Dissertation

Low-Complexity Iterative Receiver for Multi-User OFDM Systems in Time-Varying MIMO Channels

ausgeführt zum Zwecke der Erlangung des akademischen Grades
einer Doktorin der technischen Wissenschaften

eingereicht an der
Technischen Universität Wien
Fakultät für Elektrotechnik und Informationstechnik

von

Charlotte Dumard

Wien, 30. April 2009

Technische
Universität
Wien

Vienna
University of
Technology

Supervisor
Prof. Gerald Matz
Institut für Nachrichtentechnik und Hochfrequenztechnik
Technische Universität Wien, Austria

Examiner
Prof. Monica Nicoli
Dipartimento di Elettronica e Informazione (DEI)
Politecnico di Milano, Italy

FTW Dissertation Series

Charlotte Dumard

Low-Complexity Iterative Receiver for Multi-User OFDM Systems in Time-Varying MIMO Channels

ftw Creating Communication Technologies

telecommunications research center vienna

This work was carried out with funding from the Vienna Science and Technology Fund (Wiener Wissenschafts-, Forschungs- und Technologiefonds, WWTF) within the projects "Future Mobile Communications Systems - Mathematical Modeling, Analysis, and Algorithms for Multi Antenna Systems" and "Cooperative Communications in Traffic Telematics"
The Telecommunications Research Center Vienna (FTW) is supported by the Austrian Government and the City of Vienna within the competence center program COMET.

Supervisor at FTW: Thomas Zemen

This thesis has been prepared using LaTeX.

April 2009
1. Auflage

Herausgeber: FTW Forschungszentrum Telekommunikation Wien GmbH
Printed in Austria
ISBN 3-902477-19-9

Abstract

Achieving high rates at high velocity is one of the ambitious goals of future wireless communications. Increasing demand for communications at vehicular speeds in turn requires highly complex processing at the base station to estimate the fast changing channel and decode large amounts of data. Real-time processing requirements or energy constraints limit the amount of signal processing affordable at the terminals. Thus, sub-optimal methods and algorithms permitting reduction of computational complexity while still keeping sustainable signal quality are highly desirable.

The NP-complete optimum *maximum a posteriori* sequence detector can be approximated by an iterative receiver, performing channel estimation and parallel interference cancelation (PIC) followed by *linear minimum mean square error* (LMMSE) detection. The two LMMSE filters required for channel estimation and for multi-user detection are a common source of complexity in scientific computing. This work investigates two approaches, which aim at further reducing complexity while avoiding strong performance degradation.

The first approach explores linear detection, using Krylov subspace methods to approximate the output of an LMMSE filter. An LMMSE filter is derived for detection in the user space, i.e. using matched filtering prior to PIC. Combined with the Krylov subspace method, this allows joint antenna detection while achieving considerable complexity reduction.

A second approach investigates non-linear *maximum likelihood* multi-user detection. Focus is put on its low-complexity implementation by means of a sphere decoder. Making use of the channel basis expansion, an efficient implementation of a sphere decoder which is more suitable to fast varying channels is developed. Applied to computations of log-likelihood ratios, the reduced-rank sphere decoder results in a receiver achieving one magnitude less complexity at no performance loss.

This thesis aims at presenting a fair comparison between these two approaches in order to facilitate determining a trade-off between complexity and an acceptable performance degradation, depending on the system parameters and the available hardware.

Acknowledgement

This is the place to thank the numerous people who, directly or not, contributed to the completion of this thesis. First I would like to thank Thomas Zemen for his supervision at FTW and for providing me with support, ideas and encouragements through my doctoral years. I would also like to thank Gerald Matz and Monica Nicoli for their attentive reading and constructive feedback on this work, pointing out many questions. Special thanks to Monica for taking the time to come to Vienna for the defense. Many thanks also to Florian Kaltenberger, Christoph Mecklenbräuker and the team members of the Math+MIMO and COCOMINT projects for their fruitful contributions and collaborations.

