstechnik strebt nach immer höheren Datenraten und immer größerer Zuverlässigkeit der Übertragung, um immer anspruchsvollere Dienste zur Verfügung stellen zu können. Systeme mit mehreren Sendeantennen und mehreren Empfangsantennen, d.h. *multiple-input multiple-output* (MIMO)-Systeme, sind eine der vielversprechendsten Technologien, um diesen Ansprüchen gerecht zu werden. Gegenüber Systemen mit nur einer Sende- und Empfangsantenne lässt sich mit MIMO-Systemen bei gleicher Übertragungsbandbreite und Sendeleistung sowohl die Datenrate steigern als auch die Übertragungszuverlässigkeit verbessern. Diese Gewinne erfordern jedoch viel rechenaufwändigere Datendetektionsverfahren auf der Empfängerseite. Insbesondere optimale Detektion kann sehr leicht zu exzessivem Rechenaufwand führen. Konventionelle suboptimale Detektionsverfahren hingegen sind sehr recheneffizient, liegen aber weit unter der Leistungsfähigkeit optimaler Detektion. Aus diesem Grund besteht ein großer Bedarf an recheneffizienten Detektionsverfahren, die diesen Leistungsverlust reduzieren.

In dieser Dissertation werden neuartige recheneffiziente MIMO-Detektionsverfahren, welche annähernd die Leistungsfähigkeit optimaler Detektion erreichen, vorgestellt und untersucht. Zunächst wird gezeigt, dass der große Leistungsverlust konventioneller suboptimaler Detektionsverfahren hauptsächlich auf “böartige” Realisierungen des MIMO-Übertragungskanals zurückzuführen ist. Diese Erkenntnis motiviert die Einführung eines idealisierten Modells solcher böartiger MIMO-Übertragungskanäle, das eine vereinfachte Ausführung des optimalen Detektors ermöglicht. Daraus ergeben sich mit einigen Modifikationen recheneffiziente Detektionsverfahren, die robust gegenüber böartigen Übertragungskanälen sind und annähernd die Leistungsfähigkeit optimaler Detektion erreichen. Der zweite Beitrag dieser Dissertation betrifft ein weit verbreitetes Detektionsverfahren mit Entscheidungsrückkopplung. Die Leistungsfähigkeit dieses Verfahrens wird sehr stark durch die Reihenfolge, in der die Entscheidungen und Rückkopplungen durchgeführt werden, beeinflusst. Für gewöhnlich ist diese Reihenfolge nur durch die Eigenschaften des (statischen) MIMO-Übertragungskanals bestimmt. In dieser Arbeit wird nun ein Verfahren vorgeschlagen, bei dem die Reihenfolge der Entscheidungen und Rückkopplungen auch basierend auf dem Empfangsvektor (zusätzlich zum MIMO-Übertragungskanal) gewählt wird. Diese “dynamische” Wahl der Reihenfolge führt zu einer erheblichen Leistungssteigerung dieses Verfahrens und erlaubt somit, auf recheneffiziente Weise annähernd die Leistungsfähigkeit optimaler Detektion zu erreichen. Zuletzt werden die vorgestellten Konzepte und Methoden auf “weiche” (soft-output) Detektion erweitert, wodurch eine zusätzliche Leistungssteigerung in codierten MIMO-Systemen ermöglicht wird.

Acknowledgments

First, I would like to thank my advisor Franz Hlawatsch for his continuous support and excellent guidance over the last several years. Due to his careful proofreading, technical content and presentation of this thesis have been improved significantly. It is also my pleasure to express my thanks to Alex Gershman, who kindly agreed to act as a referee.

Very special thanks go to Harold Artés and Gerald Matz. The close collaboration with them has been (and still is) very important for me. In fact, their patient support has been essential for my professional development. Their technical contributions to this thesis are also gratefully acknowledged.

Finally, I want to thank Ayse Adalan for putting much effort in proofreading certain parts of this thesis and for pointing out many useful suggestions for improvement.

Contents

1	Introduction	1
1.1	Motivation and Problem Formulation	2
1.2	MIMO Wireless Systems	2
1.3	Contributions and Outline	3
1.4	Notation	5
1.5	System Model	5
1.5.1	MIMO Spatial Multiplexing Systems	6
1.5.2	Linear Dispersion Codes	7
1.6	Data Detection	8
1.7	Algorithm Assessment	8
1.7.1	Simulation Setup	9
1.7.2	Error Rate Performance	9
1.7.3	Computational Complexity	9
1.8	State of the Art	11
1.8.1	Maximum Likelihood Detection	11
1.8.2	Equalization-Based Detection	12
1.8.3	Nulling-and-Cancelling	13
2	Geometry-Based Detectors for Spatial Multiplexing	17
2.1	The “Bad Channel” Effect	18
2.1.1	Detector Performance and Bad Channels	18
2.1.2	Geometrical Analysis	19
2.2	The Idealized Bad Channel Model	21
2.2.1	Formulation of the IBC	22
2.2.2	ML Detection for the IBC	23
2.3	An Efficient ML Detector for the IBC	25
2.3.1	The Reduced Search Set	25
2.3.2	Partitioning of the Reference Line	26
2.3.3	Size of Reduced Search Set	28
2.3.4	Efficient Determination of Reduced Search Set	28

2.3.5	Efficient Calculation of Distances	30
2.3.6	Summary of ML Detection for the IBC	31
2.3.7	Computational Complexity	32
2.4	The Line Search Detector	32
2.4.1	Algorithm Statement	33
2.4.2	Discussion	34
2.4.3	Computational Complexity	34
2.5	The Sphere Projection Algorithm	35
2.5.1	Construction of the Additional Search Set	36
2.5.2	Algorithm Summary and Complexity	38
2.5.3	SPA Variants	39
2.6	Simulation Results	39
2.6.1	SER Performance	40
2.6.2	Computational Complexity	42
2.6.3	Discussion	44
2.7	Summary and Conclusions	45

3 Dynamic Nulling-and-Cancelling 47

3.1	Fundamentals of Dynamic Nulling-and-Cancelling	48
3.1.1	MAP Approach to Detection and Layer Sorting	49
3.1.2	Gaussian Approximation	50
3.2	Data-Detection Stage	50
3.3	Layer-Sorting Stage	52
3.3.1	Dynamic Layer Sorting	52
3.3.2	Discussion	54
3.4	The DNC Algorithm	56
3.4.1	Algorithm Statement	56
3.4.2	Efficient Implementation	57
3.4.3	Computational Complexity	57
3.5	Error Performance of DNC	58
3.5.1	Symbol Error Probability of a Two-Layer BPSK System	58
3.5.2	Discussion	60
3.6	LLR-based NC	61
3.6.1	Data Detection Stage	61
3.6.2	Layer-Sorting Stage	61
3.6.3	Discussion	62
3.7	Simulation Results	62
3.7.1	SER Performance for Spatial Multiplexing Systems	62
3.7.2	SER Performance for Systems Using Linear Dispersion Codes	65
3.7.3	Computational Complexity	66
3.7.4	DNC versus LLR-based NC	67
3.7.5	DNC versus Geometry-Based Detection	68
3.8	Summary and Conclusions	71

4	Soft-Output Detection Algorithms	73
4.1	MIMO Bit-Interleaved Coded Modulation	74
4.1.1	MIMO-BICM System Model	74
4.1.2	Soft-Output Detection	75
4.2	MMSE- and ZF-Based Soft-Output Detection	76
4.2.1	Gaussian Approximation	77
4.2.2	MMSE-Based Soft-Output Detection	77
4.2.3	ZF-Based Soft-Output Detection	78
4.2.4	Computational Complexity and Discussion	78
4.3	Soft-Output SPA	79
4.3.1	Basic Idea of SSPA	79
4.3.2	Re-Quantization Property of PSK with Gray Labeling	80
4.3.3	SSPA Details	84
4.3.4	Algorithm Summary and Computational Complexity	86
4.4	Simulation Results	87
4.4.1	Packet Error Rate Performance	87
4.4.2	Computational Complexity	90
4.5	Summary and Conclusions	91
5	Conclusions and Outlook	93
5.1	Conclusions	94
5.2	Outlook	97
	Bibliography	99
	List of Abbreviations	107

1

Introduction

MULTIPLE-INPUT MULTIPLE-OUTPUT (MIMO) systems are currently one of the most active research areas in wireless communications. Here, both the transmitter and the receiver employ multiple antennas, which yields great advantages over single-antenna communication systems [1–3]. In particular, with MIMO communications the link reliability can be improved and the data rate can be increased without the need for additional bandwidth or transmit power. These MIMO gains rely on the availability of *spatial diversity* and the possibility of *spatial multiplexing*. Spatial diversity can be used to combat channel fading (which improves the link reliability) and spatial multiplexing enables a parallel transmission of multiple data streams (which increases the data rate). Up to a certain extent these gains can be achieved simultaneously [4]. Thus, MIMO is one of the most promising technologies for next-generation wireless systems that have an increased demand for data rate, quality of service, and bandwidth efficiency. This is also reflected by recent efforts to incorporate the MIMO technology into wireless standards (such as IEEE 802.11n, IEEE 802.16, and the high-speed downlink packet access (HSDPA) transmission mode of UMTS).

1.1 Motivation and Problem Formulation

The full potential of MIMO communications can only be achieved with increased hardware costs as compared to single-antenna systems. This is mainly due to the requirement of multiple radio-frequency chains as well as due to significantly more complex baseband signal processing algorithms. In particular, one of the most challenging tasks of MIMO signal processing with respect to the computational requirements is data detection at the receiver side. Here, the transmitted data has to be detected (recovered) with low probability of error. For high-rate MIMO transmission schemes using spatial multiplexing, optimum data detection can easily become prohibitively complex [5] since one has to deal with a very strong spatial interference of the multiple transmitted data streams. In general, no algorithms are known that perform optimum detection with a computational complexity that behaves polynomially in the number of antennas [6, 7]. Thus, in recent years considerable research effort has been devoted to investigate and develop *efficient* data detection algorithms for MIMO systems with a (comparatively) moderate complexity. This includes advanced optimum algorithms (like sphere-decoding [7–9]) and low-complexity *suboptimum* detectors. Most of the suboptimum algorithms were originally developed in the context of multi-user detection [6, 10]. In the MIMO context, however, conventional low-complexity algorithms suffer from a significant performance degradation compared to optimum performance [11] and parts of the MIMO gains (such as diversity) are lost. Thus, there is a strong demand for computationally efficient suboptimum data detection algorithms that can achieve near-optimum performance in MIMO wireless communication systems. In this thesis, such improved detection methods are proposed and investigated.

1.2 MIMO Wireless Systems

MIMO wireless systems employ multiple antennas at the transmitter and at the receiver side. In comparison to single-antenna communication systems, MIMO systems exhibit various significant gains [1–3]. One can distinguish between *diversity gain*, *multiplexing gain*, and *array gain* [11]. The diversity gain and the array gain can also be exploited by single-input multiple-output (SIMO) or multiple-input single-output (MISO) wireless systems.

- **Diversity gain:** If multiple copies of an information signal are received through different fading links, a diversity gain can be achieved. Through an appropriate combination of these multiple received signals, deep channel fades (i.e. the case that the signal power drops significantly) can be reduced and the link reliability can be improved. There are various different notions of diversity that are just distinguished with respect to their physical interpretation. In the MIMO, SIMO, or MISO context, we are dealing with *spatial* diversity since multiple copies of a signal are obtained through multiple antennas that are spatially separated. In the SIMO case, just *receive* diversity is available which can be utilized through a coherent combination of all received signals (if channel state information is available at the receiver). Exploiting *transmit* diversity in the MISO case can be more involved since the transmitter may not have accurate channel state

information. However, via the use of appropriate space-time codes [3,12] also transmit diversity can be exploited. In the MIMO case, both transmit and receive diversity are available.

- **Multiplexing gain:** The multiplexing gain refers to several spatial dimensions that can be offered by MIMO systems. These spatial dimensions can then be used to transmit data in parallel which results in a gain in data rate. In fact, the capacity (i.e. the ultimate limit for the rate of reliable data transmission) of a MIMO channel can increase *linearly* in the minimum number of transmit and receive antennas [1,2]. This gain is unique to MIMO, i.e. SIMO or MISO system do not offer any multiplexing gain. A multiplexing gain can be realized by a scheme called *spatial multiplexing* (SM), where independent data streams are transmitted in parallel from each transmit antenna (i.e. they are multiplexed in space). In literature, this is also known as the Vertical Bell Laboratories Layered Space-Time (V-BLAST) [13,14] architecture.
- **Array gain:** Finally, multiple-antenna systems also offer an array gain, which refers to an increase of the average signal-to-noise ratio as compared to single-antenna systems. It can also be realized in SIMO or MISO systems with appropriate channel state information.

It is important to note that these MIMO gains can be achieved simultaneously – but just up to a certain extent. There is a fundamental tradeoff between the diversity gain and the multiplexing gain [4], i.e. they cannot be maximized at the same time by a certain transmission scheme. For example, in SM systems, a large multiplexing gain is achieved while no transmit diversity can be exploited and just receive diversity is available. On the other hand, space-time codes intend to extract much diversity (in particular, transmit diversity) while the exploitation of multiplexing gains is secondary.

1.3 Contributions and Outline

In this thesis, novel detection algorithms for efficient near-optimum, i.e. *maximum likelihood* (ML), detection of MIMO systems are proposed and investigated. In Chapter 2 and Chapter 3, *hard-output* detectors are proposed while Chapter 4 focuses on *soft-output* detectors in coded MIMO systems. The detailed contributions and the outline are as follows.

- The remainder of this chapter describes the notation, the linear MIMO system model, and the algorithm assessment. Furthermore, we also provide the state of the art with a review of the major existing data detection techniques for MIMO systems for further reference.
- The starting point of **Chapter 2** (“Geometry-Based Detectors for Spatial Multiplexing”) is the investigation of why conventional suboptimum data detection algorithms (in particular, linear and nulling-and-cancelling schemes) suffer from a significant performance loss compared to (optimum) ML detection in SM systems. Here, we adopt the viewpoint that in particular “bad” channels with a large condition number are to a great extent responsible for their inferior performance. We then introduce a novel channel model that captures essential parts of these bad

channels in an idealized way. It is shown that ML detection for such idealized bad channels can be performed efficiently (i.e. with a cubic complexity in the number of antennas). Based on the ML detector for idealized bad channels, the novel *line-search detector* (LSD) [15,16] is proposed. It is able to achieve near-ML performance for general (non-idealized) channels; however, it still turns out to be quite computationally intensive. We then focus on constant-modulus alphabets, which yields a further reduction of computational complexity. The resulting novel algorithm we refer to as the *sphere-projection algorithm* (SPA) [15,17]. It can be seen as an efficient non-linear add-on to conventional suboptimum detectors making them robust to bad channels. The proposed SPA variants are shown to achieve near-ML performance with low computational complexity. Extensions and applications of the concepts introduced in this chapter can be found in [18–20].

- In **Chapter 3** (“Dynamic Nulling-and-Cancelling”) we focus on the very popular and well-known suboptimum data detection technique based on decision-feedback detection [21,22] (or often referred to as nulling-and-cancelling (NC) in the MIMO context). Here, the order in which the nulling (detection) and cancelling (feedback) operations are carried out has a significant influence on the overall performance. Conventional NC employs the layer-wise post-equalization signal-to-noise ratios (PSNRs) as reliability measures to perform the layer sorting. These PSNRs are *average* quantities that do not depend on the current received vector. In this chapter, we propose the novel *dynamic nulling-and-cancelling* (DNC) [23,24] data detection technique that uses approximate a-posteriori probabilities (APPs) as measures of layer reliability. The approximate APPs are obtained from the exact APPs via a Gaussian approximation for the post-equalization interference. This results in a NC technique based on *minimum mean-square error* (MMSE) detection with an improved “dynamic” layer-sorting rule that exploits the information contained in the current received vector. We furthermore show that a recently proposed (also dynamic) “LLR-based NC scheme” can also be derived by means of a Gaussian approximation for the post-equalization interference; however, in contrast to our DNC scheme, the post-equalization interference has to be assumed as uncorrelated. We analyze the error performance and computational complexity of the DNC method [23,25]. In particular, we derive an expression for the symbol error probability of the first layer-decoding step of DNC based on some simplifying assumptions. This analysis as well as experimental results show the general superiority of DNC over conventional NC. Moreover, it reveals the conditions under which this performance advantage will be most significant. We also show that DNC can achieve near-ML performance at low computational cost. Finally, DNC is compared with the proposed geometry-based detectors of Chapter 2 and we demonstrate that DNC is superior to LLR-based NC.
- In **Chapter 4** (“Soft-Output Detection Algorithms”) *soft-output* detectors for MIMO *bit-interleaved coded modulation* (BICM) systems are investigated and proposed. In contrast to the hard-output detectors of Chapter 2 and Chapter 3, these soft-output detectors calculate approximate log-likelihood ratios (LLRs) for each of the transmitted coded bits. These LLRs

are then provided to a soft-in channel decoder. Although conceptually different, this chapter heavily relies on the algorithms and approaches introduced in Chapter 2 and Chapter 3. In particular, the major contribution of this chapter is the extension of the hard-output SPA of Chapter 2 to its soft version [26]. Its efficient implementation is specifically tailored to phase-shift keying (PSK) alphabets employing Gray labeling. The soft-output extension of the LSD can be found in [27]. Furthermore, we will use the concept of the DNC of Chapter 3 (i.e. the use of a Gaussian approximation for the post-equalization interference) to rederive [28] the conventional zero-forcing (ZF) and MMSE-based soft-output detectors. This derivation shows that MMSE-based soft-output detection can be seen as an extension of ZF-based soft-output detection to correlated post-equalization interference, which explains MMSE's significant performance advantage. All the detectors are compared for orthogonal frequency division multiplexing (OFDM) based MIMO-BICM systems, where also real-world MIMO channels are used for the simulations. In particular, it is demonstrated that the soft-output SPA significantly outperforms the ZF-based and MMSE-based detectors and that it can achieve a performance close to the soft-output extension of the sphere decoding algorithm for ML detection.

1.4 Notation

A matrix, vector, and scalar are denoted with \mathbf{A} , \mathbf{a} , and a (or A), respectively. The element at the n th row and m th column of \mathbf{A} is written as $(\mathbf{A})_{n,m}$ and \mathbf{a}_m denotes the m th column of \mathbf{A} . The real and imaginary parts of a complex number a are referred to as $\text{Re}\{a\}$ and $\text{Im}\{a\}$ (i.e. $a = \text{Re}\{a\} + j\text{Im}\{a\}$) or sometimes as a_R and a_I , respectively. The transpose and the conjugate transpose of \mathbf{A} are given by \mathbf{A}^T and $\mathbf{A}^H = \mathbf{A}^{T*}$, respectively. Here, \mathbf{A}^* denotes the element-wise complex conjugation of \mathbf{A} . The identity matrix is denoted as \mathbf{I} and the n th unit vector is given by \mathbf{e}_n . Furthermore, $\text{E}[a]$ denotes the expected value of the random variable a , and $\boldsymbol{\mu}_{\mathbf{a}} = \text{E}[\mathbf{a}]$, $\mathbf{R}_{\mathbf{a}} = \text{E}[\mathbf{a}\mathbf{a}^H]$, and $\mathbf{C}_{\mathbf{a}} = \text{E}[(\mathbf{a} - \boldsymbol{\mu}_{\mathbf{a}})(\mathbf{a} - \boldsymbol{\mu}_{\mathbf{a}})^H]$ denote the mean, the correlation matrix, and the covariance matrix of the random vector \mathbf{a} , respectively. Finally, $\|\mathbf{a}\|$ is the Euclidian norm of the vector \mathbf{a} and upper-case calligraphic letters, e.g. \mathcal{A} , refer to sets.

1.5 System Model

Throughout this thesis, we assume a *linear MIMO model* with M inputs and N outputs where the transmitted data vector $\mathbf{d} \triangleq (d_1 \cdots d_M)^T$ of size M and the received vector $\mathbf{r} \triangleq (r_1 \cdots r_N)^T$ of size $N \geq M$ are related according to

$$\mathbf{r} = \mathbf{H}\mathbf{d} + \mathbf{w} \quad (1.1)$$

for any given time instant (or channel use). Here, the $N \times M$ matrix \mathbf{H} denotes the MIMO system matrix and $\mathbf{w} \triangleq (w_1 \cdots w_N)^T$ denotes additive noise. The noise components w_n , $n = 1, \dots, N$, are assumed as zero-mean, statistically independent, and Gaussian with variance $\sigma_{\mathbf{w}}^2$, i.e. $\mathbf{R}_{\mathbf{w}} = \sigma_{\mathbf{w}}^2 \mathbf{I}$.

The data vector components d_m (here, $m \in \{1, \dots, M\}$ indexes the *layer*) are assumed zero-mean, statistically independent with unit variance, i.e. $\mathbf{R}_d = \mathbf{I}$, and are uniformly drawn from the same symbol alphabet $\mathcal{A} \triangleq \{a_0, \dots, a_{|\mathcal{A}|-1}\}$. All quantities involved in (1.1) are either real- or complex-valued. In the complex-valued case, the noise components w_n are assumed as circularly symmetric complex Gaussian.

We define the *signal-to-noise ratio* (SNR) associated to (1.1) as the ratio of the total received power to the total noise power, i.e.

$$\text{SNR} \triangleq \frac{\mathbb{E}\{\|\mathbf{H}\mathbf{d}\|^2\}}{\mathbb{E}\{\|\mathbf{w}\|^2\}}. \quad (1.2)$$

If system model (1.1) is complex-valued, it is sometimes advantageous (e.g., for nulling-and-cancelling based MIMO detection schemes) to use an equivalent real-valued representation of (1.1) (see, e.g., [29, 30]), that is obtained by considering real and imaginary parts separately, i.e.

$$\begin{pmatrix} \text{Re}\{\mathbf{r}\} \\ \text{Im}\{\mathbf{r}\} \end{pmatrix} = \begin{pmatrix} \text{Re}\{\mathbf{H}\} & -\text{Im}\{\mathbf{H}\} \\ \text{Im}\{\mathbf{H}\} & \text{Re}\{\mathbf{H}\} \end{pmatrix} \begin{pmatrix} \text{Re}\{\mathbf{d}\} \\ \text{Im}\{\mathbf{d}\} \end{pmatrix} + \begin{pmatrix} \text{Re}\{\mathbf{w}\} \\ \text{Im}\{\mathbf{w}\} \end{pmatrix}. \quad (1.3)$$

This real-valued representation can again be written in form of (1.1). The equivalent real-valued data symbols (i.e. the components of the the real-valued data vector in (1.3)) are again statistically independent if the symbols d_m are statistically independent and if they are uniformly drawn from a QAM symbol alphabet. Thus, for example, a complex-valued system with statistically independent and uniformly drawn 4-QAM symbols can be reformulated as a real-valued system of double size with statistically independent and uniformly drawn BPSK symbols.

There are many different transmission schemes that can be written in form of the linear model (1.1). The specific choice of the transmission scheme determines the corresponding physical interpretation and the structure of \mathbf{H} . We now briefly describe those schemes that will appear in this thesis; our main focus will be on SM systems.

1.5.1 MIMO Spatial Multiplexing Systems

For an SM system (e.g., [11, 14, 31]) with M_T transmit antennas and M_R receive antennas, the m th data symbol $d_m \in \mathcal{A}$ is directly transmitted on the m th transmit antenna (see Figure 1.1), which is also often referred to as the V-BLAST [13, 14] architecture.

For a *frequency-flat* channel, the input/output relation between transmit antenna m and receive antenna n can be described by a complex-valued fading coefficient $h_{n,m}$. At receive antenna n the superposition of all transmitted data symbols d_m , $m = 1, \dots, M_T$, corrupted by additive noise w_n is observed, i.e.

$$r_n = \sum_{m=1}^{M_T} h_{n,m} d_m + w_n.$$

Evidently, in vector/matrix notation we obtain (1.1), where system matrix \mathbf{H} directly corresponds to the *MIMO channel matrix* with $(\mathbf{H})_{n,m} = h_{n,m}$, and M and N represent the number of transmit antennas and the number of receive antennas, M_T and M_R , respectively, i.e. $M = M_T$, $N = M_R$.

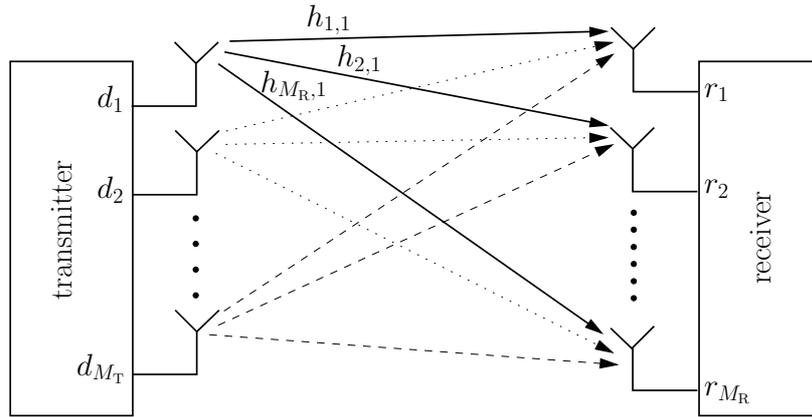


Figure 1.1: MIMO spatial multiplexing system.

For a *frequency-selective* channel, the input/output relation between transmit antenna m and receive antenna n can be described by a complex-valued impulse response $h_{n,m}[l]$, $l = 0, \dots, L-1$, with L channel taps. Thus, the simple multiplication operation of frequency-flat channels is now replaced with a more involved time-discrete convolution. However, via the use of an *orthogonal frequency division multiplexing* (OFDM) modulator at each transmit antenna and an OFDM demodulator at each receive antenna (i.e. MIMO-OFDM), the frequency-selective MIMO channel can be decomposed into parallel frequency-flat MIMO channels (e.g., [11, 31, 32]). In fact, by employing a MIMO-OFDM system with K subcarriers, one obtains

$$\mathbf{r}_k = \mathbf{H}_k \mathbf{d}_k + \mathbf{w}_k, \quad k = 0, \dots, K-1, \quad (1.4)$$

for each subcarrier k , where the elements of \mathbf{H}_k are given by the discrete Fourier transform of the channel impulse responses $h_{n,m}[l]$, i.e.

$$(\mathbf{H}_k)_{n,m} = \sum_{l=0}^{L-1} h_{n,m}[l] e^{-j2\pi \frac{lk}{K}}, \quad k = 0, \dots, K-1.$$

In this case, the linear model (1.1) holds for each subcarrier separately.

1.5.2 Linear Dispersion Codes

For linear dispersion (LD) codes [33], Q complex-valued QAM data symbols are transmitted during T consecutive channel uses during which the flat-fading channel realization is assumed to stay constant. In contrast to SM, the data symbols are in general not directly transmitted from specific transmit antennas. However, they are spread over time (i.e. over the T channel uses) and space (i.e. the M_T transmit antennas) using certain modulation/dispersion matrices, which can be designed such that

the rate or the diversity is maximized. The resulting system model is real-valued and is given by (1.1) with $M = 2Q$ and $N = 2M_{\text{R}}T$ (here, similar to (1.3), the real and imaginary parts are considered separately). The system matrix \mathbf{H} corresponds to the so called *equivalent channel matrix* that depends on the channel realization *and* on the considered LD code. In fact, LD codes also comprise SM systems and linear space-time codes (e.g., [34], including orthogonal designs [35]) as special cases.

1.6 Data Detection

One of the most challenging tasks of a MIMO receiver is *data detection*. Our main focus will be on *hard-output* detectors. Here, based on the received vector \mathbf{r} , the transmitted data vector \mathbf{d} (or equivalently, each transmitted data symbol $d_m \in \mathcal{A}$) has to be detected (recovered), and the result is some hard-decision $\hat{\mathbf{d}}$ (with $\hat{d}_m \in \mathcal{A}$) about \mathbf{d} . In Chapter 4, we will also consider so-called *soft-output* detectors that are embedded into coded MIMO systems. Here, the data detector calculates soft-decisions (e.g., log-likelihood ratios) about the coded bits that are associated with the transmitted data symbols $d_m \in \mathcal{A}$ via a certain labeling. Throughout this thesis, we assume that the data detector not only has access to the received vector \mathbf{r} , but also has *perfect knowledge* about the system matrix \mathbf{H} and the noise variance $\sigma_{\mathbf{w}}^2$. In practice, corresponding estimates can be obtained via separate (e.g., training-based) channel and noise variance estimation algorithms, which represent an important research area by itself (see, e.g., [36–39]). For example, a study of the impact of estimation errors on the performance of data detection algorithms can be found in [40]. However, without explicit estimation of the MIMO channel coefficients, \mathbf{H} may also be known by using unitary differential space-time modulation using the Cayley transform [41], where (1.1) then corresponds to an associated “linearized” system model.

In general, a data detection algorithm has to be designed with respect to two important targets: (i) *Small error rate* (i.e. good performance), and (ii) *low computational complexity*. However, there is a tradeoff between these two targets. For example, the *optimum* detector in terms of best performance (minimum error probability) is known to be very computational intensive. On the other hand, *suboptimum* detection algorithms (like equalization-based detectors) may be computationally very efficient; however, their performance is in general far inferior to that of optimum detection. Thus, the task is to design data detection algorithms that achieve a “good” – with respect to the application in mind – tradeoff between performance and computational complexity. The suboptimum algorithms proposed in this thesis aim at achieving near-optimum performance with a computational complexity that is significantly smaller than that required by the optimum detector.

1.7 Algorithm Assessment

The proposed data detection algorithms will be assessed and compared in terms of error rate performance and computational complexity. Mainly, this will be done via MATLAB simulations for various different MIMO systems.

1.7.1 Simulation Setup

Throughout, the presented MATLAB simulation results are based on the linear MIMO model (1.1), where no model errors or possible implementation losses (e.g., due to fixed-point number representations [5]) are considered. Also, the knowledge about the system matrix \mathbf{H} and the noise variance $\sigma_{\mathbf{w}}^2$ is considered as perfect. The main part of the simulations is carried out for *identically and independently distributed (iid) complex Gaussian MIMO fading channels* (e.g., [11, 42]). For an SM system, for example, this means that the entries of the channel matrix \mathbf{H} (i.e. the scalar fading coefficients $h_{n,m}$) are modeled as statistically independent circularly symmetric complex Gaussian random variables. Evidently, this channel model does not capture any spatial correlations [43, 44] between the fading coefficients of different antenna pairs, but it can be seen as an accurate model for rich scattering (e.g., indoor) environments with a large enough antenna spacing [42]. But we also show some simulation results for real-world MIMO channels that were measured at Vienna International Airport [45]. Furthermore, for all simulation results we considered MIMO systems with $M = N \geq 4$ employing various different symbol alphabets. For $M < N$, the performance difference between the various detectors is in general less pronounced and near-optimum performance can be achieved more easily. We do not consider the case $M > N$ since this would imply that we have more unknowns than equations in (1.1).

1.7.2 Error Rate Performance

For the hard-output detectors, we will focus on the uncoded *symbol error rate* (SER) performance, where no additional error correction (or channel) code is taken into account. For the soft-output detectors of Chapter 4 (where we will consider coded MIMO systems) we will focus on the *packet error rate* (PER) performance. Most of these error rate results are shown as curves versus the SNR (1.2), where an important performance characteristic is the slope of the corresponding double-logarithmic error rate curves in the high-SNR regime. It directly reflects the *diversity* (e.g., [11, 42]) that can be exploited by the corresponding system (including the detection algorithm). In an uncoded SM system, for example, the maximum spatial diversity is given by the number of receive antennas M_{R} . This maximum diversity is available if all channel coefficients $h_{n,m}$ of the channel matrix \mathbf{H} fade independently (which is the case for iid Gaussian channels), because then each data symbol d_m is transmitted over M_{R} independent scalar fading channels $h_{n,m}$, $n = 1, \dots, M_{\text{R}}$ (cf. (1.1)). The larger M_{R} , the smaller is the probability that all these channels fade simultaneously, and thus the reliability of data detection can be improved. If the available diversity in an SM system is M_{R} , the SER of the optimal hard-output detector decays like $\text{SNR}^{-M_{\text{R}}}$ in the high-SNR regime [3, 4]. This corresponds to a slope of $-M_{\text{R}}$ of the double-logarithmic SER-versus-SNR curve. In general, if the error rate decays like $\text{SNR}^{-\delta}$ in the high-SNR regime we say that the system can exploit δ th-order diversity.

1.7.3 Computational Complexity

The computational complexity of the various algorithms will be assessed in terms of the required number of *floating point operations* (flops). We will provide the corresponding *complexity orders* of

the algorithms and *flops measurement* results of specific MATLAB V5.3 implementations.

The complexity order is given by means of the “big Oh” notation $\mathcal{O}(\cdot)$ (see, e.g., [46]), which describes the *complexity scaling behavior* of an algorithm depending on one or more system parameters. It means that the argument of $\mathcal{O}(\cdot)$ times some constant is larger than the exact computational cost for parameter values which are large enough. In our setting, the crucial parameters are M and N (i.e. the size of system model (1.1)), and $|\mathcal{A}|$ (i.e. the size of the symbol alphabet \mathcal{A}). Here, for simplicity, we will always assume $M = N$. Both M and $|\mathcal{A}|$ determine the rate of the system, i.e. the bits per channel use that are transmitted over the MIMO system, cf. (1.1). In fact, $\mathcal{O}(\cdot)$ yields very general and simple expressions that quickly reveal the basic complexity behavior of the various algorithms. However, its practical meaning may be limited. In particular, for MIMO systems of moderate size, constants and lower order contributions to the computational cost may also be relevant. This is the reason why we also considered flops measurements for specific MATLAB implementations of the algorithms. With the complexity orders and the flops measurement results a good initial estimate about the computational complexity of the algorithms is obtained – in particular, if one is interested in floating-point digital signal processor (DSP) implementations. However, for implementations on dedicated hardware (e.g., very-large-scale integration (VLSI)) [5], just limited conclusions can be drawn.

The computational complexity of each algorithm will be split into the following two components:

- The “*preparation complexity*” C_{prep} which includes all operations that only have to be performed if \mathbf{H} changes.
- The “*vector complexity*” C_{vector} which includes all operations that are performed once for each received vector \mathbf{r} .

This distinction is crucial, since the contribution of these two components to the overall computational complexity may be very different (see also, e.g., [5]). Usually, C_{vector} tends to dominate the overall complexity while C_{prep} is less critical. For example, for an SM system operating in a low-mobility scenario, the channel matrix \mathbf{H} stays in fact constant over many subsequent channel uses (i.e. subsequent transmissions of data vectors) and operations contributing to C_{prep} have to be performed much less frequently than those contributing to C_{vector} . Even if \mathbf{H} changes from one channel use to the next one (e.g., in a MIMO-OFDM system, where each channel use corresponds to a certain subcarrier, see (1.4)), correlations between different realizations of \mathbf{H} can be exploited such that C_{prep} is reduced [32, 47]. However, in certain packet-based applications also C_{prep} can be decisive [48].

Finally, some of the considered data detection algorithms have a C_{vector} complexity that strongly depends on the specific realization of \mathbf{H} . Since \mathbf{H} is random, their C_{vector} is random as well. In this case, we provide average and maximum (over many realizations of \mathbf{H}) C_{vector} results. For practical system designs the maximum C_{vector} complexity can be decisive. In particular, this is the case if buffer sizes are small or if \mathbf{H} changes too slowly to perform a complexity averaging during run-time.

1.8 State of the Art

As a relevant background and for further reference, we will now review the most important hard-output data detection techniques for MIMO systems. In Chapter 4, we will provide the background on the corresponding soft-output techniques. The major hard-output detectors are linear equalization followed by quantization (e.g., [11, 42]), nulling-and-cancelling (also known as decision-feedback detection, successive interference cancellation, or V-BLAST detection) [13, 14, 22], and maximum likelihood (ML) detection (e.g., [6, 9, 11]) including sphere-decoding [8, 9, 49, 50]. The first two techniques are *suboptimum* but significantly less complex than ML detection. In fact, many algorithms that are proposed in the literature are based on these fundamental techniques, which, to a large part, were originally developed and proposed in the context of multiuser detection [6, 10].

1.8.1 Maximum Likelihood Detection

Maximum likelihood (ML) detection is optimal in the sense of minimizing the average probability of error $P[\hat{\mathbf{d}} \neq \mathbf{d}]$ when all data vectors are equally likely. For our system model (1.1) and with the assumptions made in Section 1.5, the ML detector is given by (see, e.g., [9])

$$\hat{\mathbf{d}}_{\text{ML}} = \arg \min_{\mathbf{d} \in \mathcal{D}} \{\|\mathbf{r} - \mathbf{H}\mathbf{d}\|^2\}. \quad (1.5)$$

Here, $\mathcal{D} = \mathcal{A}^M$ denotes the set of all possible transmitted data vectors \mathbf{d} . The cardinality of \mathcal{D} is $|\mathcal{D}| = |\mathcal{A}|^M$ and thus grows exponentially with M .

ML detection corresponds to a *nonconvex* optimization problem because \mathcal{D} is not a convex set [51, 52]. Therefore, numerical standard algorithms for convex optimization are not applicable. For general \mathbf{H} and general \mathbf{r} , there are no known algorithms that perform ML detection with a complexity that behaves polynomially in M (e.g., [7, 53]). For example, the straightforward solution of (1.5) by comparing $\|\mathbf{r} - \mathbf{H}\mathbf{d}\|^2$ for all $\mathbf{d} \in \mathcal{D}$ (often referred to as exhaustive-search ML detection) has a computational complexity that grows exponentially in M . In fact, its complexity may be already excessive for moderate values of M and constellation size $|\mathcal{A}|$. In general, ML detection can fully exploit all of the available diversity [3, 4]. That is, for SM systems with iid Gaussian channels, ML detection achieves a diversity order of M_R (cf. Section 1.7.2).

A very promising and more efficient alternative to exhaustive-search ML detection is *Fincke and Post's sphere decoding algorithm* (FPSD) [8] for ML detection and its numerous optimum [7, 9, 54, 55] as well as suboptimum [18, 56, 57] variants. With sphere decoding, the *average* computational complexity (i.e., averaged over a sufficient number of realizations of \mathbf{H}) of ML detection can be quite low – in particular, for practical values of M , $|\mathcal{A}|$, and for a sufficiently high SNR [50]. However, as recently shown in [58], the average complexity of FPSD still grows exponentially in M for SM systems with iid Gaussian channels. Throughout this thesis, the FPSD implementation of the ML detector serves as a performance benchmark. A nice flowchart of the FPSD can be found in [49] (see also the detailed discussion in [9] for finite constellations).

1.8.2 Equalization-Based Detection

In equalization-based (or linear) detection, an estimate of the transmitted data vector \mathbf{d} is formed as $\mathbf{y} = \mathbf{G}\mathbf{r}$ by using an equalizer \mathbf{G} . The detected data vector $\hat{\mathbf{d}}$ is then obtained through componentwise quantization $\hat{\mathbf{d}} = \mathcal{Q}_{\mathcal{A}}\{\mathbf{y}\}$ according to the used symbol alphabet \mathcal{A} , i.e.

$$\hat{d}_m = \arg \min_{a \in \mathcal{A}} |y_m - a|, \quad m = 1, \dots, M. \quad (1.6)$$

For the *zero-forcing* (ZF) detector, \mathbf{G} is given by the pseudo-inverse [59] of \mathbf{H} , i.e.,

$$\mathbf{G}_{\text{ZF}} \triangleq \mathbf{H}^\# = (\mathbf{H}^H \mathbf{H})^{-1} \mathbf{H}^H. \quad (1.7)$$

(For the last expression, we assumed that $N \geq M$ and that \mathbf{H} has full rank.) Thus, the result of ZF equalization (before quantization) is

$$\mathbf{y}_{\text{ZF}} = \mathbf{H}^\# \mathbf{r} = (\mathbf{H}^H \mathbf{H})^{-1} \mathbf{H}^H \mathbf{r} = \mathbf{d} + \tilde{\mathbf{w}}, \quad (1.8)$$

which is the transmitted data vector \mathbf{d} corrupted by the transformed noise $\tilde{\mathbf{w}} = \mathbf{H}^\# \mathbf{w}$. This means that the interference caused by \mathbf{H} is completely removed (“forced to zero”). In general, however, the transformed noise $\tilde{\mathbf{w}}$ is larger than \mathbf{w} (“noise enhancement”) and $\tilde{\mathbf{w}}$ is *correlated* with correlation matrix

$$\mathbf{R}_{\tilde{\mathbf{w}}} = \sigma_w^2 (\mathbf{H}^H \mathbf{H})^{-1}. \quad (1.9)$$

The ZF-equalized received vector \mathbf{y}_{ZF} can be seen as the solution to a *relaxed* ML problem (cf. (1.5)), where the data set \mathcal{D} underlying ML detection is relaxed to the convex set \mathbb{C}^M [52]:

$$\mathbf{y}_{\text{ZF}} = \arg \min_{\mathbf{y} \in \mathbb{C}^M} \{\|\mathbf{r} - \mathbf{H}\mathbf{y}\|^2\}.$$

The noise enhancement effect plaguing the ZF equalizer can be reduced by using the *minimum mean-square error* (MMSE) equalizer

$$\mathbf{G}_{\text{MMSE}} \triangleq (\mathbf{H}^H \mathbf{H} + \sigma_w^2 \mathbf{I})^{-1} \mathbf{H}^H, \quad (1.10)$$

which is the equalizer \mathbf{G} minimizing the mean-square error $E\{\|\mathbf{G}\mathbf{r} - \mathbf{d}\|^2\}$ [60]. Thus, the result of MMSE equalization is

$$\mathbf{y}_{\text{MMSE}} = (\mathbf{H}^H \mathbf{H} + \sigma_w^2 \mathbf{I})^{-1} \mathbf{H}^H \mathbf{r}. \quad (1.11)$$

This can again be seen as the solution to a relaxed ML problem, with the distance $\|\mathbf{r} - \mathbf{H}\mathbf{y}\|^2$ augmented by a penalty term $\sigma_w^2 \|\mathbf{y}\|^2$ that prevents \mathbf{y} from growing too large, i.e.

$$\mathbf{y}_{\text{MMSE}} = \arg \min_{\mathbf{y} \in \mathbb{C}^M} \{\|\mathbf{r} - \mathbf{H}\mathbf{y}\|^2 + \sigma_w^2 \|\mathbf{y}\|^2\}.$$

There also exist more sophisticated detection techniques based on the principle of relaxing the ML problem (e.g., semidefinite relaxation techniques [52, 61]).

ZF or MMSE equalization alone does not, in general, imply a loss of information, i.e. the ML detector (1.5) could still be based on \mathbf{y}_{ZF} or \mathbf{y}_{MMSE} . For example, the ML distance $\|\mathbf{r} - \mathbf{H}\mathbf{d}\|^2$ can be written in terms of \mathbf{y}_{ZF} according to

$$\|\mathbf{r} - \mathbf{H}\mathbf{d}\|^2 = (\mathbf{y}_{\text{ZF}} - \mathbf{d})^H \mathbf{H}^H \mathbf{H} (\mathbf{y}_{\text{ZF}} - \mathbf{d}) + \|\mathbf{r}\|^2 - \|\mathbf{H}\mathbf{y}_{\text{ZF}}\|^2, \quad (1.12)$$

and thus ML detection (1.5) based on \mathbf{y}_{ZF} becomes

$$\hat{\mathbf{d}}_{\text{ML}} = \arg \min_{\mathbf{d} \in \mathcal{D}} \{(\mathbf{d} - \mathbf{y}_{\text{ZF}})^H \mathbf{H}^H \mathbf{H} (\mathbf{d} - \mathbf{y}_{\text{ZF}})\}. \quad (1.13)$$

This can be interpreted as the ML detector for an identity channel corrupted by noise having correlation matrix (1.9). We may view (1.13) as “ML detection after ZF equalization” or “ML detection in the ZF-equalized domain,” as opposed to “direct ML detection” according to (1.5). Note that (1.5) and (1.13) are strictly equivalent. This also shows that for an orthogonal \mathbf{H} , i.e. $\mathbf{H}^H \mathbf{H} \propto \mathbf{I}$, the ML detector in (1.13) becomes $\hat{\mathbf{d}}_{\text{ML}} = \arg \min_{\mathbf{d} \in \mathcal{D}} \{\|\mathbf{d} - \mathbf{y}_{\text{ZF}}\|^2\}$ and thus is equal to the ZF solution $\hat{\mathbf{d}}_{\text{ZF}} = \mathcal{Q}_{\mathcal{A}}\{\mathbf{y}_{\text{ZF}}\}$. In this special case, the ML problem is very simple and can be solved efficiently (this also motivates the design of orthogonal space-time block codes, e.g., [34, 35]).

In general, however, ZF or MMSE equalization is strongly suboptimal due to the *componentwise quantization* of \mathbf{y}_{ZF} or \mathbf{y}_{MMSE} since the correlation of the transformed noise is not taken into account. In fact, for SM systems with iid Gaussian channels, ZF or MMSE detection can only exploit a diversity of order $M_{\text{R}} - M_{\text{T}} + 1$ [4] (while ML detection can exploit all of the available diversity of M_{R}). Thus, for $M_{\text{T}} \approx M_{\text{R}}$ their diversity gain is small. One explanation is that equalization-based detection “uses up” the degrees of freedom that would otherwise offer diversity (e.g., [4, 62]). However, their computational complexity is very low. The task with highest complexity is the calculation of the equalizer matrix \mathbf{G} that just contributes to the preparation complexity C_{prep} (cf. Section 1.7.3). Thus, for $M = N$, C_{prep} behaves as $\mathcal{O}(M^3)$. Furthermore, the vector complexity C_{vector} is governed by the equalization step with complexity $\mathcal{O}(M^2)$. Note that MMSE detection differs from ML or ZF detection in that it requires an estimate of the noise variance.

1.8.3 Nulling-and-Cancelling

In contrast to linear detection, nulling-and-cancelling (NC) uses a serial decision-feedback approach to detect the layers one after another [11, 13, 14, 63]. At each decoding step, a single layer is detected and the corresponding contribution to the received vector \mathbf{r} is subtracted; the other layers that have not been detected yet are “nulled out” (equalized) using a ZF or MMSE equalizer.

At the first decoding step, ZF or MMSE equalization based detection is applied to a certain layer $m_1 \in \{1, \dots, M\}$, yielding $\hat{d}_{m_1} = \mathcal{Q}_{\mathcal{A}}\{(\mathbf{G}\mathbf{r})_{m_1}\}$. Then, the interference corresponding to \hat{d}_{m_1} is subtracted from \mathbf{r} :

$$\mathbf{r}^{(2)} = \mathbf{r} - \mathbf{h}_{m_1} \hat{d}_{m_1},$$

where \mathbf{h}_{m_1} denotes the m_1 th column of the system matrix \mathbf{H} . If the decision \hat{d}_{m_1} was correct, i.e. $\hat{d}_{m_1} = d_{m_1}$, we obtain the reduced system model

$$\mathbf{r}^{(2)} = \mathbf{H}^{(2)}\mathbf{d}^{(2)} + \mathbf{w}. \quad (1.14)$$

Here, the reduced system matrix $\mathbf{H}^{(2)}$ of size $N \times (M - 1)$ is the system matrix \mathbf{H} with the m_1 th column removed, and the reduced data vector $\mathbf{d}^{(2)}$ of size $M - 1$ is the data vector \mathbf{d} with the m_1 th component removed. At the second decoding step, we perform a re-indexing of the remaining layers, i.e. $\{1, \dots, M\} \setminus \{m_1\} \rightarrow \{1, \dots, M - 1\}$, and we detect a specific layer $d_{m_2}^{(2)}$ with $m_2 \in \{1, \dots, M - 1\}$. This detection is based on the reduced system model in (1.14), i.e., $\hat{d}_{m_2}^{(2)} = \mathcal{Q}_A\{(\mathbf{G}^{(2)}\mathbf{r}^{(2)})_{m_2}\}$, where $\mathbf{G}^{(2)}$ denotes the ZF or MMSE equalizer corresponding to $\mathbf{H}^{(2)}$. Subsequently, the interference corresponding to $\hat{d}_{m_2}^{(2)}$ is subtracted from $\mathbf{r}^{(2)}$. This detection-subtraction procedure is repeated until all M layers are detected.

It is seen that NC attempts to progressively clean \mathbf{r} from the interference caused by the components already detected. At each new detection step, additional degrees of diversity become available provided that all previous decisions were correct. The performance of NC depends crucially on the *order* of the layers m_1, \dots, m_M . To minimize error propagation effects and to optimally support the processing of unreliable layers by means of the additional degrees of diversity that become available in the reduced system models, more reliable layers should be detected first. Therefore, the layers are commonly ordered (sorted) using the *layer-wise ZF or MMSE post-equalization SNRs* (PSNRs) as measures of layer reliability [13, 63, 64] (also known as V-BLAST ordering). Let us focus on the first decoding step (i.e. the full system model). By \mathbf{g}_m^H denoting the m th row of the equalizer \mathbf{G} , the m th component of the equalized received vector $\mathbf{y} = \mathbf{G}\mathbf{r}$ can be written as $y_m = \mathbf{g}_m^H(\mathbf{H}\mathbf{d} + \mathbf{w})$. Here, y_m is composed of a desired signal component (due to d_m), an interference term (due to $d_{m'}, m' \neq m$), and filtered noise. Thus, the resulting PSNR of the m th layer is given by

$$\text{SNR}_m \triangleq \frac{\mathbb{E}\{|\mathbf{g}_m^H \mathbf{h}_m d_m|^2\}}{\mathbb{E}\{|\sum_{m' \neq m} \mathbf{g}_m^H \mathbf{h}_{m'} d_{m'}|^2\} + \mathbb{E}\{|\mathbf{g}_m^H \mathbf{w}|^2\}}, \quad (1.15)$$

and we thus have

$$m_1^{\text{NC}} = \arg \max_{m \in \{1, \dots, M\}} \text{SNR}_m \quad (1.16)$$

for the first layer in the decoding process. For ZF and MMSE equalization, respectively, (1.15) results in (see, e.g., [65])

$$\text{SNR}_{\text{ZF}, m} \triangleq \frac{1}{\sigma_{\mathbf{w}}^2 [(\mathbf{H}^H \mathbf{H})^{-1}]_{m, m}}, \quad \text{and} \quad (1.17)$$

$$\text{SNR}_{\text{MMSE}, m} \triangleq \frac{1}{\sigma_{\mathbf{w}}^2 [(\mathbf{H}^H \mathbf{H} + \sigma_{\mathbf{w}}^2 \mathbf{I})^{-1}]_{m, m}} - 1. \quad (1.18)$$

The remaining layer-sorting steps are analogous, however, by using the corresponding reduced system matrices (e.g., by using $\mathbf{H}^{(2)}$ for the second layer decoding step). In general, NC with this PSNR-based layer sorting significantly outperforms NC without layer sorting, although it is still far inferior to ML detection. A completely equivalent formulation of the NC scheme can be obtained via the QR-decomposition of \mathbf{H} [66]. It is more efficient but has the drawback that PSNR layer sorting cannot be performed directly. The layer sorting employed for QR-based NC is often inferior [67, 68].

For SM systems with iid Gaussian channels, the diversity that can be exploited by NC is again $M_R - M_T + 1$ [4, 62, 69] (i.e. the same as for equalization-based detection). This is because the diversity exploited by NC is limited by the first step of the detection process, which is equalization-based detection applied to the full system model and thus has diversity order $M_R - M_T + 1$. In the practically relevant SNR regime (not extremely high SNRs), however, higher slopes of the SER-versus-SNR curve can be achieved.

The preparation complexity C_{prep} of NC is dominated by the calculation of the M equalizers $\mathbf{G}^{(s)}$, $s = 1, \dots, M$, that correspond to subsequently reduced system models. Thus, C_{prep} scales as $\mathcal{O}(M^4)$ assuming $M = N$. The vector complexity C_{vector} includes M equalization and detection steps (each for a single component) and M interference cancellation steps, which gives $\mathcal{O}(M^2)$ for C_{vector} . Compared to linear detection, in particular C_{prep} is increased. However, C_{prep} of NC can again be reduced to $\mathcal{O}(M^3)$ by using the recursive implementation of [70].

2

Geometry-Based Detectors for Spatial Multiplexing

THE starting point of the proposed geometry-based detectors [15–17], is the examination why suboptimal detection schemes (in particular, the equalization-based and nulling-and-cancelling (NC) schemes) fail to exploit all the diversity that is available in SM systems, cf. Section 1.8. Here, we adopt an approach to explain their inferior performance by comparing the decision regions of these schemes with the decision regions of the ML detector. The “improper” decision regions of suboptimal schemes are no problem for channel realizations with a condition number near to 1 (note that for condition number 1, zero-forcing equalization followed by componentwise quantization is equivalent to ML detection). However, for channel realizations with a large condition number, a significant performance degradation occurs. In fact, it turns out that these “bad”¹ channel realizations with large condition number are to a great extent responsible for the inferior average performance of suboptimal detection algorithms.

Motivated by this insight, we introduce an idealized model for bad channels that allows a substantially simplified implementation of ML detection. When extended to nonidealized channels, this yields the novel *line-search detector* (LSD). A subsequent simplification for constant modulus alphabets results in the novel *sphere-projection algorithm* (SPA). The SPA is an efficient nonlinear add-on to standard suboptimal detection schemes that makes these schemes robust to bad channel realization. For SM systems of practical interest (e.g., 6 transmit antennas and 6 receive antennas), the detection schemes obtained by this approach are demonstrated to yield excellent performance at low computational cost.

This chapter is organized as follows. In Section 2.1, the effects of bad channels on suboptimum detection algorithms are discussed. An idealized model for bad channels is introduced in Section

¹The term “bad” refers to the poor performance of suboptimal detection schemes for SM systems. Note, however, that these channels are not necessarily “bad” in the sense of, e.g., low channel capacity.

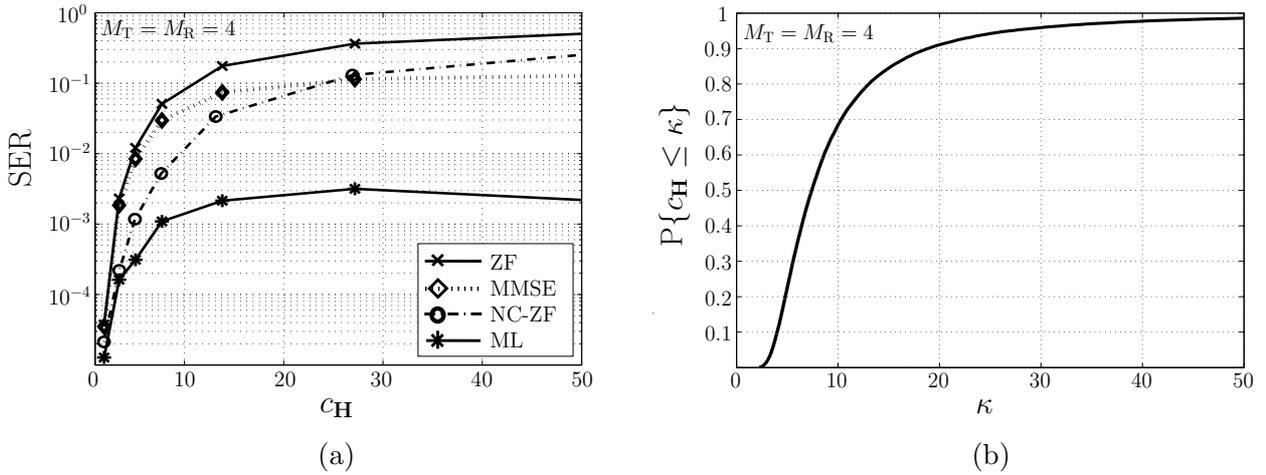


Figure 2.1: Detector performance and channel condition number. (a) SER performance of various detection schemes versus condition number c_H of the respective channel realization for a $M_T = M_R = 4$ MIMO channel with 4-QAM modulation and a fixed SNR of 15dB. (b) Cumulative distribution function of the condition number c_H .

Section 2.2, and an efficient ML detection algorithm for this model is developed in Section 2.3. In Section 2.4, this detection algorithm is extended to arbitrary MIMO channels that yields the LSD (for which it will not be ML anymore). In Section 2.5, a simplification of this latter detection algorithm yields the SPA. Finally, simulation results presented in Section 2.6 show that the proposed algorithms (in particular the SPA schemes) can yield near-ML performance with significantly less computational complexity than that required by the FPSD algorithm for ML detection.

2.1 The “Bad Channel” Effect

In this section, we will demonstrate that the inferior performance of suboptimal detection compared to ML detection is mainly caused by the occurrence of bad channel realizations.

2.1.1 Detector Performance and Bad Channels

In what follows, we will use the singular value decomposition (SVD) $\mathbf{H} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^H$, where the diagonal matrix $\mathbf{\Sigma}$ contains the singular values σ_m , $m = 1, \dots, M_T$, of \mathbf{H} and the matrices \mathbf{U} and \mathbf{V} contain, respectively, the left and right singular vectors of \mathbf{H} as columns [59]. We assume that the σ_m are indexed in nonincreasing order. The *condition number* $c_H = \sigma_1/\sigma_{M_T} \geq 1$ is the ratio of the largest to smallest singular value. For a bad (poorly conditioned) channel, c_H is large.

Experiments suggest that the performance of suboptimal detection schemes strongly depends on the channel’s condition number c_H . In Figure 2.1(a), the symbol error rate (SER) performance of the various detection schemes is shown versus the condition number of the channel realization. In this simulation, we used a $M_T = M_R = 4$ channel with iid Gaussian channel matrix entries, 4-QAM

modulation, and an SNR of 15 dB. It can be seen that there is a significant performance gap between linear (i.e., ZF or MMSE) and ML detection for $c_{\mathbf{H}}$ about 4 or larger, and also between ZF-based NC (with layer sorting using the PSNRs, cf. (1.17)) and ML detection for $c_{\mathbf{H}}$ about 8 or larger.

The impact of this behavior on the *average* SER performance of suboptimal detection of course depends on the probability with which bad channels occur. In Figure 2.1(b), we show the cumulative distribution function (cdf) of $c_{\mathbf{H}}$ estimated in the course of the simulation described above. It is seen that the probability that $c_{\mathbf{H}}$ exceeds a value of 8, 10 and 20 is about 50%, 30% and 9%, respectively. This suggests that bad channels occur frequently enough to cause a significant degradation of the average performance of suboptimal detection schemes. Theoretical investigations show that for an increasing number of transmit and receive antennas the cdf is essentially expanded (scaled) in the $c_{\mathbf{H}}$ direction [71]. We can thus expect the bad channel effect to become even worse for an increasing number of antennas.

While the performance of ML detection is fairly robust to bad channel realizations, it is noteworthy that the computational complexity of the FPSD algorithm for ML detection significantly increases for bad channels [8, 29]. Thus, bad channels either yield a poor performance of suboptimum detection schemes or result in an increased computational complexity of ML detection. Current detection algorithms are unable to achieve near-ML performance for bad channels at low computational cost.

2.1.2 Geometrical Analysis

The starting point for developing the following improved detection methods is a geometrical analysis of the decision regions of suboptimal detection methods in the case of bad channel realizations.

ZF Detection

We first consider linear detection based on ZF equalization. As we also discussed in Section 1.8.2, for a perfectly conditioned channel, i.e., $c_{\mathbf{H}} = 1$, we have $\mathbf{H}^H \mathbf{H} \propto \mathbf{I}$, and the ML detector in (1.13) becomes equal to the ZF detector solution $\hat{\mathbf{d}}_{\text{ZF}} = \mathcal{Q}_{\mathcal{A}}\{\mathbf{y}_{\text{ZF}}\}$. On the other hand, for a poorly conditioned channel $\mathbf{H}^H \mathbf{H}$ is quite different from being proportional to \mathbf{I} . Thus, the components of $\tilde{\mathbf{w}} = \mathbf{H}^{\#} \mathbf{w}$ are generally correlated, and the ZF solution $\hat{\mathbf{d}}_{\text{ZF}}$ must be expected to be far away from the optimal ML solution² $\hat{\mathbf{d}}_{\text{ML}}$.

For a geometrical analysis, we consider the probability density function (pdf) of the ZF-filtered Gaussian noise vector $\tilde{\mathbf{w}}$. The contour surfaces of this pdf are hyperellipsoids [72]. Using $\mathbf{H} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^H$, the correlation matrix $\mathbf{R}_{\tilde{\mathbf{w}}} = \sigma_{\mathbf{w}}^2 (\mathbf{H}^H \mathbf{H})^{-1}$ (cf. (1.9)) of $\tilde{\mathbf{w}}$ can be written as

$$\mathbf{R}_{\tilde{\mathbf{w}}} = \sigma_{\mathbf{w}}^2 \mathbf{V}\mathbf{\Sigma}^{-2}\mathbf{V}^H.$$

The m th principal axis of the hyperellipsoids is such that its direction is given by the m th eigenvector

²Note that even if $c_{\mathbf{H}} > 1$, it is possible that \mathbf{H} has orthogonal columns, however, with different norms. In this case, $\mathbf{H}^H \mathbf{H}$ is diagonal, which means that the components of $\tilde{\mathbf{w}}$ are still uncorrelated and ZF detection will still be optimal. However, simulations show that for poorly conditioned channel realizations a situation close to this case is very unlikely.

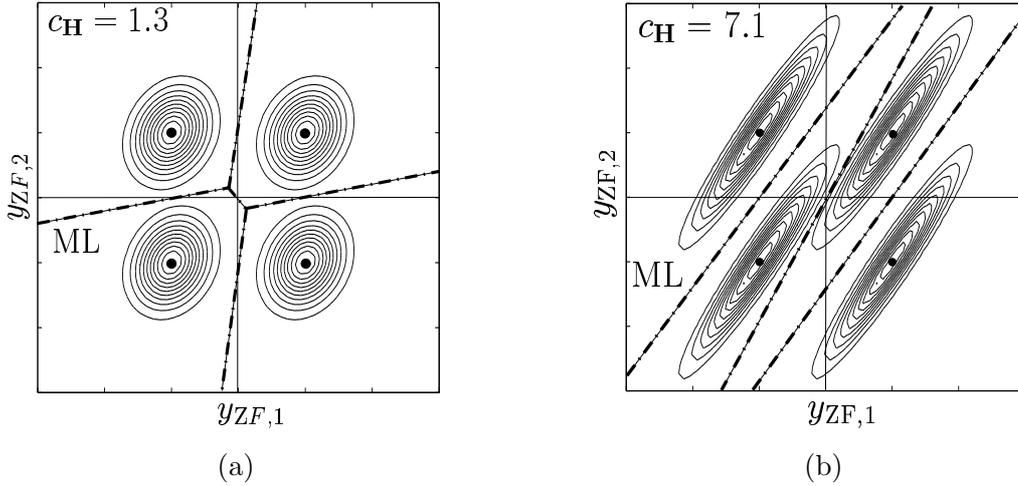


Figure 2.2: Probability density function of \mathbf{y}_{ZF} and ZF and ML decision regions in the ZF-equalized domain for a $M_T = M_R = 2$ channel with BPSK modulation. (a) Channel realization with condition number 1.3. (b) Channel realization with condition number 7.1. The ZF decision regions are the four quadrants; the ML decision regions are indicated by dash-dotted lines.

\mathbf{v}_m of $\mathbf{R}_{\tilde{\mathbf{w}}}$, which is equal to the m th column of \mathbf{V} , and its length is proportional to the square root of the m th eigenvalue of $\mathbf{R}_{\tilde{\mathbf{w}}}$ [72], which is equal to

$$\sigma_{\tilde{w},m} = \frac{\sigma_w}{\sigma_m}. \quad (2.1)$$

Thus, ZF equalization results in a *distortion* of the noise pdf with respect to the spherical geometry of the pdf of the original noise vector \mathbf{w} .

For illustration, Figure 2.2 shows the pdf of the received vector after ZF equalization, \mathbf{y}_{ZF} , for two different realizations of a real-valued $M_T = M_R = 2$ channel with condition numbers 1.3 and 7.1, respectively. The modulation format is BPSK. This figure also shows the ZF decision regions (the four quadrants) and the ML decision regions (indicated by dash-dotted lines). The ZF and ML decision regions are similar for the “good” channel with condition number 1.3, whereas they are dramatically different for the “bad” channel with condition number 7.1. Indeed, in the latter case the ML decision regions are nicely matched to the distorted noise pdf, but the ZF decision regions are not because they correspond to simple componentwise quantization. In particular, the boundary lines of the ML decision regions differ mainly by offsets that are orthogonal to the dominant principal axis \mathbf{v}_{M_T} (corresponding to the dominant eigenvalue σ_{w,M_T}^2 of $\mathbf{R}_{\tilde{\mathbf{w}}}$ and, thus, to the dominant noise component). This is intuitive, since any shift in the received vector in the direction of the dominant noise component is very likely caused by noise. For bad channels, it is thus desirable that the decision regions are approximately invariant to shifts in the direction of the dominant principal axis \mathbf{v}_{M_T} . In general, the decision regions of linear detection schemes cannot have this property because their boundary lines always go through the origin.

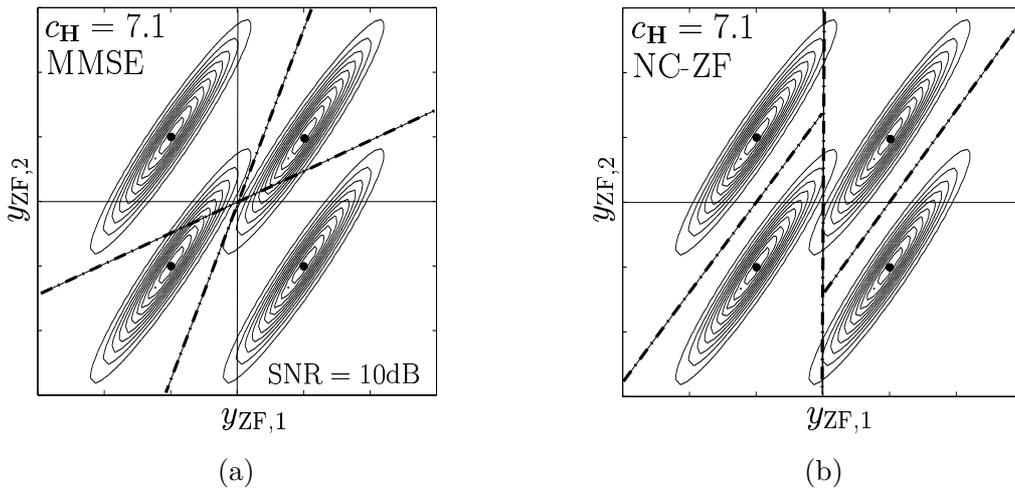


Figure 2.3: Same as Figure 2.2(b), but with MMSE decision regions (a) and ZF-based NC decision regions (b) instead of ML decision regions.

MMSE Detection

Figure 2.3(a) shows the decision regions of MMSE equalization based detection in the ZF equalized domain³ for the bad channel realization (condition number 7.1). The angles of the boundary lines of the MMSE decision regions are better matched to the distorted noise pdf than those of the ZF decision regions; however, the boundary lines still go through the origin and thus cannot implement the offsets that would allow them to become similar to the ML decision regions.

NC Detection

Finally, Figure 2.3(b) shows the decision regions (again represented in the ZF equalized domain) for ZF-based NC. Because the first symbol is simply ZF detected, one boundary of the corresponding decision region is fixed to the ordinate, which is again quite different from the ML decision regions. For successive symbols, offsets can be realized only to a certain extent. As a consequence, ZF-based NC detection performs better than ZF detection but is still significantly poorer than ML detection.

2.2 The Idealized Bad Channel Model

The previous results suggest that the average performance of suboptimal detection schemes can be improved by making these schemes robust to bad channels. Specifically, the decision regions should be made approximately invariant to a shift in the direction of the dominant noise axis. As a basis for

³We represent and compare the decision regions of all detectors in a common domain, namely, the domain obtained after ZF equalization. ZF equalization by itself does not imply any loss of optimality and has the advantage that the symbols are at the correct positions.

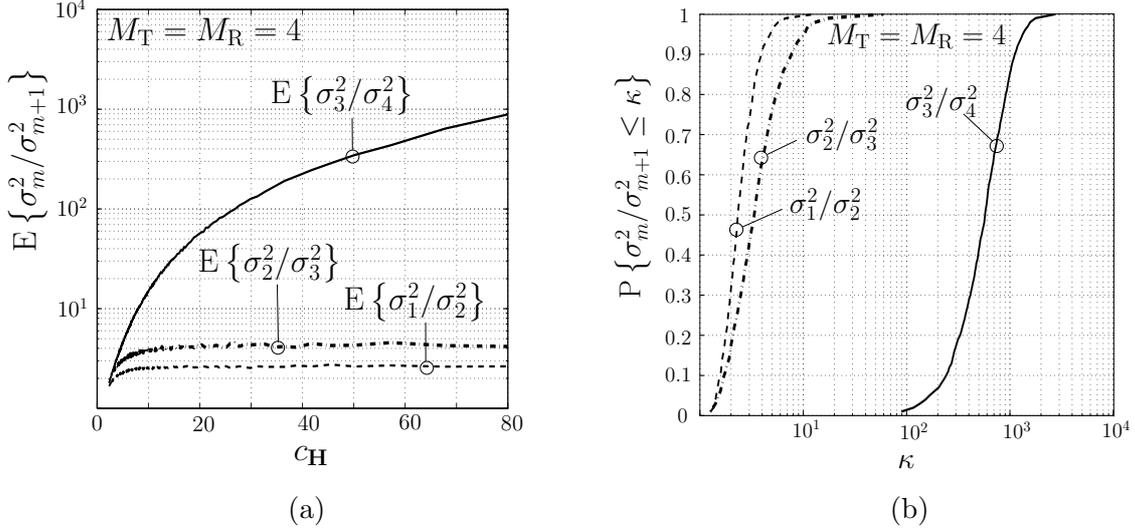


Figure 2.4: Ratio of noise variances $\sigma_m^2/\sigma_{m+1}^2$ in the direction of consecutive principal axes in the ZF-equalized domain. (a) Average ratios $E\{\sigma_m^2/\sigma_{m+1}^2\}$ versus $c_{\mathbf{H}}$ of the respective channel realization for a $M_T = M_R = 4$ iid Gaussian channel. (b) cdf of the ratios $\sigma_m^2/\sigma_{m+1}^2$ for $c_{\mathbf{H}} = 70$.

such a modification, we will first formulate the *idealized bad channel* (IBC) model. It will allow us to derive an efficient near-ML detection algorithm for bad channels and it represents the basis for the LSD and the SPA.

2.2.1 Formulation of the IBC

Our IBC model can be motivated as follows. According to (2.1), i.e., $\sigma_{\tilde{\mathbf{w}},m}^2 = \sigma_{\mathbf{w}}^2/\sigma_m^2$, the ratios of the noise variances in the directions of consecutive principal axes are given by $\sigma_{\tilde{\mathbf{w}},m+1}^2/\sigma_{\tilde{\mathbf{w}},m}^2 = \sigma_m^2/\sigma_{m+1}^2$. Figure 2.4(a) shows estimates of the *average* ratios $E\{\sigma_m^2/\sigma_{m+1}^2\}$ for a $M_T = M_R = 4$ iid Gaussian channel versus $c_{\mathbf{H}} = \sigma_1^2/\sigma_4^2$ (i.e., the $E\{\sigma_m^2/\sigma_{m+1}^2\}$ were estimated by averaging over different realizations of \mathbf{H} with a given condition number $c_{\mathbf{H}}$). It can be seen that for a bad channel (i.e., large $c_{\mathbf{H}}$), on average, σ_3^2/σ_4^2 is much larger than the other ratios σ_2^2/σ_3^2 and σ_1^2/σ_2^2 . Thus, on average, σ_4^2 is *much smaller* than the other singular values $\sigma_1^2, \sigma_2^2, \sigma_3^2$ or, equivalently, the largest principal-axis noise variance $\sigma_{\tilde{\mathbf{w}},4}^2 = \sigma_{\mathbf{w}}^2/\sigma_4^2$ is *much larger* than the other principal-axis noise variances $\sigma_{\tilde{\mathbf{w}},1}^2, \sigma_{\tilde{\mathbf{w}},2}^2, \sigma_{\tilde{\mathbf{w}},3}^2$. Simulations show that a similar behavior is exhibited also by higher-dimensional channels. To demonstrate that the ratios $\sigma_m^2/\sigma_{m+1}^2$ are well concentrated around their mean $E\{\sigma_m^2/\sigma_{m+1}^2\}$, Figure 2.4(b) shows the cdf of $\sigma_m^2/\sigma_{m+1}^2$ for $c_{\mathbf{H}} = 70$.

This suggests that for a bad channel, the largest noise variance (corresponding to the smallest singular value σ_{M_T} of \mathbf{H} and the associated principal axis \mathbf{v}_{M_T}) dominates all the other noise variances (see also [73] in context of multi-user precoding) and, hence, causes the main part of the bad channel effects that plague suboptimal detection. Therefore, we approximate a bad channel \mathbf{H} by the IBC model $\tilde{\mathbf{H}}$ that is constructed by setting the smallest singular value equal to zero and the remaining

singular values equal to the largest singular value:

$$\tilde{\mathbf{H}} \triangleq \mathbf{U}\tilde{\Sigma}\mathbf{V}^H, \quad \text{with } \tilde{\sigma}_{M_T} = 0 \text{ and } \tilde{\sigma}_1 = \tilde{\sigma}_2 = \cdots = \tilde{\sigma}_{M_T-1} = \sigma_1. \quad (2.2)$$

Note that whereas the singular vectors \mathbf{v}_m of the respective channel realization \mathbf{H} —and thus also the principal axis directions of the ZF-domain noise $\tilde{\mathbf{w}}$ as well—are maintained, the principal-axis noise variances are modified because we use an *infinite* dominant noise variance $\tilde{\sigma}_{\tilde{\mathbf{w}}, M_T}^2 = \sigma_{\mathbf{w}}^2 / \tilde{\sigma}_{M_T}^2 = \infty$ and *equal* remaining noise variances $\tilde{\sigma}_{\tilde{\mathbf{w}}, m}^2 = \sigma_{\mathbf{w}}^2 / \tilde{\sigma}_m^2 = \sigma_{\mathbf{w}}^2 / \sigma_1^2 = \sigma_{\tilde{\mathbf{w}}, 1}^2$ for $m = 1, \dots, M_T - 1$. (Actually, we will see presently that the values of these remaining noise variances do not matter as long as they are finite and equal.) Indeed, approximating a channel \mathbf{H} with the IBC $\tilde{\mathbf{H}}$ is equivalent to approximating the hyperellipsoids constituting the contour surfaces of the pdf of $\tilde{\mathbf{w}}$ by hypercylinders of infinite length whose axis is the dominant noise axis \mathbf{v}_{M_T} . Of course, actual channel realizations will not conform to the IBC. In particular, iid Gaussian channels \mathbf{H} have full rank with probability one [71]. Our IBC approximation is motivated by the desire to make the decision regions invariant to a shift in the direction of \mathbf{v}_{M_T} , as described in Subsection 2.1.2.

2.2.2 ML Detection for the IBC

Although the IBC $\tilde{\mathbf{H}}$ is only a rough approximation of a bad channel \mathbf{H} , it models an essential part of the “bad channel” effects. We now consider ML detection for the IBC (termed *IML detection*); this will later serve as a basis for developing an efficient near-ML detector for bad channels. A different application (multi-user precoding) of our IBC approximation can be found in [20].

According to (1.13), the IML decision rule is

$$\hat{\mathbf{d}}_{\text{IML}} = \arg \min_{\mathbf{d} \in \mathcal{D}} \left\{ (\mathbf{d} - \mathbf{y}_{\text{ZF}})^H \tilde{\mathbf{H}}^H \tilde{\mathbf{H}} (\mathbf{d} - \mathbf{y}_{\text{ZF}}) \right\}. \quad (2.3)$$

Using (2.2), we have

$$\tilde{\mathbf{H}}^H \tilde{\mathbf{H}} = \mathbf{V}\tilde{\Sigma}^2\mathbf{V}^H = \sigma_1^2 (\mathbf{I} - \mathbf{v}_{M_T}\mathbf{v}_{M_T}^H) = \sigma_1^2 \mathbf{P}_{\mathbf{v}_{M_T}}^\perp,$$

where $\mathbf{P}_{\mathbf{v}_{M_T}}^\perp \triangleq \mathbf{I} - \mathbf{v}_{M_T}\mathbf{v}_{M_T}^H$ denotes the orthogonal projector onto the space orthogonal to \mathbf{v}_{M_T} . Thus, (2.3) becomes

$$\hat{\mathbf{d}}_{\text{IML}} = \arg \min_{\mathbf{d} \in \mathcal{D}} \left\{ (\mathbf{d} - \mathbf{y}_{\text{ZF}})^H \mathbf{P}_{\mathbf{v}_{M_T}}^\perp (\mathbf{d} - \mathbf{y}_{\text{ZF}}) \right\} = \arg \min_{\mathbf{d} \in \mathcal{D}} \left\{ \left\| \mathbf{P}_{\mathbf{v}_{M_T}}^\perp (\mathbf{d} - \mathbf{y}_{\text{ZF}}) \right\|^2 \right\}, \quad (2.4)$$

where we have used $\mathbf{P}_{\mathbf{v}_{M_T}}^\perp = \mathbf{P}_{\mathbf{v}_{M_T}}^{\perp H} \mathbf{P}_{\mathbf{v}_{M_T}}^\perp$. Because of the projector $\mathbf{P}_{\mathbf{v}_{M_T}}^\perp$ the norm of the component of $\mathbf{d} - \mathbf{y}_{\text{ZF}}$ perpendicular to \mathbf{v}_{M_T} is minimized, while the component of $\mathbf{d} - \mathbf{y}_{\text{ZF}}$ in the direction of \mathbf{v}_{M_T} (the dominant noise direction) is completely suppressed in this minimization. Thus, as desired, the IML decision regions are invariant to this latter component.

For an alternative formulation and geometrical interpretation of the IML decision rule (2.4), let us define the *reference line* \mathcal{L} as the straight line that is parallel to the dominant noise axis \mathbf{v}_{M_T} and whose offset from the origin is \mathbf{y}_{ZF} , i.e.

$$\mathcal{L}: \quad \mathbf{y}_{\mathcal{L}}(\alpha) \triangleq \alpha \mathbf{v}_{M_T} + \mathbf{y}_{\text{ZF}}, \quad (2.5)$$

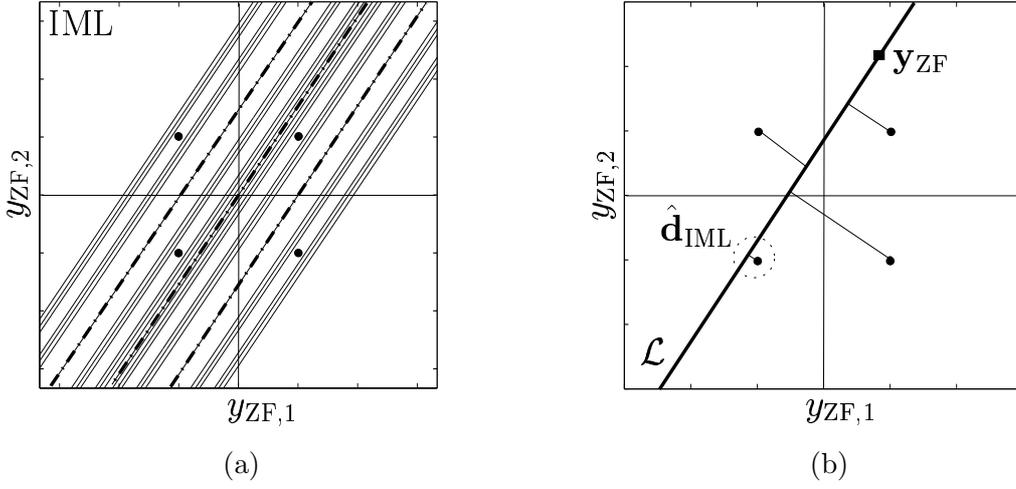


Figure 2.5: Illustration of IML detection for a real-valued $M_T = M_R = 2$ channel and BPSK modulation. (a) pdf of \mathbf{y}_{ZF} for the IBC corresponding to the bad channel of Figure 2.2 and Figure 2.3 and IML decision regions indicated by dash-dotted lines. (b) Reference-line geometry of IML detection according to (2.7).

where $\alpha \in \mathbb{C}$ denotes the line parameter. We maintain that the norm $\|\mathbf{P}_{\mathbf{v}_{M_T}}^\perp(\mathbf{d} - \mathbf{y}_{ZF})\|$ minimized in (2.4) is equal to the distance of \mathbf{d} from \mathcal{L} . Indeed, let $\mathbf{y}_{\mathcal{L}}(\alpha_{\min}(\mathbf{d}))$ denote the point of \mathcal{L} closest to a given data vector \mathbf{d} , i.e.

$$\alpha_{\min}(\mathbf{d}) \triangleq \arg \min_{\alpha \in \mathbb{C}} \left\{ \|\mathbf{d} - \mathbf{y}_{\mathcal{L}}(\alpha)\|^2 \right\} = \arg \min_{\alpha \in \mathbb{C}} \left\{ \|\mathbf{d} - (\alpha \mathbf{v}_{M_T} + \mathbf{y}_{ZF})\|^2 \right\} = \mathbf{v}_{M_T}^H (\mathbf{d} - \mathbf{y}_{ZF}). \quad (2.6)$$

Then, the distance of \mathbf{d} from \mathcal{L} is given by $\|\mathbf{d} - \mathbf{y}_{\mathcal{L}}(\alpha_{\min}(\mathbf{d}))\|$, and this distance can easily be shown to be equal to $\|\mathbf{P}_{\mathbf{v}_{M_T}}^\perp(\mathbf{d} - \mathbf{y}_{ZF})\|$:

$$\begin{aligned} \|\mathbf{d} - \mathbf{y}_{\mathcal{L}}(\alpha_{\min}(\mathbf{d}))\| &= \|\mathbf{d} - \mathbf{y}_{\mathcal{L}}(\mathbf{v}_{M_T}^H (\mathbf{d} - \mathbf{y}_{ZF}))\| = \|\mathbf{d} - \mathbf{v}_{M_T}^H (\mathbf{d} - \mathbf{y}_{ZF}) \mathbf{v}_{M_T} - \mathbf{y}_{ZF}\| \\ &= \|(\mathbf{I} - \mathbf{v}_{M_T} \mathbf{v}_{M_T}^H)(\mathbf{d} - \mathbf{y}_{ZF})\| = \|\mathbf{P}_{\mathbf{v}_{M_T}}^\perp(\mathbf{d} - \mathbf{y}_{ZF})\|. \end{aligned}$$

Thus, as claimed, ML detection for the IBC is equivalent to finding the data vector $\mathbf{d} \in \mathcal{D}$ that minimizes the distance $\|\mathbf{d} - \mathbf{y}_{\mathcal{L}}(\alpha_{\min}(\mathbf{d}))\|$ from the reference line \mathcal{L} :

$$\hat{\mathbf{d}}_{IML} = \arg \min_{\mathbf{d} \in \mathcal{D}} \left\{ \|\mathbf{d} - \mathbf{y}_{\mathcal{L}}(\alpha_{\min}(\mathbf{d}))\|^2 \right\}. \quad (2.7)$$

Therefore, $\hat{\mathbf{d}}_{IML}$ is the data vector closest to the reference line \mathcal{L} . This formulation will be essential for developing an efficient ML detection algorithm in Section 2.3.

To continue the example of Figure 2.2 and Figure 2.3, Figure 2.5(a) depicts the pdf of \mathbf{y}_{ZF} for the IBC associated with the bad channel realization with $c_{\mathbf{H}} = 7.1$. The IML decision regions are also shown. The geometry of the reference-line formulation of IML detection (2.7) is illustrated in Figure 2.5(b) for one specific realization of \mathbf{y}_{ZF} .

2.3 An Efficient ML Detector for the IBC

In this section, we develop an efficient ML detection algorithm with complexity $\mathcal{O}(M_T^3)$ for the IBC. Thus, it turns out that ML detection for the IBC is *not* exponentially complex in the number of transmit antennas M_T . This is possible, since the structure induced by the IBC is very strong. We first show that the reference-line formulation of the IML detector derived in the previous section allows a significant reduction of the search set \mathcal{D} .

2.3.1 The Reduced Search Set

Let \mathcal{Z}_r be the ZF decision region (in the ZF-equalized domain) corresponding to a data vector $\mathbf{d}^{(r)} \in \mathcal{D}$. That is, for any $\mathbf{y} \in \mathcal{Z}_r$ the ZF decision is $\hat{\mathbf{d}}_{\text{ZF}} = \mathbf{d}^{(r)}$. Because in the ZF-equalized domain the ZF decision is a simple componentwise quantization, every $\mathbf{y} \in \mathcal{Z}_r$ is closer to $\mathbf{d}^{(r)}$ than to any other data vector $\mathbf{d}^{(r')} \in \mathcal{D}$, $r' \neq r$. We shall also say that $\mathbf{d}^{(r)}$ is “the data vector corresponding to \mathcal{Z}_r ”, where

$$\mathcal{Z}_r \triangleq \{\mathbf{y} \mid \|\mathbf{d}^{(r)} - \mathbf{y}\|^2 \leq \|\mathbf{d}^{(r')} - \mathbf{y}\|^2, r' \neq r\}. \quad (2.8)$$

THEOREM 1. *The ML detector for the IBC in (2.4) is equivalent to*

$$\hat{\mathbf{d}}_{\text{IML}} = \arg \min_{\mathbf{d} \in \tilde{\mathcal{D}}} \left\{ \|\mathbf{P}_{\mathbf{v}_{M_T}}^\perp (\mathbf{d} - \mathbf{y}_{\text{ZF}})\|^2 \right\}, \quad (2.9)$$

where the reduced search set $\tilde{\mathcal{D}} \subset \mathcal{D}$ is given by all data vectors $\mathbf{d}^{(r)} \in \mathcal{D}$ that correspond to ZF decision regions \mathcal{Z}_r pierced by the reference line \mathcal{L} in (2.5).

Proof. As before, let $\mathbf{y}_{\mathcal{L}}(\alpha_{\min}(\mathbf{d}))$ denote the point of \mathcal{L} closest to \mathbf{d} , so that $\|\mathbf{d} - \mathbf{y}_{\mathcal{L}}(\alpha_{\min}(\mathbf{d}))\|$ expresses the distance between \mathbf{d} and \mathcal{L} (cf. (2.6)). Assume (proof by contradiction) that $\hat{\mathbf{d}}_{\text{IML}}$ does *not* correspond to any ZF decision region that is pierced by \mathcal{L} , i.e., no point of \mathcal{L} lies in the ZF decision region of $\hat{\mathbf{d}}_{\text{IML}}$. Hence, in particular, $\mathbf{y}_{\mathcal{L}}(\alpha_{\min}(\hat{\mathbf{d}}_{\text{IML}}))$ does not lie in the ZF decision region of $\hat{\mathbf{d}}_{\text{IML}}$, and thus it must lie in the ZF decision region \mathcal{Z}_r of some other data vector $\mathbf{d}^{(r)} \in \mathcal{D}$, $\mathbf{d}^{(r)} \neq \hat{\mathbf{d}}_{\text{IML}}$. That is, $\mathbf{y}_{\mathcal{L}}(\alpha_{\min}(\hat{\mathbf{d}}_{\text{IML}}))$ is closer to $\mathbf{d}^{(r)}$ than to $\hat{\mathbf{d}}_{\text{IML}}$, i.e.,

$$\|\mathbf{d}^{(r)} - \mathbf{y}_{\mathcal{L}}(\alpha_{\min}(\hat{\mathbf{d}}_{\text{IML}}))\| < \|\hat{\mathbf{d}}_{\text{IML}} - \mathbf{y}_{\mathcal{L}}(\alpha_{\min}(\hat{\mathbf{d}}_{\text{IML}}))\|.$$

For this other data vector $\mathbf{d}^{(r)}$, we also have

$$\|\mathbf{d}^{(r)} - \mathbf{y}_{\mathcal{L}}(\alpha_{\min}(\mathbf{d}^{(r)}))\| \leq \|\mathbf{d}^{(r)} - \mathbf{y}_{\mathcal{L}}(\alpha_{\min}(\hat{\mathbf{d}}_{\text{IML}}))\|$$

because $\mathbf{y}_{\mathcal{L}}(\alpha_{\min}(\mathbf{d}^{(r)}))$ is the point of \mathcal{L} closest to $\mathbf{d}^{(r)}$, and thus $\mathbf{y}_{\mathcal{L}}(\alpha_{\min}(\hat{\mathbf{d}}_{\text{IML}}))$ cannot be closer to $\mathbf{d}^{(r)}$ than $\mathbf{y}_{\mathcal{L}}(\alpha_{\min}(\mathbf{d}^{(r)}))$. Combining the two inequalities from above yields

$$\|\mathbf{d}^{(r)} - \mathbf{y}_{\mathcal{L}}(\alpha_{\min}(\mathbf{d}^{(r)}))\| < \|\hat{\mathbf{d}}_{\text{IML}} - \mathbf{y}_{\mathcal{L}}(\alpha_{\min}(\hat{\mathbf{d}}_{\text{IML}}))\|,$$

which means that $\mathbf{d}^{(r)}$ is closer to \mathcal{L} than $\hat{\mathbf{d}}_{\text{IML}}$. But we know from (2.7) that $\hat{\mathbf{d}}_{\text{IML}}$ is the data vector closest to \mathcal{L} . Hence, we have a contradiction, and the proof is complete. \square

Thus, the reduced search set $\tilde{\mathcal{D}}$ consists of all data vectors $\mathbf{d} \in \mathcal{D}$ whose ZF decision regions are pierced by the reference line \mathcal{L} . To characterize $\tilde{\mathcal{D}}$, it is sufficient to specify for each $\mathbf{d} \in \tilde{\mathcal{D}}$ an arbitrary point \mathbf{y} of the decision region corresponding to \mathbf{d} , since \mathbf{d} can easily be recovered from \mathbf{y} by using componentwise quantization according to the symbol alphabet \mathcal{A} , i.e. $\mathbf{d} = \mathcal{Q}_{\mathcal{A}}\{\mathbf{y}\}$. The idea behind the algorithm proposed next is that these points can be found in \mathcal{L} , and thus the search for $\tilde{\mathcal{D}}$ can be restricted to \mathcal{L} .

2.3.2 Partitioning of the Reference Line

The intersection of the ZF decision regions with the reference line \mathcal{L} induces a partitioning of \mathcal{L} . We will now show that for a wide class of symbol alphabets, this partitioning can be calculated easily. Subsequently, this will serve as a basis for an efficient determination of the reduced search set $\tilde{\mathcal{D}}$.