Now, there are a lot more people I want to thank for making my doctoral years at FTW so exciting. To start with, special recognition is due to Markus Kommenda for giving me the chance to join FTW almost five years ago, and for endless discussions and advice on photography. For contributing to the extremely friendly environment at FTW, I am very grateful to my current and former colleagues. These include in no special order Laira Bernadó, Claudio Weidmann, Petros Elia, Jossy Sayir, for providing the necessary brain supply in the form of turkish (or was it greek?) coffee; Amira Gutmann-Trieb, for the unfrequent but precious long (girl-) talks we would have around lunch; Fabio Ricciato, Erwin Riegler and Maxime Guillaud, for their true friendship and all the time spent off-work together.

Savory thanks goes to the Italian crew and its pasta cooking, and to Sebastian Egger and the *FTW kitchen* members for providing great lunches (and some dinners) during my time at FTW. Very special thanks to Pavle Belanović, just for never stopping pavling around, and particularly for taking care of the organization on my defense day during his vacation.

At last I will not forget to mention those people who most believe in me, despite having no clue as to what I am doing: my parents and siblings for being so proud although fearing having to explain what my job actually is; Joachim Stern for proof-reading my thesis, and being the only one capable of re-motivating me when I was that close to give it up, even after quitting asking me to repair his mobile phone.

Contents

Chapter 1
Introduction

1.1 Motivation

Future mobile communication networks need to be designed to respond to an increasing demand from the user's side. This demand includes high rate data transmissions, as online music or video streaming, or communications that keep reliable when the user is moving at vehicular speeds. With bandwidth and power constraints, signal processing at both mobile and base units is not straightforward, to be able to treat such an amount of information. Additionally, a constantly rising number of users and fast changing channels need to be taken into account. How to increase data rates is partly answered, for instance, by using multiple input multiple output (MIMO) or multi-carrier systems. However, introducing diversity increases the number of parameters of the system (number of antennas, users, subcarriers) and further enhances the complexity of the signal processing at the receiver. Hence, efficient low-complexity algorithms are needed. In this thesis we focus on the uplink of a multi-carrier (MC) code division multiple access (CDMA) system based on orthogonal frequency division multiplexing (OFDM) and more particularly on low-complexity channel estimation and multi-user detection at the receiver.

1.2 State of the Art

MIMO systems have proven to significantly improve capacity [FG98]. However, multiple antennas at the transmitters enhance multi-access interference (MAI) at the receiver, which adds up to inter-symbol interference (ISI) in frequency selective channels. Using OFDM allows suppressing the ISI by means of a cycle prefix.

The optimum multi-user detector is the *maximum a posteriori* (MAP) sequence detector, that minimizes the error probability of all user information sequences. The MAP detector is known to be NP-complete [GJ79] and becomes prohibitively complex if not unfeasible in time-varying systems. A sub-optimal approach based on an iterative

structure of the receiver makes use of soft information feed back to refine channel estimation and multi-user detection [ARAS99, TKS02, KST04]. Such a receiver for MC-CDMA is presented in [Weh05], performing interference cancelation followed by linear minimum mean square error (LMMSE) detection [Hay05]. It has close to optimal performance while allowing considerable complexity reduction. The soft information gained about the transmitted data symbols is used to enhance the channel estimation and data detection in consecutive iterations. For channel estimation, a second LMMSE filter estimates the projection parameters of the time-varying channel represented through a Slepian basis expansion [Sle78, ZM05].

Multiple antennas at the receiver can be exploited to perform joint antenna detection and virtually increase the spreading factor, following the *resource pooling* result [HT01]. The drawback of joint antenna detection is the extremely high computational complexity due to increasing dimensions of the system. The matrix inversion that is necessary to compute the two LMMSE filters needed for channel estimation and multi-user detection largely determines the computational complexity of the receiver.

We aim at implementing a receiver performing joint antenna detection with lower complexity, such that it becomes feasible in a practical system. At first, we consider linear detection and make use of Krylov subspace methods to approximate the LMMSE filter output with low-complexity. The next step consists in non-linear detection and low-complexity implementation of a *maximum likelihood* (ML) detector by means of sphere decoding.

Krylov subspace methods [vdV03, Saa03, MS00, KS99] are projection methods, allowing to solve a linear system with low complexity by trading accuracy for efficiency. It has long been used in signal processing, e.g. for beamforming [KT04, KH06] or detection [CMD06, DU07], where computational complexity reduction is shown.