We assume that the symbol alphabet \mathcal{A} is “*line-structured*”. By this we mean that the boundaries of the symbol decision regions that are associated with the quantization operation $\mathcal{Q}_{\mathcal{A}}\{y\}$ of some complex value $y \in \mathbb{C}$ (cf. (1.6)) are given by *straight lines*. Examples of line-structured alphabets are QAM and PSK alphabets but not, e.g., an hexagonal alphabet. The boundary lines of the symbol decision regions for the 4-QAM and the 8-PSK alphabet are illustrated in Figure 4.2(a) and Figure 4.3(a), respectively.

Let P denote the number of these boundary lines. The p th boundary line in the complex-valued symbol domain can be written as

$$b^{(p)}(\beta) = \beta u^{(p)} + o^{(p)}, \quad \beta \in \mathbb{R}, \quad p = 1, \dots, P, \quad (2.10)$$

where $u^{(p)} \in \mathbb{C}$ and $o^{(p)} \in \mathbb{C}$ define direction and offset, respectively. For given $u^{(p)}$ and $o^{(p)}$ we move along $b^{(p)}$ by varying β . For example, the 4-QAM alphabet $\mathcal{A} = 1/\sqrt{2} \{1 + j, -1 + j, -1 - j, 1 - j\}$ has $P = 2$ orthogonal boundary lines that are given by the real and imaginary axis described by $u^{(1)} = 1$, $o^{(1)} = 0$, and $u^{(2)} = j$, $o^{(2)} = 0$, respectively (cf. Figure 4.2(a)).

We now consider the partitioning of the reference line \mathcal{L} that is induced by the ZF decision regions \mathcal{Z}_r (cf. (2.8)), which are bounded by $M_{\text{T}}P$ hyperplanes. The (m, p) th hyperplane is obtained by setting the m th component of a vector \mathbf{y} equal to the p th boundary line (2.10). The partitioning of \mathcal{L} is induced by the intersection of \mathcal{L} with all hyperplanes. The intersection with the (m, p) th boundary hyperplane yields a straight boundary line $\mathcal{B}^{(m,p)} \subset \mathbb{C}$ for the line parameter α that can be calculated by equating (2.10) and the m th component of (2.5), i.e., $y_{\mathcal{L},m}(\alpha) = b^{(p)}(\beta)$. This gives

$$\alpha v_{M_{\text{T}},m} + y_{\text{ZF},m} = \beta u^{(p)} + o^{(p)},$$

and we obtain

$$\mathcal{B}^{(m,p)} : \quad \alpha = \alpha^{(m,p)}(\beta) = \beta \underbrace{\frac{u^{(p)}}{v_{M_{\text{T}},m}}}_{a^{(m,p)}} + \underbrace{\frac{1}{v_{M_{\text{T}},m}}(o^{(p)} - y_{\text{ZF},m})}_{b^{(m,p)}} = \beta a^{(m,p)} + b^{(m,p)}, \quad (2.11)$$

which describes a straight line in \mathbb{C} with direction $a^{(m,p)}$ and offset $b^{(m,p)}$. If we parameterize the reference line $\mathbf{y}_{\mathcal{L}}(\alpha)$ by $\alpha = \alpha^{(m,p)}$, the m th component of $\mathbf{y}_{\mathcal{L}}(\alpha)$ is equal to the p th boundary line

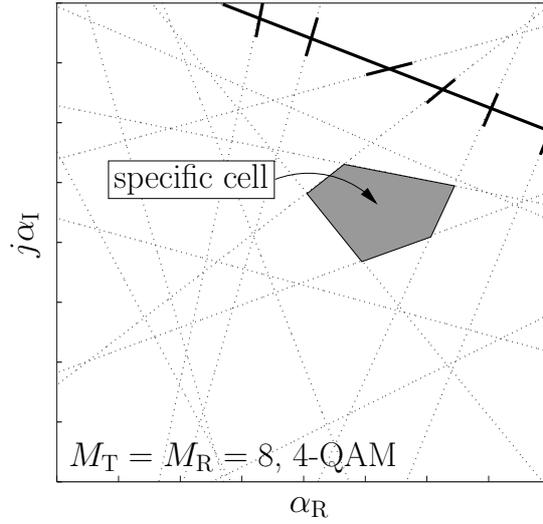


Figure 2.6: Boundary lines $\mathcal{B}^{(m,p)}$ in the line parameter $\alpha = \alpha_R + j\alpha_I$ of the reference line \mathcal{L} for a $M_T = M_R = 8$ channel and 4-QAM modulation. (The boldface line segment will be addressed at a later point.)

(2.10) of the symbol decisions regions. Thus, the p th boundary line appears rotated and shifted in the parameter α of the reference line $\mathbf{y}_{\mathcal{L}}(\alpha)$, which corresponds to the intersection of the (m,p) th hyperplane of the ZF decision regions with $\mathbf{y}_{\mathcal{L}}(\alpha)$. Furthermore, it is easy to verify that the angles between the P boundary lines $\mathcal{B}^{(m,p)}$, $p = 1, \dots, P$ corresponding to the m th component are equal to the angles between the boundary lines $b^{(p)}(\beta)$ in the symbol domain.

The superposition of all the $M_T P$ boundary lines $\mathcal{B}^{(m,p)}$, $m = 1, \dots, M_T$, $p = 1, \dots, P$ partition the line parameter α (and thus the reference line \mathcal{L}) into cells, and each of these cells correspond to the intersection of a ZF decision region with \mathcal{L} . In Figure 2.6, this cell partitioning of α is illustrated for a $M_T = M_R = 8$ channel and 4-QAM modulation; we obtain $M_T P = 16$ boundary lines. As shown in Section 2.3.1, the reduced search set $\tilde{\mathcal{D}}$ consists of all data vectors $\mathbf{d} \in \mathcal{D}$ whose ZF decision regions are pierced by \mathcal{L} . Thus, we have $|\tilde{\mathcal{D}}|$ elementary cells $\mathcal{C}^{(r)} \subset \mathbb{C}$, $r = 1, \dots, |\tilde{\mathcal{D}}|$, where the specific cell $\mathcal{C}^{(r)}$ corresponds to the intersection of the ZF decision region \mathcal{Z}_r with \mathcal{L} . Thus, all points of a given $\mathcal{C}^{(r)}$ will lead to the same ZF-quantized data vector. More specifically, for any $\alpha \in \mathcal{C}^{(r)}$ the ZF decision (quantization) of the corresponding vector $\mathbf{y}_{\mathcal{L}}(\alpha)$ leads to the same result $\mathbf{d}^{(r)}$, i.e.,

$$\mathbf{d}^{(r)} = \mathcal{Q}_{\mathcal{A}}\{\mathbf{y}_{\mathcal{L}}(\alpha)\}, \quad \text{for any } \alpha \in \mathcal{C}^{(r)}.$$

The data vector $\mathbf{d}^{(r)}$ then belongs to the reduced search set $\tilde{\mathcal{D}}$ because it corresponds to a ZF decision region that is pierced by \mathcal{L} . Therefore, any collection of arbitrary cell points—one point for each cell—defines the reduced search set $\tilde{\mathcal{D}}$. Note that the ZF equalized received vector $\mathbf{y}_{\text{ZF}} = \mathbf{H}^{\#} \mathbf{r}$ is obtained for $\alpha = 0$, see (2.5). Thus, the result of ZF detection, $\hat{\mathbf{d}}_{\text{ZF}} = \mathcal{Q}_{\mathcal{A}}\{\mathbf{y}_{\text{ZF}}\}$, also belongs to the reduced search set $\tilde{\mathcal{D}}$.

2.3.3 Size of Reduced Search Set

The results of the previous section allow us to derive an upper bound on the size $|\tilde{\mathcal{D}}|$ of the reduced search set (this upper bound will be tight for BPSK and 4-QAM alphabets). To find this upper bound, recall that the cell partitioning is defined by the $M_{\text{T}}P$ boundary lines in (2.11). Assuming that $k - 1$ boundary lines are given, it is easy to verify that each additional boundary line yields at most k new cells. The additional boundary line will yield exactly k new cells if and only if it intersects at *different* points with *all* the $k - 1$ previous lines. Thus,

$$|\tilde{\mathcal{D}}| \leq |\tilde{\mathcal{D}}|_{\max} = 1 + \sum_{k=1}^{M_{\text{T}}P} k = \frac{(M_{\text{T}}P)^2}{2} + \frac{M_{\text{T}}P}{2} + 1. \quad (2.12)$$

This upper bound is tight, i.e. $|\tilde{\mathcal{D}}| = |\tilde{\mathcal{D}}|_{\max}$, for BPSK and 4-QAM alphabets, since there will be no parallel boundary lines and only two boundary lines will intersect in a particular point. If there are parallel boundary lines (e.g., 16-QAM) and/or if more than two boundary lines intersect in the same point (e.g., 8-PSK), then $|\tilde{\mathcal{D}}| < |\tilde{\mathcal{D}}|_{\max}$.

Comparing $|\tilde{\mathcal{D}}|_{\max}$ with $|\mathcal{D}| = |\mathcal{A}|^{M_{\text{T}}}$, we see that for reasonably large $|\mathcal{A}|$ and M_{T} we have $|\tilde{\mathcal{D}}| \ll |\mathcal{D}|$. For example, for a $M_{\text{T}} = M_{\text{R}} = 8$ channel and 4-QAM modulation we obtain $|\tilde{\mathcal{D}}| = |\tilde{\mathcal{D}}|_{\max} = 137$ pierced decision regions (i.e. data vectors in $\tilde{\mathcal{D}}$) out of a total of $|\mathcal{D}| = 4^8 = 65536$ possible transmitted data vectors. One of these $|\tilde{\mathcal{D}}|$ data vectors is the ML solution for the IBC (cf. (2.9)). This illustrates the reduction of complexity achieved by the reduced search set $\tilde{\mathcal{D}}$.

Of course, the problem remains to find all the data vectors in $\tilde{\mathcal{D}}$ efficiently. The next section provides a solution to this problem.

2.3.4 Efficient Determination of Reduced Search Set

According to the previous section, the reduced search set $\tilde{\mathcal{D}}$ can be determined by finding an arbitrary point α out of each cell $\mathcal{C}^{(r)} \subset \mathbb{C}$, since the corresponding vector $\mathbf{y}_{\mathcal{L}}(\alpha)$ for $\alpha \in \mathcal{C}^{(r)}$ defines the data vector $\mathbf{d}^{(r)} \in \tilde{\mathcal{D}}$ associated to $\mathcal{C}^{(r)}$ by $\mathbf{d}^{(r)} = \mathcal{Q}_{\mathcal{A}}\{\mathbf{y}_{\mathcal{L}}(\alpha)\}$. To find points α out of each cell in an efficient manner, we suggest to systematically search the reference line along each boundary line.

To search along the (m, p) th boundary line $\mathcal{B}^{(m, p)}$ (2.11), we calculate the intersection points $\alpha_{\text{int}}^{(m, p; m', p')}$ of $\mathcal{B}^{(m, p)}$ with all $\mathcal{B}^{(m', p')}$, $(m', p') \neq (m, p)$ that are not parallel to $\mathcal{B}^{(m, p)}$. Equating $\alpha^{(m, p)}(\beta) = \beta a^{(m, p)} + b^{(m, p)}$ with $\alpha^{(m', p')}(\beta') = \beta' a^{(m', p')} + b^{(m', p')}$ and solving for β yields

$$\beta^{(m, p; m', p')} = \frac{a_{\text{R}}^{(m', p')} b_{\text{I}}^{(m', p')} + a_{\text{I}}^{(m', p')} (b_{\text{R}}^{(m, p)} - b_{\text{R}}^{(m', p')})}{a_{\text{R}}^{(m', p')} a_{\text{I}}^{(m, p)} - a_{\text{R}}^{(m, p)} a_{\text{I}}^{(m', p')}} ,$$

where the subscripts R and I denote the real and imaginary parts of a and b , respectively, and the intersection point of $\mathcal{B}^{(m, p)}$ and $\mathcal{B}^{(m', p')}$ is obtained as

$$\alpha_{\text{int}}^{(m, p; m', p')} = \beta^{(m, p; m', p')} a^{(m, p)} + b^{(m, p)}. \quad (2.13)$$

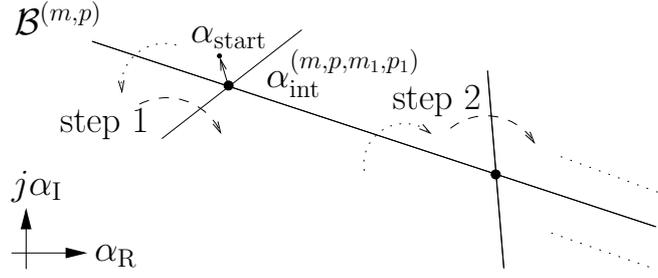


Figure 2.7: Algorithm for determining the data vectors $\mathbf{d}^{(r)} \in \tilde{\mathcal{D}}$ corresponding to all cells bounded by the (m, p) th boundary line $\mathcal{B}^{(m,p)}$ (cf. the boldface line in Figure 2.6). Based on the first intersection point $\alpha_{\text{int}}^{(m,p;m_1,p_1)}$, the first cell point α_{start} and the associated first data vector $\mathbf{d}^{(1)}$ are calculated. All remaining data vectors $\mathbf{d}^{(r)} \in \tilde{\mathcal{D}}$ associated with $\mathcal{B}^{(m,p)}$ are uniquely determined by the remaining intersection points, using the hops indicated by the dotted and dashed arrows.

As we move along $\mathcal{B}^{(m,p)}$, the intersection points with the other boundary lines (calculated previously) tell us where we cross the border from one cell to the next.⁴ We can sort these intersection points according to, e.g., monotonically increasing real parts, and let $\alpha_{\text{int}}^{(m,p;m_1,p_1)}$ be the intersection point with the smallest real part, i.e.

$$(m_1, p_1) = \arg \min_{m', p'} \alpha_{\text{int}, \text{R}}^{(m,p;m',p')}.$$

Figure 2.7 shows $\mathcal{B}^{(m,p)}$ (this could be the boldface line depicted in Figure 2.6) and the intersection point $\alpha_{\text{int}}^{(m,p;m_1,p_1)}$. To move from $\alpha_{\text{int}}^{(m,p;m_1,p_1)}$ into the first cell, we add a small offset α_{offset} and obtain the new point $\alpha_{\text{start}} = \alpha_{\text{int}}^{(m,p;m_1,p_1)} + \alpha_{\text{offset}}$. The vector in $\mathbb{C}^{M_{\text{T}}}$ on the reference line \mathcal{L} corresponding to α_{start} is given by $\mathbf{y}^{(1)} = \mathbf{y}_{\mathcal{L}}(\alpha_{\text{start}}) = \alpha_{\text{start}} \mathbf{v}_{M_{\text{T}}} + \mathbf{y}_{\text{ZF}}$. We then obtain our first data vector $\mathbf{d}^{(1)} \in \tilde{\mathcal{D}}$ associated with $\mathcal{B}^{(m,p)}$ as $\mathbf{d}^{(1)} = \mathcal{Q}_{\mathcal{A}}\{\mathbf{y}^{(1)}\}$.

The remaining data vectors $\mathbf{d}^{(r)} \in \tilde{\mathcal{D}}$ along $\mathcal{B}^{(m,p)}$ are now determined by alternately “hopping” over $\mathcal{B}^{(m,p)}$ and an intersecting boundary line as illustrated in Fig. 2.7. At each intersection, we perform one *step* consisting of two *hops*. The first hop (indicated by the dotted arrow) is over $\mathcal{B}^{(m,p)}$, i.e., the current search line; this corresponds to an update of the m th data vector component across the p th boundary. The second hop (indicated by the dashed arrow) is over $\mathcal{B}^{(m_1,p_1)}$, i.e., our first intersecting boundary line; this corresponds to an update of the m_1 th data vector component across the p_1 th boundary.

These data vector component updates can be performed without calculating new α values or corresponding $\mathbf{y}_{\mathcal{L}}(\alpha)$ vectors (i.e., no hops are actually implemented). Suppose we just obtained a specific data vector $\mathbf{d}^{(r)} \in \tilde{\mathcal{D}}$ and wish to determine the next data vector $\mathbf{d}^{(r+1)} \in \tilde{\mathcal{D}}$ corresponding to the cell we would reach by hopping over the intersecting boundary line $\mathcal{B}^{(m',p')}$. Now by definition,

⁴For simplicity of exposition, we assume that the intersection points of any two boundary lines are different. This holds for arbitrary QAM constellations; however, it does not hold for PSK constellations with $|\mathcal{A}| > 4$ because there all the P boundary lines in the symbol plane intersect at the origin, and this geometry is maintained in \mathcal{P} . For example, for 8-PSK we obtain M_{T} intersection points, each of which is the intersection of $P = 4$ boundary lines. However, these points are known *a priori* and the following algorithm can easily be extended to this situation.

$\mathcal{B}^{(m',p')}$ corresponds to the intersection of the (m', p') th boundary hyperplane of the ZF decision regions with the reference line \mathcal{L} . If the line parameter α crosses $\mathcal{B}^{(m',p')}$, the m' th component of $\mathbf{y}_{\mathcal{L}}(\alpha)$ crosses the p' th boundary line of the symbol decision regions (all other components of $\mathbf{y}_{\mathcal{L}}(\alpha)$ remain in the same symbol decision region). Thus, to determine $\mathbf{d}^{(r+1)}$ we just have to update the m' th component of $\mathbf{d}^{(r)}$ according to the p' th boundary line of the symbol decision regions. Using $\mathbf{d}^{(r)}$, this update for $\mathbf{d}^{(r+1)}$ can be written as

$$d_m^{(r+1)} = \begin{cases} d_m^{(r)}, & \text{if } m \neq m', \\ d_{m'}^{(r)} + \Delta d_{m',p'}^{(r)}, & \text{if } m = m', \end{cases}$$

for $m = 1 \dots M_T$, or equivalently

$$\mathbf{d}^{(r+1)} = \mathbf{d}^{(r)} + \Delta d_{m',p'}^{(r)} \mathbf{e}_{m'}, \quad (2.14)$$

where $\mathbf{e}_{m'}$ refers to the m' th unit vector. Here, the update value $\Delta d_{m',p'}^{(r)}$ depends on the m' th component of $\mathbf{d}^{(r)}$ and on the p' th boundary line. For example, for 4-QAM we have

$$\Delta d_{m',p'}^{(r)} = \begin{cases} -2j d_{m',I}^{(r)}, & \text{if } p' = 1, \\ -2 d_{m',R}^{(r)}, & \text{if } p' = 2. \end{cases}$$

Here, the update simply amounts to flipping the imaginary part (for $p' = 1$) or the real part (for $p' = 2$) of the m' th component of $\mathbf{d}^{(r)}$.

If we perform this systematic search along arbitrary $M_T P - 1$ boundary lines $\mathcal{B}^{(m,p)}$, it is guaranteed that all fundamental cells $\mathcal{C}^{(r)}$ for α are taken into account and thus all data vectors in $\tilde{\mathcal{D}}$ are found. It is evident that several data vectors will be multiple obtained. The search along $\mathcal{B}^{(m,p)}$ yields at most $2M_T P$ different data vectors, which results in a total maximum of $2M_T P(M_T P - 1)$ obtained data vectors. Compared with $|\tilde{\mathcal{D}}|_{\max} = (M_T P)^2/2 + M_T P/2 + 1$ (cf. (2.12)), we see that on average each data vector $\mathbf{d} \in \tilde{\mathcal{D}}$ is approximately found 4 times. For example, for a $M_T = M_R = 8$ channel and a 4-QAM alphabet we find 480 data vectors, of which 137 are different. But most of the multiply checked data vectors are obtained through efficient single component updates according to (2.14). This, in particular, enables an efficient recursive calculation of the corresponding distances.

2.3.5 Efficient Calculation of Distances

According to Theorem 1, we have to minimize the distance

$$\psi_{\text{IML}}^2(\mathbf{d}^{(r)}) \triangleq \|\mathbf{P}_{\mathbf{v}_{M_T}}^\perp (\mathbf{d}^{(r)} - \mathbf{y}_{\text{ZF}})\|^2 \quad (2.15)$$

over all data vectors $\mathbf{d}^{(r)} \in \tilde{\mathcal{D}}$ to find the ML solution for the IBC. Using $\mathbf{P}_{\mathbf{v}_{M_T}}^\perp = \mathbf{I} - \mathbf{v}_{M_T} \mathbf{v}_{M_T}^H$ and $\xi(\mathbf{d}^{(r)}) \triangleq \mathbf{v}_{M_T}^H (\mathbf{d}^{(r)} - \mathbf{y}_{\text{ZF}})$, this distance can be written as

$$\psi_{\text{IML}}^2(\mathbf{d}^{(r)}) = \|\mathbf{d}^{(r)} - \mathbf{y}_{\text{ZF}} - \xi(\mathbf{d}^{(r)}) \mathbf{v}_{M_T}\|^2,$$

which has a computational effort of $\mathcal{O}(M_T)$. In the previous section, we have shown how to determine $\tilde{\mathcal{D}}$ efficiently. We now present an efficient recursive algorithm for calculating $\psi_{\text{IML}}^2(\mathbf{d}^{(r)})$ based on the single component update according to (2.14).

Again, we systematically search along the boundary line $\mathcal{B}^{(m,p)}$. Suppose that $\psi_{\text{IML}}^2(\mathbf{d}^{(r)})$ and $\xi(\mathbf{d}^{(r)})$ has already been determined, and that the next data vector $\mathbf{d}^{(r+1)}$ corresponds to a hop over the intersecting boundary line $\mathcal{B}^{(m',p')}$ (cf. Figure 2.7). As was shown, $\mathbf{d}^{(r+1)}$ is a neighbor of $\mathbf{d}^{(r)}$ that differs from $\mathbf{d}^{(r)}$ only in the m' th component, namely, by $\Delta d_{m',p'}^{(r)}$. Thus, the distance for $\mathbf{d}^{(r+1)}$ is

$$\begin{aligned} \psi_{\text{IML}}^2(\mathbf{d}^{(r+1)}) &= \|\mathbf{P}_{\mathbf{v}_{M_T}}^\perp (\mathbf{d}^{(r)} + \Delta d_{m',p'}^{(r)} \mathbf{e}_{m'} - \mathbf{y}_{\text{ZF}})\|^2 \\ &= \|\mathbf{P}_{\mathbf{v}_{M_T}}^\perp (\mathbf{d}^{(r)} - \mathbf{y}_{\text{ZF}})\|^2 + \|\mathbf{P}_{\mathbf{v}_{M_T}}^\perp \Delta d_{m',p'}^{(r)} \mathbf{e}_{m'}\|^2 \\ &\quad + 2 \operatorname{Re}\left\{ (\mathbf{d}^{(r)} - \mathbf{y}_{\text{ZF}})^H \mathbf{P}_{\mathbf{v}_{M_T}}^{\perp H} \mathbf{P}_{\mathbf{v}_{M_T}}^\perp \Delta d_{m',p'}^{(r)} \mathbf{e}_{m'} \right\} \\ &= \psi_{\text{IML}}^2(\mathbf{d}^{(r)}) + \Delta^{(r)} \end{aligned} \quad (2.16)$$

with

$$\Delta^{(r)} = (1 - |v_{M_T, m'}|^2) |\Delta d_{m',p'}^{(r)}|^2 + 2 \operatorname{Re}\left\{ (d_{m'}^{(r)} - y_{\text{ZF}, m'} - v_{M_T, m'} \xi(\mathbf{d}^{(r)})) \Delta d_{m',p'}^{(r)*} \right\}. \quad (2.17)$$

Furthermore, $\xi(\mathbf{d}^{(r)})$ can be updated as $\xi(\mathbf{d}^{(r+1)}) = \xi(\mathbf{d}^{(r)}) + v_{M_T, m'}^* \Delta d_{m',p'}^{(r)}$. This recursion is initialized by calculating $\psi_{\text{IML}}^2(\mathbf{d}^{(r)})$ and $\xi(\mathbf{d}^{(r)})$ in a straightforward manner. The two updates, i.e. the calculation of $\psi_{\text{IML}}^2(\mathbf{d}^{(r+1)})$ and $\xi(\mathbf{d}^{(r+1)})$, have a complexity that is independent from M_T .

2.3.6 Summary of ML Detection for the IBC

We have now developed the main elements of our efficient ML detection algorithm for the IBC. Note that we still assume that the actual channel has IBC structure, i.e. $\mathbf{H} = \tilde{\mathbf{H}}$ (cf. (2.2)). In an implementation of this algorithm, the recursive procedure described in Section 2.3.4 and Section 2.3.5 has to be performed for all $M_T P$ boundary lines except for the last one (whose data vectors have all been processed before), so that all data vectors $\mathbf{d} \in \tilde{\mathcal{D}}$ are taken into account and the ML solution according to (2.9) is obtained.

The principal steps of our algorithm can finally be summarized as follows.

- Preparation steps:
 - Determine the right singular vector \mathbf{v}_{M_T} corresponding the zero singular value of $\tilde{\mathbf{H}}$. For general (non-IBC) channels, this step will be adressed in more detail in Section 2.4.
 - Calculate the ZF equalizer $\mathbf{G}_{\text{ZF}} = \tilde{\mathbf{H}}^\#$.
- Calculate the ZF equalized received vector $\mathbf{y}_{\text{ZF}} = \mathbf{G}_{\text{ZF}} \mathbf{r}$.
- Determine all $M_T P$ boundary lines $\mathcal{B}^{(m,p)}$ in \mathcal{L} (see (2.11)).
- Calculate all intersection points (2.13) and order them for each $\mathcal{B}^{(m,p)}$, e.g., according to increasing real parts.
- For each $\mathcal{B}^{(m,p)}$, determine the associated data vectors $\mathbf{d}^{(r)} \in \tilde{\mathcal{D}}$ and their distances $\psi_{\text{IML}}^2(\mathbf{d}^{(r)})$ as discussed in Section 2.3.4 and Section 2.3.5. The processing associated to $\mathcal{B}^{(m,p)}$ is summarized in Figure 2.8.
- Find $\hat{\mathbf{d}}_{\text{IML}} \in \tilde{\mathcal{D}}$ that has minimum distance $\psi_{\text{IML}}^2(\mathbf{d}^{(r)})$, $r = 1, \dots, |\tilde{\mathcal{D}}|$.

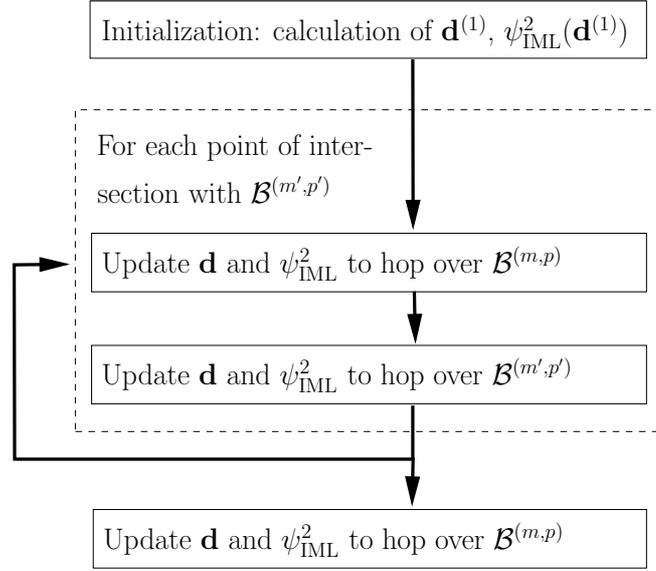


Figure 2.8: Processing associated to the (m,p) th boundary line $\mathcal{B}^{(m,p)}$.

2.3.7 Computational Complexity

As discussed in Section 1.7.3, we split the computational complexity into two components: (i) Preparation complexity C_{prep} , and (ii) vector complexity C_{vector} . For $M_{\text{T}} = M_{\text{R}}$, we now analyze how the complexity of the proposed ML detector for the IBC depends on the parameters $|\mathcal{A}|$ and M_{T} that determine the transmission rate. The preparation complexity C_{prep} is caused by the computation of \mathbf{G}_{ZF} and $\mathbf{v}_{M_{\text{T}}}$, which has the dominant complexity of $\mathcal{O}(M_{\text{T}}^3)$ (see, e.g., [59] when the SVD of $\tilde{\mathbf{H}}$ is employed). The vector complexity C_{vector} is composed as follows:

- Determination of all $M_{\text{T}}P$ boundary lines in \mathcal{L} : complexity $\mathcal{O}(M_{\text{T}}P)$.
- Calculation of all intersection points: complexity $\mathcal{O}(M_{\text{T}}^2P^2)$.
- Calculation of $M_{\text{T}}P$ initial data vectors and their associated distances: complexity $\mathcal{O}(M_{\text{T}}^2P)$.
- Calculation of roughly $2M_{\text{T}}^2P^2$ distance updates according to (2.16): complexity $\mathcal{O}(M_{\text{T}}^2P^2)$.

Hence, the overall vector complexity C_{vector} of our algorithm is $\mathcal{O}(M_{\text{T}}^2P^2)$. This is quadratic in the number of transmit antennas M_{T} and, since usually $P \propto |\mathcal{A}|$, quadratic in the symbol alphabet size $|\mathcal{A}|$. Thus, in contrast to ML detection for an arbitrary realization \mathbf{H} , ML detection for the IBC $\tilde{\mathbf{H}}$ is *not* exponentially complex in the number of antennas M_{T} . This is due to the strong structure induced by the IBC model.

2.4 The Line Search Detector

The algorithm presented in the last section performs ML detection for the IBC. This detector is only of theoretical interest because actual channel realizations will not conform to the IBC. Thus, we now

extend our algorithm to an arbitrary, i.e. non-IBC, MIMO channel \mathbf{H} .

2.4.1 Algorithm Statement

Specifically, we propose to use the IBC approximation $\tilde{\mathbf{H}}$ of \mathbf{H} to find the reduced search set $\tilde{\mathcal{D}}$, and then to minimize the “true” ML distance

$$\psi_{\text{ML}}^2(\mathbf{d}^{(r)}) \triangleq \|\mathbf{r} - \mathbf{H}\mathbf{d}^{(r)}\|^2$$

over all data vectors $\mathbf{d}^{(r)}$ in the reduced search set $\tilde{\mathcal{D}}$. The resulting algorithm we refer to as the *line search detector* (LSD). The corresponding detected data vector $\hat{\mathbf{d}}_{\text{LSD}}$ is given by

$$\hat{\mathbf{d}}_{\text{LSD}} \triangleq \arg \min_{\mathbf{d}^{(r)} \in \tilde{\mathcal{D}}} \psi_{\text{ML}}^2(\mathbf{d}^{(r)}). \quad (2.18)$$

Note that $\hat{\mathbf{d}}_{\text{LSD}}$ is *not* guaranteed to be equal to the ML decision $\hat{\mathbf{d}}_{\text{ML}}$ since we minimize ψ_{ML}^2 not over the whole data set \mathcal{D} of size $|\mathcal{A}|^{M_T}$ (which would have a computational complexity that is exponential in M_T) but over the reduced set $\tilde{\mathcal{D}}$. On the other hand, the complexity of our algorithm is only $\mathcal{O}(M_T^3)$ as will be shown presently.

The algorithm consists of similar steps as the ML detector for the IBC (cf. Section 2.3.6) but with some important differences:

- Calculate $\mathbf{G}_{\text{ZF}} = \mathbf{H}^\# = \mathbf{D}_{\text{ZF}} \mathbf{H}^H$, where $\mathbf{D}_{\text{ZF}} \triangleq (\mathbf{H}^H \mathbf{H})^{-1}$, and $\mathbf{y}_{\text{ZF}} = \mathbf{G}_{\text{ZF}} \mathbf{r}$. That is, \mathbf{y}_{ZF} is now calculated via the true non-IBC channel realization \mathbf{H} which is assumed to have full column rank.
- Determine the dominant eigenvector \mathbf{v}_{M_T} of \mathbf{D}_{ZF} . The vector \mathbf{v}_{M_T} is the right singular vector corresponding to the smallest singular vector of \mathbf{H} . Furthermore, \mathbf{v}_{M_T} is also the eigenvector of the inverse Gram matrix \mathbf{D}_{ZF} associated to its largest singular value. Thus, \mathbf{v}_{M_T} can be efficiently approximated by means of a few iterations of the power method [59], which is given by the following recursive formula:

$$\mathbf{v}_{M_T}^{(z)} = \frac{\mathbf{D}_{\text{ZF}} \mathbf{v}_{M_T}^{(z-1)}}{\|\mathbf{D}_{\text{ZF}} \mathbf{v}_{M_T}^{(z-1)}\|}, \quad z = 1 \dots Z. \quad (2.19)$$

Here, $\mathbf{v}_{M_T}^{(z)}$ denotes an approximation to \mathbf{v}_{M_T} obtained at the z th iteration of the power method. This recursion is initialized with a random vector for $\mathbf{v}_{M_T}^{(0)}$. For the determination of the reduced search set $\tilde{\mathcal{D}}$ we simply set $\mathbf{v}_{M_T} = \mathbf{v}_{M_T}^{(Z)}$. As we will demonstrate in Section 2.6, a few iterations (e.g., $Z = 4$) are completely sufficient. In particular, for a bad channel realization with only one small singular value (i.e., when the IBC is a good model for bad channel realizations) the power method just requires very few iterations to yield very good approximations to \mathbf{v}_{M_T} . In all other cases (e.g., $\sigma_1 \approx \sigma_2 \dots \approx \sigma_{M_T}$), the power method performs rather poor, i.e. with a few iterations no accurate approximation to \mathbf{v}_{M_T} is obtained. This, however, has no significant impact on the performance of the LSD, since a very inaccurate IBC model for determining $\tilde{\mathcal{D}}$ does not require

a very accurate \mathbf{v}_{M_T} . Thus, for the preparation steps of the LSD no SVD is required. The only additional preparation effort compared to ZF equalization is the approximation of \mathbf{v}_{M_T} via the power method.

- Using the IBC approximation of \mathbf{H} , find the reduced search set $\tilde{\mathcal{D}}$ as explained in Section 2.3.4.
- Search $\tilde{\mathcal{D}}$ for the data vector minimizing ψ_{ML}^2 . Here, we can again calculate the distances efficiently by means of a recursion similar to (2.16). Recall that through hopping over the intersecting boundary line $\mathcal{B}^{(m',p')}$ the data vector $\mathbf{d}^{(r+1)}$ is obtained from $\mathbf{d}^{(r)}$ via the update (2.14) of the m' th component by $\Delta d_{m',p'}^{(r)}$. The associated distance update results in

$$\psi_{\text{ML}}^2(\mathbf{d}^{(r+1)}) = \psi_{\text{ML}}^2(\mathbf{d}^{(r)}) + \Delta^{(r)}, \quad (2.20)$$

where the update term $\Delta^{(r)}$ is given by

$$\Delta^{(r)} = \|\mathbf{h}_{m'}\|^2 |\Delta d_{m',p'}^{(r)}|^2 - 2 \operatorname{Re} \left\{ \boldsymbol{\xi}(\mathbf{d}^{(r)})^H \mathbf{h}_{m'} \Delta d_{m',p'}^{(r)} \right\}. \quad (2.21)$$

Here, the vector $\boldsymbol{\xi}(\mathbf{d}^{(r)}) \triangleq \mathbf{r} - \mathbf{H}\mathbf{d}^{(r)}$ can be calculated recursively by using

$$\boldsymbol{\xi}(\mathbf{d}^{(r+1)}) = \boldsymbol{\xi}(\mathbf{d}^{(r)}) - \mathbf{h}_{m'} \Delta d_{m',p'}^{(r)},$$

and $\|\mathbf{h}_{m'}\|^2$ can be precalculated.

2.4.2 Discussion

There are two cases where the LSD is optimum, i.e. $\hat{\mathbf{d}}_{\text{LSD}}$ in (2.18) and the ML solution $\hat{\mathbf{d}}_{\text{ML}}$ in (1.5) coincide:

1. An *orthogonal channel*: Here, ZF detection is optimum, i.e. $\hat{\mathbf{d}}_{\text{ML}} = \hat{\mathbf{d}}_{\text{ZF}}$, and because $\hat{\mathbf{d}}_{\text{ZF}} \in \tilde{\mathcal{D}}$ (see Section 2.3.2) also $\hat{\mathbf{d}}_{\text{LSD}} = \hat{\mathbf{d}}_{\text{ZF}}$.
2. An *IBC*: Here, $\hat{\mathbf{d}}_{\text{LSD}} = \hat{\mathbf{d}}_{\text{IML}} = \hat{\mathbf{d}}_{\text{ML}}$.

We can thus expect near-ML performance for very good channels and for bad channels with a single dominant noise axis. Evidently, for good channel realizations, the LSD (including the determination of $\tilde{\mathcal{D}}$, etc.) is superfluous, and close-to optimum performance can already be achieved with any conventional suboptimum detector (like the simple ZF detector). This could motivate to perform the LSD just for bad channel realizations and not for every channel realization, which would allow to reduce the average (over many channel realization) computational complexity of the LSD.

2.4.3 Computational Complexity

For $M_T = M_R$, we now study the complexity order of the LSD. Again, the preparation complexity C_{prep} and the vector complexity C_{vector} are treated separately (cf. Section 1.7.3).

The preparation complexity C_{prep} is composed as follows:

- Calculation of $\mathbf{D}_{\text{ZF}} = (\mathbf{H}^H \mathbf{H})^{-1}$ and $\mathbf{G}_{\text{ZF}} = \mathbf{D}_{\text{ZF}} \mathbf{H}^H$: complexity $\mathcal{O}(M_{\text{T}}^3)$.
- Calculation of $\mathbf{v}_{M_{\text{T}}}$ via the power method (cf. (2.19)): complexity $\mathcal{O}(M_{\text{T}}^2)$.
- Calculation of $\|\mathbf{h}_m\|^2$ for $m = 1, \dots, M_{\text{T}}$ (cf. (2.21)): complexity $\mathcal{O}(M_{\text{T}}^2)$.

Thus, the dominant complexity of the preparatory steps is $\mathcal{O}(M_{\text{T}}^3)$. The following steps account to the vector complexity C_{vector} .

- Determination of all $M_{\text{T}}P$ boundary lines in \mathcal{L} and their intersection points: complexity $\mathcal{O}(M_{\text{T}}^2 P^2)$.
- Calculation of $M_{\text{T}}P$ initial data vectors and their associated distances: complexity $\mathcal{O}(M_{\text{T}}^3 P)$.
- Calculation of roughly $2 M_{\text{T}}^2 P^2$ distance updates according to (2.20) and (2.21): complexity $\mathcal{O}(M_{\text{T}}^3 P^2)$.

Hence, the overall vector complexity C_{vector} of the LSD is $\mathcal{O}(M_{\text{T}}^3 P^2)$. This is cubic in the number of transmit antennas M_{T} and, since usually $P \propto |\mathcal{A}|$, quadratic in the symbol alphabet size $|\mathcal{A}|$. The increased C_{vector} complexity of the LSD compared to the C_{vector} complexity of the ML detector for the IBC (cf. Section 2.3.7) is due to the computationally more intensive distance calculations (cf. (2.17) and (2.21)).

2.5 The Sphere Projection Algorithm

Although the LSD has a complexity order that is just cubic in the number of antennas, flops measurements will show that the LSD is still computationally rather intensive for SM systems of practical interest. In particular, the construction of the reduced search set involves many computations. In this section, we will achieve a substantial reduction of complexity through a further reduction of the data search set. Therefor we constrain the employed symbol alphabet to have *constant modulus* property, i.e. $|d_m| = 1$, $m = 1, \dots, M$, (including arbitrary PSK constellations like 4-QAM and BPSK). For an efficient construction of a reduced search set, we explicitly take this property into account. The resulting *sphere-projection algorithm* (SPA) [15, 17] can be viewed as a simple nonlinear add-on to an existing suboptimal scheme such as ZF, MMSE, or NC detection. This add-on improves the error-rate and diversity-gain performance of the suboptimal detector by making it robust to the bad channel effect discussed in Section 2.1.1.

The add-on construction of the SPA is as follows. Let $\hat{\mathbf{d}}$ denote the result of any conventional suboptimal detector. This result can be expected to be reasonably good for a good channel. In order to improve the performance for bad channels, we consider a suitably chosen *additional search set* $\mathcal{D}_+ \subset \mathcal{D}$ of valid data vectors \mathbf{d} that are potentially better than $\hat{\mathbf{d}}$ in the sense of a smaller ML distance $\|\mathbf{r} - \mathbf{H}\mathbf{d}\|^2$. We then minimize $\|\mathbf{r} - \mathbf{H}\mathbf{d}\|^2$ over the search set \mathcal{D}_{SP} that consists of $\hat{\mathbf{d}}$ and all data vectors in \mathcal{D}_+ :

$$\hat{\mathbf{d}}_{\text{SP}} \triangleq \arg \min_{\mathbf{d} \in \mathcal{D}_{\text{SP}}} \|\mathbf{r} - \mathbf{H}\mathbf{d}\|^2, \quad \text{with } \mathcal{D}_{\text{SP}} \triangleq \{\hat{\mathbf{d}}\} \cup \mathcal{D}_+. \quad (2.22)$$

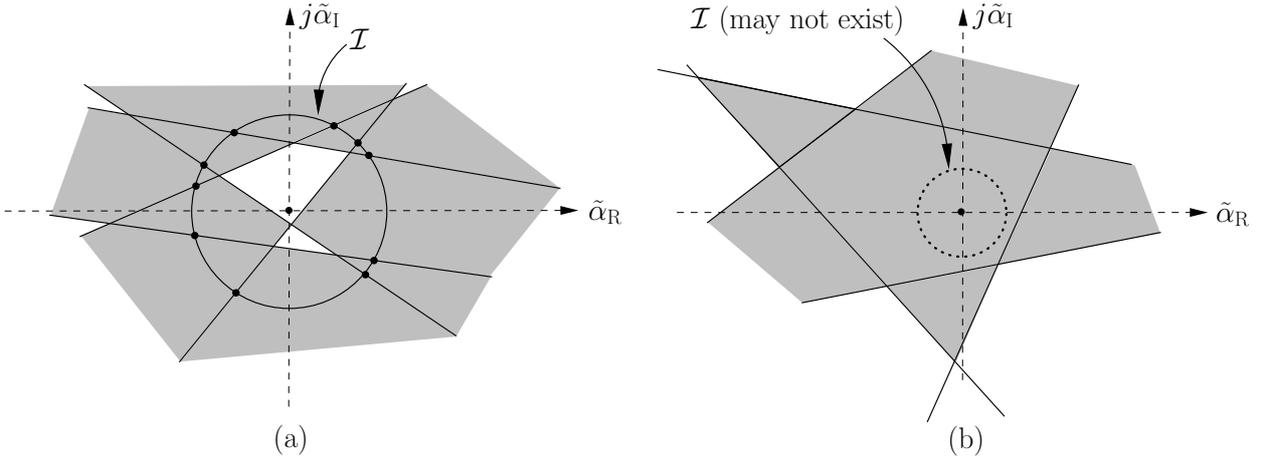


Figure 2.9: Geometry underlying the SPA. (a) Some boundary lines intersect the intersection circle \mathcal{I} (only possible for Case 1), (b) No boundary line intersects \mathcal{I} (possible for Case 1 and Case 2). The shaded regions indicate (some of) the cells corresponding to data vectors in \mathcal{D}_+ .

The SPA is an add-on to the given suboptimal detector because $\hat{\mathbf{d}}$ is calculated and included in the total search set \mathcal{D}_{SP} .

Due to the constant modulus property of the symbol alphabet, all data vectors \mathbf{d} are located on an M_T -dimensional “data hypersphere” \mathcal{H} about the origin, with radius $R = \sqrt{M_T}$. This geometry will allow a simple construction of \mathcal{D}_+ that uses a projection onto the data hypersphere \mathcal{H} (this explains the name SPA).

2.5.1 Construction of the Additional Search Set

The additional search set \mathcal{D}_+ has to be constructed such that it improves the detector performance in the bad channel case. Therefore, let us consider a bad channel whose dominant noise component in the direction of \mathbf{v}_{M_T} (i.e., of the reference line \mathcal{L}) is much larger than all other noise components. This channel can again be approximated by an IBC. Because the ML detector for the IBC chooses the data vector with minimum distance from \mathcal{L} (see Section 2.2.2), it makes sense to construct \mathcal{D}_+ as a set of data vectors that are close to \mathcal{L} . On the other hand, we know that all data vectors are located on the data hypersphere \mathcal{H} . With the SPA, we attempt to combine these two properties by choosing data vectors for \mathcal{D}_+ that are close to the intersection $\mathcal{L} \cap \mathcal{H}$. However, this intersection does not always exist. Therefore, we have to distinguish between two cases:

- *Case 1:* If \mathcal{L} intersects \mathcal{H} , we choose \mathcal{D}_+ to consist of data vectors $\mathbf{d} \in \mathcal{H}$ located at or at least close to the intersection $\mathcal{L} \cap \mathcal{H}$.
- *Case 2:* If \mathcal{L} does not intersect \mathcal{H} , we choose \mathcal{D}_+ to consist of data vectors $\mathbf{d} \in \mathcal{H}$ that are close to \mathcal{L} .

In the following, we shall elaborate on both cases.

Case 1: \mathcal{L} intersects \mathcal{H}

By using the decomposition $\mathbf{y}_{\text{ZF}} = \mathbf{y}_{\mathbf{v}_{M_T}} + \mathbf{y}_{\mathbf{v}_{M_T}}^\perp$, where $\mathbf{y}_{\mathbf{v}_{M_T}} = (\mathbf{v}_{M_T}^H \mathbf{y}_{\text{ZF}}) \mathbf{v}_{M_T}$ is colinear with \mathbf{v}_{M_T} and $\mathbf{y}_{\mathbf{v}_{M_T}}^\perp = \mathbf{P}_{\mathbf{v}_{M_T}}^\perp \mathbf{y}_{\text{ZF}}$ is orthogonal to \mathbf{v}_{M_T} , the reference line \mathcal{L} in (2.5) can be rewritten as

$$\mathcal{L}: \quad \mathbf{y}_{\mathcal{L}}(\tilde{\alpha}) = \tilde{\alpha} \mathbf{v}_{M_T} + \mathbf{y}_{\mathbf{v}_{M_T}}^\perp, \quad \text{with} \quad \tilde{\alpha} = \alpha + \mathbf{v}_{M_T}^H \mathbf{y}_{\text{ZF}} \in \mathbb{C}. \quad (2.23)$$

The intersection $\mathcal{L} \cap \mathcal{H}$ corresponds to the equation $\|\mathbf{y}_{\mathcal{L}}(\tilde{\alpha})\|^2 = R^2$ or, equivalently,

$$|\tilde{\alpha}|^2 + \|\mathbf{y}_{\mathbf{v}_{M_T}}^\perp\|^2 = R^2. \quad (2.24)$$

This defines an *intersection circle* \mathcal{I} for the transformed line parameter $\tilde{\alpha}$ with radius $\tilde{R} \triangleq \sqrt{R^2 - \|\mathbf{y}_{\mathbf{v}_{M_T}}^\perp\|^2}$ (note that $R^2 - \|\mathbf{y}_{\mathbf{v}_{M_T}}^\perp\|^2 \geq 0$ because we assumed that the intersection $\mathcal{L} \cap \mathcal{H}$ exists).

Next, we find all cells $\mathcal{C} \subset \mathcal{L}$ that are pierced by the intersection circle \mathcal{I} . We first calculate all $M_T P$ boundary lines $\mathcal{B}^{(m,p)}$. Afterwards, we calculate the intersection points—if they exist—of each $\mathcal{B}^{(m,p)}$ (see (2.11)) with \mathcal{I} (defined by (2.24)), i.e.,

$$|\alpha^{(m,p)}(\beta) + \mathbf{v}_{M_T}^H \mathbf{y}_{\text{ZF}}|^2 = \tilde{R}^2, \quad (2.25)$$

where $\alpha^{(m,p)}(\beta) = \beta a^{(m,p)} + b^{(m,p)}$. These intersection points are illustrated in Figure 2.9(a). (Here, we assume that at least one boundary line intersects \mathcal{I} . If no such intersection exists although \mathcal{L} intersects \mathcal{H} —cf. Figure 2.9(b)—the processing for Case 2 described further below has to be performed.) The intersection points are then sorted according to their angle. Starting with angle $\phi = 0$ we have $\tilde{\alpha} = \tilde{R}$ and the first data vector in \mathcal{D}_+ is obtained as $\mathbf{d}^{(1)} = \mathcal{Q}_{\mathcal{A}}\{\mathbf{y}_{\mathcal{L}}(\tilde{R})\}$. Now we move along \mathcal{I} from one intersection point to the next, which again corresponds to hops over subsequent boundary lines. In this process, we apply the data component update procedure from Section 2.3.4 to obtain the remaining data vectors $\mathbf{d}^{(r)} \in \mathcal{D}_+$, and we use the recursion (2.20) to efficiently calculate the distances $\psi_{\text{ML}}^2(\mathbf{d}^{(r)}) = \|\mathbf{r} - \mathbf{H}\mathbf{d}^{(r)}\|^2$ in (2.22).

In Case I where \mathcal{L} intersects \mathcal{H} , the size of the resulting additional search set \mathcal{D}_+ is bounded as $|\mathcal{D}_+| \leq 2 M_T P$ because each one of the $M_T P$ boundary lines $\mathcal{B}^{(m,p)}$ has 0 or 2 intersection points with \mathcal{I} . The worst case, $|\mathcal{D}_+| = 2 M_T P$, occurs if and only if all boundary lines intersect \mathcal{I} .

Case 2: \mathcal{L} does not intersect \mathcal{H}

If $R^2 \leq \|\mathbf{y}_{\mathbf{v}_{M_T}}^\perp\|^2$, \mathcal{L} and \mathcal{H} do not intersect (see Figure 2.9(b)). Evidently, $\mathbf{y}_{\mathbf{v}_{M_T}}^\perp$ (which according to (2.23) corresponds to $\tilde{\alpha} = 0$) is the point of \mathcal{L} with minimum distance from \mathcal{H} . We thus take $\mathbf{d}^{(1)} = \mathcal{Q}_{\mathcal{A}}\{\mathbf{y}_{\mathbf{v}_{M_T}}^\perp\}$ as the first data vector in \mathcal{D}_+ . Simulation results indicate that it is advantageous to include also the nearest neighbors of $\mathbf{d}^{(1)}$ in \mathcal{D}_+ . These additional data vectors can easily be found by substituting the nearest-neighbor symbols for the individual components of $\mathbf{d}^{(1)}$. For PSK constellations with $P \geq 2$ (like 4-QAM and 8-PSK), we obtain 2 nearest neighbors for each data vector component, yielding $|\mathcal{D}_+| = 2M_T + 1$. For BPSK data modulation ($P = 1$) we have $|\mathcal{D}_+| = M_T + 1$. The distances $\psi_{\text{ML}}^2(\mathbf{d}^{(r)}) = \|\mathbf{r} - \mathbf{H}\mathbf{d}^{(r)}\|^2$ in (2.22) can again be calculated recursively by using (2.20).

The same procedure is used if \mathcal{L} and \mathcal{H} intersect but no boundary line intersects the intersection circle \mathcal{I} (cf. the discussion of Case 1 above).

Discussion

In both Case 1 and Case 2, an important aspect of the SPA is that we look for the point(s) on the data hypersphere \mathcal{H} that is/are closest to the reference line \mathcal{L} . In a certain sense, this corresponds to a projection onto the hypersphere \mathcal{H} , which explains the name “sphere-projection algorithm.” In Case 1 where \mathcal{L} and \mathcal{H} intersect, the projection points are given by the intersection circle \mathcal{I} .

Note that the reduced search $\mathcal{D}_{\text{SP}} \triangleq \{\hat{\mathbf{d}}\} \cup \mathcal{D}_+$ used in (2.22) contains at most $2M_T P + 1$ data vectors. Thus, in the worst case $|\mathcal{D}_{\text{SP}}|$ increases just *linearly* in the number of transmit antennas and, since usually $P \propto |\mathcal{A}|$, *linearly* in the symbol alphabet size $|\mathcal{A}|$. In contrast, the size of the reduced search set $\tilde{\mathcal{D}}$ used for the LSD increases quadratically in both M_T and P (cf. (2.12)). For example, for a $M_T = M_R = 8$ channel and 4-QAM modulation ($P = 2$) we obtained $|\tilde{\mathcal{D}}| = 137$ out of a total of $|\mathcal{D}| = 4^8 = 65536$ possible transmitted data vectors. For the SPA, we now have $|\mathcal{D}_{\text{SP}}| \leq 33$. This illustrates the significant reduction of complexity achieved by exploiting the constant modulus property of the symbol alphabet. Note however that – in contrast to the LSD – the SPA is not guaranteed to be optimum for the IBC.

2.5.2 Algorithm Summary and Complexity

Having explained the efficient construction of the additional search set \mathcal{D}_+ , we now summarize the use of the SPA as an add-on to a suboptimal detector and discuss its computational complexity. The principal steps of the SPA are as follows.

- Calculate $\hat{\mathbf{d}}$, the result of any given suboptimal detector, and the associated distance $\|\mathbf{r} - \mathbf{H}\hat{\mathbf{d}}\|^2$.
- Calculate $\mathbf{D}_{\text{ZF}} = (\mathbf{H}^H \mathbf{H})^{-1}$ and $\mathbf{y}_{\text{ZF}} = \mathbf{D}_{\text{ZF}} \mathbf{H}^H \mathbf{r}$.
- Calculate the dominant eigenvector \mathbf{v}_{M_T} of \mathbf{D}_{ZF} using several iterations of the power method according to (2.19).
- Calculate all boundary lines $\mathcal{B}^{(m,p)}$ in \mathcal{L} .
- Determine the additional search set \mathcal{D}_+ and calculate the corresponding distances
 - by considering all data vectors whose cells are pierced by the intersection circle \mathcal{I}
 - or, if \mathcal{I} does not exist or if no boundary line intersects \mathcal{I} , by considering $\mathbf{d}^{(1)} = \mathcal{Q}_{\mathcal{A}}\{\mathbf{y}_{\mathbf{v}_{M_T}}^\perp\}$ and its nearest neighbors.
- Find the minimal distance (including the distance obtained for $\hat{\mathbf{d}}$).

Here, $\hat{\mathbf{d}}$ may be the result of ZF detection, MMSE detection, ZF-based NC, or MMSE-based NC (see Section 1.8); the resulting SPA variants will be referred to as SPA-ZF, SPA-MMSE, SPA-NC-ZF, and SPA-NC-MMSE detection, respectively.

The computational complexity of the SPA add-on (not counting the calculation of $\hat{\mathbf{d}}$) can be assessed as follows. The preparation complexity C_{prep} has the dominant complexity of $\mathcal{O}(M_T^3)$ due

to the calculation of $\mathbf{G}_{\text{ZF}} = \mathbf{D}_{\text{ZF}}\mathbf{H}^H$. The vector complexity C_{vector} is essentially composed of \mathbf{y}_{ZF} calculation with complexity $\mathcal{O}(M_{\text{T}}^2)$, two distance calculations with complexity $\mathcal{O}(M_{\text{T}}^2)$ and, in the worst case, of $2M_{\text{T}}P - 1$ efficient distance updates with total complexity $\mathcal{O}(M_{\text{T}}^2P)$. Hence, the vector complexity of $\mathcal{O}(M_{\text{T}}^3P^2)$ for the LSD is reduced to $\mathcal{O}(M_{\text{T}}^2P)$ for the SPA, which is due to the smaller reduced search set of the SPA. An experimental assessment of SPA's complexity will be provided in Subsection 2.6.2.

2.5.3 SPA Variants

To use the SPA as an add-on to ZF detection, we choose $\hat{\mathbf{d}} = \mathcal{Q}_{\mathcal{A}}\{\mathbf{y}_{\text{ZF}}\}$. The resulting SPA-ZF algorithm is the most efficient SPA variant because calculation of \mathbf{y}_{ZF} is already part of the SPA.

To use the SPA as an add-on to MMSE detection, we choose $\hat{\mathbf{d}} = \hat{\mathbf{d}}_{\text{MMSE}} = \mathcal{Q}_{\mathcal{A}}\{\mathbf{y}_{\text{MMSE}}\}$ where $\mathbf{y}_{\text{MMSE}} = (\mathbf{H}^H\mathbf{H} + \sigma_{\mathbf{w}}^2\mathbf{I})^{-1}\mathbf{H}^H\mathbf{r}$ is the result of MMSE equalization. Furthermore, we replace the reference line \mathcal{L} in (2.23) by the ‘‘MMSE-based reference line’’

$$\mathcal{L}' : \mathbf{y}(\tilde{\alpha}) = \tilde{\alpha}\mathbf{v}_{M_{\text{T}}} + \mathbf{P}_{\mathbf{v}_{M_{\text{T}}}}^{\perp}\mathbf{y}_{\text{MMSE}}, \quad \text{with } \tilde{\alpha} = \alpha + \mathbf{v}_{M_{\text{T}}}^H\mathbf{y}_{\text{MMSE}} \in \mathbb{C}.$$

This reference line \mathcal{L}' is again parallel to the dominant principal axis $\mathbf{v}_{M_{\text{T}}}$ but its offset is the component of \mathbf{y}_{MMSE} (rather than of \mathbf{y}_{ZF}) perpendicular to $\mathbf{v}_{M_{\text{T}}}$. The resulting SPA-MMSE algorithm can be obtained simply by replacing in the SPA-ZF algorithm $\mathbf{D}_{\text{ZF}} = (\mathbf{H}^H\mathbf{H})^{-1}$ with $\mathbf{D}_{\text{MMSE}} \triangleq (\mathbf{H}^H\mathbf{H} + \sigma_{\mathbf{w}}^2\mathbf{I})^{-1}$. The rationale for the SPA-MMSE algorithm is that on average \mathbf{y}_{MMSE} will be closer to the transmitted data vector \mathbf{d} than \mathbf{y}_{ZF} . Note that applying the power method to \mathbf{D}_{MMSE} instead of \mathbf{D}_{ZF} requires slightly more iterations in order to obtain accurate results for $\mathbf{v}_{M_{\text{T}}}$. This is because the ‘‘regularization term’’ $\sigma_{\mathbf{w}}^2\mathbf{I}$ decreases the ratio of the largest to the second largest singular value of $(\mathbf{H}^H\mathbf{H} + \sigma_{\mathbf{w}}^2\mathbf{I})^{-1}$ as compared to $(\mathbf{H}^H\mathbf{H})^{-1}$, which slows down the convergence of the power method [59].

Finally, to combine the SPA with NC detection (either ZF-based or MMSE-based), we propose to execute the SPA-ZF or SPA-MMSE algorithm, respectively, except for the use of the NC detection result for $\hat{\mathbf{d}}$. As we will demonstrate in the next section, the SPA-NC-MMSE algorithm is able to yield near-ML performance with significantly less computational cost than that required by the FPSD algorithm for ML detection.

2.6 Simulation Results

We now present simulation results in order to assess the error-rate performance and computational complexity of the proposed LSD and SPA variants in comparison to the conventional ZF, MMSE, NC, and ML schemes. In our simulations, we considered SM systems with iid Gaussian channels (see also Section 1.7) employing 4-QAM symbols. In Section 3.7.5 we will also show some performance results using 8-PSK symbols.

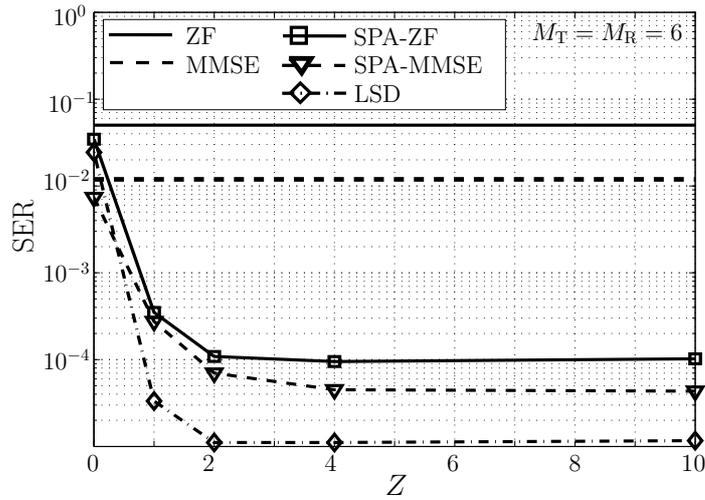


Figure 2.10: SER versus the number of power method iterations Z , for a $M_T = M_R = 6$ channel and 4-QAM modulation at an SNR of 20 dB. $Z = 0$ corresponds to a randomly chosen axis \mathbf{v}_{M_T} .

2.6.1 SER Performance

We now investigate the uncoded symbol error rate (SER) performance of the various detection schemes.

SER versus Number of Power Method Iterations

First, we study the dependence of the SER on the number of power method iterations Z that is used to approximate \mathbf{v}_{M_T} (cf. (2.19)).

Fig. 2.10 shows the SER versus Z performance for LSD, SPA-ZF, and SPA-MMSE for a $M_T = M_R = 6$ SM system at an SNR of 20 dB. For comparison, the SER achieved with conventional ZF and MMSE detection is indicated by the horizontal lines. It is seen that for $Z \geq 1$, the proposed detectors yield significant performance improvements. For $Z = 0$, we used a randomly chosen \mathbf{v}_{M_T} ; as can be expected, in this case just negligible performance improvements over ZF or MMSE detection are achieved. We also can observe that due to the regularization of the MMSE equalizer, SPA-MMSE requires more iterations of the power method than the SPA-ZF and the LSD. Increasing Z beyond 4 does not yield any additional performance improvement. Thus, we chose $Z = 4$ in all simulations presented below.

SER versus SNR Performance

Figure 2.11 and Figure 2.12 show the SER versus SNR performance of the various proposed and standard detectors for a $M_T = M_R = 4$ and a $M_T = M_R = 6$ SM system, respectively. For the NC algorithms, we used layer sorting maximizing the post-equalization SNRs at each detection step (see Section 1.8.3 and, in particular, (1.16)). The following conclusions can be drawn from these results.

- The LSD performs substantially better than the standard suboptimal detectors.

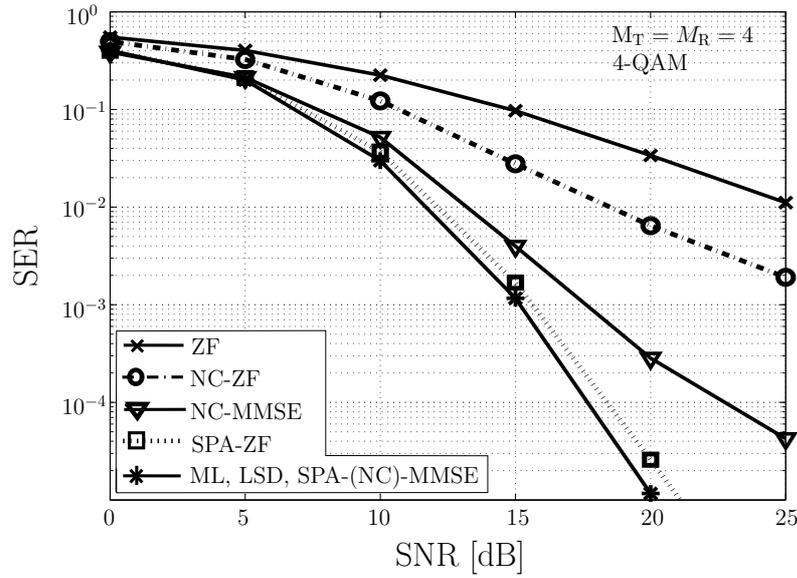


Figure 2.11: *SER versus SNR performance of the various proposed and standard detectors for a $M_T = M_R = 4$ channel using 4-QAM modulation.*

- The SPA add-on significantly improves the performance of the standard suboptimal detectors.
- The SPA schemes using ZF or MMSE detection outperform their respective NC counterparts.
- The MMSE-based schemes (lin., NC, or SPA) outperform their respective ZF-based counterparts. Note that the MMSE-based schemes are different from the ZF-based schemes in that they require knowledge about the noise variance.
- For the $M_T = M_R = 4$ channel (see Figure 2.11), the SPA-MMSE, SPA-NC-MMSE, and the LSD perform practically as well as the ML detector (the corresponding SER curves in Figure 2.11 are indistinguishable), and the SPA-ZF detector achieves a performance that is very close to ML performance. The proposed detectors perform better than the conventional schemes.
- For the $M_T = M_R = 6$ channel (see Figure 2.12), the SPA-NC-MMSE detector achieves a performance that is very close to ML performance. The LSD outperforms the NC-MMSE detector whose performance is intermediate between that of the SPA-MMSE detector (which performs slightly better) and that of the SPA-ZF detector (which performs slightly worse).
- The performance of the SPA-ZF and SPA-MMSE detectors is close to that of the LSD. This shows that our strategy for constructing \mathcal{D}_+ works well.
- A comparison of the results obtained for the $M_T = M_R = 4$ and $M_T = M_R = 6$ channels suggests that for increasing channel size, the performance of the proposed algorithms degrades (compared to ML performance). This is due to the IBC approximation underlying our algorithms. Specifically, for increasing channel size the probability that two or more principal axes are dominant

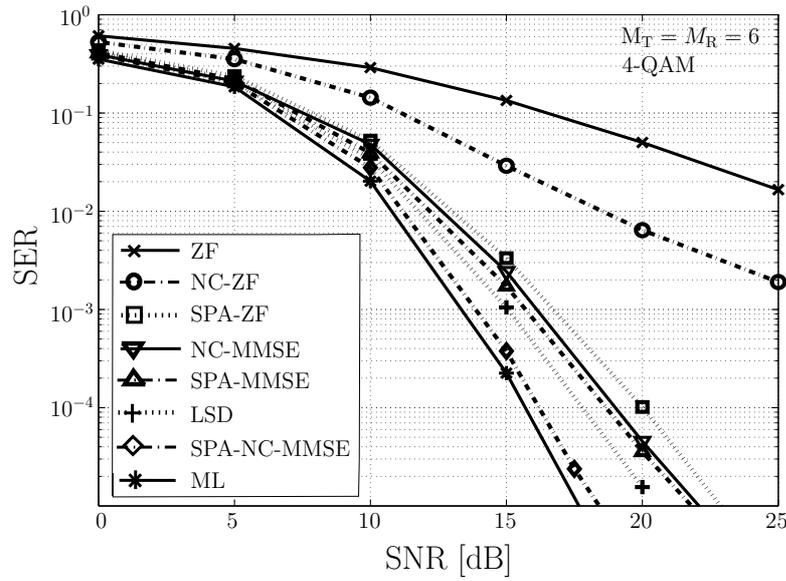


Figure 2.12: SER versus SNR performance of the various proposed and standard detectors for a $M_T = M_R = 6$ channel using 4-QAM modulation.

(rather than just one as assumed for IBCs) increases, so that the IBC approximation becomes less accurate.

2.6.2 Computational Complexity

The complexity order $\mathcal{O}(\cdot)$ results of the proposed LSD and the SPA were presented in Section 2.4.3 and Section 2.5.2, respectively. Essentially, the overall complexity of both detectors scales with M_T^3 . Thus, due to the exponential scaling in M_T of the ML detector, for large enough M_T the LSD and the SPA are less computational intensive than the ML detector. To provide more insight about the computational complexity of the various algorithms than the $\mathcal{O}(\cdot)$ results, we performed kflops measurements of specific MATLAB implementations (cf. Section 1.7.3). We considered three different SM systems ($M_T = M_R = 4, 6, 8$) with 4-QAM modulation. The corresponding kflops estimates are displayed in Table 2.1. ML detection was performed efficiently by means of the FPSD algorithm [8, 49] that is based on the equivalent real-valued system model (1.3). Its complexity was measured at an SNR of 8 dB. The complexity of the other schemes is independent of the SNR. NC was implemented using the efficient recursive matrix inversion algorithm of [70] (see also the more detailed discussion in Section 3.4.2 of Chapter 3).

The computational complexity is split into preparation complexity C_{prep} (Table 2.1(a)) and vector complexity C_{vector} (Table 2.1(b)). Each table presents only one value for both linear detectors (ZF and MMSE; denoted as “lin.”) and only one value for both linear detectors combined with the SPA (SPA-ZF and SPA-MMSE; denoted as “SPA-lin.”), because there is virtually no difference in complexity. The preparation complexity of the FPSD is due to the QR-decomposition [59] of the equivalent real-

(a)

measured kflops – preparation complexity C_{prep}						
channel	FPSD	LSD	lin.	NC	SPA-lin.	SPA-NC
(4, 4)	1.9	2.3	1.7	1.9	2.3	2.6
(6, 6)	6.1	6.2	4.7	5.3	6.2	6.8
(8, 8)	14.0	12.7	10.2	11.4	12.7	14.0

(b)

measured kflops – vector complexity C_{vector}									
channel	FPSD		LSD	lin.	NC	SPA-lin.		SPA-NC	
	av.	max.				av.	max.	av.	max.
(4, 4)	2.6	18.2	12.9	0.3	0.6	1.5	2.4	1.9	2.7
(6, 6)	11.0	79.2	40.3	0.7	1.5	2.8	4.8	3.9	5.9
(8, 8)	44.1	364	91.8	1.2	3.2	4.5	7.9	7.0	10.4

Table 2.1: Measured complexity in kflops for (a) preparation complexity C_{prep} and (b) vector complexity C_{vector} .

valued channel matrix.

The complexity of FPSD is random and strongly depends on the channel realization \mathbf{H} ; in particular, for a bad channel realization, the maximum complexity exceeds the average complexity by a large amount. Thus, in addition to the average FPSD complexity, Table 2.1(b) shows the maximum FPSD complexity obtained during 1000 simulation runs. A maximum complexity is also provided for SPA-lin. and SPA-NC; it refers to the case where all boundary lines intersect the intersection circle \mathcal{I} (cf. Case 1 in Section 2.5.1).

From Table 2.1, the following conclusions can be drawn.

- The complexity of the SPA detectors is much lower than that of the FPSD and the LSD. In particular, for the $M_T = M_R = 6$ SM system, SPA-NC-MMSE requires significantly less computational effort than FPSD and LSD, even though SPA-NC-MMSE achieves near-ML performance.
- The overall complexity of the SPA detectors is higher than that of the standard suboptimal detectors. Of course, the computational cost is increased due to the SPA add-on.
- On average, the complexity of FPSD is very low. As a comparison, the overall complexity of exhaustive-search ML detection is 43 kflops and 1425 kflops for the $M_T = M_R = 4$ and $M_T = M_R = 6$ SM system, respectively. However, the maximum complexity of FPSD is much larger than its average complexity. For practical system designs the maximum vector complexity can be decisive.
- In contrast to FPSD, the LSD has a fixed complexity independent of the channel realization. According to Table 2.1(b), the vector complexity of the LSD is larger than the average complexity

of FPSD and smaller than the maximum complexity of FPSD. Thus, on average, the LSD is more computationally intensive than the FPSD implementation of ML detection; however, LSD's worst case complexity is smaller. The reason for this is as follows. In contrast to the LSD, the FPSD trades complexity with the condition number of the channel realization. If the channel is good, the complexity of the FPSD is very low. If the channel is bad, the computational complexity of FPSD is very high. Indeed, the maximum C_{vector} result of the FPSD in Table 2.1(b) corresponds to a bad channel realization. Contrary, the LSD performs ML detection for a bad channel (i.e. the IBC) irrespective of the actual channel realization. Even if the channel realization is good (and simple ZF detection is sufficient to perform ML detection), the LSD performs ML detection for the IBC. This results in a higher average complexity of the LSD compared to the FPSD. However, for bad channel realization the LSD is more efficient than the FPSD, which is indicated by the smaller complexity of the LSD compared to the maximum complexity of the FPSD.

- All detectors have a similar C_{prep} . For LSD and the SPA detectors, C_{prep} is slightly higher than that for the other suboptimum detectors due to the additional calculation of \mathbf{v}_{M_T} .
- For the SPA-lin. detectors, the average C_{vector} is about twice that of the NC detectors but significantly lower than that of the FPSD.
- The C_{vector} complexity of the LSD and the SPA variants scales much slower with increasing number of antennas than the C_{vector} complexity of the FPSD. For twice the system size (going from the $M_T = M_R = 4$ system to the $M_T = M_R = 8$ system), the LSD and the SPA variants require about 7.2 and 3.5 times more computations, respectively, while the FPSD requires about 17 times more computations on average.

2.6.3 Discussion

The error-rate and computational complexity results demonstrate that the proposed algorithms can achieve near-ML performance with a much lower computational complexity than that required by the FPSD. In particular, the FPSD suffers from a random complexity that can be very large in the worst-case. The LSD, on the other hand, has the advantage that it has a predetermined complexity that is smaller than the worst-case complexity of the FPSD. However, the LSD suffers from suboptimum performance (in particular for large MIMO systems) and a quite large average complexity (in particular for small MIMO systems) as compared to the average complexity of the FPSD. As discussed above, this is due to the fact that the FPSD is able to trade complexity with the condition number of the channel realization while the LSD always performs data detection for a bad channel irrespective of the actual channel realization. However, this disadvantage of the LSD could be circumvented by the following strategy. If the channel condition number lies above a threshold, we perform the LSD (here we can expect near-ML performance); if the channel condition number lies below a threshold, we perform conventional suboptimum (e.g., ZF) detection. The resulting algorithm has a random complexity with a worst-case complexity given by the predetermined complexity of the LSD, but has a much

smaller (depending on the choice of the threshold) average complexity. Another interesting extension of the LSD principle can be found in [18, 19]. Here, the LSD is directly incorporated into the FPSD algorithm, which allows to achieve the complexity advantages of both algorithms simultaneously.

The SPA is clearly more attractive than both the FPSD and the LSD. It allows to get close to ML performance with significantly less average and worst-case complexity. The SPA-lin. detectors also outperform their NC counterparts using layer-sorting. However, this is achieved with a larger computational cost (about a factor of two to three). For the $M_T = M_R = 6$ MIMO system, for example, the larger complexity of SPA-MMSE compared to NC-MMSE may not pay off, since the performance gains are moderate. However, for smaller systems (e.g., $M_T = M_R = 4$) or for the ZF-based schemes, SPA-lin. yields huge performance gains over NC. Finally, combining the SPA with NC is very attractive, since near-ML performance can be achieved efficiently also for the $M_T = M_R = 6$ SM system.

2.7 Summary and Conclusions

The starting point of this chapter was an analysis of the effect of bad (poorly conditioned) channels on suboptimal detectors for SM systems. The performance of standard suboptimal detection schemes severely degrades compared to the performance of the ML detector when bad channel realizations occur. We found that this inferior performance is due to the inability of linear detectors to properly adapt their decision regions to the noise statistics in the ZF domain. In addition, bad channels lead to a high computational complexity of the sphere-decoding algorithm for ML detection.

Based on an idealized bad channel (IBC) model we then presented new detection methods that are robust to bad channels. The IBC captures bad channel effects in a simplified form in that it just models a *single* small singular value of the channel realization. The initial form of the new detection approach was an efficient ML detector for the IBC model. Subsequently, we extended this detector to be suitable for arbitrary (nonidealized) channels. This detector we referred to as the line search detector (LSD). The LSD is optimum for good (i.e. orthogonal) as well as idealized bad channels. Although its complexity scales just cubically with the number of antennas, we found that it still can be computationally intensive. In particular, for SM systems of moderate size, the complexity of the LSD was demonstrated to be smaller than the worst-case complexity (over many channel realizations) required by the sphere decoding algorithm, but larger than its average complexity. A significant reduction of the computational complexity was achieved with the *sphere-projection algorithm* (SPA) that is a computationally efficient, nonlinear add-on to standard suboptimal detectors. Here, a constant modulus symbol alphabet is required. The SPA add-on significantly improves the error-rate and diversity-gain performance of a suboptimal detector by making it robust to bad channel realizations.

Simulations showed that the LSD and the SPA outperform ZF-based and MMSE-based nulling-and-cancelling and that they can achieve near-ML performance. Their performance is best for SM systems of moderate size, while for increasing system size their performance degrades compared to the performance of ML detection. For example, for an SM system with 6 transmit antennas and 6

receive antennas, the SPA is able to yield near-ML performance with significantly less computational complexity than that required by the sphere-decoding algorithm for ML detection.

Dynamic Nulling-and-Cancelling

IN the previous chapter about geometry-based detectors for SM systems we observed that NC with layer sorting using the layerwise post-equalization SNRs (PSNRs) as a reliability criterion [63, 64] (see also Section 1.8.3) is an attractive suboptimum data detection technique. NC is computationally very efficient with a performance that is much better than that of the equalization-based (linear) techniques. However, its performance is far inferior to that of ML detection and it also suffers from a performance loss compared to geometry-based detection. It is interesting to observe that the PSNRs are just *average* quantities that do not depend on the received vector—more specifically, they depend on the channel realization and on the noise variance, but neither on the transmitted data vector nor on the noise realization.

In this chapter, we propose an improved NC technique that we term *dynamic nulling-and-cancelling* (DNC) [23–25]. The basic idea of the DNC scheme is to use the layerwise a-posteriori probabilities (APPs) as reliability measures to perform the layer sorting. To keep the complexity low, we use *approximate* APPs that are constructed by means of a Gaussian approximation for the residual post-equalization interference. This approach is inspired and motivated by [74], where an iteratively updated Gaussian approximation for the post-equalization interference was used in the context of multiuser detection. As we will show, the Gaussian approximation results in an MMSE-based nulling technique with a significantly improved yet very simple layer-sorting rule that is “dynamic” in that it depends on the current received vector, in contrast to the “static” (average) PSNR-based layer sorting employed by conventional NC.

DNC is in general more complex than conventional NC because the layers are sorted anew for each received vector and not just for an entire data block during which the system matrix is constant. To keep the extra complexity small, we use an efficient recursive technique for matrix inversion that was proposed for conventional NC in [70]. We will demonstrate that the resulting efficient DNC implementation can yield near-ML performance for a wide range of system sizes and SNRs with a

computational cost that is much smaller than that required by the sphere-decoding algorithm for ML detection.

Furthermore, we investigate and compare the performance of NC and DNC both analytically and experimentally. We study the conditions under which the performance gains of DNC over NC can be expected to be substantial or rather small. We show that the performance gains will be strongest when conventional PSNR-based sorting fails to exploit all degrees of freedom that are available for layer sorting. On the other hand, we also show that dynamic layer sorting will almost reduce to conventional PSNR sorting when PSNR sorting has a strong preference for a specific layer.

A layer-sorting rule that also depends on the current received vector was recently proposed in [75]. Here, log-likelihood ratios (LLRs) are considered for layer sorting of ZF-based NC. The resulting LLR-based NC scheme can be seen as a ZF analogue of our (MMSE-based) DNC algorithm. In addition to that, we will demonstrate that LLR-based NC, too, can be derived with a Gaussian approximation for the post-equalization interference; however, in contrast to our DNC scheme, the interference is assumed to be uncorrelated. Thus, DNC can also be seen as an extension of LLR-based NC to correlated post-equalization interference, which explains its significant performance advantage.

This chapter is organized as follows. In Section 3.1, we formulate the basic principle of the novel DNC technique. The two stages of DNC—data detection and layer sorting—are then developed in Section 3.2 and Section 3.3, respectively. In Section 3.4, a statement of the overall DNC algorithm is provided, an efficient method for recursive matrix inversion is discussed, and the algorithm’s computational complexity is analyzed. The error performance of DNC is studied for a simple special case in Section 3.5. In Section 3.6 the relation between DNC and LLR-based NC is established and discussed. Finally, in Section 3.7 we assess the SER performance and computational complexity of DNC through numerical simulations carried out for SM systems and for MIMO systems using LD codes. Within Section 3.7, DNC is also compared to LLR-based NC and to the geometry-based data detectors that were proposed in the previous chapter.

3.1 Fundamentals of Dynamic Nulling-and-Cancelling

We now describe the basic principle of the novel DNC technique, namely, the construction of the approximate APP and its use for symbol detection and “dynamic” layer sorting. We will consider the linear MIMO model described in Section 1.5. In contrast to the geometry-based detectors (see Chapter 2), the proposed DNC algorithm is not specifically tailored to SM systems. Thus, the matrix \mathbf{H} of model (1.1) denotes the system matrix (and not necessarily the channel matrix as for SM systems) and has dimension $M \times N$.

We will look at the *first* decoding step, where we detect a layer m_1 in favor of a symbol \hat{d}_{m_1} with high reliability. Layer m_1 is then used to perform the interference cancellation as in conventional NC (see Section 1.8.3). The subsequent detection and interference-cancellation steps are analogous, however with a reduced number of active layers. A complete statement of the DNC algorithm will be provided in Section 3.4.

3.1.1 MAP Approach to Detection and Layer Sorting

For the m th layer, $m \in \{1, \dots, M\}$, the optimum decision on the data symbol $d_m \in \mathcal{A}$ is given by the maximum a-posteriori (MAP) rule that maximizes the APP¹ $P[d_m = a | \mathbf{y}_{\text{ZF}}]$ [76]:

$$\hat{d}_m \triangleq \arg \max_{a \in \mathcal{A}} P[d_m = a | \mathbf{y}_{\text{ZF}}]. \quad (3.1)$$

The resulting maximum APP $P[d_m = \hat{d}_m | \mathbf{y}_{\text{ZF}}]$ characterizes the reliability of the symbol decision \hat{d}_m . Recall that for NC more reliable layers should be detected before less reliable ones (cf. Section 1.8.3). Therefore, the layers are commonly sorted using the ZF or MMSE PSNRs (see (1.17) and (1.18), respectively) as measures of layer reliability [13, 63, 64]. Our approach to layer sorting now is to first calculate the optimum symbol \hat{d}_m for each layer m and then choose the layer m for interference cancellation for which the reliability of this optimum symbol decision (APP for $d_m = \hat{d}_m$) is maximum, i.e.

$$m_1 \triangleq \arg \max_{m \in \{1, \dots, M\}} P[d_m = \hat{d}_m | \mathbf{y}_{\text{ZF}}]. \quad (3.2)$$

For conventional NC, (1.16) is used for layer sorting. Decoding layer m_1 in favor of \hat{d}_{m_1} has maximum reliability at this stage, and we subsequently use this result for interference cancellation.

The complexity of calculating the APP $P[d_m = a | \mathbf{y}_{\text{ZF}}]$ required in (3.1) is exponential in M . A significant reduction of complexity can be obtained by an approximation. We first observe that by Bayes' rule, the APP can be rewritten in terms of the conditional pdf $f(\mathbf{y}_{\text{ZF}} | d_m = a)$ as

$$P[d_m = a | \mathbf{y}_{\text{ZF}}] = \frac{f(\mathbf{y}_{\text{ZF}} | d_m = a)}{\sum_{a' \in \mathcal{A}} f(\mathbf{y}_{\text{ZF}} | d_m = a')}, \quad (3.3)$$

where it has been assumed that all data symbols are transmitted equally likely, i.e., $P[d_m = a] = 1/|\mathcal{A}|$ for all $a \in \mathcal{A}$. We have

$$f(\mathbf{y}_{\text{ZF}} | d_m = a) \propto \sum_{\mathbf{d} \in \mathcal{D}(a, m)} f(\mathbf{y}_{\text{ZF}} | \mathbf{d}), \quad (3.4)$$

where $\mathcal{D}(a, m)$ denotes the set of all $\mathbf{d} \in \mathcal{D}$ for which $d_m = a$. Because according to $\mathbf{y}_{\text{ZF}} = \mathbf{d} + \tilde{\mathbf{w}}$ (see (1.8)) $f(\mathbf{y}_{\text{ZF}} | \mathbf{d})$ is a Gaussian pdf, it follows from (3.4) that $f(\mathbf{y}_{\text{ZF}} | d_m = a)$ is a Gaussian mixture pdf.

Under the condition that $d_m = a$, we can reformulate $\mathbf{y}_{\text{ZF}} = \mathbf{d} + \tilde{\mathbf{w}}$ as

$$\mathbf{y}_{\text{ZF}} = a \mathbf{e}_m + \sum_{\substack{m'=1 \\ m' \neq m}}^M d_{m'} \mathbf{e}_{m'} + \tilde{\mathbf{w}}, \quad \text{for } d_m = a, \quad (3.5)$$

where \mathbf{e}_m denotes the m th M -dimensional unit vector. This shows that for $d_m = a$, \mathbf{y}_{ZF} is equal (up to a shift by $a\mathbf{e}_m$) to the *post-equalization interference*² for the m th layer, $\sum_{m' \neq m} d_{m'} \mathbf{e}_{m'} + \tilde{\mathbf{w}}$.

¹The APP can equivalently be conditioned on the result of ZF equalization $\mathbf{y}_{\text{ZF}} = \mathbf{H}^\# \mathbf{r}$ rather than on the received vector \mathbf{r} since ZF equalization without quantization does not imply any loss of information (cf. Section 1.8.2).

²Note that the m th layer $y_{\text{ZF}, m}$ of the ZF equalized received vector is commonly referred as interference free since the corresponding data symbol d_m is just corrupted by additive noise, i.e. $y_{\text{ZF}, m} = d_m + \tilde{w}_m$, and not by any other data symbols $d_{m'}$, $m' \neq m$. However, here we adopt the viewpoint that $y_{\text{ZF}, m}$ interferes with $y_{\text{ZF}, m'}$, $m' \neq m$, since $\tilde{\mathbf{w}}$ is correlated. We consider $y_{\text{ZF}, m'} = d_{m'} + \tilde{w}_{m'}$, $m' \neq m$, and \tilde{w}_m as the post-equalization interference for the m th layer.

3.1.2 Gaussian Approximation

We now use a Gaussian approximation for the post-equalization interference to obtain a computationally efficient approximation to (3.3). More specifically, we approximate the Gaussian mixture pdf $f(\mathbf{y}_{\text{ZF}}|d_m=a)$ by the Gaussian pdf³ [72]

$$\tilde{f}_m(\mathbf{y}_{\text{ZF}}|d_m=a) \triangleq \frac{1}{\pi^M \det(\mathbf{C}_m)} e^{-(\mathbf{y}_{\text{ZF}}-\boldsymbol{\mu}_m)^H \mathbf{C}_m^{-1} (\mathbf{y}_{\text{ZF}}-\boldsymbol{\mu}_m)}. \quad (3.6)$$

Here, the mean $\boldsymbol{\mu}_m$ and the covariance \mathbf{C}_m are chosen consistent with the true pdf $f(\mathbf{y}_{\text{ZF}}|d_m=a)$, i.e., $\boldsymbol{\mu}_m \triangleq \mathbb{E}[\mathbf{y}_{\text{ZF}}|d_m=a]$ and $\mathbf{C}_m \triangleq \mathbb{E}[(\mathbf{y}_{\text{ZF}}-\boldsymbol{\mu}_m)(\mathbf{y}_{\text{ZF}}-\boldsymbol{\mu}_m)^H|d_m=a]$. From (3.5), we obtain

$$\boldsymbol{\mu}_m = a \mathbf{e}_m, \quad \mathbf{C}_m = \mathbf{I} - \mathbf{e}_m \mathbf{e}_m^T + \mathbf{R}_{\tilde{\mathbf{w}}}. \quad (3.7)$$

The Gaussian pdf $\tilde{f}_m(\mathbf{y}_{\text{ZF}}|d_m=a)$ is now completely determined, and the APP in (3.3) is approximated by

$$\mathbb{P}[d_m=a|\mathbf{y}_{\text{ZF}}] \approx \frac{\tilde{f}_m(\mathbf{y}_{\text{ZF}}|d_m=a)}{\sum_{a' \in \mathcal{A}} \tilde{f}_m(\mathbf{y}_{\text{ZF}}|d_m=a')}.$$

Using this Gaussian approximation, the maximization in (3.1) that yielded the optimum symbol for the m th layer is replaced with

$$\hat{d}_m \triangleq \arg \max_{a \in \mathcal{A}} \left\{ \frac{\tilde{f}_m(\mathbf{y}_{\text{ZF}}|d_m=a)}{\sum_{a' \in \mathcal{A}} \tilde{f}_m(\mathbf{y}_{\text{ZF}}|d_m=a')} \right\} = \arg \max_{a \in \mathcal{A}} \tilde{f}_m(\mathbf{y}_{\text{ZF}}|d_m=a). \quad (3.8)$$

Furthermore, the maximization in (3.2) that yielded the optimum layer is replaced with⁴

$$m_1 \triangleq \arg \max_{m \in \{1, \dots, M\}} \left\{ \frac{\tilde{f}_m(\mathbf{y}_{\text{ZF}}|d_m=\hat{d}_m)}{\sum_{a \in \mathcal{A}} \tilde{f}_m(\mathbf{y}_{\text{ZF}}|d_m=a)} \right\}. \quad (3.9)$$

In what follows, these new definitions for \hat{d}_m and m_1 will be used rather than the original definitions (3.1) and (3.2).

The equations (3.8) and (3.9) define, respectively, the *data-detection stage* and *layer-sorting stage* of DNC. In the next two sections, we will develop the calculations corresponding to these two stages.

3.2 Data-Detection Stage

Using the Gaussian approximation (3.6) together with (3.7), we can write the approximate MAP decision (3.8) as

$$\hat{d}_m = \arg \max_{a \in \mathcal{A}} \left\{ \frac{1}{\pi^M \det(\mathbf{C}_m)} e^{-(\mathbf{y}_{\text{ZF}}-a\mathbf{e}_m)^H \mathbf{C}_m^{-1} (\mathbf{y}_{\text{ZF}}-a\mathbf{e}_m)} \right\}$$

³We assume that $\mathbf{C}_m = \mathbf{I} - \mathbf{e}_m \mathbf{e}_m^T + \mathbf{R}_{\tilde{\mathbf{w}}}$ is nonsingular (this is guaranteed if $\sigma_{\tilde{\mathbf{w}}}^2 \neq 0$ and \mathbf{H} has full rank).