In [CMD06], the projection coefficients onto the Krylov subspace are computed as universal weights, based on the self-averaging properties of random matrices modeling the channel. The *a priori* random eigenvalues of the channel matrix can be described by averaging over sufficiently large samples. The eigenvalue distribution of the channel matrix converges to a deterministic distribution when its dimensions grow to infinity. Universal weights are thus computed independently of the received signal. However, the authors in [CMD06] do not take into account an iterative scheme using interference cancelation. In such a case, the projection computations are not common to all users

anymore, and no computational complexity reduction can be achieved this way.

In [DU07], the authors use the Lanczos algorithm [Saa03] to approximate the Wiener filter in an iterative receiver for a single user single input multiple output system. Their iterative scheme uses an adjusted mean of the signal based on *a priori* information to cancel the multi-path interference. The computational complexity using the Lanczos algorithm in [DU07] scales quadratic with the length of the observation vector.

So far, using conventional parallel interference schemes to detect each user leads to multiple LMMSE filters and as a consequence the computational complexity for multi-user detection grows polynomial with the number of users. Joint detection of all users can be done using the user space signal obtained after matched filtering [HWS04]. This way, a unique LMMSE filter is required for multi-user detection, partly solving the complexity issue for the iterative receiver mentioned above.

Non linear ML detection searches for the closest point to the received signal, according to a given metric. ML detection is an exhaustive search over the complete alphabet of all possible transmitted vectors and might become prohibitively complex for larger alphabets or an increasing number of transmit antennas.

Originally developed to search for vectors within a lattice [Poh81, FP85], sphere decoders aim at performing ML detection without requiring exhaustive search [AEVZ02, HV05a, HV05b]. Although they appear to have exponential complexity when the system parameters tend to infinity [JO05a], considerable complexity reduction might be achieved in a practical case, if the radius of the sphere is well defined.

Two innovative enumerations have been presented in 1985 by Fincke and Pohst [FP85] and in 1994 by Schnorr and Euchner [SE94]. They differ by the way sorting of the nodes is performed. In the former, the nodes are sorted according to their increasing values in a unique way independent on the search performed. The latter improves sorting by ordering the nodes according to their partial search metric. In such a case, sorting is more efficient, nevertheless it differs for every new search and *a fortiori* for each new channel realization.

Sphere decoders have been first applied to ML detection in [VB93, Mow94] and used in further application like multiple antenna systems [VH02] or CDMA systems [Bru01, BB03]. A further simplified suboptimal sphere decoder is presented in [Bru04], achieving close to optimal performance and more complexity reduction. Extension to computations of soft information [HtB03, BGBF03], MIMO-OFDM systems [SMH06] have been introduced, and a parallel implementation of a soft output sphere decoder is given in [JO05b].

Based on the Fincke-Pohst and Schnorr-Euchner algorithms, numerous works have been published, that mainly focus on speeding up the sphere decoder (i.e. reducing its complexity by visiting less nodes). Refined criteria like new ordering of the nodes, new decision criterium, use of different metrics [SB08], or finding the optimal radius [DEGC03] have been the focus of research on sphere decoding in order to reduce complexity. A general framework [MGDC06] includes various possible improvements for sphere decoding.

Discussion on the complexity of a sphere decoder for an OFDM system is presented in [BRC08]. The authors of [BRC08] modify the sphere decoder by making use of the Toeplitz structure of the matrix to reduce computational complexity. They define a block sphere decoder taking place after a preprocessing stage of the sphere decoder. Although it leads to a considerable reduction of the complexity of the preprocessing stage, the tree search itself has greater complexity. The authors come to the conclusion that the block sphere decoder would not be suitable for a realistic frequency selective environment.

1.3 Outline and Contributions

This thesis focuses on deriving low-complexity alternatives to the iterative receiver performing LMMSE detection presented in [Weh05], and is based on the work partly published in the following papers:

- C. Dumard and T. Zemen. Double Krylov subspace approximation for low-complexity iterative multi-user decoding and time-variant channel estimation. In *Proc. IEEE Workshop on Signal Processing Advances in Wireless Com. (SPAWC)*, New York, NY, USA, June 2005.

- C. Dumard and T. Zemen. Krylov approximation for multi-user detection with parallel interference cancellation in chip and user space. In *Proc. Winter School on Coding and Information Theory*, Bratislava, Slovakia, February 2005.