⁴Whereas the denominator in (3.3), $\sum_{a \in \mathcal{A}} f(\mathbf{y}_{\text{ZF}}|d_m=a) = |\mathcal{A}| f(\mathbf{y}_{\text{ZF}})$, does not depend on the layer index m , the denominator in (3.9), $\sum_{a \in \mathcal{A}} \tilde{f}_m(\mathbf{y}_{\text{ZF}}|d_m=a) = |\mathcal{A}| \tilde{f}_m(\mathbf{y}_{\text{ZF}})$, depends on m .

$$\begin{aligned}
&= \arg \max_{a \in \mathcal{A}} \left\{ -(\mathbf{y}_{\text{ZF}} - a\mathbf{e}_m)^H \mathbf{C}_m^{-1} (\mathbf{y}_{\text{ZF}} - a\mathbf{e}_m) \right\} \\
&= \arg \max_{a \in \mathcal{A}} \left\{ 2 \operatorname{Re} \{ \mathbf{y}_{\text{ZF}}^H \mathbf{C}_m^{-1} \mathbf{e}_m a \} - |a|^2 \mathbf{e}_m^T \mathbf{C}_m^{-1} \mathbf{e}_m \right\}.
\end{aligned} \tag{3.10}$$

The matrix inversion lemma [59] applied to $\mathbf{C}_m^{-1} = (\mathbf{I} - \mathbf{e}_m \mathbf{e}_m^T + \mathbf{R}_{\tilde{\mathbf{w}}})^{-1}$ yields

$$\mathbf{C}_m^{-1} = \mathbf{W} \left(\mathbf{I} + \frac{\mathbf{e}_m \mathbf{e}_m^T \mathbf{W}}{1 - W_{m,m}} \right), \tag{3.11}$$

with

$$\mathbf{W} \triangleq (\mathbf{I} + \mathbf{R}_{\tilde{\mathbf{w}}})^{-1} = (\mathbf{I} + \sigma_{\tilde{\mathbf{w}}}^2 (\mathbf{H}^H \mathbf{H})^{-1})^{-1} \tag{3.12}$$

and $W_{m,m} \in \mathbb{R}$ denoting the m th diagonal element of \mathbf{W} . The $M \times M$ matrix \mathbf{W} is termed *Wiener estimator* in [77]; it converts ZF equalization (1.8) into MMSE equalization (1.11) [77]:

$$\mathbf{W} \mathbf{y}_{\text{ZF}} = \mathbf{y}_{\text{MMSE}}.$$

Using this result and (3.11), we obtain

$$\begin{aligned}
\mathbf{y}_{\text{ZF}}^H \mathbf{C}_m^{-1} \mathbf{e}_m &= \mathbf{y}_{\text{ZF}}^H \mathbf{W} \mathbf{e}_m \left(1 + \frac{\mathbf{e}_m^T \mathbf{W} \mathbf{e}_m}{1 - W_{m,m}} \right) \\
&= y_{\text{MMSE},m}^* \left(1 + \frac{W_{m,m}}{1 - W_{m,m}} \right) \\
&= \frac{y_{\text{MMSE},m}^*}{1 - W_{m,m}}
\end{aligned} \tag{3.13}$$

and

$$\begin{aligned}
\mathbf{e}_m^T \mathbf{C}_m^{-1} \mathbf{e}_m &= \mathbf{e}_m^T \mathbf{W} \mathbf{e}_m \left(1 + \frac{\mathbf{e}_m^T \mathbf{W} \mathbf{e}_m}{1 - W_{m,m}} \right) \\
&= W_{m,m} \left(1 + \frac{W_{m,m}}{1 - W_{m,m}} \right) \\
&= \frac{W_{m,m}}{1 - W_{m,m}}.
\end{aligned} \tag{3.14}$$

Thus, the maximization in (3.10) simplifies to

$$\hat{d}_m = \arg \max_{a \in \mathcal{A}} \left\{ \frac{1}{1 - W_{m,m}} \left(2 \operatorname{Re} \{ y_{\text{MMSE},m}^* a \} - W_{m,m} |a|^2 \right) \right\}. \tag{3.15}$$

The eigenvalues $\lambda_{W,m}$ of \mathbf{W} satisfy $0 \leq \lambda_{W,m} < 1$. Because $W_{m,m}$ can be written as a quadratic form induced by \mathbf{W} and this quadratic form is bounded by the minimum and maximum eigenvalues of \mathbf{W} , we obtain the following two (equivalent) inequalities:

$$0 \leq W_{m,m} < 1, \quad 1 \leq \frac{1}{1 - W_{m,m}} < \infty.$$

Assuming that $W_{m,m} \neq 0$ for $m = 1, \dots, M$ (which holds if \mathbf{H} has full rank), we can rewrite (3.15) as

$$\hat{d}_m = \arg \max_{a \in \mathcal{A}} \left\{ \frac{W_{m,m}}{1 - W_{m,m}} \left(2 \frac{\operatorname{Re} \{ y_{\text{MMSE},m}^* a \}}{W_{m,m}} - |a|^2 \right) \right\}$$

$$\begin{aligned}
&= \arg \min_{a \in \mathcal{A}} \left\{ \frac{|y_{\text{MMSE},m}|^2}{W_{m,m}^2} - 2 \frac{\text{Re}\{y_{\text{MMSE},m}^* a\}}{W_{m,m}} + |a|^2 \right\} \\
&= \arg \min_{a \in \mathcal{A}} \eta_m^2(a),
\end{aligned} \tag{3.16}$$

with the “unbiased distance”

$$\eta_m(a) \triangleq \left| \frac{y_{\text{MMSE},m}}{W_{m,m}} - a \right|. \tag{3.17}$$

The minimization in (3.16) is known as *unbiased MMSE detection* [78]; it will hereafter be denoted as

$$\hat{d}_m = \mathcal{Q}_{\mathcal{A},u}\{y_{\text{MMSE},m}\}. \tag{3.18}$$

The terms “unbiased distance” and “unbiased MMSE detection” reflect the fact that, in contrast to the MMSE estimate $y_{\text{MMSE},m}$, the scaled MMSE estimate $\frac{y_{\text{MMSE},m}}{W_{m,m}}$ is conditionally unbiased given d_m , i.e., $\text{E}\{\frac{y_{\text{MMSE},m}}{W_{m,m}} - d_m | d_m\} = 0$ or equivalently $\text{E}\{\frac{y_{\text{MMSE},m}}{W_{m,m}} | d_m\} = d_m$ [78]. In general, the error probability of unbiased MMSE detection (3.18) is slightly smaller than that of conventional MMSE detection (see, e.g., [5]). (Recall that for conventional MMSE detection, each component $y_{\text{MMSE},m}$ of the MMSE-equalized received vector \mathbf{y}_{MMSE} is quantized according to (1.6).) Unbiased and conventional MMSE detection are however fully equivalent for constant-modulus signaling, i.e., when $|a|$ is equal for all symbols $a \in \mathcal{A}$.

Thus, our development in this section has shown that the data-detection stage of DNC—that is, approximate MAP detection using the Gaussian approximation for the post-equalization interference—is equivalent to unbiased MMSE detection, which is computationally simple.

3.3 Layer-Sorting Stage

We now develop the calculations corresponding to the layer-sorting stage. This stage is the determination of the most reliable layer according to (3.9), with $\hat{d}_m = \mathcal{Q}_{\mathcal{A},u}\{y_{\text{MMSE},m}\}$ according to (3.18).

3.3.1 Dynamic Layer Sorting

With (3.6) and (3.7), the maximization in (3.9) becomes

$$m_1 = \arg \max_{m \in \{1, \dots, M\}} \left\{ \frac{\exp(-(\mathbf{y}_{\text{ZF}} - \hat{d}_m \mathbf{e}_m)^H \mathbf{C}_m^{-1} (\mathbf{y}_{\text{ZF}} - \hat{d}_m \mathbf{e}_m))}{\sum_{a \in \mathcal{A}} \exp(-(\mathbf{y}_{\text{ZF}} - a \mathbf{e}_m)^H \mathbf{C}_m^{-1} (\mathbf{y}_{\text{ZF}} - a \mathbf{e}_m))} \right\}. \tag{3.19}$$

Using (3.13) and (3.14), this can be shown to be equivalent to

$$m_1 = \arg \min_{m \in \{1, \dots, M\}} \sum_{a \in \mathcal{A} \setminus \{\hat{d}_m\}} e^{g(a,m)},$$

where $\mathcal{A} \setminus \{\hat{d}_m\}$ refers to the set of all symbols $a \in \mathcal{A}$ that are not equal to \hat{d}_m and

$$g(a, m) \triangleq \frac{2 \text{Re}\{y_{\text{MMSE},m}^* (a - \hat{d}_m)\} - W_{m,m} (|a|^2 - |\hat{d}_m|^2)}{1 - W_{m,m}}.$$

Taking the logarithm of the expression to be minimized and applying the log-max approximation (e.g., [79, 80])

$$\log\left(\sum_i e^{x_i}\right) \approx \max_i x_i, \quad (3.20)$$

we obtain

$$m_1 = \arg \min_{m \in \{1, \dots, M\}} \left\{ \log \left(\sum_{a \in \mathcal{A} \setminus \{\hat{d}_m\}} e^{g(a, m)} \right) \right\} \approx \arg \min_{m \in \{1, \dots, M\}} \left\{ \max_{a \in \mathcal{A} \setminus \{\hat{d}_m\}} g(a, m) \right\}.$$

This simplifying approximation will be used in the following, and accordingly m_1 is considered to be redefined by the last expression. We can reformulate this expression in terms of the unbiased distance $\eta_m^2(\cdot)$ in (3.17):

$$m_1 = \arg \max_{m \in \{1, \dots, M\}} \left\{ \frac{W_{m,m}}{1 - W_{m,m}} I_m \right\}, \quad (3.21)$$

with the *instantaneous reliability factor* (IRF)

$$I_m \triangleq \min_{a \in \mathcal{A} \setminus \{\hat{d}_m\}} \{ \eta_m^2(a) \} - \eta_m^2(\hat{d}_m). \quad (3.22)$$

To obtain an illuminating reformulation of this result, we consider the layerwise MMSE PSNR (see (1.18))

$$\text{SNR}_{\text{MMSE}, m} = \frac{1}{\text{MSE}_m} - 1 \quad (3.23)$$

of the m th layer, where MSE_m is the minimum MSE of the m th layer [60, 72], i.e.,

$$\text{MSE}_m \triangleq \text{E}\{|y_{\text{MMSE}, m} - d_m|^2\} = \sigma_{\mathbf{w}}^2 \left((\mathbf{H}^H \mathbf{H} + \sigma_{\mathbf{w}}^2 \mathbf{I})^{-1} \right)_{m,m},$$

which can be written as

$$\text{MSE}_m = \sigma_{\mathbf{w}}^2 \left(\sum_{m'=1}^M \frac{1}{\sigma_{m'}^2 + \sigma_{\mathbf{w}}^2} \mathbf{v}_{m'} \mathbf{v}_{m'}^H \right)_{m,m} = \sigma_{\mathbf{w}}^2 \sum_{m'=1}^M \frac{1}{\sigma_{m'}^2 + \sigma_{\mathbf{w}}^2} |(\mathbf{v}_{m'})_m|^2.$$

Here, the $\sigma_{m'}^2$'s and $\mathbf{v}_{m'}$'s denote, respectively, the eigenvalues and eigenvectors of $\mathbf{H}^H \mathbf{H}$. MSE_m can be related to $W_{m,m}$ as follows. We first note that $\mathbf{I} + \sigma_{\mathbf{w}}^2 (\mathbf{H}^H \mathbf{H})^{-1} = \sum_{m'=1}^M \left(1 + \frac{\sigma_{\mathbf{w}}^2}{\sigma_{m'}^2} \right) \mathbf{v}_{m'} \mathbf{v}_{m'}^H$. From (3.12), we then have

$$\begin{aligned} W_{m,m} &= \left(\sum_{m'=1}^M \frac{\sigma_{m'}^2}{\sigma_{m'}^2 + \sigma_{\mathbf{w}}^2} \mathbf{v}_{m'} \mathbf{v}_{m'}^H \right)_{m,m} = \sum_{m'=1}^M \frac{\sigma_{m'}^2}{\sigma_{m'}^2 + \sigma_{\mathbf{w}}^2} |(\mathbf{v}_{m'})_m|^2 \\ &= \sum_{m'=1}^M |(\mathbf{v}_{m'})_m|^2 - \sigma_{\mathbf{w}}^2 \sum_{m'=1}^M \frac{1}{\sigma_{m'}^2 + \sigma_{\mathbf{w}}^2} |(\mathbf{v}_{m'})_m|^2 \\ &= 1 - \text{MSE}_m. \end{aligned}$$

Inserting this into (3.23), we obtain $\text{SNR}_{\text{MMSE}, m}$ in terms of $W_{m,m}$:

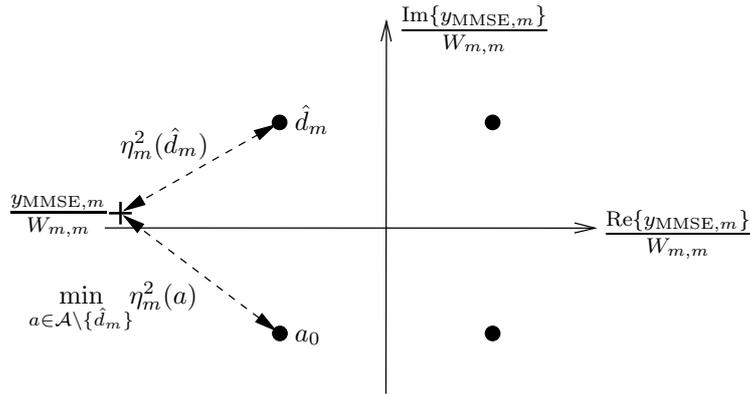


Figure 3.1: Situation where the IRF can lead to a significant performance improvement of dynamic layer sorting over conventional layer sorting.

$$\text{SNR}_{\text{MMSE},m} = \frac{W_{m,m}}{1 - W_{m,m}}. \quad (3.24)$$

Thus, (3.21) can be written in terms of $\text{SNR}_{\text{MMSE},m}$ as

$$m_1 = \arg \max_{m \in \{1, \dots, M\}} \{ \text{SNR}_{\text{MMSE},m} I_m \}. \quad (3.25)$$

For constant-modulus (in particular, PSK) symbol alphabets, (3.25) can be shown to simplify as

$$m_1 = \arg \max_{m \in \{1, \dots, M\}} \left\{ \text{SNR}_{\text{MMSE},m} \min_{a \in \mathcal{A} \setminus \{\hat{d}_m\}} \text{Re} \left\{ \frac{y_{\text{MMSE},m}^*}{W_{m,m}} (\hat{d}_m - a) \right\} \right\}. \quad (3.26)$$

3.3.2 Discussion

The quantity maximized in (3.25), $\text{SNR}_{\text{MMSE},m} I_m$, is the proposed reliability measure of the detected symbol $\hat{d}_m = \mathcal{Q}_{\mathcal{A},u}\{y_{\text{MMSE},m}\}$ of the m th layer. This quantity consists of two factors:

- The first factor is the MMSE PSNR, which expresses the *average* reliability of the m th layer. This factor depends on the channel realization \mathbf{H} and on the noise variance $\sigma_{\mathbf{w}}^2$, but neither on the transmitted data vector \mathbf{d} nor on the noise realization \mathbf{w} .
- The second factor is the IRF in (3.22), which can be rewritten as

$$I_m = \min_{a \in \mathcal{A} \setminus \{\hat{d}_m\}} \{ \eta_m^2(a) \} - \min_{a \in \mathcal{A}} \{ \eta_m^2(a) \} \geq 0.$$

The IRF is seen to compare the smallest unbiased distance, $\eta_m^2(\hat{d}_m) = \min_{a \in \mathcal{A}} \{ \eta_m^2(a) \}$, with the second smallest unbiased distance, $\min_{a \in \mathcal{A} \setminus \{\hat{d}_m\}} \{ \eta_m^2(a) \}$. Thus, it expresses the *instantaneous reliability* of the m th layer decision \hat{d}_m . It depends on \mathbf{d} and \mathbf{w} via $y_{\text{MMSE},m}$ (cf. (3.17)).

With conventional NC, the layers are sorted simply according to maximum $\text{SNR}_{\text{MMSE},m}$ or, equivalently (see (3.23)), according to minimum MSE_m . That is, instead of (3.25) one has (cf. (1.16))

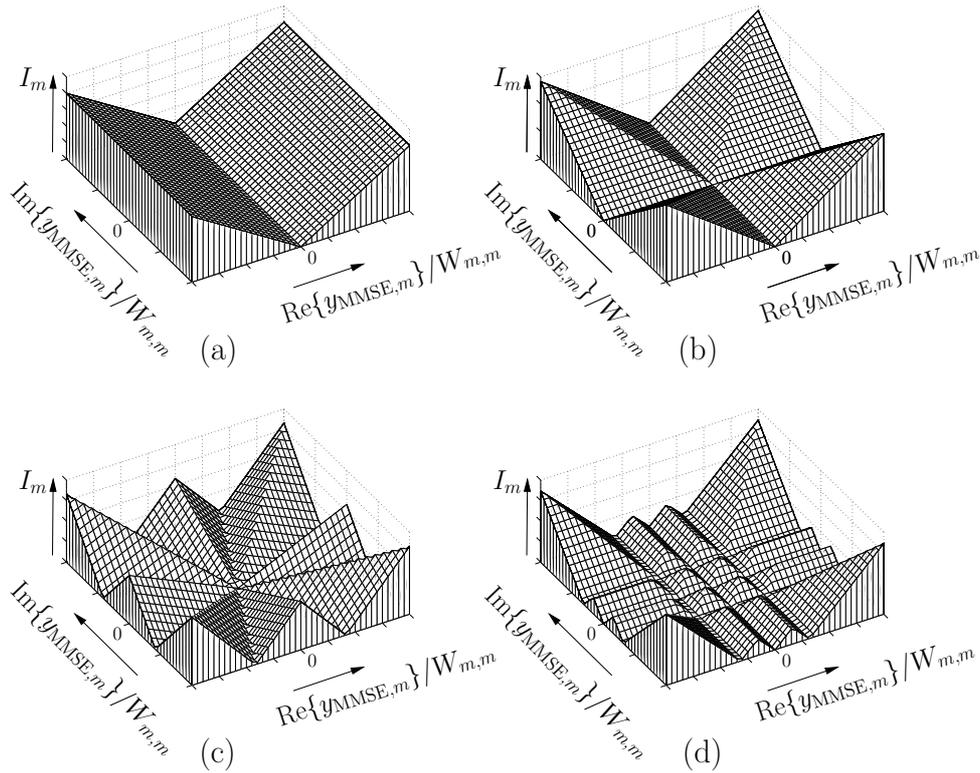


Figure 3.2: IRF I_m as a function of $y_{\text{MMSE},m}/W_{m,m}$ for various symbol alphabets. (a) BPSK, (b) 4-QAM, (c) 8-PSK, and (d) 16-QAM.

$$m_1^{\text{NC}} = \arg \max_{m \in \{1, \dots, M\}} \text{SNR}_{\text{MMSE},m} = \arg \min_{m \in \{1, \dots, M\}} \text{MSE}_m.$$

The new DNC layer-sorting rule (3.25) additionally takes into account the IRF I_m . Whereas $\text{SNR}_{\text{MMSE},m}$ and MSE_m merely measure the average reliability of MMSE equalization, the IRF measures the instantaneous reliability of the subsequent detection (quantization) process. To appreciate the beneficial influence of the IRF, consider the situation shown in Figure 3.1 for a 4-QAM symbol alphabet. We assume that $\text{SNR}_{\text{MMSE},m}$ is large but, for a specific received vector \mathbf{r} , $y_{\text{MMSE},m}/W_{m,m}$ happens to be close to a boundary of the symbol decision regions. This means that layer m is very unreliable because the unbiased distance for the detected symbol \hat{d}_m (cf. (3.17)) is close to the unbiased distance for some other symbol $a_0 \neq \hat{d}_m$, i.e., $\eta_m^2(\hat{d}_m) \approx \eta_m^2(a_0)$. It follows that the IRF is small, and hence DNC correctly treats this layer as unreliable. In contrast, because of the large $\text{SNR}_{\text{MMSE},m}$, conventional NC would erroneously treat this layer as reliable.

In Figure 3.2, the IRF is depicted as a function of $y_{\text{MMSE},m}/W_{m,m}$ for some symbol alphabets. It is seen that the IRF is a piecewise linear function of $y_{\text{MMSE},m}/W_{m,m}$. Furthermore, as discussed above, the IRF is small if $y_{\text{MMSE},m}/W_{m,m}$ is close to a boundary of the symbol decision regions; it is zero on these boundaries.

3.4 The DNC Algorithm

In the previous two sections, we developed the calculations of the DNC algorithm for the first decoding step. We now present a formal statement of the complete DNC algorithm, discuss its efficient implementation using a recursive matrix inversion technique, and analyze its computational complexity.

3.4.1 Algorithm Statement

The calculations to be performed at the s th decoding step of DNC, where $s \in \{1, \dots, M\}$, are summarized below (cf. the review of conventional NC in Section 1.8.3). In what follows, let $\mathbf{H}^{(s)}$ and $\mathbf{r}^{(s)}$ denote, respectively, the reduced system matrix and the interference-cleaned received vector obtained from the previous $((s-1)$ th) decoding step, with initialization according to $\mathbf{H}^{(1)} = \mathbf{H}$ and $\mathbf{r}^{(1)} = \mathbf{r}$.

1. *Precalculations:* Calculate

$$\mathbf{D}_{\text{MMSE}}^{(s)} \triangleq (\mathbf{H}^{(s)H} \mathbf{H}^{(s)} + \sigma_{\mathbf{w}}^2 \mathbf{I})^{-1}, \quad (3.27)$$

and perform MMSE equalization:

$$\mathbf{y}_{\text{MMSE}}^{(s)} = \mathbf{D}_{\text{MMSE}}^{(s)} \mathbf{H}^{(s)H} \mathbf{r}^{(s)}.$$

Next, for all active layers $m = 1, \dots, M - s + 1$, calculate

$$\text{SNR}_{m, \text{MMSE}}^{(s)} = \frac{1}{\text{MSE}_m^{(s)}} - 1, \quad W_{m,m}^{(s)} = 1 - \text{MSE}_m^{(s)},$$

where $\text{MSE}_m^{(s)} = \sigma_{\mathbf{w}}^2 (\mathbf{D}_{\text{MMSE}}^{(s)})_{m,m}$.

2. *Data detection:* Perform unbiased MMSE detection for all active layers, i.e., calculate

$$\hat{d}_m^{(s)} = \mathcal{Q}_{\mathcal{A}, \mathbf{u}}^{(s)} \{y_{\text{MMSE}, m}^{(s)}\} = \arg \min_{a \in \mathcal{A}} \eta_m^{(s)2}(a), \quad m = 1, \dots, M - s + 1,$$

with $\eta_m^{(s)}(a) = \left| \frac{y_{\text{MMSE}, m}^{(s)}}{W_{m,m}^{(s)}} - a \right|$.

3. *Layer sorting:* Determine the most reliable layer index $m_s \in \{1, \dots, M - s + 1\}$ according to

$$m_s = \arg \max_{m \in \{1, \dots, M - s + 1\}} \{\text{SNR}_{\text{MMSE}, m}^{(s)} I_m^{(s)}\},$$

where $I_m^{(s)} = \min_{a \in \mathcal{A} \setminus \{\hat{d}_m^{(s)}\}} \{\eta_m^{(s)2}(a)\} - \eta_m^{(s)2}(\hat{d}_m^{(s)})$.

4. *Decoding and interference cancellation:* Decode layer m_s in favor of $\hat{d}_{m_s}^{(s)}$ and use this result for interference cancellation:

$$\mathbf{r}^{(s+1)} = \mathbf{r}^{(s)} - \mathbf{h}_{m_s}^{(s)} \hat{d}_{m_s}^{(s)},$$

where $\mathbf{h}_{m_s}^{(s)}$ denotes the m_s th column of $\mathbf{H}^{(s)}$. Finally, form $\mathbf{H}^{(s+1)}$ by removing $\mathbf{h}_{m_s}^{(s)}$ from $\mathbf{H}^{(s)}$, and re-index the remaining active layers, i.e., $\{1, \dots, M - s + 1\} \setminus \{m_s\} \rightarrow \{1, \dots, M - s\}$. The vector $\mathbf{r}^{(s+1)}$, matrix $\mathbf{H}^{(s+1)}$, and active-layer index set $\{1, \dots, M - s\}$ then form the input of the next ($s \rightarrow s+1$) decoding step.

3.4.2 Efficient Implementation

The computational complexity of DNC is dominated by the calculation of the matrices $\mathbf{D}_{\text{MMSE}}^{(s)}$, $s = 1, \dots, M$ in (3.27). Whereas $\mathbf{D}_{\text{MMSE}}^{(1)}$ depends only on \mathbf{H} and $\sigma_{\mathbf{w}}^2$, the subsequently calculated matrices $\mathbf{D}_{\text{MMSE}}^{(s)}$ for $s = 2, \dots, M$ depend on the received vector \mathbf{r} via m_s . Thus, the calculation of $\mathbf{D}_{\text{MMSE}}^{(1)}$ contributes to the preparation complexity C_{prep} , whereas the calculation of $\mathbf{D}_{\text{MMSE}}^{(s)}$, $s = 2, \dots, M$, contribute to the vector complexity C_{vector} (cf. Section 1.7.3). With conventional MMSE-based NC, on the other hand, layer sorting just depends on the system matrix \mathbf{H} , and hence the calculation of all matrices $\mathbf{D}_{\text{MMSE}}^{(s)}$, $s = 1, \dots, M$, contribute to C_{prep} . For example, in the case of a SM system and a block-fading channel model, \mathbf{H} (the MIMO channel matrix) does not change during a block of consecutive channel uses. In such cases, C_{vector} is more decisive than C_{prep} , and thus, DNC is more complex than NC.

Fortunately, the complexity of computing the matrices $\mathbf{D}_{\text{MMSE}}^{(s)}$ for a given received vector \mathbf{r} can be significantly reduced by means of a recursive algorithm that was proposed for conventional NC in [70] (see also [81]). Indeed, $\mathbf{D}_{\text{MMSE}}^{(s+1)}$ can be calculated from $\mathbf{D}_{\text{MMSE}}^{(s)}$ as

$$\mathbf{D}_{\text{MMSE}}^{(s+1)} = \begin{pmatrix} \mathbf{A}_1 & \mathbf{A}_2 \\ \mathbf{A}_3 & \mathbf{A}_4 \end{pmatrix} - \frac{1}{\alpha} \begin{pmatrix} \mathbf{a}_1 \\ \mathbf{a}_2 \end{pmatrix} (\mathbf{a}_1^H \ \mathbf{a}_2^H), \quad (3.28)$$

where \mathbf{A}_1 through \mathbf{A}_4 , \mathbf{a}_1 and \mathbf{a}_2 , and α are parts of $\mathbf{D}_{\text{MMSE}}^{(s)}$ as shown below:

$$\mathbf{D}_{\text{MMSE}}^{(s)} = \begin{array}{c} \begin{array}{c} \downarrow \\ m_s \end{array} \\ \begin{array}{|c|c|c|} \hline \mathbf{A}_1 & \mathbf{a}_1 & \mathbf{A}_2 \\ \hline \mathbf{a}_1^H & \alpha & \mathbf{a}_2^H \\ \hline \mathbf{A}_3 & \mathbf{a}_2 & \mathbf{A}_4 \\ \hline \end{array} \begin{array}{c} \leftarrow m_s \end{array} \end{array}$$

The recursion (3.28) is initialized by $\mathbf{D}_{\text{MMSE}}^{(1)} = (\mathbf{H}^H \mathbf{H} + \sigma_{\mathbf{w}}^2 \mathbf{I})^{-1}$.

3.4.3 Computational Complexity

We will now determine the complexity order $\mathcal{O}(\cdot)$ of C_{prep} and C_{vector} for the DNC technique, assuming $N = M$ for simplicity. In Section 3.7.3, the complexity of the DNC will be assessed by means of measurements.

The preparation complexity C_{prep} is caused by the computation of the $M \times M$ matrix $\mathbf{D}_{\text{MMSE}}^{(1)} = (\mathbf{H}^H \mathbf{H} + \sigma_{\mathbf{w}}^2 \mathbf{I})^{-1}$, and thus $C_{\text{vector}} = \mathcal{O}(M^3)$. The vector complexity C_{vector} is composed as follows:

- The complexity of calculating the $M - 1$ matrices $\mathbf{D}_{\text{MMSE}}^{(s)}$ of size $(M - s + 1) \times (M - s + 1)$, $s = 2, \dots, M$ is $\mathcal{O}(M^4)$ when a direct calculation is used and $\mathcal{O}(M^3)$ when the recursive algorithm of the previous section is used.

- We have to perform M equalization, detection, and layer-sorting steps for system models of size $(M - s + 1) \times (M - s + 1)$, $s = 1, \dots, M$; the complexity of these operations is $\mathcal{O}(|\mathcal{A}|M^3)$.
- Finally, the M interference cancellation steps have a complexity of $\mathcal{O}(M^2)$.

Thus, the overall vector complexity C_{vector} of DNC using the efficient recursive calculation of the matrices $\mathbf{D}_{\text{MMSE}}^{(s)}$ is $\mathcal{O}(|\mathcal{A}|M^3)$. This is cubic in the number of layers M and linear in the symbol alphabet size $|\mathcal{A}|$. This complexity order of C_{vector} is larger than that of the SPA and conventional NC (both have a C_{vector} that scales quadratically in the number of layers, cf. Section 2.5.2 and Section 1.8.3). As will be demonstrated in Section 3.7.3, for SM systems of practical interest, the vector complexity C_{vector} of DNC turns out to be twice as large as C_{vector} of NC.

In general, with DNC, C_{vector} is much more significant relative to C_{prep} than with NC. Fortunately, the recursive calculation of the matrices $\mathbf{D}_{\text{MMSE}}^{(s)}$ yields a strong reduction of C_{vector} and the resulting overall complexity of DNC is significantly smaller than that required by the FPSD (see Section 3.7.3). We note that the recursive calculation of the $\mathbf{D}_{\text{MMSE}}^{(s)}$'s yields larger benefits for DNC than for conventional NC, since for NC only C_{prep} is reduced.

3.5 Error Performance of DNC

In Section 3.3.2, we argued that dynamic layer sorting based on the IRF should lead to a performance advantage of DNC over conventional MMSE-based NC. We will now demonstrate this performance advantage by studying the error performance of the first layer-decoding step of DNC. The first layer-decoding step is important because it has a decisive impact on the overall error performance of NC schemes.

3.5.1 Symbol Error Probability of a Two-Layer BPSK System

For mathematical tractability, we consider the simple special case of two layers using BPSK modulation. The two components of \mathbf{y}_{MMSE} are assumed statistically independent and Gaussian. (Our numerical simulations in Section 3.7.1 will demonstrate that the results obtained under these simplifying assumptions are consistent with the performance observed when the assumptions are not satisfied.)

The system matrix \mathbf{H} is considered fixed.

For BPSK modulation, dynamic layer sorting (3.26) simplifies to

$$m_1 = \arg \max_{m \in \{1, \dots, M\}} \{\text{SNR}_{\text{MMSE}, m} \hat{d}_m z_m\}, \quad \text{with } z_m \triangleq \frac{\text{Re}\{y_{\text{MMSE}, m}\}}{W_{m, m}}, \quad (3.29)$$

where \hat{d}_m denotes the result of conventional MMSE detection, i.e.,

$$\hat{d}_m = \mathcal{Q}_{\mathcal{A}}\{y_{\text{MMSE}, m}\} = \text{sgn}(\text{Re}\{y_{\text{MMSE}, m}\}) = \text{sgn}(z_m). \quad (3.30)$$

It follows that

$$m_1 = \arg \max_{m \in \{1, \dots, M\}} \{\text{SNR}_{\text{MMSE}, m} |z_m|\}. \quad (3.31)$$

To calculate the symbol error probability for the m th layer, we note that

$$z_m = d_m + n_m \quad (3.32)$$

where n_m is statistically independent of d_m . Under the Gaussian approximation for the post-equalization interference, n_m is Gaussian with zero mean and variance $\sigma_{n_m}^2 = 1/(2 \text{SNR}_{\text{MMSE},m})$. Thus, the error probability of MMSE detection for the m th layer is given by [21]

$$P[\hat{d}_m \neq d_m] = Q(\sqrt{2 \text{SNR}_{\text{MMSE},m}}), \quad (3.33)$$

where $Q(\cdot)$ denotes the Q-function. This result is valid for both NC and DNC. For NC, it motivates layer sorting according to the maximum PSNR, since the decoding process for the first layer should be most reliable. Thus, the error probability of the first layer-decoding step of conventional NC is given by

$$P_{\text{NC},1}[\mathcal{E}] = Q(\sqrt{2 \text{SNR}_{\text{MMSE},\max}}), \quad (3.34)$$

where $\text{SNR}_{\text{MMSE},\max} \triangleq \max_{m \in \{1, \dots, M\}} \text{SNR}_{\text{MMSE},m}$ denotes the maximum PSNR.

For DNC, calculation of the error probability cannot be based on (3.33) because of the dynamic layer sorting employed. In the following, we derive the error probability of the first layer-decoding step of DNC, $P_{\text{DNC},1}[\mathcal{E}]$, for the case of two statistically independent layers using BPSK modulation ($d_1, d_2 \in \{-1, 1\}$). Because of symmetry, $P_{\text{DNC},1}[\mathcal{E}]$ is equal to the conditional error probability given any specific choice of transmitted symbols d_1 and d_2 , e.g., $d_1 = d_2 = 1$:

$$P_{\text{DNC},1}[\mathcal{E}] = P_{\text{DNC},1}[\mathcal{E} | d_1 = d_2 = 1]. \quad (3.35)$$

To simplify notation we define the PSNR ratio $\rho_{2,1} \triangleq \frac{\text{SNR}_{\text{MMSE},2}}{\text{SNR}_{\text{MMSE},1}}$. Using (3.31) and (3.30), an error in the first layer-decoding step occurs either if DNC decodes in favor of layer 1 ($|z_1| \geq \rho_{2,1}|z_2|$) and makes a detection error ($\hat{d}_1 \neq 1$ or equivalently $z_1 \leq 0$), or if DNC decodes in favor of layer 2 ($|z_2| \geq \frac{1}{\rho_{2,1}}|z_1|$) and makes a detection error ($\hat{d}_2 \neq 1$ or equivalently $z_2 \leq 0$). Thus, the conditional error event given that $d_1 = d_2 = 1$ is

$$\mathcal{E}_c = \left[(|z_1| \geq \rho_{2,1}|z_2|) \wedge (z_1 \leq 0) \right] \vee \left[\left(|z_2| \geq \frac{1}{\rho_{2,1}}|z_1| \right) \wedge (z_2 \leq 0) \right].$$

It can be shown that this is equivalent to the event $z_1 \leq -\rho_{2,1}z_2$. Hence, (3.35) becomes

$$P_{\text{DNC},1}[\mathcal{E}] = P[z_1 \leq -\rho_{2,1}z_2 | d_1 = d_2 = 1].$$

Using (3.32) and assuming that z_1 and z_2 are statistically independent and Gaussian, we obtain further

$$\begin{aligned} P_{\text{DNC},1}[\mathcal{E}] &= \frac{1}{2\pi \sigma_{n_1} \sigma_{n_2}} \int_{-\infty}^{\infty} \int_{-\infty}^{-\rho_{2,1}z_2} \exp\left(-\frac{1}{2} \left(\frac{z_1-1}{\sigma_{n_1}}\right)^2\right) \exp\left(-\frac{1}{2} \left(\frac{z_2-1}{\sigma_{n_2}}\right)^2\right) dz_1 dz_2 \\ &= \frac{1}{\sqrt{2\pi} \sigma_{n_2}} \int_{-\infty}^{\infty} Q\left(\frac{\rho_{2,1}z_2 + 1}{\sigma_{n_1}}\right) \exp\left(-\frac{1}{2} \left(\frac{z_2-1}{\sigma_{n_2}}\right)^2\right) dz_2 \\ &= \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{\infty} Q\left(\sqrt{\frac{\text{SNR}_{\text{MMSE},2}}{\text{SNR}_{\text{MMSE},1}}} x + \sqrt{\frac{2}{\text{SNR}_{\text{MMSE},1}}} (\text{SNR}_{\text{MMSE},1} + \text{SNR}_{\text{MMSE},2})\right) e^{-x^2/2} dx, \end{aligned}$$

where we used $\sigma_{n_m} = 1/\sqrt{2 \text{SNR}_{\text{MMSE},m}}$ and $\rho_{2,1} = \frac{\text{SNR}_{\text{MMSE},2}}{\text{SNR}_{\text{MMSE},1}}$. Finally, by applying the identity (e.g. [6])

$$\frac{1}{\sqrt{2\pi}} \int_{-\infty}^{\infty} \text{Q}(\lambda x + \mu) e^{-x^2/2} dx = \text{Q}\left(\frac{\mu}{\sqrt{1 + \lambda^2}}\right),$$

we obtain the final expression

$$P_{\text{DNC},1}[\mathcal{E}] = \text{Q}\left(\sqrt{2(\text{SNR}_{\text{MMSE},1} + \text{SNR}_{\text{MMSE},2})}\right). \quad (3.36)$$

3.5.2 Discussion

Comparing (3.34) and (3.36), we see that the error probability of NC is determined by the *maximum* of the two PSNRs whereas the error probability of DNC is determined by the *sum* of the two PSNRs. We can draw the following conclusions.

- Since $\text{SNR}_{\text{MMSE},1} + \text{SNR}_{\text{MMSE},2} \geq \max\{\text{SNR}_{\text{MMSE},1}, \text{SNR}_{\text{MMSE},2}\}$, the error probability of DNC is upper bounded by the error probability of NC:

$$P_{\text{DNC},1}[\mathcal{E}] \leq P_{\text{NC},1}[\mathcal{E}].$$

- In the limiting cases $\rho_{2,1} \rightarrow \infty$ and $\frac{1}{\rho_{2,1}} \rightarrow \infty$, the two error probabilities become equal, i.e., $P_{\text{DNC},1}[\mathcal{E}] \rightarrow P_{\text{NC},1}[\mathcal{E}]$. Hence, the error performance of DNC is similar to that of NC if one of the PSNRs is very dominant. In fact, in that case DNC effectively decodes in favor of the layer with maximum PSNR, and thus it becomes equivalent to NC.
- The performance advantage of dynamic layer sorting is most significant for equal PSNRs, i.e., when $\rho_{2,1} = 1$. Here, $P_{\text{NC},1}[\mathcal{E}] = \text{Q}(\sqrt{2 \text{SNR}_{\text{MMSE},1}})$ and $P_{\text{DNC},1}[\mathcal{E}] = \text{Q}(\sqrt{4 \text{SNR}_{\text{MMSE},1}})$, corresponding to a 3-dB SNR advantage of DNC over NC. In fact, for equal PSNRs NC randomly selects some layer and thus does not exploit the possibility of layer sorting. DNC exploits this degree of freedom, and thus achieves better performance.

The simulation results in Section 3.7.1 suggest that these conclusions are approximately true also when the simplifying assumptions under which they were derived are not satisfied.

In [82] it has been shown that for an increasing size of the system matrix, *all* MMSE PSNRs associated with an iid system matrix converge to the *same* deterministic value. In the case of SM systems (where the system matrix equals the channel matrix), we can thus expect that the performance advantage of DNC over NC is stronger for a larger number of transmit and receive antennas. For a small system, on the other hand, the variations in the PSNRs will be large, and thus there may be channel realizations for which the performance advantage of DNC is only marginal (see Section 3.7.1).

Finally, we expect strong average performance advantages of DNC over NC if for each realization of the system matrix \mathbf{H} the PSNRs are grouped into subsets of equal PSNRs. In such a case, NC merely performs a sorting between these subsets; within each subset a layer is randomly selected. This again corresponds to an effective loss of degrees of freedom in performing the layer sorting. Examples are the equivalent real-valued system model of SM systems using QAM signaling (see (1.3)) and systems using certain LD codes (cf. Section 1.5.2).

3.6 LLR-based NC

We now show that LLR-based NC proposed in [75] can also be derived by means of a Gaussian approximation for the post-equalization interference. However, different to the DNC, the post-equalization interference is assumed to be uncorrelated. That is, the matched covariance matrix $\mathbf{C}_m = \mathbf{I} - \mathbf{e}_m \mathbf{e}_m^T + \mathbf{R}_{\tilde{\mathbf{w}}}$ in (3.7) is replaced with $\tilde{\mathbf{C}}_m$, where

$$(\tilde{\mathbf{C}}_m)_{i,m'} \triangleq \begin{cases} (\mathbf{C}_m)_{i,m'}, & \text{for } i = m', \\ 0, & \text{otherwise.} \end{cases} \quad (3.37)$$

We now develop the data detection stage (cf. Section 3.2) and the layer-sorting stage (cf. Section 3.3) using $\tilde{\mathbf{C}}_m$ instead of \mathbf{C}_m . As it was done for the development of the DNC in Section 3.2 and Section 3.3, we again just focus on the first layer decoding step.

3.6.1 Data Detection Stage

Similar to (3.10), we have

$$\hat{d}_m = \arg \max_{a \in \mathcal{A}} \left\{ 2 \operatorname{Re} \{ \mathbf{y}_{\text{ZF}}^H \tilde{\mathbf{C}}_m^{-1} \mathbf{e}_m a \} - |a|^2 \mathbf{e}_m^T \tilde{\mathbf{C}}_m^{-1} \mathbf{e}_m \right\},$$

which can be further simplified as

$$\begin{aligned} \hat{d}_m &= \arg \max_{a \in \mathcal{A}} \left\{ 2 \operatorname{Re} \{ y_{\text{ZF},m}^* (\mathbf{R}_{\tilde{\mathbf{w}}})_{m,m} a \} - |a|^2 (\mathbf{R}_{\tilde{\mathbf{w}}})_{m,m} \right\} \\ &= \arg \max_{a \in \mathcal{A}} \left\{ 2 \operatorname{Re} \{ y_{\text{ZF},m}^* a \} - |a|^2 \right\} \\ &= \arg \min_{a \in \mathcal{A}} \gamma_m^2(a), \end{aligned}$$

where $\gamma_m(a) \triangleq |y_{\text{ZF},m} - a|$. Thus, the data detection stage with the correlations in \mathbf{y}_{ZF} being neglected is equivalent to ZF detection

$$\hat{d}_m = \mathcal{Q}_{\mathcal{A}} \{ y_{\text{ZF},m} \}. \quad (3.38)$$

3.6.2 Layer-Sorting Stage

Using $\tilde{\mathbf{C}}_m$ instead of \mathbf{C}_m and $\hat{d}_m = \mathcal{Q}_{\mathcal{A}} \{ y_{\text{ZF},m} \}$, (3.19) becomes

$$m_1 = \arg \max_{m \in \{1, \dots, M\}} \left\{ \frac{\exp(-(\mathbf{y}_{\text{ZF}} - \hat{d}_m \mathbf{e}_m)^H \tilde{\mathbf{C}}_m^{-1} (\mathbf{y}_{\text{ZF}} - \hat{d}_m \mathbf{e}_m))}{\sum_{a \in \mathcal{A}} \exp(-(\mathbf{y}_{\text{ZF}} - a \mathbf{e}_m)^H \tilde{\mathbf{C}}_m^{-1} (\mathbf{y}_{\text{ZF}} - a \mathbf{e}_m))} \right\}.$$

Using the log-max approximation (3.20), this can be equivalently written as

$$m_1 = \arg \max_{m \in \{1, \dots, M\}} \{ \text{SNR}_{\text{ZF},m} I_m \}, \quad (3.39)$$

where $\text{SNR}_{\text{ZF},m}$ is the ZF post-equalization SNR (see (1.17)), and where the IRF I_m is now given by

$$I_m = \min_{a \in \mathcal{A} \setminus \{\hat{d}_m\}} \{ \gamma_m^2(a) \} - \gamma_m^2(\hat{d}_m).$$

3.6.3 Discussion

The obtained data detection and layer-sorting stage in (3.38) and (3.39), respectively, is the “simplified LLR-based V-BLAST detector” of [75]. Evidently, LLR-based NC can be seen as a ZF-analogue of the proposed DNC scheme; it can be formally obtained by replacing all MMSE-based quantities (equalizer, unbiased distance, post-equalization SNR) with their respective ZF-based counterparts. Our derivation using the Gaussian approximation for the post-equalization interference also reveals that the proposed DNC detector is an extension of the LLR-based NC detector to correlated post-equalization interference. This provides an explanation of the significant performance advantage of DNC over LLR-based NC, which will be demonstrated in Section 3.7.4. This performance advantage is in fact obtained with no increase in computational complexity. However, in contrast to DNC, the result of LLR-based NC is invariant to the value of the noise variance, i.e. $\text{SNR}_{\text{ZF},m}$ in (3.39) can be replaced with $1/((\mathbf{H}^H \mathbf{H})^{-1})_{m,m}$ (cf. (1.17)). Thus, LLR-based NC has the advantage that no knowledge about the noise variance is required.

Our use of the recursive calculation of $\mathbf{D}_{\text{MMSE}}^{(s)}$, $s = 1, \dots, M$, (see Section 3.4.2) to reduce the computational complexity of DNC can also be applied to reduce the complexity of LLR-based NC in a straightforward way. The same recursive algorithm can be used for calculating $\mathbf{D}_{\text{ZF}}^{(s)} \triangleq (\mathbf{H}^H(s) \mathbf{H}(s))^{-1}$ instead of $\mathbf{D}_{\text{MMSE}}^{(s)} = (\mathbf{H}^H(s) \mathbf{H}(s) + \sigma_{\mathbf{w}}^2 \mathbf{I})^{-1}$. Furthermore, our investigation of the error performance of the first layer decoding step of DNC (see Section 3.5) applies similarly to LLR-based NC. In (3.36), the MMSE PSNRs $\text{SNR}_{\text{MMSE},1}$ and $\text{SNR}_{\text{MMSE},2}$ are simply replaced by their corresponding ZF counterparts $\text{SNR}_{\text{ZF},1}$ and $\text{SNR}_{\text{ZF},2}$, respectively.

3.7 Simulation Results

We will now present simulation results to assess the SER performance and computational complexity of the proposed DNC technique. At the beginning, we compare DNC with conventional MMSE-based NC using PSNR sorting (see Section 1.8.3, referred to as NC-MMSE) and with ML detection. In Section 3.7.4 we then compare DNC with LLR-based NC, and finally, in Section 3.7.5 we compare DNC and LLR-based NC with the geometry-based detectors that were proposed in Chapter 2. In our simulations, we used iid Gaussian MIMO channels.

3.7.1 SER Performance for Spatial Multiplexing Systems

For an SM system (see Section 1.5.1), the system matrix \mathbf{H} is the MIMO channel matrix and the size of the system model is given by the numbers of transmit and receive antennas, i.e., $M = M_{\text{T}}$ and $N = M_{\text{R}}$.

SER Performance of First Layer-Decoding Step

We first corroborate our theoretical error probability results about the error performance of the first layer-decoding step of Section 3.5. We considered a MIMO system with $M_{\text{T}} = M_{\text{R}} = 8$ and a 4-QAM

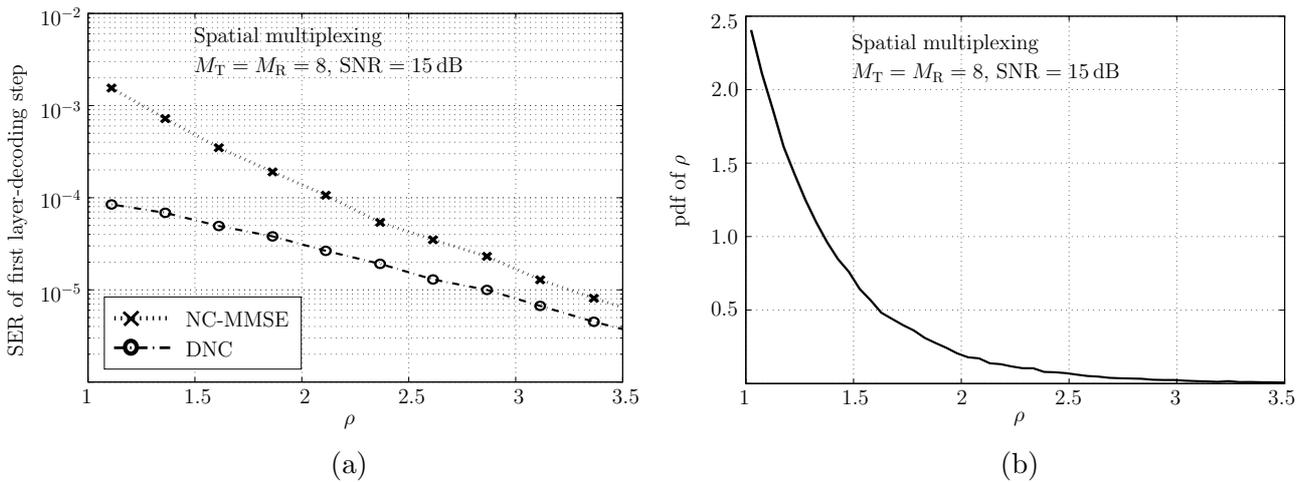


Figure 3.3: Simulation results demonstrating the performance of the first layer-decoding step for DNC and conventional MMSE-based NC, for an SM system with $M_T = M_R = 8$, 4-QAM symbol alphabet, and an SNR of 15 dB. (a) SER versus the PSNR ratio ρ (see text) corresponding to the respective channel realization, (b) estimated pdf of ρ .

symbol alphabet. This system does not comply with the simplifying assumptions made in Section 3.5. Figure 3.3(a) shows the simulated SER (at a channel SNR of 15 dB) of the first layer-decoding step for DNC and for conventional MMSE-based NC with layer sorting according to maximum PSNR. The SER is plotted versus the ratio of the largest PSNR to the second largest PSNR (this ratio is denoted as ρ). It can be seen that the performance advantage of DNC over NC is largest when the two largest PSNRs are nearly equal, i.e., for $\rho \approx 1$. In that case, DNC achieves an SER reduction by a factor of about 20. On the other hand, the SER reduction becomes quite small when one of the PSNRs is dominant (e.g., for $\rho = 3$ the SER is reduced by just a factor of about 2). This behavior is consistent with our theoretical results in Section 3.5.

The impact of this behavior on the *average* SER performance of the first layer-decoding step of course depends on the pdf of the PSNR ratio ρ . In Figure 3.3(b), we show an estimated pdf that has been computed in the course of the simulation described above. It can be seen that small values of ρ are most likely. Thus we can expect that for the first layer-decoding step, the average SER of DNC is significantly smaller than the average SER of NC-MMSE. This is important since the first layer-decoding step has a decisive impact on the overall SER performance.

Overall SER Performance

We now investigate the overall SER performance of the DNC scheme. The performance of DNC and NC-MMSE was evaluated for both the (standard) complex-valued system model (1.1) and the corresponding equivalent real-valued system model (1.3). The real schemes will be denoted as DNC-R and NC-MMSE-R, respectively. For (D)NC schemes, other than for ML detection and linear equalization based detection, the complex and real implementations generally exhibit different SER

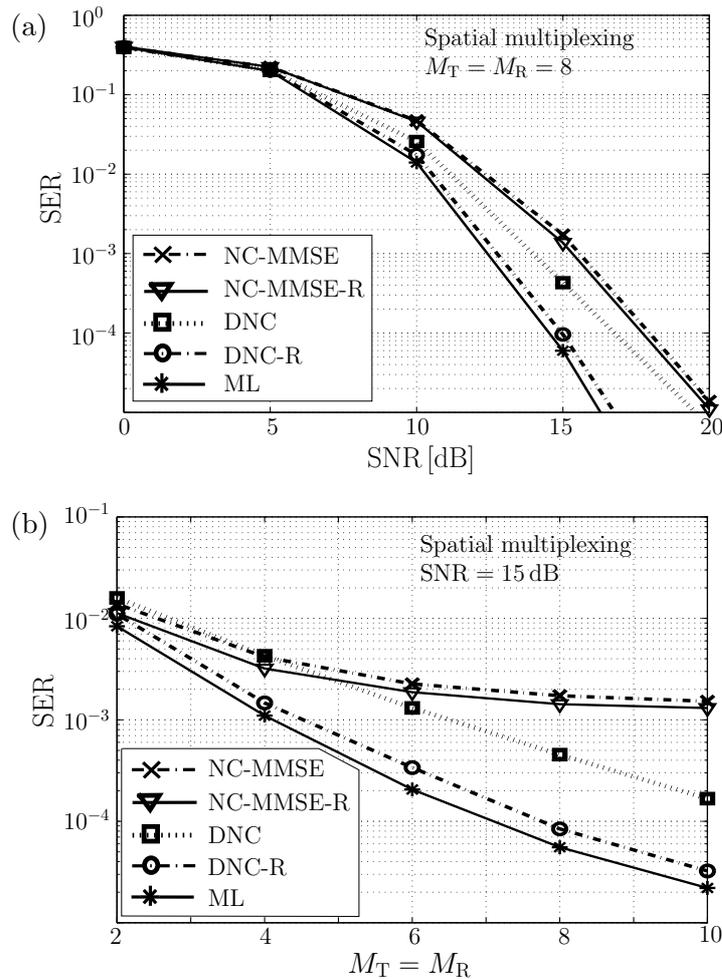


Figure 3.4: SER performance of the proposed DNC detectors (DNC, DNC-R) and of NC-MMSE, NC-MMSE-R, and ML detection for SM systems using 4-QAM symbols. (a) SER-versus-SNR performance for $M_T = M_R = 8$, (b) SER versus $M_T = M_R$ performance at an SNR of 15 dB.

performance. This is because in the real implementation, the layer sorting can be done independently for the real part and the imaginary part (see, e.g., [30] for conventional NC).

For an SM system of size $M_T = M_R = 8$ and with 4-QAM symbols, Figure 3.4(a) shows the SER-versus-SNR performance of the DNC and DNC-R detectors and of NC-MMSE, NC-MMSE-R, and ML. Figure 3.4(b) shows the SER versus $M_T = M_R$ performance at an SNR of 15 dB. The following conclusions can be drawn from these results.

- DNC-R achieves near-ML performance over a wide range of SNRs (see Figure 3.4(a)) and over a wide range of system sizes (see Figure 3.4(b)).
- DNC-R performs significantly better than DNC. To understand this behavior, consider the case where, e.g., the real part of a given layer is reliable but the imaginary part is very unreliable (such a situation has been shown in Figure 3.1). Because DNC-R performs reliability estimation

and layer sorting separately for the real part and the imaginary part, it is able to correctly adapt to this situation. In contrast, DNC does not distinguish between the reliabilities of the real part and the imaginary part, and thus the layer sorting is the same for both parts.

- NC-MMSE-R performs only slightly better than NC-MMSE. This can be understood as follows. For NC-MMSE-R, at the *first* decoding step, the estimated reliabilities (PSNRs) are the same for the real and imaginary parts of a given layer. Thus, the layer sorting at the first decoding step cannot be done differently for the real and imaginary parts. However, interference cancellation is performed separately, which may result in different layer orders for the real and imaginary parts in *subsequent* detection and interference cancellation steps. In contrast, with NC-MMSE the real and imaginary parts are always canceled jointly. This explains the slight performance advantage of NC-MMSE-R over NC-MMSE (see [30] for ZF-based NC).
- DNC-R performs substantially better than NC-MMSE-R and NC-MMSE. This is because DNC-R employs dynamic layer sorting which, moreover, is carried out separately for the real and imaginary parts.
- DNC outperforms NC-MMSE-R and NC-MMSE for $M_T = M_R \geq 6$.
- The performance advantage of DNC over NC-MMSE is greater for larger system sizes. As explained in Section 3.5, for increasing system size the PSNRs of the various layers become more similar, and thus NC-MMSE is increasingly unable to perform an appropriate layer sorting.

3.7.2 SER Performance for Systems Using Linear Dispersion Codes

Next, we consider a MIMO system using a linear dispersion (LD) code (see Section 1.5.2). We used the LD code in [33, equation (31)] with $M_T = M_R = T = 3$, $Q = 9$, and 4-QAM symbols (here referred to as LDC1), and the LD code discussed in [33, p. 1818] with $M_T = 8$, $M_R = 4$, $T = 8$, $Q = 32$, and 16-QAM symbols (here referred to as LDC2). The size of the equivalent channel matrices associated with LDC1 and LDC2 is 18×18 and 64×64 , respectively. Fig. 3.5(a) and (b) shows the SER-versus-SNR performance of DNC, NC, and ML detection using LDC1 and LDC2, respectively. The following conclusions can be drawn from these results.

- DNC can achieve near-ML performance. At an SER of 10^{-2} , the SNR loss of DNC compared to ML detection is just about 0.8 dB for LDC1 and 1.5 dB for LDC2.
- DNC performs substantially better than NC-MMSE. This can be explained by the structure of the system matrix \mathbf{H} . Indeed, at the first decoding step, the MMSE PSNRs of *all* layers are exactly equal for LDC1 and very similar for LDC2 (99% of all realizations of \mathbf{H} have $\rho < 1.5$).

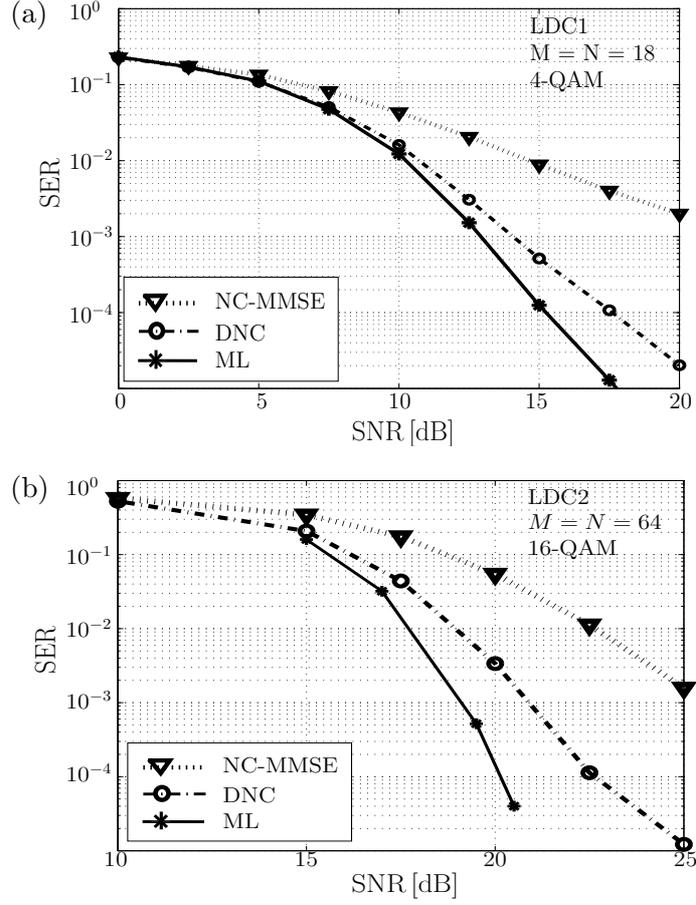


Figure 3.5: *SER-versus-SNR performance of the proposed DNC detector and of NC-MMSE and ML for MIMO systems using LD codes. (a) First LD code (see text) with $M_T = M_R = 3$ and 4-QAM symbols, (b) second LD code with $M_T = 8$, $M_R = 4$, and 16-QAM symbols. (The SER curve for the ML detector in part (b) was adapted from [33, Figure 9] via the approximate relation $SER \approx 4 \cdot BER$.)*

3.7.3 Computational Complexity

Next, we complement the complexity analysis of Section 3.4.3 by presenting empirical estimates of the computational complexity of DNC, NC, and ML detection. For SM systems with equal numbers of transmit and receive antennas $M_T = M_R \in \{4, 6, 8\}$ and 4-QAM symbols, Table 3.1 shows kflop estimates that were measured using MATLAB (the corresponding SER performance was shown in Figure 3.4). We again distinguish between the preparation complexity C_{prep} (Table 3.1(a)) and the vector complexity C_{vector} (Table 3.1(b)). As for the results in Table 2.1 in Section 2.6.2, FPSD was used for ML detection and the efficient recursive matrix inversion algorithm (discussed in detail in Section 3.4.2) was used for conventional NC (thus the numbers for FPSD and NC in Table 3.1 are the same as in Table 2.1). The following conclusions can be drawn from Table 3.1.

- The vector complexity C_{vector} of DNC-R is significantly smaller than both the average and maximum C_{vector} of FPSD (even though DNC-R achieves near-ML performance).

(a)

measured kflops – preparation complexity C_{prep}					
channel	FPSD	DNC-R	DNC	NC-R	NC
(4, 4)	1.9	2.5	1.7	2.7	1.9
(6, 6)	6.1	7.9	4.8	8.8	5.3
(8, 8)	14.0	18.1	10.2	20.2	11.4

(b)

measured kflops – vector complexity C_{vector}						
channel	FPSD		DNC-R	DNC	NC-R	NC
	av.	max.				
(4, 4)	2.6	18.2	1.4	1.2	0.7	0.6
(6, 6)	11.0	79.2	4.4	3.3	2.3	1.5
(8, 8)	44.1	364	10.1	6.9	5.1	3.2

Table 3.1: Measured computational complexity (in kflops) of the various detection techniques for SM systems. (a) Preparation complexity C_{prep} , (b) vector complexity C_{vector} .

- For DNC(-R), C_{vector} is about twice as large as for NC(-R). The complexity of DNC-R is about three times larger than that of NC.
- The computational complexity of the real implementations (DNC-R, NC-R) is larger than that of the complex-valued counterparts (DNC, NC). This is due to the double system size, even though some savings are made possible by the real calculations. This additional computational complexity of DNC-R yields a substantial performance improvement over DNC while the performance improvement of NC-R over NC is only marginal (cf. Figure 3.4). Thus, for NC, the real-valued implementation may be not be justifiable.
- C_{prep} is slightly smaller for DNC(-R) than for NC(-R), because with DNC(-R) a part of C_{prep} is transferred to C_{vector} .
- In contrast to FPSD, the complexity of DNC(-R) and NC(-R) is predetermined; it does not depend on the specific channel or noise realization.

3.7.4 DNC versus LLR-based NC

We now compare the performance of DNC to LLR-based NC [75] (see also Section 3.6). We consider a $M_T = M_R = 6$ SM system with 4-QAM symbols. Both, DNC and LLR-based NC were applied to the equivalent real-valued system model according to (1.3) (denoted as DNC-R and LLR-NC-R, respectively). As a reference, we also show the results for ML detection and conventional NC-ZF and NC-MMSE (both employing PSNR sorting). In fact, LLR-NC-R requires the same computational effort as DNC-R (thus, the results for DNC(-R) in Table 3.1 also hold for LLR-NC(-R)); however, in contrast to DNC, LLR-based NC does not require any knowledge about the noise variance.

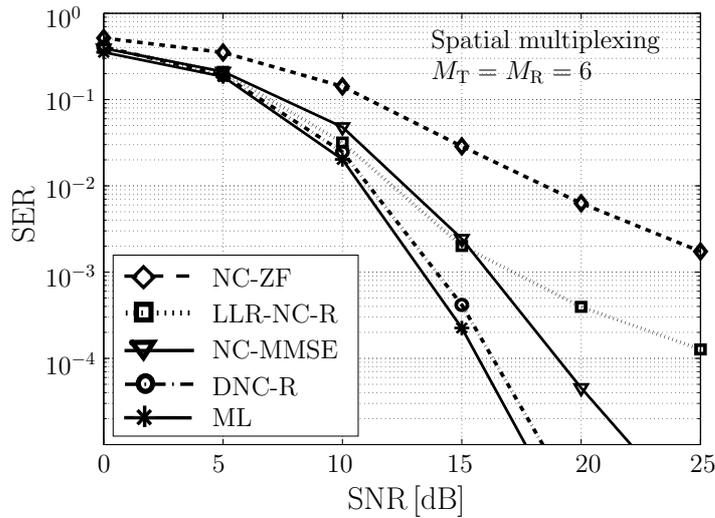


Figure 3.6: *SER versus SNR performance of DNC-R and LLR-NC-R for a $M_T = M_R = 6$ SM system using 4-QAM symbols.*

The following conclusions can be drawn from Figure 3.6.

- LLR-NC-R outperforms conventional NC-ZF. This is due to dynamic layer sorting employed by LLR-NC.
- DNC-R performs significantly better than LLR-NC-R. As discussed in Section 3.6, DNC can be seen as an extension of LLR-NC to correlated post-equalization interference. This explains its significant performance advantage.
- In the low SNR regime LLR-NC-R performs slightly better than conventional NC-MMSE. In the high SNR regime NC-MMSE performs significantly better than LLR-NC-R.

3.7.5 DNC versus Geometry-Based Detection

In the previous sections we observed that DNC can achieve near-ML performance with low computational effort. In Chapter 2 this was also claimed for the proposed geometry-based detectors (in particular for the SPA). We now summarize and discuss the performance and complexity results of DNC and geometry-based detection. Exemplary, Figure 3.7 summarizes the SER-versus-SNR performance results for the $M_T = M_R = 6$ SM system with 4-QAM symbols. Here, the results are shown separately for ZF-based (see Figure 3.7(a)) and MMSE-based (see Figure 3.7(b)) schemes. Furthermore, for the ZF-based schemes, LLR-NC-R as the ZF analogue of DNC-R is considered, and, in addition, we also implemented the SPA as an add-on to LLR-NC-R (denoted as SPA-LLR-NC-R). As a reference, ML performance is shown in both figures. Apart from the computational complexity of SPA-LLR-NC-R, all corresponding complexity results can be found in Table 2.1 and Table 3.1, respectively. Note that LLR-NC-R has virtually the same computational complexity as DNC-R.

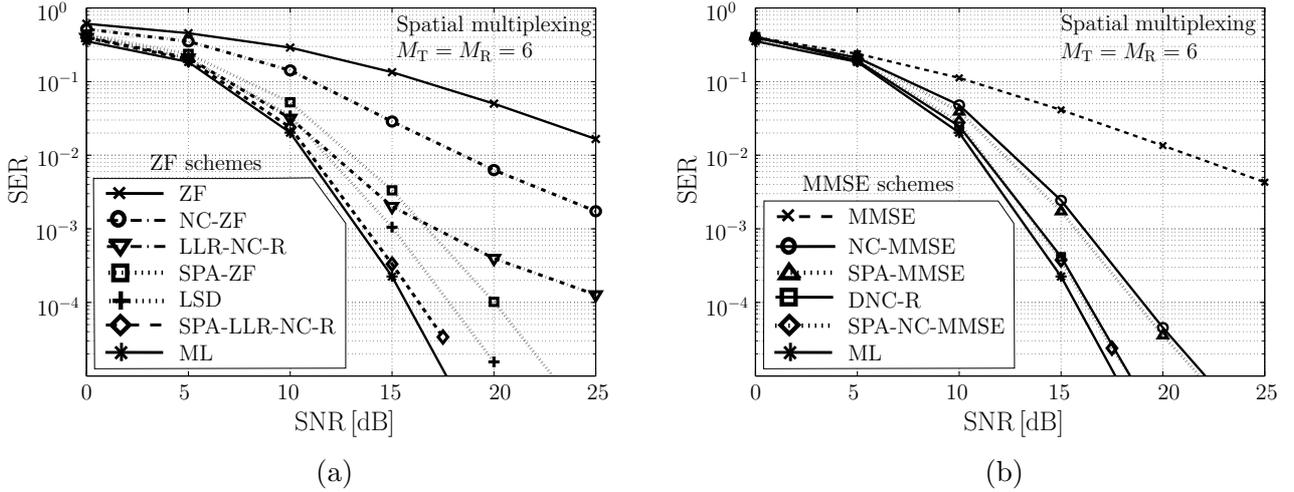


Figure 3.7: SER versus SNR performance of various standard and proposed detectors (DNC and geometry-based detectors) for a $M_T = M_R = 6$ SM system using 4-QAM symbols. (a) ZF-based schemes, (b) MMSE-based schemes.

The following conclusions can be drawn from Figure 3.7 .

- Both SPA-NC-MMSE and DNC-R achieve (roughly the same) near-ML performance (cf. Figure 3.7(b)) with much lower computational cost than FPSD and LSD. SPA-NC-MMSE has a larger maximum C_{vector} but a smaller average C_{vector} than DNC-R.
- For the ZF schemes, only the SPA add-on to LLR-NC-R is able to achieve near-ML performance (cf. Figure 3.7(a)).
- DNC-R performs significantly better than SPA-MMSE. Here, DNC-R has a complexity similar to the maximum C_{vector} of SPA-MMSE, but, on average, SPA-MMSE is much more efficient. (For the $M_T = M_R = 4$ system, however, both algorithms achieve near-ML performance, cf. Figure 2.11 and Figure 3.4(b), but with DNC-R having a slight advantage in complexity.)
- SPA-ZF outperforms LLR-NC-R in the high SNR regime and suffers from a small performance degradation in the low SNR regime (cf. Figure 3.7(a)). In particular, SPA-ZF achieves a higher diversity than LLR-NC-R while having a smaller average complexity than LLR-NC-R.

Discussion and Further Results

For moderate sized SM systems (which are of greatest practical interest), both the DNC and the SPA algorithms are a very good choice to achieve near-ML performance at low computational cost. They are also superior to the LSD, whose computational complexity is much larger than that of the DNC and the SPA algorithms. Furthermore, the DNC has a slight complexity advantage over the SPA algorithms and for larger MIMO systems the performance of DNC will even improve relative to the SPA algorithms. This is due to the fact that the IBC approximation (the basis for the SPA) becomes

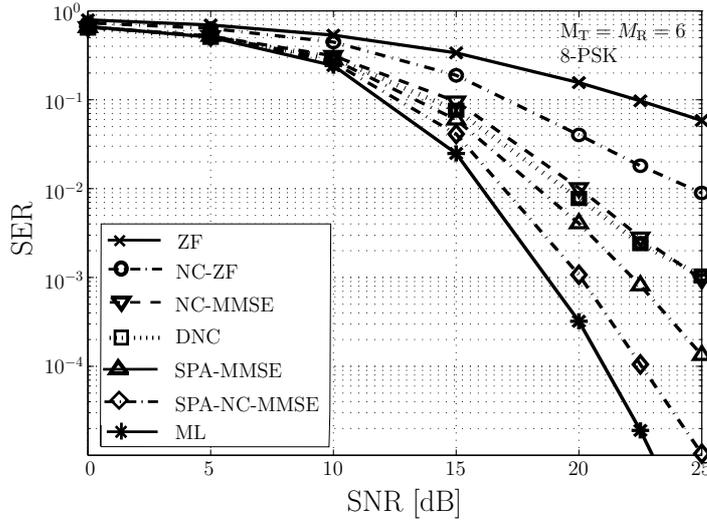


Figure 3.8: *SER versus SNR performance of DNC and various SPA schemes for a $M_T = M_R = 6$ SM system using 8-PSK symbols. Here, DNC-R cannot be performed since 8-PSK symbols do not allow for a separate decoding of real and imaginary parts of the data symbols.*

less accurate while DNC-R can still achieve near-ML performance. Note however that DNC requires the knowledge of the noise variance; the ZF analogue of DNC-R (the LLR-NC-R algorithm) is in general not able to achieve near-ML performance, and the ZF variants of the SPA are superior to LLR-NC-R.

Up to now, we considered QAM symbol alphabets that allow for a separate decoding of the real and imaginary parts of the data symbols. The corresponding DNC implementation was based on the equivalent real-valued system model. We showed that in this case, DNC (i.e. DNC-R) can achieve near-ML performance and outperforms the SPA-lin. detectors. Figure 3.4 shows that the performance improvement of DNC over conventional NC can be quite small if this separate decoding is not utilized. For an 8-PSK symbol alphabet, for example, such a separate decoding is not possible and DNC cannot be expected to achieve near-ML performance (since DNC-R cannot be performed). This is supported by Figure 3.8, which shows DNC and SPA performance results for a $M_T = M_R = 6$ SM system employing 8-PSK symbols. The following conclusions can be drawn.

- Indeed, DNC cannot achieve near-ML performance and it has just a slight performance advantage over conventional NC. Thus, even by exploiting dynamic layer sorting, near-ML performance cannot be achieved in this case.
- In contrast to DNC, SPA-NC-MMSE can achieve near-ML performance.
- The SPA-MMSE detector outperforms DNC.

Thus, the SPA and the DNC show a different performance relative to each other depending on the considered system model. In addition to that, they exhibit the following different features:

- The SPA is specifically tailored to SM systems while the DNC is a very general algorithm that can be used for any type of a linear MIMO model (including, e.g., LD codes).
- The DNC is not restricted to any type of symbol alphabet while the SPA can only be used for constant modulus alphabets. However, DNC achieves best performance if the real and imaginary parts of the data symbols can be decoded separately, which, e.g., is not possible for $|\mathcal{A}|$ -PSK alphabets with $|\mathcal{A}| > 4$.
- In contrast to the DNC, the SPA is an add-on to any suboptimum data detection algorithm. The SPA itself can be combined with DNC or LLR-NC (e.g., to achieve near-ML performance, the SPA was employed as an add-on to LLR-NC-R in Figure 3.7(a)).
- By construction, the DNC operates on symbol level while the SPA add-on operates on data vector level.
- The computational complexity of the DNC is predetermined (i.e. it does not depend on the specific channel, data, or noise realization), while the complexity of the SPA is random.

3.8 Summary and Conclusions

In this chapter we proposed the *dynamic nulling-and-cancelling* (DNC) technique for MIMO detection. DNC is based on the principle that at each decoding step, the symbol and layer with maximum approximate a-posteriori probability (APP) is detected and canceled. The approximate APP is constructed via a Gaussian approximation for the post-equalization interference. This results in an MMSE nulling technique and a “dynamic” layer-sorting rule that is superior to conventional layer sorting based on the “static” (average) post-equalization SNRs because it also exploits the information contained in the current received vector. We also showed that a recently proposed NC technique using log-likelihood ratios (LLRs) for layer sorting, too, can be derived with a Gaussian approximation for the post-equalization interference; however, in contrast to our DNC scheme, the interference is assumed to be uncorrelated. Thus, our DNC scheme can be seen as an extension of LLR-based NC to correlated post-equalization interference.

The performance advantages of DNC over conventional MMSE-based NC were demonstrated both analytically and numerically. Specifically, we showed that the largest performance gains are obtained when the post-equalization SNRs of all layers are similar. Whereas in this case NC cannot perform a meaningful layer sorting (it effectively selects some layer at random), DNC exploits the information provided by the instantaneous-reliability factor for layer sorting and thus achieves better performance. This advantage of DNC is particularly pronounced in the case of large SM systems, in the case of MIMO systems using certain linear dispersion codes, and in the case of QAM symbols that allow for a separate decoding of the real and imaginary parts of the data symbols. The additional computations required by dynamic layer sorting were strongly reduced by a recursive calculation of the MMSE equalizer matrices.

Our simulation results showed that DNC significantly outperforms LLR-based NC and that it can yield near-ML performance for a wide range of system sizes and channel SNRs with a much smaller computational complexity than that required by the sphere-decoding algorithm for ML detection. However, if the real and the imaginary parts of the data symbols cannot be decoded separately (in particular for higher-order PSK alphabets), near-ML performance is in general not achieved by the DNC technique; in this case, MMSE-based NC employing the SPA as an add-on achieves better performance results. DNC has the advantage that it can be applied to arbitrary MIMO systems employing arbitrary symbol alphabets while the SPA is specifically tailored to SM systems with constant modulus alphabets. For SM systems of moderate size, DNC even turns out to be slightly more efficient (in particular, with respect to the worst-case computational requirements) than the several SPA variants.

4

Soft-Output Detection Algorithms

IN the previous two chapters we investigated *hard-output* data detection algorithms that provide hard decisions about the transmitted data symbols and we investigated their performance for *uncoded* MIMO systems, where no (outer) channel code was taken into account. However, any practical wireless system employs some sort of channel coding to allow for a reliable communication. For MIMO systems, the most important coding strategies are space-time trellis coding [3] and *bit-interleaved coded modulation* (BICM) [83–85]. We will focus on MIMO-BICM since it is very well suited for MIMO fading channels (including OFDM-based MIMO systems [86]), where it has been shown to outperform space-time trellis coding [87].

Although hard-output detectors can directly be employed for MIMO-BICM systems as well, the overall system performance can be improved by using *soft-output* instead of hard-output detectors [88]. In this chapter we will investigate and propose such soft-output detectors for MIMO-BICM systems. Here, usually a two-stage receiver is employed that consists of a detector and a channel decoder that are separated by an interleaver. The detector provides soft (or hard) decisions about the coded bits, which then form the input to the channel decoder. A soft-output detector provides *reliability* estimates of the coded bits, which are usually expressed as *log-likelihood ratios* (LLRs). This is in contrast to a hard-output detector (e.g., any detector from Chapter 2 or Chapter 3 followed by a demapper), which just would provide a binary hard decision. We do not consider any iterative (“turbo”) detection and decoding processing [80, 89, 90], which would result in a further improvement of MIMO-BICM system performance, however, with the drawback of an increased computational effort.

Similar to optimum hard-output detection, soft-output detection with an exact LLR calculation is computationally very intensive and becomes prohibitive for large MIMO systems. Thus, there is a strong demand for efficient soft-output detectors that calculate *approximate* LLRs. In this chapter, we extend the algorithms and concepts of Chapter 2 and Chapter 3 to provide approximate LLRs of

coded bits instead of suboptimum hard symbol decisions. In particular, we will extend the hard-output SPA (see Section 2.5) to its soft version (which we will refer to as the soft-output SPA (SSPA) [26]). The soft-output extension of the LSD can be found in [27]. Furthermore, we will use the concept of the DNC (i.e. the use of a Gaussian approximation for the post-equalization interference to get a computational efficient expression for the reliability of a certain data symbol) to re-derive [28] the very efficient equalization-based (i.e. ZF- and MMSE-based) soft-output detectors that can also be found in [91–93]. In fact, our derivation shows that MMSE-based soft-output detection can be seen as an extension of ZF-based soft-output detection to correlated post-equalization interference.

The various soft-output detectors will be assessed for OFDM-based MIMO-BICM systems using synthetic and measured MIMO channels. Our simulation results demonstrate that the SSPA can efficiently achieve a performance close to the performance of the soft-output extension (or *list extension*) of Fincke-Phost's sphere decoding algorithm (LFPSD) [80] for ML detection. Furthermore, we will show that the equalization-based soft-output detectors suffer from a significant performance loss compared to the SSPA and LFPSD while requiring significantly less computational complexity.

This chapter is organized as follows. In Section 4.1 we will provide some background on MIMO-BICM and soft-output detection. In Section 4.2 we derive the ZF- and the MMSE-based soft-output detectors using a Gaussian approximation of the post-equalization interference. Section 4.3 is devoted to the soft-output extension of the SPA. Finally, simulation results are provided in Section 4.4.

4.1 MIMO Bit-Interleaved Coded Modulation

In this section we provide some background on MIMO-BICM systems employing soft-output detection and soft-input channel decoding.

4.1.1 MIMO-BICM System Model

We consider a MIMO-BICM system as illustrated in Figure 4.1 (e.g., [86,87]). At the BICM transmitter, a stream of information bits is passed through a standard convolutional encoder and an interleaver Π . The resulting stream of coded and interleaved bits is then demultiplexed into M bit streams that are partitioned into groups of $\log_2|\mathcal{A}|$ bits. At each time instant, M of these groups (each associated to a different data stream) are mapped to the complex-valued data symbols $d_m \in \mathcal{A}$, $m = 1, \dots, M$, which are then transmitted over the MIMO system acting as (cf. system model (1.1))

$$\mathbf{r} = \mathbf{H}\mathbf{d} + \mathbf{w}. \quad (4.1)$$

In this setup, the coding and interleaving is performed across subsequent channel uses, i.e. the bits that belong to a single codeword are spread over many subsequent data transmissions, where each just carries $M \log_2|\mathcal{A}|$ coded bits. Thus, if the encoding length (and thus the interleaver length) is sufficiently large, any correlation between the coded bits that belong to the same \mathbf{d} can be neglected. If we consider an OFDM-based MIMO system (cf. (1.4)), where (4.1) holds for each subcarrier, the coding and interleaving is performed across the subcarriers [86, 87]. If K denotes the number of

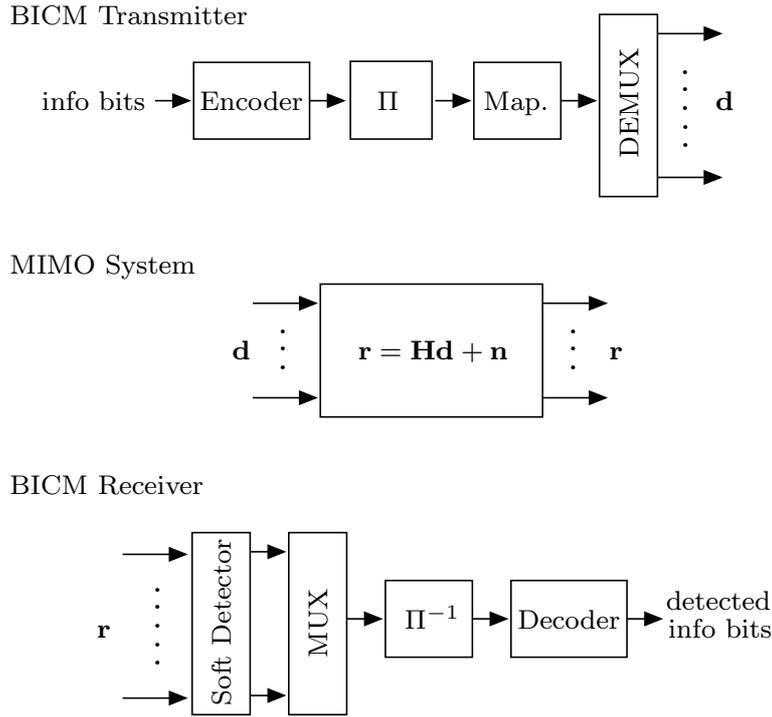


Figure 4.1: MIMO system using bit-interleaved coded modulation.

subcarriers, a block of KM data symbols is transmitted or, equivalently, each codeword consists of $KM \log_2 |\mathcal{A}|$ bits.

At the MIMO-BICM receiver, the soft-output detector uses the received vector \mathbf{r} and the knowledge of \mathbf{H} and $\sigma_{\mathbf{w}}^2$ to calculate an LLR for each of the $M \log_2 |\mathcal{A}|$ coded bits that are associated with a transmitted data vector \mathbf{d} . Here, no structure of the channel code is taken into account. The resulting LLRs (from all layers and channel uses) are multiplexed into a single stream, deinterleaved (using deinterleaver Π^{-1}), and are then used for soft-in Viterbi decoding [88].

4.1.2 Soft-Output Detection

Let $b_{m,i}$ with $i = 1, \dots, \log_2 |\mathcal{A}|$ denote the coded bits of the m th layer (i.e., after encoding, interleaving, demultiplexing), to which the symbol $d_m \in \mathcal{A}$ is associated via some labeling (e.g., Gray labeling). We assume that the code bits $b_{m,i}$ are equally likely and statistically independent, which is a good approximation for a sufficiently large encoder (and interleaver) length.

The LLR of $b_{m,i}$ is given by

$$\Lambda_{m,i} \triangleq \log \left(\frac{f(\mathbf{r} | b_{m,i} = 1)}{f(\mathbf{r} | b_{m,i} = 0)} \right), \quad (4.2)$$

which can be further written as

$$\Lambda_{m,i} = \log \left(\frac{\sum_{\mathbf{d} \in \mathcal{D}_{m,i}^1} e^{-\frac{1}{\sigma_{\mathbf{w}}^2} \|\mathbf{r} - \mathbf{H}\mathbf{d}\|^2}}{\sum_{\mathbf{d} \in \mathcal{D}_{m,i}^0} e^{-\frac{1}{\sigma_{\mathbf{w}}^2} \|\mathbf{r} - \mathbf{H}\mathbf{d}\|^2}} \right). \quad (4.3)$$

Here, $\mathcal{D}_{m,i}^b$ denotes the set of all possible transmit data vectors \mathbf{d} whose label at layer m and bit position i equals $b \in \{0, 1\}$. That is, $\mathcal{D}_{m,i}^b$ consists of all data vectors \mathbf{d} with $d_m \in \mathcal{A}_i^b$ and $d_{m'} \in \mathcal{A}$, $m' \neq m$, i.e.

$$\mathcal{D}_{m,i}^b \triangleq \left\{ \mathbf{d} \mid d_m \in \mathcal{A}_i^b, d_{m'} \in \mathcal{A}, m' \neq m \right\}, \quad (4.4)$$

where the *reduced alphabet* $\mathcal{A}_i^b \subset \mathcal{A}$ is obtained by retaining only those symbols of alphabet \mathcal{A} whose bit label at position i equals b . A significant complexity reduction of evaluating the exact LLR expression (4.3) can be obtained by applying the log-max approximation (see (3.20)). We obtain

$$\Lambda_{m,i} \approx \frac{1}{\sigma_w^2} [\lambda_{m,i}^0 - \lambda_{m,i}^1], \quad \lambda_{m,i}^b \triangleq \min_{\mathbf{d} \in \mathcal{D}_{m,i}^b} \psi_{\text{ML}}^2(\mathbf{d}), \quad (4.5)$$

with $\psi_{\text{ML}}^2(\mathbf{d}) = \|\mathbf{r} - \mathbf{H}\mathbf{d}\|^2$. This can be expected to yield just a very small performance degradation compared to (4.3) [83, 94]. With (4.5), the LLR is expressed as a difference of minimized ML distances $\psi_{\text{ML}}^2(\mathbf{d})$. The first term is $\psi_{\text{ML}}^2(\mathbf{d})$ minimized over all possible data vectors with $d_m \in \mathcal{A}_i^0$; the second term is $\psi_{\text{ML}}^2(\mathbf{d})$ minimized over all possible data vectors with $d_m \in \mathcal{A}_i^1$. If \mathbf{d} with $d_m \in \mathcal{A}_i^0$ achieves a smaller $\psi_{\text{ML}}^2(\mathbf{d})$ than \mathbf{d} with $d_m \in \mathcal{A}_i^1$, $\Lambda_{m,i}$ is negative, which expresses the fact that $b_{m,i}$ is more likely to be zero than one. If, however, \mathbf{d} with $d_m \in \mathcal{A}_i^1$ achieves a smaller $\psi_{\text{ML}}^2(\mathbf{d})$, $\Lambda_{m,i}$ is positive, and $b_{m,i}$ is more likely to be one than zero.

The computational complexity of the soft-output detector that computes (4.5) still behaves exponentially in M and can be excessive for practical values of $|\mathcal{A}|$ and M [80]. Thus, various efficient MIMO-BICM soft-output detection algorithms providing approximate LLRs have been proposed, such as the list extension of the Fincke-Phost sphere decoding (LFPSD) algorithm [80], other variants based on sphere decoding [79, 95, 96], and algorithms based on ZF or MMSE equalization [92, 97].

4.2 MMSE- and ZF-Based Soft-Output Detection

In the following, we re-derive [17] the MMSE-based and the ZF-based soft-output detectors by means of a Gaussian approximation for the post-equalization interference (as it was also employed to derive the DNC in Chapter 3). The MMSE soft-output detector can be found in [91, 93, 97], and the ZF-based soft-output detector was proposed in [92]. Both detectors are now obtained under the common framework of a Gaussian approximation for the post-equalization interference and in contrast to [91–93, 97] we do not a-priori assume any specific equalizer structure. We will show (similar to the relation between DNC and LLR-based NC, cf. Section 3.6) that the MMSE soft-output detector can be seen as an extension of the ZF soft-output detector to correlated post-equalization interference, which explains its significant performance advantage. Both algorithms exhibit a similar structure that consists of an equalization step (ZF or MMSE, respectively) followed by a *per-layer* LLR calculation. This results in a significantly reduced computational complexity. In particular, the complexity of the equalization-based detectors is just a fraction of that required by the LFPSD. However, the resulting performance loss is substantial (this will be demonstrated in Section 4.4).

4.2.1 Gaussian Approximation

We start by reformulating¹ the LLR in terms of $\mathbf{y}_{\text{ZF}} = \mathbf{G}_{\text{ZF}}\mathbf{r}$ instead of \mathbf{r} . With (1.12) and $\mathbf{R}_{\tilde{\mathbf{w}}} = \sigma_{\tilde{\mathbf{w}}}^2(\mathbf{H}^H\mathbf{H})^{-1}$, and we can express the LLR (4.3) as

$$\begin{aligned}\Lambda_{m,i} &= \log \left(\frac{\sum_{\mathbf{d}: d_m \in \mathcal{A}_i^1} e^{-(\mathbf{y}_{\text{ZF}} - \mathbf{d})^H \mathbf{R}_{\tilde{\mathbf{w}}}^{-1} (\mathbf{y}_{\text{ZF}} - \mathbf{d})}}{\sum_{\mathbf{d}: d_m \in \mathcal{A}_i^0} e^{-(\mathbf{y}_{\text{ZF}} - \mathbf{d})^H \mathbf{R}_{\tilde{\mathbf{w}}}^{-1} (\mathbf{y}_{\text{ZF}} - \mathbf{d})}} \right) \\ &= \log \left(\frac{f(\mathbf{y}_{\text{ZF}} | b_m^{(i)} = 1)}{f(\mathbf{y}_{\text{ZF}} | b_m^{(i)} = 0)} \right).\end{aligned}$$

Inserting

$$f(\mathbf{y}_{\text{ZF}} | b_m^{(i)} = b) = \frac{1}{|\mathcal{A}_i^b|} \sum_{a \in \mathcal{A}_i^b} f(\mathbf{y}_{\text{ZF}} | d_m = a)$$

and using the log-max approximation (3.20), we obtain further

$$\Lambda_{m,i} \approx \max_{a \in \mathcal{A}_i^1} \log f(\mathbf{y}_{\text{ZF}} | d_m = a) - \max_{a \in \mathcal{A}_i^0} \log f(\mathbf{y}_{\text{ZF}} | d_m = a). \quad (4.6)$$

As it was done in the derivation of the DNC (cf. Section 3.1.2), we now apply a Gaussian approximation for the post-equalization interference. This yields a computationally efficient expression for (4.6). That is, $f(\mathbf{y}_{\text{ZF}} | d_m = a)$ is approximated by a Gaussian pdf $\tilde{f}_m(\mathbf{y}_{\text{ZF}} | d_m = a)$ with mean $\boldsymbol{\mu}_m$ and covariance matrix \mathbf{C}_m (cf. (3.6)). Thus, the LLR in (4.6) is approximated according to

$$\Lambda_{m,i} \approx \tilde{\Lambda}_{m,i} \triangleq \max_{a \in \mathcal{A}_i^1} \log \tilde{f}_m(\mathbf{y}_{\text{ZF}} | d_m = a) - \max_{a \in \mathcal{A}_i^0} \log \tilde{f}_m(\mathbf{y}_{\text{ZF}} | d_m = a). \quad (4.7)$$

4.2.2 MMSE-Based Soft-Output Detection

For MMSE-based soft-output detection the mean $\boldsymbol{\mu}_m$ and the covariance matrix \mathbf{C}_m of $\tilde{f}_m(\mathbf{y}_{\text{ZF}} | d_m = a)$ (cf. (3.6)) are both matched to the mean and the covariance matrix of the post-equalization interference, i.e. $\boldsymbol{\mu}_m = a \mathbf{e}_m$ and $\mathbf{C}_m = \mathbf{I} - \mathbf{e}_m \mathbf{e}_m^T + \mathbf{R}_{\tilde{\mathbf{w}}}$ (see (3.7)).

Using these results for $\boldsymbol{\mu}_m$ and \mathbf{C}_m and inserting (3.6) into (4.7), we obtain the MMSE-based soft-output detector as

$$\tilde{\Lambda}_{\text{MMSE},m,i} \triangleq \min_{a \in \mathcal{A}_i^0} q_m(a) - \min_{a \in \mathcal{A}_i^1} q_m(a), \quad (4.8)$$

where

¹The following derivation could also be performed without going into the ZF domain; however, in that case the calculations would be more involved.

$$\begin{aligned}
q_m(a) &\triangleq (\mathbf{y}_{\text{ZF}} - a \mathbf{e}_m)^H \mathbf{C}_m^{-1} (\mathbf{y}_{\text{ZF}} - a \mathbf{e}_m) \\
&= \mathbf{y}_{\text{ZF}}^H \mathbf{C}_m^{-1} \mathbf{y}_{\text{ZF}} - 2 \operatorname{Re} \{ \mathbf{y}_{\text{ZF}}^H \mathbf{C}_m^{-1} \mathbf{e}_m a \} + |a|^2 \mathbf{e}_m^T \mathbf{C}_m^{-1} \mathbf{e}_m.
\end{aligned} \tag{4.9}$$

The term $\mathbf{y}_{\text{ZF}}^H \mathbf{C}_m^{-1} \mathbf{y}_{\text{ZF}}$ does not depend on a and thus can be disregarded in (4.8). By using (3.13) and (3.14) for the terms $\mathbf{y}_{\text{ZF}}^H \mathbf{C}_m^{-1} \mathbf{e}_m$ and $\mathbf{e}_m^T \mathbf{C}_m^{-1} \mathbf{e}_m$, respectively, and by applying the identity

$$\operatorname{SNR}_{\text{MMSE},m} = \frac{W_{m,m}}{1 - W_{m,m}}$$

for the MMSE post-equalization SNR (cf. (3.24)), it can be easily verified that (4.8) simplifies to

$$\tilde{\Lambda}_{\text{MMSE},m,i} = \operatorname{SNR}_{\text{MMSE},m} \left[\min_{a \in \mathcal{A}_i^0} \eta_m^2(a) - \min_{a \in \mathcal{A}_i^1} \eta_m^2(a) \right], \tag{4.10}$$

where $\eta_m(a)$ refers to the unbiased distance

$$\eta_m(a) = \left| \frac{y_{\text{MMSE},m}}{W_{m,m}} - a \right|$$

as in (3.17).

4.2.3 ZF-Based Soft-Output Detection

For ZF-based soft-output detection, we again use $\boldsymbol{\mu}_m = a \mathbf{e}_m$ for $\tilde{f}_m(\mathbf{y}_{\text{ZF}} | d_m = a)$, but all correlations in \mathbf{y}_{ZF} are neglected, i.e. \mathbf{C}_m is replaced with $\tilde{\mathbf{C}}_m$ given by (3.37), where all nondiagonal elements of $\mathbf{C}_m = \mathbf{I} - \mathbf{e}_m \mathbf{e}_m^T + \mathbf{R}_{\tilde{\mathbf{w}}}$ are set to zero.

Thus, the quadratic form $q_m(a)$ in (4.8) is now replaced with $\tilde{q}_m(a)$, i.e.

$$\tilde{\Lambda}_{\text{ZF},m,i} \triangleq \min_{a \in \mathcal{A}_i^0} \tilde{q}_m(a) - \min_{a \in \mathcal{A}_i^1} \tilde{q}_m(a), \tag{4.11}$$

where

$$\begin{aligned}
\tilde{q}_m(a) &\triangleq (\mathbf{y}_{\text{ZF}} - a \mathbf{e}_m)^H \tilde{\mathbf{C}}_m^{-1} (\mathbf{y}_{\text{ZF}} - a \mathbf{e}_m) \\
&= \mathbf{y}_{\text{ZF}}^H \tilde{\mathbf{C}}_m^{-1} \mathbf{y}_{\text{ZF}} - 2 \operatorname{Re} \{ \mathbf{y}_{\text{ZF}}^H \tilde{\mathbf{C}}_m^{-1} \mathbf{e}_m a \} + |a|^2 \mathbf{e}_m^T \tilde{\mathbf{C}}_m^{-1} \mathbf{e}_m.
\end{aligned}$$

By simplifying this expression and inserting into (4.11), we obtain the ZF-based soft-output detector as

$$\tilde{\Lambda}_{\text{ZF},m,i} = \operatorname{SNR}_{\text{ZF},m} \left[\min_{a \in \mathcal{A}_i^0} \gamma_m^2(a) - \min_{a \in \mathcal{A}_i^1} \gamma_m^2(a) \right]. \tag{4.12}$$

Here, $\gamma_m(a) = |y_{\text{ZF},m} - a|$ and $\operatorname{SNR}_{\text{ZF},m}$ denotes the ZF post-equalization SNR (1.17).

4.2.4 Computational Complexity and Discussion

The final expressions for MMSE-based (4.10) and ZF-based (4.12) soft-output detection, together with the expressions for $\eta_m(a)$ and $\gamma_m(a)$ in terms of $y_{\text{MMSE},m}$ and $y_{\text{ZF},m}$, respectively, show that

these detectors consist of an equalization step and subsequent *per-layer* LLR calculation, i.e. after equalization each layer is processed separately. The preparation complexity C_{prep} is dominated by the calculation of \mathbf{G}_{ZF} or \mathbf{G}_{MMSE} , respectively, with complexity $\mathcal{O}(M^3)$ (again assuming $M = N$). The vector complexity C_{vector} is mainly composed of the equalization step with complexity $\mathcal{O}(M^2)$.

Evidently, the structure of these soft-output detectors is very similar to that of the log-max soft-output detector in (4.5). All three detectors compute a difference of two distances, where one distance corresponds to the respective bit being 0 and the other corresponds to that bit being 1. However, the distances and the pre-factors in the corresponding approximate LLR expressions are defined differently.

Finally, our derivation shows that the MMSE-based soft-output detector can be seen as an extension of the ZF-based soft-output detector to correlated post-equalization interference. This provides an explanation of the significant performance advantage of MMSE-based over ZF-based soft-output detection (this will be demonstrated in Section 4.4). Note that the noise variance $\sigma_{\mathbf{w}}^2$ can be ignored in the pre-factors of the log-max (4.5) and ZF-based (4.12) soft-output detector. For these two detectors, $\sigma_{\mathbf{w}}^2$ is an irrelevant scaling factor that has no influence on the performance of the subsequent soft-in Viterbi decoder (that maximizes the sum over all LLRs). For the MMSE-based soft-output detector, however, $\sigma_{\mathbf{w}}^2$ is more than just a scaling factor (e.g., $\sigma_{\mathbf{w}}^2$ is also used in the calculation of the MMSE equalizer matrix). Consequently, the ZF and log-max detectors have the advantage over MMSE detection that knowledge of the noise variance is not required.

4.3 Soft-Output SPA

In Chapter 2 we proposed (hard-output) geometry-based data detection algorithms for SM systems. In particular, the SPA (based on constant modulus alphabets) was shown to be a very attractive detection algorithm (see Section 2.5). For SM systems of moderate size, it can achieve near-ML performance at low computational cost. We now extend the SPA to provide soft outputs (i.e. LLRs) instead of hard decisions. The resulting method we refer to as the *soft-SPA* (SSPA) [26]. It is specifically tailored to PSK alphabets employing *Gray labeling*. This labeling is most commonly used in practice and optimum for BICM systems [84]. If one considers BICM systems employing iterative detection and decoding, other labeling strategies (e.g., set partitioning labeling) can achieve better performance results [98]. The soft version of the LSD (see Section 2.4) can be found in [28]. As for the SPA, we again focus on SM systems (cf. (1.5.1)), where $M = M_{\text{T}}$, and $N = M_{\text{R}}$.

4.3.1 Basic Idea of SSPA

First of all, max-log soft-output detection (4.5) is closely related to the ML detector (cf. (1.5))

$$\hat{\mathbf{d}}_{\text{ML}} = \arg \min_{\mathbf{d} \in \mathcal{D}} \|\mathbf{r} - \mathbf{H}\mathbf{d}\|^2 = \arg \min_{\mathbf{d} \in \mathcal{D}} \psi_{\text{ML}}^2(\mathbf{d}). \quad (4.13)$$

Denoting the i th bit of the m th component of $\hat{\mathbf{d}}_{\text{ML}}$ as $(\hat{\mathbf{d}}_{\text{ML}})_{m,i} = b$, it follows that $\hat{\mathbf{d}}_{\text{ML}} \in \mathcal{D}_{m,i}^b$ and hence we immediately obtain

$$\lambda_{m,i}^b = \psi_{\text{ML}}^2(\hat{\mathbf{d}}_{\text{ML}}). \quad (4.14)$$

However, the computation of $\hat{\mathbf{d}}_{\text{ML}}$ in (4.13) is computationally very intensive and one must still compute the term $\lambda_{m,i}^{\bar{b}}$ in (4.5) ($\bar{b} = 1-b$ denotes bit flipping).

The idea behind the SSPA is as follows. As discussed in detail in Section 2.5, the SPA is an efficient approximation to the ML detector. It replaces \mathcal{D} with a reduced search set \mathcal{D}_{SP} to obtain a hard decision

$$\hat{\mathbf{d}}_{\text{SP}} = \arg \min_{\mathbf{d} \in \mathcal{D}_{\text{SP}}} \psi_{\text{ML}}^2(\mathbf{d}) \quad (4.15)$$

such that

$$\psi_{\text{ML}}^2(\hat{\mathbf{d}}_{\text{SP}}) \approx \psi_{\text{ML}}^2(\hat{\mathbf{d}}_{\text{ML}}).$$

This together with (4.14) yields the approximation

$$\lambda_{m,i}^b \approx \tilde{\lambda}_{m,i}^b \triangleq \psi_{\text{ML}}^2(\hat{\mathbf{d}}_{\text{SP}}), \quad \text{for } b = (\hat{\mathbf{d}}_{\text{SP}})_{m,i}. \quad (4.16)$$

It remains to calculate a similar approximation for $\lambda_{m,i}^{\bar{b}}$. A comparison of (4.5) and (4.14) reveals that such an approximation could in principle be obtained by applying the SPA with $\mathcal{D}_{m,i}^{\bar{b}}$ instead of \mathcal{D} , i.e.,

$$\lambda_{m,i}^{\bar{b}} \approx \tilde{\lambda}_{m,i}^{\bar{b}} \triangleq \arg \min_{\mathbf{d} \in \mathcal{D}_{\text{SP},m,i}^{\bar{b}}} \psi_{\text{ML}}^2(\mathbf{d}) \quad (4.17)$$

Here, $\mathcal{D}_{\text{SP},m,i}^{\bar{b}}$ denotes the SPA reduced search set corresponding to $\mathcal{D}_{m,i}^{\bar{b}}$, i.e. the SPA is performed with the reduced alphabet $\mathcal{A}_i^{\bar{b}}$ instead of \mathcal{A} for layer m . In general, computing (4.17) requires a second SPA pass. However, by employing PSK alphabets with Gray labeling we will show that such a second SPA pass can be circumvented and $\tilde{\lambda}_{m,i}^{\bar{b}}$ can be calculated with moderate additional effort using intermediate results of the first SPA pass that is used to calculate (4.16). In fact, all vectors $\tilde{\mathbf{d}} \in \mathcal{D}_{\text{SP},m,i}^{\bar{b}}$ can be obtained by a *re-quantization* to $\mathcal{A}_i^{\bar{b}}$ of the vectors $\mathbf{d} \in \mathcal{D}_{\text{SP}}$ and the associated distances $\psi_{\text{ML}}^2(\tilde{\mathbf{d}})$ can be computed via simple updates of $\psi_{\text{ML}}^2(\mathbf{d})$, $\mathbf{d} \in \mathcal{D}_{\text{SP}}$.

4.3.2 Re-Quantization Property of PSK with Gray Labeling

The efficient implementation of the SSPA is based on the following fundamental *re-quantization property*. Here, the quantization of $y \in \mathbb{C}$ with respect to the reduced alphabet $\mathcal{A}_i^{\bar{b}}$, denoted as

$$\mathcal{Q}_{\mathcal{A}_i^{\bar{b}}}\{y\} = \arg \min_{a \in \mathcal{A}_i^{\bar{b}}} |y - a|,$$

remains unchanged when it is preceded by the quantization with respect to \mathcal{A} , i.e.,

$$\mathcal{Q}_{\mathcal{A}_i^{\bar{b}}}\{y\} = \mathcal{Q}_{\mathcal{A}_i^{\bar{b}}}\{\mathcal{Q}_{\mathcal{A}}\{y\}\}. \quad (4.18)$$

This means, *the decision boundaries for $\mathcal{A}_i^{\bar{b}}$ form a subset of the boundaries for \mathcal{A}* . Thus, any quantization with respect to $\mathcal{A}_i^{\bar{b}}$ can be performed using \mathcal{A} -quantized values.

Evidently, this property depends on the alphabet and on the labeling employed. In the following we show that PSK alphabets whose size is a power of two (including, e.g., BPSK, 4-QAM (QPSK), and 8-PSK alphabets) with Gray labeling have this re-quantization property. At first, we discuss that

the data symbols in the reduced alphabet $\mathcal{A}_i^{\bar{b}} \subset \mathcal{A}$ have to satisfy a certain angular property. In fact, the angular separation of the symbols in $\mathcal{A}_i^{\bar{b}}$ has to be an odd multiple of the basic angular separation $\Delta\theta \triangleq 2\pi/|\mathcal{A}|$ of the data symbols in \mathcal{A} . We subsequently show that Gray labeling yields this angular property of the data symbols in $\mathcal{A}_i^{\bar{b}}$.

Re-Quantization Property of PSK: Angular Property

In the following we assume that \mathcal{A} is a PSK alphabet with size $|\mathcal{A}| \geq 4$ that is a power of two. If $|\mathcal{A}| = 2$, i.e. \mathcal{A} is a BPSK alphabet, re-quantization property (4.18) holds since the corresponding reduced alphabets \mathcal{A}_0^0 and \mathcal{A}_0^1 have no boundary lines at all (since the decision region is \mathbb{C} in each case) and thus forms a trivial subset of the boundary line for \mathcal{A} . We now represent the PSK data symbols $a_t = e^{j\theta_t} \in \mathcal{A}$, $t = 0, \dots, |\mathcal{A}| - 1$, by their associated angles θ_t (up to arbitrary multiples of 2π), which we describe recursively using

$$\theta_{(t+1) \bmod |\mathcal{A}|} = \theta_t + \Delta\theta, \quad (4.19)$$

where $(\cdot) \bmod |\mathcal{A}|$ denotes the modulo operation with respect to $|\mathcal{A}|$, i.e. we have $\theta_1 = \theta_0 + \Delta\theta$, $\theta_2 = \theta_1 + \Delta\theta$, up to $\theta_0 = \theta_{|\mathcal{A}|-1} + \Delta\theta$. Thus, the symbols $a_t \in \mathcal{A}$, $t = 0, \dots, |\mathcal{A}| - 1$, are sorted in a counter-clockwise manner on the unit circle and two subsequent symbols (in a cyclic manner, i.e. including the last symbol $a_{|\mathcal{A}|-1}$ and the first symbols a_0) are nearest neighbor symbols that are separated by the angle $\Delta\theta$. Up to a fixed angular offset, this representation of the data symbols $a_t \in \mathcal{A}$ is unique. In analogy to (4.19), the angles $\tilde{\theta}_l$ (up to arbitrary multiples of 2π) of the data symbols $\tilde{a}_l = e^{j\tilde{\theta}_l} \in \mathcal{A}_i^{\bar{b}} \subset \mathcal{A}$, $l = 0, \dots, |\mathcal{A}|/2 - 1$, can be written as

$$\tilde{\theta}_{(l+1) \bmod (|\mathcal{A}|/2)} = \tilde{\theta}_l + t_l \Delta\theta, \quad (4.20)$$

where t_l are some integers from the set $\{1, \dots, |\mathcal{A}| - 1\}$. Thus, the symbols \tilde{a}_l , $l = 0, \dots, |\mathcal{A}|/2 - 1$, of the reduced alphabet $\mathcal{A}_i^{\bar{b}}$ are also sorted in a counter-clockwise manner on the unit circle, but now with an angular separation that is a t_l -multiple of $\Delta\theta$. For \mathcal{A} and $\mathcal{A}_i^{\bar{b}}$ the boundaries of the symbol decision regions are straight line segments whose angles are determined by the angular mean of two subsequent (with respect to (4.19) and (4.20), respectively) data symbols. The angles of these boundary lines are thus given by

$$\chi_t \triangleq \theta_t + \frac{\Delta\theta}{2} \quad \text{and} \quad \tilde{\chi}_l \triangleq \tilde{\theta}_l + t_l \frac{\Delta\theta}{2}, \quad (4.21)$$

for \mathcal{A} and $\mathcal{A}_i^{\bar{b}}$, respectively. For a labeling with the re-quantization property (i.e. the decision boundaries for $\mathcal{A}_i^{\bar{b}}$ form a subset of the boundaries for \mathcal{A}), we must have that each $\tilde{\chi}_l$ is equal to χ_t for some t . Thus, for each $\tilde{\theta}_l$ and t_l we must have

$$\tilde{\theta}_l + t_l \frac{\Delta\theta}{2} = \theta_t + \frac{\Delta\theta}{2} \quad (4.22)$$

for some θ_t . Since $\theta_t - \tilde{\theta}_l$ can be any n -multiple of $\Delta\theta$ with $n \in \mathbb{N}$, we obtain

$$t_l = 2n + 1.$$

It follows that t_l has to be one of the *odd* integer numbers from the set $\{1, \dots, |\mathcal{A}| - 1\}$. If t_l is *even* for some l , the decision boundaries for $\mathcal{A}_i^{\bar{b}}$ do *not* form a subset of the boundaries for \mathcal{A} . In view of (4.20), any labeling of a PSK alphabet that allows for re-quantizations, yields reduced alphabets $\mathcal{A}_i^{\bar{b}}$, where the angles of two subsequent data symbols are separated by an *odd-multiple* of $\Delta\theta$.

Re-Quantization Property of PSK: Gray Labeling

We now show that the angular property for the data symbols $\tilde{a}_l \in \mathcal{A}_i^{\bar{b}}$ holds if Gray labeling is employed. Let $\mathbf{b}_t \triangleq (b_{t,0} \cdots b_{t,\log_2|\mathcal{A}|-1})^T$, $b_{t,i} \in \{0, 1\}$, denote the label associated with data symbol $a_t \in \mathcal{A}$. With Gray labeling, by definition, the labels of two nearest neighbor symbols differ in only one of the $\log_2|\mathcal{A}|$ bit positions [99]. That is, subsequent labels \mathbf{b}_t , $t = 0, \dots, |\mathcal{A}| - 1$, differ only in one bit position. Since we consider PSK alphabets with Gray labeling, this also includes the first label \mathbf{b}_0 and the last label $\mathbf{b}_{|\mathcal{A}|-1}$ and thus, this is often referred to as a *binary cyclic Gray labeling*. By labeling expansion it can be constructed recursively [99]. In fact, the cyclic Gray labels \mathbf{b}_t , $t = 0, \dots, |\mathcal{A}| - 1$, can be constructed based on the cyclic Gray labels \mathbf{b}'_s , $s = 0, \dots, |\mathcal{A}'| - 1$, of data symbols $a'_s \in \mathcal{A}'$, where the alphabet \mathcal{A}' has half the size of \mathcal{A} , i.e. $|\mathcal{A}| = 2|\mathcal{A}'|$. In the simplest case \mathcal{A} is a QPSK (4-QAM) alphabet and \mathcal{A}' is a BPSK alphabet. By labeling expansion, the labels \mathbf{b}_t for the data symbols $a_t \in \mathcal{A}$ are obtained by repeating each label \mathbf{b}'_s of $a'_s \in \mathcal{A}'$ and by augmenting a bit sequence that consists of alternating $[0, 1]$ and $[1, 0]$ blocks. We have

$$\left[\mathbf{b}_0 \mathbf{b}_1 \mid \mathbf{b}_2 \mathbf{b}_3 \mid \mathbf{b}_4 \mathbf{b}_5 \mid \cdots \mid \mathbf{b}_{|\mathcal{A}|-2} \mathbf{b}_{|\mathcal{A}|-1} \right] = \left[\begin{array}{cc|cc|cc| \cdots | cc} \mathbf{b}'_0 & \mathbf{b}'_0 & \mathbf{b}'_1 & \mathbf{b}'_1 & \mathbf{b}'_2 & \mathbf{b}'_2 & \cdots & \mathbf{b}'_{|\mathcal{A}'|-1} & \mathbf{b}'_{|\mathcal{A}'|-1} \\ 0 & 1 & 1 & 0 & 0 & 1 & \cdots & 1 & 0 \end{array} \right]. \quad (4.23)$$

Obviously, cyclic Gray labels \mathbf{b}'_s result in cyclic Gray labels \mathbf{b}_t . For example, a cyclic Gray labeling for a QPSK (4-QAM) constellation \mathcal{A} can be obtained via labeling expansion of the (trivial) Gray labeling of a BPSK constellation \mathcal{A}' , e.g., with $\mathbf{b}'_0 = 0$ and $\mathbf{b}'_1 = 1$ one obtains

$$\left[\mathbf{b}_0 \mathbf{b}_1 \mid \mathbf{b}_2 \mathbf{b}_3 \right] = \left[\begin{array}{cc|cc} \mathbf{b}'_0 & \mathbf{b}'_0 & \mathbf{b}'_1 & \mathbf{b}'_1 \\ 0 & 1 & 1 & 0 \end{array} \right] = \left[\begin{array}{cc|cc} 0 & 0 & 1 & 1 \\ 0 & 1 & 1 & 0 \end{array} \right]. \quad (4.24)$$

Cyclic Gray labeling also guarantees that the data symbols $\tilde{a}_l \in \mathcal{A}_i^{\bar{b}}$ satisfy the angular property. By looking at the structure of the labeling expansion (4.23), this can be easily shown as follows. In general, the labels for the data symbols $\tilde{a}_l \in \mathcal{A}_i^{\bar{b}}$ are given by certain $|\mathcal{A}|/2$ columns of $[\mathbf{b}_0 \mathbf{b}_1 \dots \mathbf{b}_{|\mathcal{A}|-1}]$. The remaining columns are the labels for the data symbols in the complementary set \mathcal{A}_i^b , where $b = 1 - \bar{b}$. We require that two subsequent data symbols $\tilde{a}_l \in \mathcal{A}_i^{\bar{b}}$ are separated by an odd-multiple of $\Delta\theta$ (see previous section). Equivalently, the corresponding subsequent labels for $\tilde{a}_l \in \mathcal{A}_i^{\bar{b}}$ are given by columns of $[\mathbf{b}_0 \mathbf{b}_1 \dots \mathbf{b}_{|\mathcal{A}|-1}]$ which are separated by an *even* number of labels for the complementary set \mathcal{A}_i^b . From the right hand side of (4.23) we observe that always *neighboring pairs* of labels provide the labels for the data symbols in $\mathcal{A}_i^{\bar{b}}$ and \mathcal{A}_i^b , respectively. It follows that two subsequent data symbols in $\tilde{a}_l \in \mathcal{A}_i^{\bar{b}}$ are always separated by an odd-multiple of $\Delta\theta$. Hence, by starting with a BPSK constellation, cyclic Gray labels for higher-order PSK alphabets are constructed via labeling expansion and it is always guaranteed that the re-quantization property (4.18) holds.

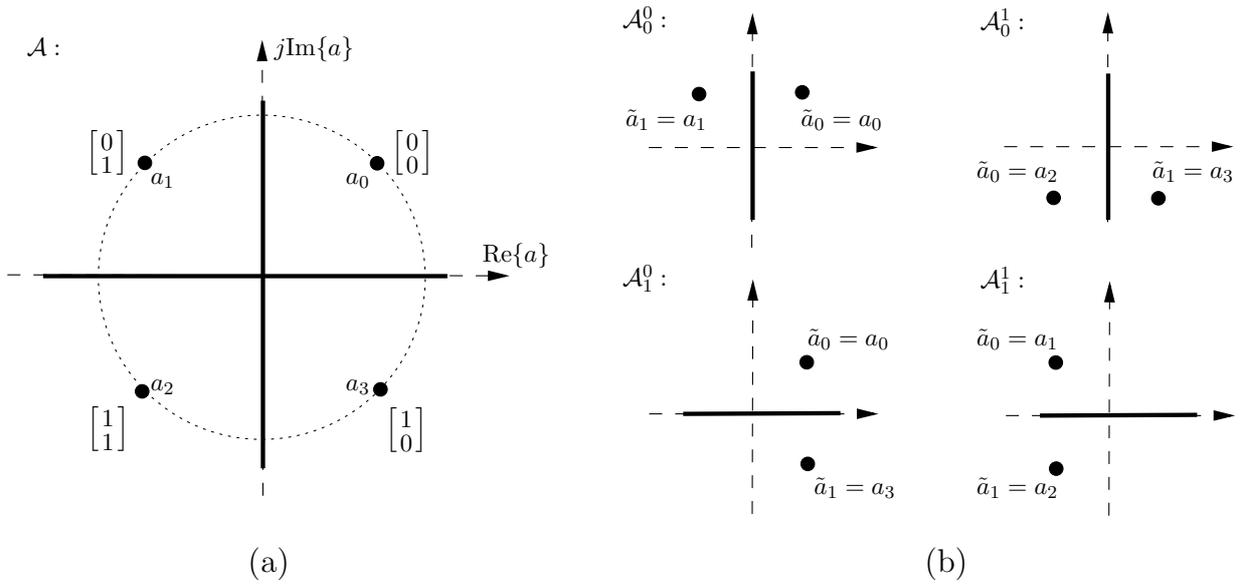


Figure 4.2: 4-QAM alphabet $\mathcal{A} = \frac{1}{\sqrt{2}} \{1 + j, -1 + j, -1 - j, 1 - j\}$ with Gray labeling. (a) \mathcal{A} with the corresponding labels, (b) reduced alphabets $\mathcal{A}_i^{\bar{b}} = \{\tilde{a}_0, \tilde{a}_1\} \subset \mathcal{A}$, for $i = 0, 1$, $\bar{b} = 0, 1$. The bold line segments are the corresponding boundaries of the symbol decision regions. Evidently, the boundaries for $\mathcal{A}_i^{\bar{b}}$ form a subset of the boundaries for \mathcal{A} .

Examples

4-QAM Alphabet. The four data symbols a_t , $t = 0, \dots, 3$, of the 4-QAM constellation $\mathcal{A} = 1/\sqrt{2} \{1 + j, -1 + j, -1 - j, 1 - j\}$ are illustrated in Figure 4.2(a). Up to a fixed angular offset, the data symbols can be described by (4.19) using $\Delta\theta = \pi/2$. The boundary lines of the symbol decision regions are given by the horizontal and vertical axis (indicated by the bold line segments). The four angles of the associated boundary line segments are given by (cf. (4.21))

$$\chi_0 = \frac{\pi}{2}, \quad \chi_1 = \pi, \quad \chi_2 = \frac{3\pi}{2}, \quad \chi_3 = 2\pi.$$

The Gray labeling of the respective symbols is also provided in Figure 4.2(a). It can be obtained using labeling expansion of the Gray labeling employed for BPSK modulation (cf. (4.24)). Figure 4.2(b) shows the four corresponding reduced symbol alphabets, namely, \mathcal{A}_0^0 , \mathcal{A}_0^1 , \mathcal{A}_1^0 , and \mathcal{A}_1^1 . For all reduced alphabets, the boundaries of the decision regions form a subset (either the horizontal or vertical axis) of the boundaries for \mathcal{A} . Evidently, a 4-QAM constellation with Gray labeling has the re-quantization property (4.18).

8-PSK Alphabet. The 8-PSK constellation is illustrated in Figure 4.3(a). Up to a fixed angular offset, this alphabet can be described by (4.19) using $\Delta\theta = \pi/4$. The boundaries of the symbol decision regions are indicated by the bold line segments whose angles can be obtained via (4.21). The Gray labeling of the respective symbols (also shown in Figure 4.3(a)) can be obtained using labeling expansion of the 4-QAM Gray labeling (see previous example). Figure 4.2(b) shows one corresponding

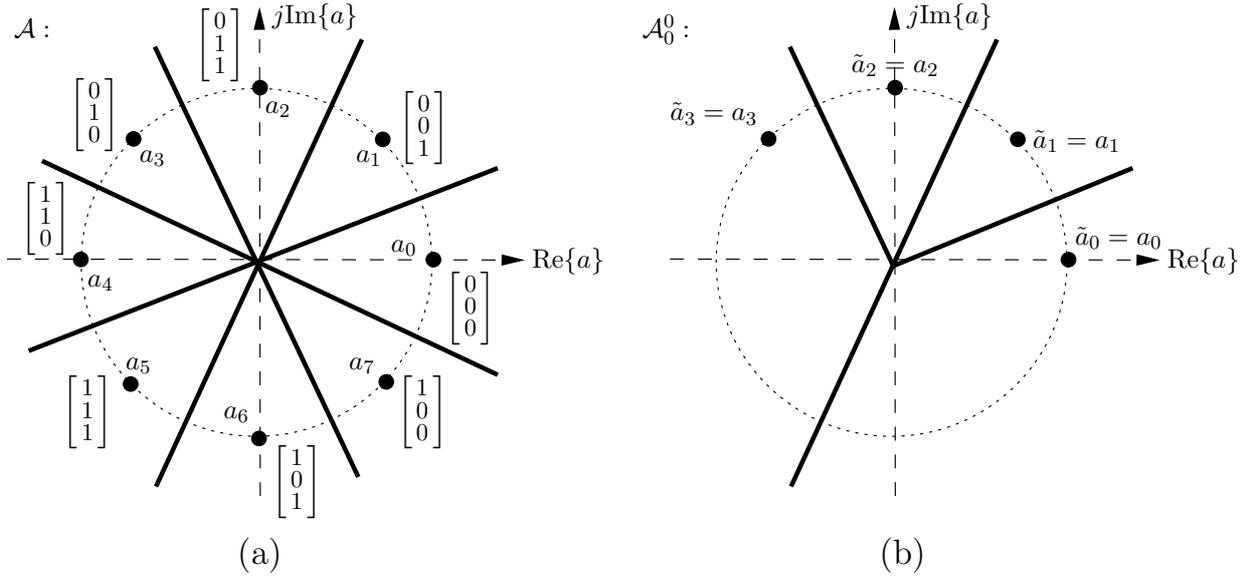


Figure 4.3: 8-PSK alphabet \mathcal{A} with Gray labeling. (a) \mathcal{A} with the corresponding labels, (b) reduced alphabet $\mathcal{A}_0^0 \subset \mathcal{A}$. The bold line segments are the corresponding boundaries of the symbol decision regions. The boundaries for \mathcal{A}_0^0 form a subset of the boundaries for \mathcal{A} .

reduced symbol alphabet, namely \mathcal{A}_0^0 , and its associated boundary lines. Again, the re-quantization property (4.18) holds.

4.3.3 SSPA Details

In the following we will discuss the efficient calculation of $\lambda_{m,i}^{\bar{b}}$ in (4.17) based on the reduced set \mathcal{D}_{SP} and the associated distance set

$$\Psi_{\text{SP}} \triangleq \{\psi_{\text{ML}}^2(\mathbf{d}), \mathbf{d} \in \mathcal{D}_{\text{SP}}\}. \quad (4.25)$$

The sets \mathcal{D}_{SP} and Ψ_{SP} are both obtained via the SPA. Recall that $\mathcal{D}_{m,i}^{\bar{b}}$ consists of all data vectors \mathbf{d} with $d_m \in \mathcal{A}_i^{\bar{b}}$ and $d_{m'} \in \mathcal{A}$, $m' \neq m$ (cf. (4.4)). Thus, when determining the reduced search set $\mathcal{D}_{\text{SP},m,i}^{\bar{b}}$ for (4.17), the main difference to the initial SPA pass is that quantization and nearest neighbor search for layer m are limited to the reduced alphabet $\mathcal{A}_i^{\bar{b}}$. The efficient implementation of the SSPA is based on the fundamental re-quantization property (4.18) of PSK alphabets with Gray labeling.

The basic operation to obtain a data vector $\tilde{\mathbf{d}} \in \mathcal{D}_{\text{SP},m,i}^{\bar{b}}$ is the re-quantization according to the reduced alphabet $\mathcal{A}_i^{\bar{b}}$ of the m th component of $\mathbf{d} \in \mathcal{D}_{\text{SP}}$ (the other components remain unchanged):

$$\tilde{d}_{m'} = \begin{cases} \mathcal{Q}_{\mathcal{A}_i^{\bar{b}}}\{d_m\}, & \text{if } m' = m, \\ d_{m'}, & \text{if } m' \neq m. \end{cases} \quad (4.26)$$

On the vector level, this re-quantization of the m th component will be denoted as $\tilde{\mathbf{d}} = \mathcal{Q}_{\mathcal{A}_i^{\bar{b}},m}\{\mathbf{d}\}$.

Recall that \mathcal{D}_{SP} consists of the result of a conventional detector $\hat{\mathbf{d}}$ and of an additional search set \mathcal{D}_+ , i.e. $\mathcal{D}_{\text{SP}} = \{\hat{\mathbf{d}}\} \cup \mathcal{D}_+$. For $\mathcal{D}_{\text{SP},m,i}^{\bar{b}}$ we have $\mathcal{D}_{\text{SP},m,i}^{\bar{b}} = \{\hat{\mathbf{d}}_{m,i}^{\bar{b}}\} \cup \mathcal{D}_{+,m,i}^{\bar{b}}$. If $\hat{\mathbf{d}}$ is the result of ZF or

MMSE detection, i.e. $\hat{\mathbf{d}} = \mathcal{Q}_{\mathcal{A}}\{\mathbf{y}\}$, where \mathbf{y} is the corresponding equalized received vector, we have $\hat{\mathbf{d}}_{m,i}^{\bar{b}} = \mathcal{Q}_{\mathcal{A}_i^{\bar{b}},m}\{\hat{\mathbf{d}}\}$ since we use the reduced symbol alphabet $\mathcal{A}_i^{\bar{b}}$ instead of \mathcal{A} for layer m . It remains to determine the additional search set $\mathcal{D}_{+,m,i}^{\bar{b}}$.

Construction of Additional Search Set

Again we distinguish between the two cases of SPA set construction (cf. Section 2.5.1).

Case 1: \mathcal{L} intersects \mathcal{H} . If \mathcal{L} intersects \mathcal{H} , each data vector $\mathbf{d}^{(r)} \in \mathcal{D}_+$ is associated to a specific cell (for the line parameter $\tilde{\alpha}$) that is pierced by the intersection circle \mathcal{I} with radius \tilde{R} (see (2.25)). The intersection circle \mathcal{I} can be described by

$$\mathcal{I}: \tilde{\alpha}_{\mathcal{I}}(\phi) \triangleq \tilde{R} e^{j\phi}, \quad \phi \in [0, 2\pi).$$

The additional search set \mathcal{D}_+ consists of all data vectors that are obtained as $\mathcal{Q}_{\mathcal{A}}\{\mathbf{y}_{\mathcal{L}}(\tilde{\alpha}_{\mathcal{I}}(\phi))\}$ for all possible angles ϕ (recall that $\mathbf{y}_{\mathcal{L}}(\tilde{\alpha}) = \tilde{\alpha} \mathbf{v}_{M_T} + \mathbf{y}_{\mathbf{v}_{M_T}}^{\perp}$). According to Section 2.5.1, the first data vector $\mathbf{d}^{(1)} \in \mathcal{D}_+$ is obtained with $\phi = 0$, i.e. $\mathbf{d}^{(1)} = \mathcal{Q}_{\mathcal{A}}\{\mathbf{y}_{\mathcal{L}}(\tilde{R})\}$. The remaining data vectors are obtained by hopping from one intersection point – between \mathcal{I} and a boundary line $\mathcal{B}^{(m,p)}$ of the symbol decision regions for \mathcal{A} – to the next intersection point that is obtained for increasing ϕ . Equivalently, the boundary lines $\mathcal{B}^{(m,p)}$ partition the angle parameter ϕ into certain intervals $\mathcal{V}^{(r)}$, and each data vector $\mathbf{d}^{(r)} \in \mathcal{D}_+$ is associated to a specific $\mathcal{V}^{(r)}$. We thus have $\mathbf{d}^{(r)} = \mathcal{Q}_{\mathcal{A}}\{\mathbf{y}_{\mathcal{L}}(\tilde{\alpha}_{\mathcal{I}}(\phi))\}$ for any $\phi \in \mathcal{V}^{(r)}$.

For the construction of $\mathcal{D}_{+,m,i}^{\bar{b}}$, we now use $\mathcal{A}_i^{\bar{b}}$ instead of \mathcal{A} for layer m . Thus, in analogy to the construction of \mathcal{D}_+ , each data vector $\tilde{\mathbf{d}}^{(r)} \in \mathcal{D}_{+,m,i}^{\bar{b}}$ is obtained as $\tilde{d}_m^{(r)} = \mathcal{Q}_{\mathcal{A}_i^{\bar{b}}}\{y_{\mathcal{L},m}(\tilde{\alpha}_{\mathcal{I}}(\phi))\}$ and $\tilde{d}_{m'}^{(r)} = \mathcal{Q}_{\mathcal{A}}\{y_{\mathcal{L},m'}(\tilde{\alpha}_{\mathcal{I}}(\phi))\}$, $m' \neq m$, for ϕ in a specific interval $\tilde{\mathcal{V}}^{(r)}$. In general, the intervals $\tilde{\mathcal{V}}^{(r)}$ (obtained with $\mathcal{A}_i^{\bar{b}}$ for layer m and \mathcal{A} for the remaining layers) are completely different from the intervals $\mathcal{V}^{(r)}$ (obtained with \mathcal{A} for all layers). This is due to the fact that the boundary lines associated with $\mathcal{A}_i^{\bar{b}}$ are different from that associated with \mathcal{A} . However, the re-quantization property (4.18) implies that an interval $\tilde{\mathcal{V}}^{(r)}$ either equals an interval $\mathcal{V}^{(r)}$ or is a *union* of adjacent intervals $\mathcal{V}^{(r)}$. The reason is that the boundary lines for $\mathcal{A}_i^{\bar{b}}$ form a subset of the boundary lines for \mathcal{A} , which simply implies that some intersection points with \mathcal{I} are no longer present for the reduced alphabet $\mathcal{A}_i^{\bar{b}}$. It follows that the *entire* set $\mathcal{D}_{+,m,i}^{\bar{b}}$ can be obtained by re-quantizing every data vector $\mathbf{d}^{(r)} \in \mathcal{D}_+$, i.e., $\tilde{\mathbf{d}}^{(r)} = \mathcal{Q}_{\mathcal{A}_i^{\bar{b}},m}\{\mathbf{d}^{(r)}\}$. In that way, data vectors $\tilde{\mathbf{d}}^{(r)}$ associated to an interval $\tilde{\mathcal{V}}^{(r)}$ that is the union of two or more intervals $\mathcal{V}^{(r)}$ will be multiply obtained. We do not take this into account, since the resulting complexity reduction would be insignificant.

Case 2: \mathcal{L} does not intersect \mathcal{H} . If \mathcal{L} and \mathcal{H} do not intersect, \mathcal{D}_+ consists of $\mathbf{d}^{(1)} = \mathcal{Q}_{\mathcal{A}}\{\mathbf{y}_{\mathbf{v}_{M_T}}^{\perp}\}$ and all its nearest-neighbor data vectors. Thus, $\tilde{\mathbf{d}}^{(1)} \in \mathcal{D}_{+,m,i}^{\bar{b}}$ is obtained as $\tilde{\mathbf{d}}^{(1)} = \mathcal{Q}_{\mathcal{A}_i^{\bar{b}},m}\{\mathbf{d}^{(1)}\}$. As for the construction of \mathcal{D}_+ , the remaining data vectors in $\tilde{\mathbf{d}}^{(r)} \in \mathcal{D}_{+,m,i}^{\bar{b}}$ can be simply found by substituting the nearest-neighbor symbols for the individual components of $\tilde{\mathbf{d}}^{(1)}$. In analogy to the Case 1 procedure we propose to construct $\mathcal{D}_{+,m,i}^{\bar{b}}$ by re-quantizing every data vector $\mathbf{d}^{(r)} \in \mathcal{D}_+$, i.e., $\tilde{\mathbf{d}}^{(r)} = \mathcal{Q}_{\mathcal{A}_i^{\bar{b}},m}\{\mathbf{d}^{(r)}\}$. This evidently yields the nearest-neighbor data vectors $\tilde{\mathbf{d}}^{(r)}$ of $\tilde{\mathbf{d}}^{(1)}$ with respect

to the components $m' \neq m$ but possibly misses the two nearest-neighbor data vectors of $\tilde{\mathbf{d}}^{(1)}$ with respect to component m . Note that this procedure is also used if \mathcal{L} does intersect \mathcal{H} but no boundary line intersects the intersection circle \mathcal{I} .

Efficient Calculation of Distances

To calculate $\tilde{\lambda}_{m,i}^{\bar{b}}$ according to (4.17), we need the distance set $\Psi_{\text{SP},m,i}^{\bar{b}}$ that is associated to $\mathcal{D}_{\text{SP},m,i}^{\bar{b}} = \{\hat{\mathbf{d}}_{m,i}^{\bar{b}}\} \cup \mathcal{D}_{+,m,i}^{\bar{b}}$ (cf. (4.25)):

$$\Psi_{\text{SP},m,i}^{\bar{b}} \triangleq \{\psi_{\text{ML}}^2(\mathbf{d}), \mathbf{d} \in \mathcal{D}_{\text{SP},m,i}^{\bar{b}}\}.$$

Similar to the efficient distance updates that were employed for the LSD and the SPA (cf. (2.20) in Section 2.4.1), the distances $\psi_{\text{ML}}^2(\mathbf{d}), \mathbf{d} \in \mathcal{D}_{\text{SP},m,i}^{\bar{b}}$, can be obtained via an update of $\psi_{\text{ML}}^2(\mathbf{d}), \mathbf{d} \in \mathcal{D}_{\text{SP}}$, provided that the vectors $\boldsymbol{\xi}(\mathbf{d}) = \mathbf{r} - \mathbf{H}\mathbf{d}, \mathbf{d} \in \mathcal{D}_{\text{SP}}$, are also stored during the first SPA pass. Consider the re-quantized data vector $\tilde{\mathbf{d}} = \mathcal{Q}_{\mathcal{A}_i^{\bar{b}},m}\{\mathbf{d}\}$ and let $\Delta_{m,i}^{\bar{b}} \triangleq \tilde{d}_m - d_m$ denote the difference between the m th components of $\tilde{\mathbf{d}}$ and \mathbf{d} . Note that the re-quantization operation (4.26) just corresponds to a simple symbol re-mapping of the m th component of \mathbf{d} and is analogous to the update operation in (2.14) employed for the LSD. We then have (cf. (2.20) and (2.21))

$$\psi_{\text{ML}}^2(\tilde{\mathbf{d}}) = \psi_{\text{ML}}^2(\mathbf{d}) + \|\mathbf{h}_m\|^2 |\Delta_{m,i}^{\bar{b}}|^2 - 2 \operatorname{Re}\{\boldsymbol{\xi}^H(\mathbf{d})\mathbf{h}_m \Delta_{m,i}^{\bar{b}}\}. \quad (4.27)$$

4.3.4 Algorithm Summary and Computational Complexity

We now summarize the SSPA and discuss its computational complexity. Note that the SSPA heavily relies on the re-quantization property of PSK alphabets with Gray labeling (see Section 4.3.2).

1. Use the conventional SPA to calculate the reduced search set \mathcal{D}_{SP} , the associated distance set Ψ_{SP} , the detector output $\hat{\mathbf{d}}_{\text{SP}}$, and $\psi_{\text{ML}}^2(\hat{\mathbf{d}}_{\text{SP}})$, the minimum element of Ψ_{SP} . For the SPA all components are drawn from the same alphabet \mathcal{A} .
2. For each coded bit $b_{m,i}, m = 1, \dots, M_T, i = 0, \dots, \log_2|\mathcal{A}| - 1$, perform the following steps:
 - Determine $b = (\hat{\mathbf{d}}_{\text{SP}})_{m,i}$ and set $\tilde{\lambda}_{m,i}^b = \psi_{\text{ML}}^2(\hat{\mathbf{d}}_{\text{SP}})$;
 - calculate $\Psi_{\text{SP},m,i}^{\bar{b}} = \{\psi_{\text{ML}}^2(\mathbf{d}), \mathbf{d} \in \mathcal{D}_{\text{SP},m,i}^{\bar{b}}\}$ by re-quantizing each $\mathbf{d} \in \mathcal{D}_{\text{SP}}$ and performing the distance update (4.27);
 - obtain $\tilde{\lambda}_{m,i}^{\bar{b}}$ as the minimum element of $\Psi_{\text{SP},m,i}^{\bar{b}}$;
 - finally, calculate the approximate LLR

$$\tilde{\Lambda}_{\text{SP},m,i} = \frac{1-2b}{\sigma_w^2} (\tilde{\lambda}_{m,i}^b - \tilde{\lambda}_{m,i}^{\bar{b}}),$$

where the factor $1-2b \in \{-1, 1\}$ serves to adjust the sign.

The computational complexity of the SSPA can be assessed as follows. The preparation complexity C_{prep} is the same as that of the conventional hard-output SPA (cf. Section 2.5.2), which has the dominant complexity of $\mathcal{O}(M_T^3)$ due to the calculation of \mathbf{D}_{ZF} or \mathbf{D}_{MMSE} , respectively. The vector

complexity C_{vector} is composed of C_{vector} of the hard-output SPA (given by $\mathcal{O}(M_T^2 P)$, cf. Section 2.5.2) and of those operations carried out to provide soft instead of hard outputs (see the second item of the SSPA summary). Here, for each of the $M_T \log_2 |\mathcal{A}|$ coded bits, at most $2M_T P + 1$ data vectors of the reduced search set \mathcal{D}_{SP} (see the discussion in Section 2.5.1) are re-quantized and the corresponding distance updates (4.27) are performed. Since the complexity of a distance update is $\mathcal{O}(M_T)$, we obtain the overall vector complexity C_{vector} of the SSPA as $\mathcal{O}(M_T^3 |\mathcal{A}| \log_2 |\mathcal{A}|)$, where $P \propto |\mathcal{A}|$ is assumed.

4.4 Simulation Results

We next provide simulation results in order to assess the error-rate performance and the computational complexity of the various soft-output detectors. We consider an OFDM-based MIMO-BICM system for frequency-selective MIMO channels. The MIMO system operates in an SM mode, i.e. we have $M = M_T$ and $N = M_R$.

As soft-output detectors, we used the SSPA in conjunction with ZF and MMSE detectors (denoted SSPA-ZF and SSPA-MMMSE, respectively) and $Z = 4$ power method iterations for the SPA (cf. (2.19)), the list extension of Fincke and Phost's sphere decoding (LFPSD) algorithm (with LLR thresholds ± 8) [80], and the ZF-based and MMSE-based soft-output detectors according to (4.12) and (4.10), respectively. We consider LFPSD as the performance benchmark, although it does not provide the exact LLR expressions according to the log-max soft-output detector in (4.5). However, it is more efficient (although still exponentially complex) than computing (4.5) (or (4.3)) directly and its associated performance loss is marginal [80]. The ZF and SSPA-ZF soft-output detectors were considered besides the MMSE and SSPA-MMSE soft-output detectors since they have the advantage that knowledge of the noise variance is not required.

4.4.1 Packet Error Rate Performance

We first provide simulation results for the packet error rate (PER) performance of the considered OFDM-based MIMO-BICM system employing the various soft-output detectors. As discussed in Section 4.1.1, the encoding and interleaving was performed over all K subcarriers of the MIMO-OFDM system. The whole packet (corresponding to a codeword that is transmitted across the K subcarriers) was considered as erroneous if any corresponding information bit at the output of the soft-in channel decoder was decoded incorrectly. In practical systems, a packet error directly reduces the effective throughput since for an erroneous packet a suitable packet re-transmission procedure has to be invoked.

We employed $K = 128$ subcarriers, a standard rate-1/2 16-state convolutional code with octal generators (23, 35), 4 bits trellis termination, random block interleavers, and Gray labeling. At the receiver, a soft-in Viterbi decoder with a traceback depth of 25 was employed for channel decoding [88].

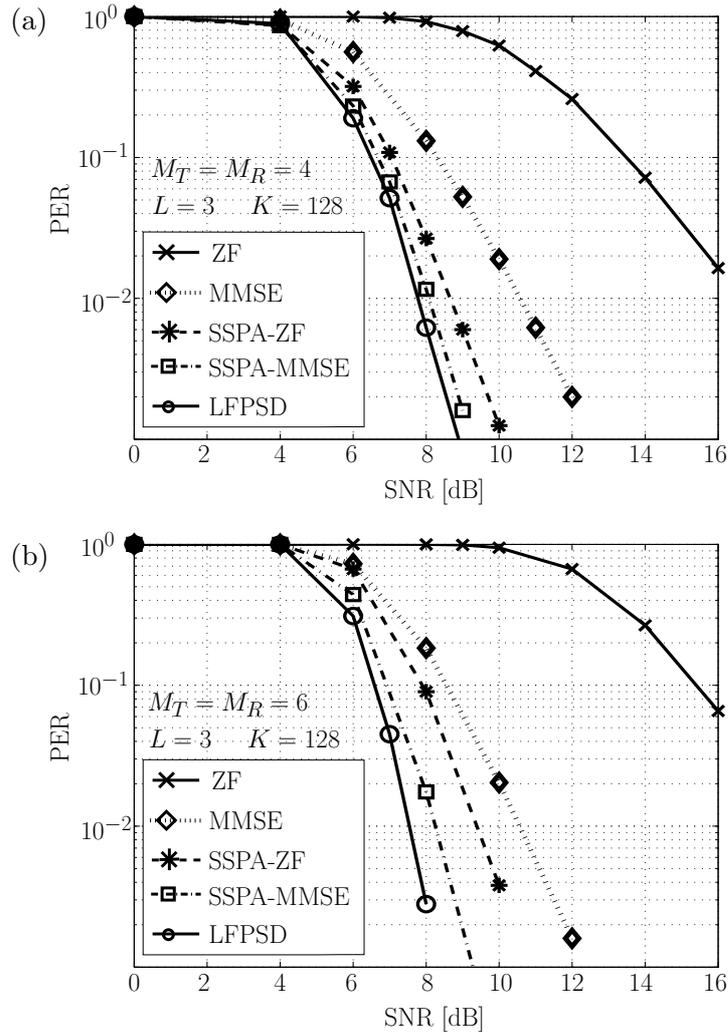


Figure 4.4: PER versus SNR performance of the various soft-output detection schemes for an OFDM-based MIMO-BICM system with a synthetic channel and 4-QAM modulation. (a) $M_T = M_R = 4$, (b) $M_T = M_R = 6$.

PER Performance for Synthetic Channels

We first consider a synthetic frequency-selective MIMO channel with $L = 3$ channel taps that were generated as spatially and temporally iid Gaussian with an uniform power-delay profile.

Figure 4.4(a) and Figure 4.4(b) show the PER versus SNR performance for a $M_T = M_R = 4$ and a $M_T = M_R = 6$ MIMO system, respectively, with 4-QAM data modulation. LFPSD used a candidate list containing 32 and 256 data vectors for the $M_T = M_R = 4$ and $M_T = M_R = 6$ system, respectively. The following conclusions can be drawn from these results:

- For the $M_T = M_R = 4$ channel, the performance of the SSPA-MMSE soft-output detector is virtually the same as that of LFPSD.
- For the $M_T = M_R = 6$ channel, the performance of SSPA-MMSE is very close to that of LFPSD.

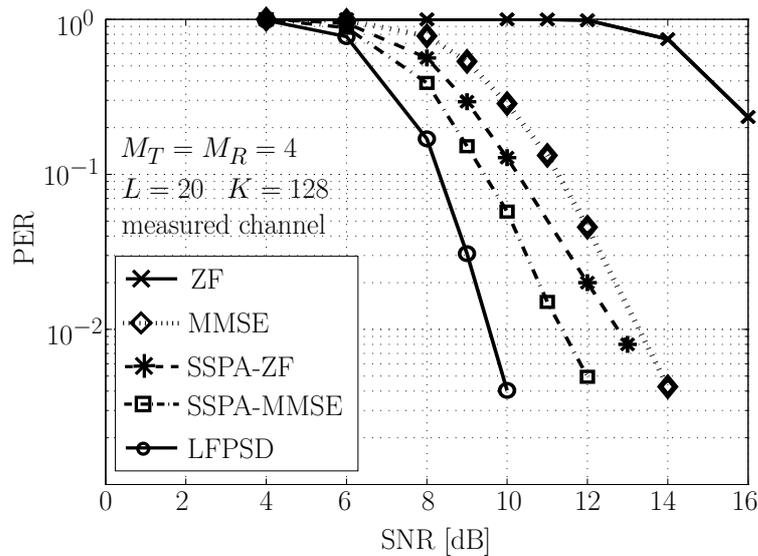


Figure 4.5: *PER versus SNR performance of the various soft-output detection schemes for an OFDM-based MIMO-BICM system for a measured MIMO channel and 4-QAM modulation.*

- The SSPA-ZF and the SSPA-MMSE detectors significantly outperform their equalization-based counterparts.
- The MMSE detector performs significantly better than the ZF detector.
- Both, the SSPA-ZF and the SSPA-MMSE detector can achieve a performance close to LFPSD performance.
- In contrast to ZF- and MMSE-based detection, SSPA-MMSE apparently does not incur any diversity loss.
- A comparison for the $M_T = M_R = 4$ case and $M_T = M_R = 6$ case suggests that for increasing channel size, the performance of the SSPA schemes degrade compared to that of LFPSD. This is similar to the behavior observed for the hard-output SPA in Section 2.6.1 and can be accounted to the fact that the IBC approximation becomes less accurate for an increasing channel size.

PER Performance for Measured Channels

Finally, we evaluate the various soft-output detectors using indoor MIMO channel measurements obtained at Vienna International Airport². A detailed description of the measurement campaign can be found in [45]. The transmitter and receiver positions were fixed and there was no line of sight. The channel data comprised 4452 impulse response snapshots of length $L = 20$. Fig. 4.5 again shows the PER versus SNR performance of the various soft-output detectors for a $M_T = M_R = 4$ MIMO

²I would like to thank N. Czik for providing the measurement data.

measured kflops – vector complexity C_{vector}					
$M_T = M_R$	LFPSD		SSPA-lin.		lin.
	av.	max.	av.	max.	
4	9.5	18.6	4.9	7.2	0.4
6	109	212	11.8	18.4	0.9
8	279	914	23.2	33.5	1.4

Table 4.1: Measured vector complexity C_{vector} (in kflops) of the various soft-output detectors.

system with 4-QAM data modulation. LFPSD used a candidate list containing 32 data vectors. The following conclusions can be drawn from these results:

- The PER performance for the measured channel (specifically the coding gain) is somewhat poorer than that for the synthetic channel (see Figure 4.4(a)).
- The performance relations of the algorithms to each other are quite similar to the synthetic case. However, the gaps between the individual PER curves are slightly larger. In particular, the gap between SSPA-MMSE and LFPSD now is larger, which can be attributed to spatial channel correlations that reduce the quality of the IBC approximation underlying the SSPA.

4.4.2 Computational Complexity

We now provide MATLAB flops measurements for the individual soft-output detectors. The results were obtained for iid Gaussian MIMO channels corresponding to a per-subcarrier complexity in the synthetic OFDM-based MIMO-BICM transmission setup. Table 4.1 shows the obtained results for the vector complexity C_{vector} . The preparation complexities are the same as for the hard-output detectors and can thus be found in Table 2.1(a). There is virtually no difference in the complexity of the ZF- and the corresponding MMSE-based schemes (both are denoted as “lin.”). LFPSD was implemented with 32 candidate data vectors for the $M_T = M_R = 4$ channel and with 256 candidate data vectors for the $M_T = M_R = 6$ and $M_T = M_R = 8$ channel. As for the (hard-output) FPSD, the complexity of LFPSD strongly depends on the channel realization and on the SNR. Again, we used an SNR of 8 dB and we provide the corresponding maximum and average complexities obtained within 1000 simulation runs. The following conclusions can be drawn from Table 4.1.

- The computational complexity required by the equalization-based (i.e. “lin.”) soft-output detectors is just a fraction of that required by the LFPSD and the SSPA.
- The average and the maximum complexity of the SSPA is much smaller than that required by the LFPSD (about a factor of 10 for the $M_T = M_R = 6$ case).
- The soft-output algorithms are computationally more intensive than their hard-output counterparts (cf. Table 2.1(b)).

4.5 Summary and Conclusions

In this chapter we investigated and proposed efficient *soft-output* detection algorithms for MIMO bit-interleaved coded modulation (BICM) systems. The task of a soft-output detector is to calculate (approximate) log-likelihood ratios (LLRs) that are provided to a soft-in channel decoder.

At the beginning, we applied a Gaussian approximation for post-equalization interference to re-derive known equalization-based soft-output detectors, which consist of an equalization step (ZF or MMSE) with a subsequent per-layer LLR calculation. Due to this per-layer processing, their computational complexity is very low. Our derivation showed that the MMSE-based detector can be seen as an extension of the ZF-based detector to correlated post-equalization interference. This extension yields a substantial performance improvement without an increase in computational complexity. However, we demonstrated that both equalization-based detectors suffer from a significant performance loss as compared to the soft-output (i.e. list) extension of the sphere-decoding algorithm.

To close the gap to sphere-decoding performance, we proposed the efficient *soft sphere-projection algorithm* (SSPA). The SSPA is based on the hard-output sphere-projection algorithm (SPA) and exploits intermediate SPA results to obtain approximate LLRs with low computational effort. The basis of the SSPA was the so-called re-quantization property, which was shown to hold for PSK alphabets with Gray labeling. Simulation results for OFDM-based MIMO-BICM systems demonstrated that the performance of the SSPA is similar to that of the soft-output sphere-decoding algorithm although its computational complexity is significantly smaller.

Conclusions and Outlook

FINALLY, we summarize the most important aspects and results of our work, present some conclusions, and provide suggestions for further research.

We proposed novel suboptimum detection algorithms for MIMO systems that are able to achieve near-optimum performance with low computational effort. We demonstrated that standard low-complexity approaches (such as equalization-based detection and nulling-and-cancelling) are in general far inferior to optimum detection. As a consequence, they are unable to exploit the full potential (such as the available diversity) of MIMO communication systems. Furthermore, optimum detection is in general computationally very intensive. This is also the case for advanced optimum detection via sphere-decoding, which has a high worst-case complexity (despite being quite efficient on average). Our proposed algorithms can be seen as improvements of conventional low-complexity schemes to reduce (or close) their gap to optimum performance. These improvements, in general, go along with an increased computational effort, which, however, is still low (in the worst case and also on average) as compared to sphere-decoding.

5.1 Conclusions

From our work about *efficient near-optimum detection algorithms for MIMO communication systems* we can draw the following conclusions.

Chapter 2 (“Geometry-Based Detectors for Spatial Multiplexing”):

- In MIMO spatial multiplexing (SM) systems, “bad” channel realizations with a large condition number are responsible for the inferior performance of conventional suboptimum detection algorithms.
- By combating the bad channel effect, the error-rate and diversity-gain performance of conventional schemes can be improved significantly.
- For moderate sized SM systems (up to six transmit and six receive antennas), modeling just one *single* small singular value of bad channels is already sufficient to capture essential parts of bad channel realizations, i.e. it is very unlikely that more than one singular value of the channel matrix is very small.
- Optimum data detection for *idealized* bad channels (which have one small singular value and the remaining singular values are all identical) can be performed *efficiently*, i.e. its complexity increases cubically and not exponentially in the number of antennas.
- For moderate sized SM systems, the proposed optimum data detector for idealized bad channels applied to arbitrary (nonidealized) channels, i.e. the *line search detector* (LSD), is still computationally intensive as compared to the sphere-decoding algorithm for optimum data detection (in particular, with respect to the *average* computational requirements).
- A constant modulus constraint on the symbol alphabet enables a substantial reduction of LSD’s computational complexity via some heuristics. For moderate sized SM systems, the resulting *sphere-projection algorithm* (SPA) add-on to conventional suboptimum detectors enables to achieve near-optimum performance with significantly less computational complexity than that required by the sphere-decoding algorithm. For example, for an SM system with six transmit and six receive antennas the SPA achieves near-optimum performance with an average complexity that is about three times smaller and a worst-case complexity that is over ten times smaller than that needed for sphere-decoding.
- With the SPA add-on, the computational effort of the respective conventional low-complexity schemes is increased (roughly by a factor of four).
- The idealized model for bad channels becomes less accurate for increasing system size. Here, it is more likely that two or more singular values of the channel are small.

- For large SM systems, idealized bad channels are not able to capture a large part of the bad channel effects that plague conventional schemes. In this case, the SPA add-on will still improve the performance of suboptimum schemes, but it will not be able to achieve near-optimum performance anymore.
- Efficient near-optimum performance is achieved by the SPA only for *moderate sized SM systems* (up to six transmit and six receive antennas) employing PSK alphabets.

Chapter 3 (“Dynamic Nulling-and-Cancelling”):

- The layerwise post-equalization SNRs (that are used by conventional NC to perform the layer sorting) are average quantities that just depend on the channel realization but not on the received vector.
- With maximum a-posteriori (MAP) detection and by employing the layerwise a-posteriori probabilities (APPs) for layer sorting, an optimum (but computationally very intensive) NC approach can be formulated.
- By using a Gaussian approximation for the post-equalization interference, the MAP-based NC approach simplifies to MMSE detection with an improved yet very simple layer-sorting rule, which is “dynamic” in that it depends on the current received vector in addition to the “static” channel realization. In fact, for dynamic layer sorting, the post-equalization SNRs are simply augmented by a dynamic quantity that provides the reliability of the detection (quantization) operation.
- If the post-equalization interference is assumed to be uncorrelated and Gaussian, the “LLR-based NC” technique (that was recently proposed in the literature) is obtained. This is an ZF-based NC scheme that also employs dynamic layer sorting.
- With dynamic layer sorting, the performance of NC schemes can be improved significantly. The performance gains will be largest when the post-equalization SNRs of all layers are similar. Here, the conventional (static) approach effectively performs a random layer sorting. On the other hand, dynamic layer sorting almost reduces to conventional layer sorting when a single post-equalization SNR is dominating.
- Efficient near-optimum performance is achieved by the proposed *dynamic nulling-and-cancelling* (DNC) scheme, i.e. MMSE-based NC with dynamic layer sorting, for SM systems (within a wide range of system sizes and channel SNRs) and for MIMO systems using certain linear-dispersion codes.
- A recursive calculation of the equalizer matrices enables an efficient implementation of DNC. This recursive approach yields larger benefits for DNC than for conventional NC.

- DNC is more complex than conventional NC (about a factor of three if DNC is applied to the equivalent real-valued system model while NC is applied to the complex-valued one) but still much more efficient than sphere-decoding.
- Compared to the SPA, DNC has the advantage that it can be applied to any linear MIMO model, while the SPA is specifically tailored to SM systems employing PSK alphabets.
- The performance of DNC is best for MIMO systems employing *QAM constellations*, which enable a separate detection of the real and imaginary parts of the data symbols. For moderate sized SM systems employing 4-QAM (QPSK) symbols, DNC also shows a slight complexity advantage over the SPA with a similar near-optimum performance.
- If, in moderate sized SM systems, a separate detection of the real and imaginary parts of the data symbols is not utilized (or not possible, e.g., due to higher order PSK symbols), DNC exhibits just a slight performance advantage over NC and performs worse than the SPA.

Chapter 4 (“Soft-Output Detection Algorithms”):

- Concepts and algorithms used for hard-output detection can be extended to the soft-output case. Here, instead of performing hard symbol decisions, the detector calculates approximate log-likelihood ratios (LLRs), which are then provided to a channel decoder. This improves the overall system performance as compared to hard detection and decoding.
- Soft-output detection is more computationally intensive than hard-output detection. In particular, both the list (i.e. soft-output) extension of the sphere-decoding algorithm and the proposed soft-output extension of the SPA (the SSPA) require significantly more computations than their hard-output counterparts.
- For moderate sized OFDM-based SM systems using bit-interleaved coded modulation (BICM), the SSPA has a similar average complexity as the *hard-output* sphere decoding algorithm (but a much smaller worst-case complexity) while its performance is close to that of the *soft-output* sphere-decoding algorithm.
- The Gaussian approximation for the post-equalization interference (used to derive the DNC) can be applied to re-derive conventional equalization-based soft-output detectors, which just perform an equalization step with a subsequent *per-layer* LLR calculation (i.e. all layers are processed independently after equalization).
- The computational complexity of the equalization-based soft-output detectors is just a fraction of that required by soft-output sphere-decoding and by the SSPA. However, the equalization-based detectors suffer from a significant performance loss.
- The SSPA shows a similar behavior as the SPA. For increasing system size, the SSPA performance degrades compared to that of the soft-output sphere-decoding algorithm.

- Spatial correlations have a degrading influence on the performance of the various schemes and they can yield an additional performance degradation of the SSPA as compared to the soft-output sphere-decoding algorithm.

5.2 Outlook

There remain several issues and possible extensions for further research.

- *Algorithm assessment:* The various algorithms were compared and investigated under idealized and simplified conditions. In particular, we assumed perfect channel state information at the receiver and to a large part we modeled the MIMO channel as uncorrelated. Furthermore, no implementation losses (e.g., due to fixed-point number representations or look-up tables for specialized functions) or model errors were taken into account. This provided us with just a basic insight into the performance and complexity behavior of the various algorithms. In practical implementations, however, there will be channel estimation errors, the transmit and receive antennas will be more or less correlated, and there will be an implementation loss and a model mismatch up to a certain degree. Thus, it remains to assess and compare the various algorithms under real-world conditions. In this thesis, a first attempt in this direction was performed by using measured (instead of synthetic) MIMO channels for a comparison of the soft-output detection algorithms.
- *SPA for higher-order QAM:* The SPA requires the symbol alphabet to have constant modulus property. In fact, this restricts its application to PSK symbol alphabets and the SPA cannot be applied to higher-order QAM constellations (like, e.g., 16-QAM). Since excellent performance results can be achieved by the SPA, an extension to higher-order QAM constellations seems to be promising. A possible approach in this direction could be to treat higher-order QAM constellations as several nested 4-QAM constellations (e.g., a 16-QAM constellation can be treated as two nested 4-QAM constellations), which all have the constant modulus property. The SPA could then be applied to these 4-QAM constellations in a multi-stage way.
- *Soft-input soft-output detection:* A large part of this thesis was devoted to *hard-output* detection algorithms. However, the overall system performance can be improved by calculating *soft* instead of hard decisions about the coded bits that are then provided to a soft-input channel decoder. Hence, in this thesis we also considered the extension of hard-output detection to soft-output detection. Here, in particular, we extended the hard-output SPA to its soft version. A soft-output extension of the DNC algorithm seems to be also promising and remains as open research. Furthermore, the performance of those coded MIMO systems with soft-output detection and soft-input decoding can be improved even further by employing the “turbo” principle of *iterative detection and decoding* [80, 89, 90]. Here, the data detector and the channel decoder exchange soft decisions about the coded bits in an iterative way. Hence, instead of just calculating soft

decisions about certain coded bits, the detector also has to accept soft decisions (i.e. a-priori information) as its input. The corresponding detectors can be referred to as *soft-input soft-output detectors*. In fact, the iterative detection and decoding approach allows to get very close to the capacity of MIMO systems [80]. Consequently, developing efficient soft-input soft-output detection algorithms is a very important research area and it seems promising to extend the various algorithms of this thesis (in particular, the SPA and the DNC) to soft-input soft-output processing.

- *Precoding* based on vector perturbation is a very promising transmission technique for wireless communications scenarios, where a transmitter uses multiple antennas to serve multiple *non-cooperative* users [73, 100, 101]. (This is similar to an SM system, where the multiple receive antennas – now corresponding to multiple users each having a single antenna – are not allowed to perform a joint processing of the received vector.) Here, the data to be transmitted is perturbed such that the transmit power is minimized. In fact, the calculation of the optimum perturbation vector can be seen as the dual problem to optimum data detection at the receiver. Consequently, optimum vector perturbation is in general very computationally intensive and there is a strong demand for efficient (approximate) vector perturbation algorithms for multi-antenna multi-user communication systems. Several efficient approximate techniques have been proposed in the literature, including, e.g., Tomlinson-Harashima precoding [100, 101], which is dual to nulling-and-cancelling at the receiver. Hence, it seems promising to apply and extend the concepts of the near-optimum data detectors of this thesis to efficient vector perturbation. As a first result we used the concept of idealized bad channels (see Section 2.2) to come up with an efficient vector perturbation algorithm based on approximate integer relations [20].

Bibliography

- [1] I. E. Telatar, "Capacity of multi-antenna Gaussian channels," *Europ. Trans. Telecomm.*, vol. 10, pp. 585–595, Nov./Dec. 1999.
- [2] G. J. Foschini, "Layered space-time architecture for wireless communication in a fading environment when using multi-element antennas," *Bell Labs Tech. J.*, vol. 1, no. 2, pp. 41–59, 1996.
- [3] V. Tarokh, N. Seshadri, and A. R. Calderbank, "Space-time codes for high data rate wireless communications: Performance criterion and code construction," *IEEE Trans. Inf. Theory*, vol. 44, pp. 744–765, March 1998.
- [4] L. Zheng and D. Tse, "Diversity and multiplexing: A fundamental tradeoff in multiple antenna channels," *IEEE Trans. Inf. Theory*, vol. 49, pp. 1073–1096, May 2003.
- [5] A. Burg, *VLSI Circuits for MIMO Communication Systems*. Konstanz: Hartung-Gorre Verlag, 2006.
- [6] S. Verdú, *Multiuser Detection*. Cambridge (UK): Cambridge Univ. Press, 1998.
- [7] E. Agrell, T. Eriksson, A. Vardy, and K. Zeger, "Closest point search in lattices," *IEEE Trans. Inf. Theory*, vol. 48, pp. 2201–2214, Aug. 2002.
- [8] U. Fincke and M. Phost, "Improved methods for calculating vectors of short length in a lattice, including a complexity analysis," *Math. Comp.*, vol. 44, pp. 463–471, April 1985.
- [9] M. Damen, H. El Gamal, and G. Caire, "On maximum-likelihood detection and the search for the closest lattice point," *IEEE Trans. Inf. Theory*, vol. 49, pp. 2389–2402, Oct. 2003.
- [10] S. Moshavi, "Multi-user detection for DS-CDMA communications," *IEEE Comm. Mag.*, pp. 124–136, Oct. 1996.
- [11] A. Paulraj, R. U. Nabar, and D. Gore, *Introduction to Space-Time Wireless Communications*. Cambridge (UK): Cambridge Univ. Press, 2003.
- [12] S. M. Alamouti, "A simple transmit diversity technique for wireless communications," *IEEE J. Sel. Areas Comm.*, vol. 16, pp. 1451–1458, Oct. 1998.
- [13] P. W. Wolniansky, G. J. Foschini, G. D. Golden, and R. A. Valenzuela, "V-BLAST: An architecture for realizing very high data rates over the rich-scattering wireless channel," in *Proc. URSI Int. Symp. on Signals, Systems and Electronics*, (Pisa, Italy), pp. 295–300, Sept. 1998.
- [14] G. D. Golden, G. J. Foschini, R. A. Valenzuela, and P. W. Wolniansky, "Detection algorithm and initial laboratory results using V-BLAST space-time communications architecture," *Elect. Lett.*, vol. 35, pp. 14–16, Jan. 1999.

-
- [15] H. Artés, D. Seethaler, and F. Hlawatsch, "Efficient detection algorithms for MIMO channels: A geometrical approach to approximate ML detection," *IEEE Trans. Signal Processing*, vol. 51, pp. 2808–2820, Nov. 2003.
- [16] D. Seethaler, H. Artés, and F. Hlawatsch, "Efficient approximate-ML detection for MIMO spatial multiplexing systems by using a 1-D nearest neighbor search," in *Proc. IEEE ISSPIT 2003*, (Darmstadt, Germany, invited paper), pp. 290–293, Dec. 2003.
- [17] D. Seethaler, H. Artés, and F. Hlawatsch, "Efficient near-ML detection for MIMO channels: The sphere-projection algorithm," in *Proc. IEEE Globecom 2003*, vol. IV, (San Francisco, CA), pp. 2089–2093, Dec. 2003.
- [18] H. Artés, "Reducing sphere decoder complexity by elliptical tree pruning," in *Proc. IEEE SPAWC 2004*, (Lisbon, Portugal), pp. 333–337, July 2004.
- [19] H. Artés, "Channel dependent tree pruning for the sphere decoder," in *Proc. IEEE ISIT 2004*, (Chicago, USA), p. 539, July 2004.
- [20] D. Seethaler and G. Matz, "Efficient vector perturbation in multi-antenna multi-user systems based on approximate integer relations," in *Proc. EUSIPCO 2006*, (Florence, Italy), Sept. 2006.
- [21] J. G. Proakis, *Digital Communications*. New York: McGraw-Hill, 3rd ed., 1995.
- [22] A. Duel-Hallen, "Decorrelating decision-feedback multiuser detector for synchronous code-division multiple access channels," *IEEE Trans. Comm.*, vol. 41, pp. 285–290, Feb. 1993.
- [23] D. Seethaler, H. Artés, and F. Hlawatsch, "Dynamic nulling-and-cancelling for efficient near-ML decoding of MIMO systems," *IEEE Trans. Signal Processing*, to appear, 2006.
- [24] D. Seethaler, H. Artés, and F. Hlawatsch, "Dynamic nulling-and-cancelling with near-ML performance," in *Proc. IEEE ICASSP 2004*, vol. IV, (Montreal, Canada), pp. 777–780, May 2004.
- [25] D. Seethaler, H. Artés, and F. Hlawatsch, "Dynamic versus conventional layer sorting for nulling-and-cancelling based MIMO detection," in *Proc. EUSIPCO 2005*, (Antalya, Turkey), Sept. 2005.
- [26] D. Seethaler, G. Matz, and F. Hlawatsch, "Efficient soft demodulation in MIMO-OFDM systems with BICM and constant modulus alphabets," in *Proc. IEEE ICASSP 2006*, vol. IV, (Toulouse, France), pp. 105–108, May 2006.
- [27] D. Seethaler, G. Matz, and F. Hlawatsch, "Low-complexity soft demodulation of MIMO-BICM using the line search detector," in *Proc. IEEE ICC 2005*, vol. IV, (Seoul, Korea), pp. 2447–2451, May 2005.
- [28] D. Seethaler, G. Matz, and F. Hlawatsch, "An efficient MMSE-based demodulator for MIMO bit-interleaved coded modulation," in *Proc. IEEE Globecom 2004*, vol. IV, (Dallas, Texas), pp. 2455–2459, Dec. 2004.
- [29] M. Damen, A. Chkeif, and J. Belfiore, "Lattice code decoder for space-time codes," *IEEE Comm. Letters*, vol. 4, pp. 161–163, May 2000.
- [30] R. F. H. Fischer and C. Windpassinger, "Real versus complex-valued equalisation in V-BLAST systems," *Elect. Lett.*, vol. 39, pp. 470–471, March 2003.
- [31] A. Goldsmith, *Wireless Communications*. Cambridge (UK): Cambridge Univ. Press, 2005.
- [32] M. Borgmann and H. Bölcskei, "Interpolation-based efficient matrix inversion for MIMO-OFDM receivers," in *Proc. Asilomar Conf. Signals, Systems and Computers*, vol. II, (Pacific Grove, CA), pp. 1941–1947, Nov. 2004.
- [33] B. Hassibi and B. Hochwald, "High-rate codes that are linear in space and time," *IEEE Trans. Inf. Theory*, vol. 48, no. 7, pp. 1804–1824, 2002.

- [34] S. Barbarossa, *Multiantenna Wireless Communication Systems*. Norwood, MA: Artech House, 2005.
- [35] V. Tarokh, H. Jafarkhani, and R. A. Calderbank, "Space-time block codes from orthogonal designs," *IEEE Trans. Inf. Theory*, vol. 45, pp. 1456–1467, July 1999.
- [36] B. Hassibi and B. Hochwald, "How much training is needed in multiple-antenna wireless links?," *IEEE Trans. Inf. Theory*, vol. 49, pp. 951–963, April 2003.
- [37] F. Dietrich and W. Utschick, "Pilot-assisted channel estimation based on second-order statistics," *IEEE Trans. Signal Processing*, vol. 53, pp. 1178–1193, March 2005.
- [38] M. Biguesh and A. B. Gershman, "Training-based MIMO channel estimation: A study of estimator tradeoffs and optimal training signals," *IEEE Trans. Signal Processing*, vol. 54, pp. 884–893, March 2006.
- [39] N. Czink, G. Matz, D. Seethaler, and F. Hlawatsch, "Improved MMSE estimation of correlated MIMO channels using a structured correlation estimator," in *Proc. IEEE SPAWC 2005*, (New York City, USA), pp. 595–599, June 2005.
- [40] M. Rupp, "On the influence of uncertainties in MIMO decoding algorithms," in *Proc. Asilomar Conf. Signals, Systems, Computers*, vol. 1, pp. 570–574, Nov. 2002.
- [41] B. Hassibi and B. Hochwald, "Cayley differential unitary space-time codes," *IEEE Trans. Inf. Theory*, vol. 48, pp. 1485–1503, June 2002.
- [42] D. Tse and P. Viswanath, *Fundamentals of Wireless Communication*. Boston (MA): Cambridge University Press, 2005.
- [43] J.P. Kermoal, L. Schumacher, K.I. Pedersen, P.E. Mogensen, and F. Frederiksen, "A stochastic MIMO radio channel model with experimental validation," *IEEE J. Sel. Ar. Comm.*, vol. 20, pp. 1211–1226, Aug. 2002.
- [44] W. Weichselberger, M. Herdin, H. Özcelik, and E. Bonek, "A stochastic MIMO channel model with joint correlation of both link ends," *IEEE Trans. Wireless Comm.*, vol. 5, pp. 90–100, Jan. 2006.
- [45] M. Herdin, *Non-Stationary Indoor MIMO Radio Channels*. PhD thesis, Vienna University of Technology, Aug. 2004.
- [46] J. Von Zur Gathen and J. Gerhard, *Modern Computer Algebra*. Cambridge (UK): Cambridge Univ. Press, 2003.
- [47] D. Cescato, M. Borgmann, H. Bölcskei, J. Hansen, and A. Burg, "Interpolation-based QR decomposition in MIMO-OFDM systems," in *Proc. IEEE SPAWC 2004*, (NY, USA), pp. 945–949, June 2005.
- [48] A. Burg, S. Haene, D. Perels, P. Luethi, N. Felber, and W. Fichtner, "Receiver design for multi-antenna wireless communications," in *Proc. IEEE PRIME 2005*, (Laussane, Switzerland), pp. 231–234, July 2005.
- [49] E. Viterbo and J. Boutros, "A universal lattice code decoder for fading channels," *IEEE Trans. Inf. Theory*, vol. 45, pp. 1639–1642, July 1999.
- [50] B. Hassibi and H. Vikalo, "On the sphere decoding algorithm I. Expected complexity," *IEEE Trans. Signal Processing*, vol. 53, pp. 2806–2818, Aug. 2005.
- [51] S. Boyd and L. Vandenberghe, *Convex Optimization*. Cambridge (UK): Cambridge Univ. Press, Dec. 2004.
- [52] W. K. Ma, T. N. Davidson, K. M. Wong, Z. Q. Luo, and P. C. Ching, "Quasi-maximum-likelihood multiuser detection using semi-definite relaxation with application to synchronous CDMA," *IEEE Trans. Signal Processing*, vol. 50, pp. 912–922, April 2002.
- [53] M. Grötschel, L. Lovász, and A. Schrijver, *Geometric Algorithms and Combinatorial Optimization*. Berlin: Springer, 2nd ed., 1993.

- [54] K. Su and I. J. Wassell, "A new ordering for efficient sphere decoding," in *Proc. IEEE ICC 2005*, (Seoul, Korea), pp. 1906–1910, May 2005.
- [55] M. Stojnic, H. Vikalo, and B. Hassibi, "A branch and bound approach to speed up the sphere decoder," in *Proc. IEEE ICASSP 2005*, (Philadelphia, USA), pp. 429–433, May 2005.
- [56] K. W. Wong, C. Y. Tsui, R. S. K. Cheng, and W. H. Mow, "A VLSI architecture of a K-best lattice decoding algorithm for MIMO channels," in *Proc. IEEE ISCAS 2002*, (Scottsdale, USA), pp. 273–276, May 2002.
- [57] B. Gowaikar, R.; Hassibi, "Efficient near-ML decoding via statistical pruning," in *Proc. IEEE ISIT*, p. 274, June/July 2003.
- [58] J. Jalden and B. Ottersten, "On the complexity of sphere decoding in digital communications," *IEEE Trans. Signal Processing*, vol. 53, pp. 1474–1484, Apr. 2005.
- [59] G. H. Golub and C. F. Van Loan, *Matrix Computations*. Baltimore: Johns Hopkins University Press, 3rd ed., 1996.
- [60] S. M. Kay, *Fundamentals of Statistical Signal Processing: Estimation Theory*. Englewood Cliffs (NJ): Prentice Hall, 1993.
- [61] A. Wiesel, Y. C. Eldar, and S. Shamai, "Semidefinite relaxation for detection of 16-QAM signaling in MIMO channels," *IEEE Signal Processing Letters*, vol. 12, pp. 653–656, Sept. 2005.
- [62] S. Loyka and F. Gagnon, "Performance analysis of the V-BLAST algorithm: An analytical approach," *IEEE Trans. Wireless Comm.*, vol. 3, pp. 1326–1337, July 2004.
- [63] G. J. Foschini, G. D. Golden, R. A. Valenzuela, and P. W. Wolniansky, "Simplified processing for high spectral efficiency wireless communication employing multi-element arrays," *IEEE J. Sel. Areas Comm.*, vol. 17, pp. 1841–1852, Nov. 1999.
- [64] B. Hassibi, "A fast square-root implementation for BLAST," in *Proc. 34th Asilomar Conf. Signals, Systems, Computers*, (Pacific Grove, CA), pp. 1255–1259, Nov./Dec. 2000.
- [65] R. W. Heath, S. Sandhu, and A. J. Paulraj, "Antenna selection for spatial multiplexing systems with linear receivers," *IEEE Commun. Letters*, vol. 5, pp. 142–144, April 2001.
- [66] G. Ginis and J. Cioffi, "On the relation between V-BLAST and GDFE," *IEEE Comm. Letters*, vol. 5, pp. 364–366, Sept. 2001.
- [67] D. Wübben, R. Böhnke, J. Rinas, V. Kühn, and K. Kammeyer, "Efficient algorithm for decoding layered space-time codes," *Electronics Letters*, vol. 37, pp. 1348–1350, Oct. 2001.
- [68] R. Böhnke, D. Wübben, V. Kühn, and K. D. Kammeyer, "Reduced complexity MMSE detection for BLAST architectures," in *Proc. IEEE Globecom 2003*, vol. 4, (San Francisco, CA), pp. 2258–2262, Dec. 2003.
- [69] W.-J. Choi, R. Negi, and J. M. Cioffi, "Combined ML and DFE decoding for the V-BLAST system," in *Proc. IEEE ICC-00*, (New Orleans, LA), pp. 18–22, June 2000.
- [70] J. Benesty, Y. Huang, and J. Chen, "A fast recursive algorithm for optimum sequential signal detection in a BLAST system," *IEEE Trans. Signal Processing*, vol. 51, pp. 1722–1730, July 2003.
- [71] A. Edelman, *Eigenvalues and Condition Numbers of Random Matrices*. PhD thesis, Massachusetts Institute of Technology, Cambridge (MA), 1989.
- [72] C. W. Therrien, *Discrete Random Signals and Statistical Signal Processing*. Englewood Cliffs (NJ): Prentice Hall, 1992.

- [73] C. B. Peel, B. M. Hochwald, and A. L. Swindlehurst, "A vector-perturbation technique for near-capacity multiantenna multiuser communication - Part I: Channel inversion and regularization," *IEEE Trans. Comm.*, vol. 53, pp. 195–202, Jan. 2005.
- [74] J. Luo, K. R. Pattipati, P. K. Willett, and F. Hasegawa, "Near-optimal multiuser detection in synchronous CDMA using probabilistic data association," *IEEE Commun. Letters*, vol. 5, pp. 361–363, Sept. 2001.
- [75] S. W. Kim and K. P. Kim, "Log-likelihood-ratio-based detection ordering in V-BLAST," *IEEE Trans. Communications*, vol. 54, pp. 302–307, Feb. 2006.
- [76] S. M. Kay, *Fundamentals of Statistical Signal Processing: Detection Theory*. Upper Saddle River (NJ): Prentice Hall, 1998.
- [77] A. Klein, G. Kaleh, and P. W. Baier, "Zero forcing and minimum mean square error equalization for multiuser detection in code-division multiple-access channels," *IEEE Trans. Veh. Technol.*, vol. 45, pp. 276–287, May 1996.
- [78] J. M. Cioffi, G. P. Dudevoir, M. V. Eyuboglu, and G. D. Forney, "MMSE decision-feedback equalizers and coding – Part I: Equalization results," *IEEE Trans. Commun.*, vol. 43, pp. 2582–2594, Oct. 1995.
- [79] S. B aro, J. Hagenauer, and M. Witzke, "Iterative detection of MIMO transmission using a list-sequential LISS detector," in *Proc. IEEE ICC 2003*, (Anchorage, AK), pp. 2653–2657, May 2003.
- [80] B. M. Hochwald and S. ten Brink, "Achieving near-capacity on a multiple-antenna channel," *IEEE Trans. Inf. Theory*, vol. 51, pp. 389–399, March 2003.
- [81] J. L opez-Vicario, C. Mecklenbr auker, and C. Ant on-Haro, "Reduced-complexity methods for throughput maximization in MIMO channels," in *Proc. IEEE ICC-2004*, vol. 5, (Paris, France), pp. 2678–2682, June 2004.
- [82] J. Evans and D. Tse, "Large system performance of linear multiuser receivers in multipath fading channels," *IEEE Trans. Inf. Theory*, vol. 46, no. 6, pp. 2059–2078, 2000.
- [83] E. Zehavi, "8-PSK trellis codes for a Rayleigh channel," *IEEE Trans. Comm.*, vol. 40, pp. 873–884, May 1992.
- [84] G. Caire, G. Taricco, and E. Biglieri, "Bit-interleaved coded modulation," *IEEE Trans. Inf. Theory*, vol. 44, pp. 927–945, May 1998.
- [85] J. J. Boutros, F. Boixadera, and C. Lamy, "Bit-interleaved coded modulations for multiple-input multiple-output channels," in *Proc. IEEE Symp. Spread Spectrum Techn. Appl.*, (Parsippany, NJ), pp. 123–126, Sept. 2000.
- [86] H. Yang, "A road to future broadband wireless access: MIMO-OFDM-based air interface," *IEEE Communications Magazine*, pp. 53–60, Jan. 2005.
- [87] S. H. M uller-Weinfurtner, "Coding approaches for multiple antenna transmission in fast fading and OFDM," *IEEE Trans. Signal Processing*, vol. 50, pp. 2442–2450, Oct. 2002.
- [88] S. Lin and D. J. Costello, *Error Control Coding*. Upper Saddle River (NJ): Prentice Hall, 2004.
- [89] J. Hagenauer, "The turbo principle: Tutorial introduction and state of the art," in *Proc. 1st Int. Symp. on Turbo Codes and Related Topics*, (Brest, France), pp. 1–11, Sept. 1997.
- [90] S. ten Brink, J. Speidel, and R. H. Yan, "Iterative demapping for QPSK modulation," *Electronics Letters*, vol. 34, pp. 1459–1460, July 1998.
- [91] A. Dejonghe and L. Vandendorpe, "Turbo equalization for multilevel modulation: An efficient low-complexity scheme," in *Proc. IEEE ICC 2002*, vol. 3, (New York City), pp. 1863–1867, May 2002.

-
- [92] M. Butler and I. Collings, "A zero-forcing approximate log-likelihood receiver for MIMO bit-interleaved coded modulation," *IEEE Commun. Letters*, vol. 8, pp. 105–107, Feb. 2004.
- [93] K.-B. Song and S. A. Mujtaba, "A low complexity space-frequency BICM MIMO-OFDM system for next-generation WLANs," in *Proc. IEEE Globecom 2003*, (San Francisco, CA), pp. 1059–1063, Dec. 2003.
- [94] P. Robertson, E. Villebrun, and P. Hoeher, "A comparison of optimal and sub-optimal MAP decoding algorithms operating in the log domain," in *Proc. IEEE ICC-1995*, (Seattle, WA), pp. 1009–1013, June 1995.
- [95] Z. Guo and P. Nilsson, "A low complexity soft-output MIMO decoding algorithm," in *Proc. IEEE Sarnoff Symp. on Advances in Wired and Wireless Commu. 2005*, (Princeton, USA), pp. 90–93, April 2004.
- [96] Y. L. C. Jong and T. J. Willink, "Iterative tree search detection for MIMO wireless systems," *IEEE Trans. Communications*, vol. 53, pp. 930–935, June 2005.
- [97] X. Wang and H. V. Poor, "Iterative (turbo) soft interference cancellation and decoding for coded CDMA," *IEEE Trans. Comm.*, vol. 47, pp. 1046–1061, July 1999.
- [98] A. Chindapol and J. A. Ritcey, "Design, analysis, and performance evaluation for BICM-ID with square QAM constellations in Rayleigh fading channels," *IEEE J. Sel. Areas Comm.*, vol. 19, pp. 944–957, May 2001.
- [99] E. Agrell, J. Lassing, E. G. Ström, and T. Ottosson, "On the optimality of the binary reflected Gray code," *IEEE Trans. Inf. Theory*, vol. 50, pp. 3170–3182, Dec. 2004.
- [100] B. M. Hochwald, C. B. Peel, and A. L. Swindlehurst, "A vector-perturbation technique for near-capacity multiantenna multiuser communication - Part II: Perturbation," *IEEE Trans. Comm.*, vol. 53, pp. 537–544, March 2005.
- [101] C. Windpassinger, R. F. H. Fischer, T. Vencel, and J. B. Huber, "Precoding in multiantenna and multiuser communications," *IEEE Trans. Wireless Comm.*, vol. 3, pp. 1305–1316, July 2004.

List of Abbreviations

APP	a-posteriori probability
BICM	bit-interleaved coded modulation
BPSK	binary phase-shift keying
cdf	cumulative distribution function
DNC	dynamic nulling-and-cancelling
FPSD	Fincke and Phost's sphere-decoding
HSDPA	high-speed downlink packet access
IBC	idealized bad channel
IML	idealized maximum likelihood
IRF	instantaneous reliability factor
LD	linear dispersion
LFPSD	list-extension of Fincke and Phost's sphere-decoding
LLR	log-likelihood ratio
LSD	line search detector
MIMO	multiple-input multiple-output
MISO	multiple-input single-output
ML	maximum likelihood
MMSE	minimum mean-square error
NC	nulling-and-cancelling
OFDM	orthogonal frequency division multiplexing
pdf	probability density function
PER	packet error rate
PSK	phase-shift keying
PSNR	post-equalization signal-to-noise-ratio
QAM	quadrature amplitude modulation
SER	symbol error rate
SIMO	single-input multiple-output
SM	spatial multiplexing
SNR	signal-to-noise-ratio
SPA	sphere projection algorithm
SSPA	soft-output sphere projection algorithm
SVD	singular value decomposition
V-BLAST	Vertical Bell Laboratories Layered Space-Time
VLSI	very-large-scale integration
ZF	zero-forcing