- C. Dumard and T. Zemen. Integration of the Krylov subspace method in an iterative multi-user detector for time-variant channels. In *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Toulouse, France, May 2006.

- C. Dumard and T. Zemen. Krylov subspace method based low-complexity MIMO multi-user receiver for time-variant channels. In *Proc. 17th IEEE International Symposium on Personal, Indoor and Mobile Radio Communication (PIMRC)*, Helsinki, Finland, September 2006, best student paper award.

- C. Dumard, F. Kaltenberger, and K. Freudenthaler. Low-cost LMMSE equalizer based on Krylov subspace methods for HSDPA. *IEEE Transactions on Wireless Communications*, 6(5):1610–1614, May 2007.

- C. Dumard and T. Zemen. Low-complexity MIMO multi-user receiver: A joint antenna detection scheme for time-varying channels. *IEEE Transactions on Signal Processing*, 56(7):2931–2940, July 2008.

- C. Dumard and T. Zemen. Sphere decoder for a MIMO multi-user MC-CDMA uplink in time-varying channels. In *Proc. 17th IEEE International Conference on Communications (ICC)*, Glasgow, Scotland, June 2007.

- C. Dumard and T. Zemen. Subspace based sphere decoder for MC-CDMA in time-varying MIMO channels. In *Proc. 18th IEEE International Symposium on Personal, Indoor and Mobile Radio Communication (PIMRC)*, Athens, Greece, September 2007.

- C. Dumard, J. Jaldén and T. Zemen. Soft sphere decoder for an iterative receiver in time-varying channels. In *16th European Signal Processing Conference (EUSIPCO)*, Lausanne, Switzerland, August 2008.

The content of individual chapters is briefly summarized here.
Part I describes the multi-user MIMO system used in this thesis, with exception for Chapter 5.
Chapter 2 provides an introduction to the uplink of an MC-CDMA system based on OFDM. Each user-receiver channel is a MIMO time-varying channel, and all users are supposed moving at the same velocity. Design of the transmitters and cooperation between transmit antennas for one particular user is detailed. The channel model based on Slepian basis expansion, allowing for low-complexity estimation of the projection coefficients is described. The multi-user detector performing parallel interference cancelation followed by LMMSE filtering is presented for various scenarios, i.e.

- parallel interference cancelation in the chip space, followed by either independent or joint processing of the received antennas (respectively single and joint antenna detection);

- parallel interference cancelation in the user space, allowing for joint detection of all transmitted signals using one LMMSE filter only.

The two parallel interference cancelation configurations are mathematically nearly equivalent, however they lead to noticeable difference in terms of complexity.

Considering the system described in Chapter 2, we aim at developing efficient low-complexity algorithms for the receiver. The remaining of the thesis consists in two parts, focusing on linear and non-linear detection respectively.
For linear detection in **Part II**, we consider Krylov subspace methods as iterative methods for the solution of linear systems of equations to approximate LMMSE filters.

Chapter 3 first gives an introduction to the Krylov subspace method (basic idea and recursive algorithm). Considering a generic LMMSE filter, explicit expressions for the computational complexity are detailed, using either exact computation of the LMMSE filter or its approximation by means of the Krylov subspace method. Closer look is given to its application to the iterative receiver described in Chapter2, and various initializations for the iterative Krylov algorithm are defined for the LMMSE filters for channel estimation and multi-user detection.

Simulations results are collected in **Chapter 4**. The simulation setup is described in details and bit error rate performance using Monte Carlo simulations are shown for a SISO and a 4×4 MIMO system. Computational complexity for the various LMMSE filters presented in Chapter 2 is discussed given closed-form expressions. Discussion of a trade-off between number of iterations of the receiver, computational complexity and performance loss using the Krylov subspace method in different scenarios, is run in order to find a 'best' solution depending on the system parameters. Using the Krylov subspace method in case of parallel interference in the chip space, parallelization of the computations into as many branches as transmit antennas is allowed, without global computational complexity reduction. For parallel interference cancelation in the user space, computational complexity is considerably reduced at the cost of a very slight loss in performance. The use of convenient initialization is shown to accelerate the convergence of the Krylov algorithm, further reducing computational complexity.

Another application of the Krylov subspace method for LMMSE is shown in **Chapter 5**, namely to High Speed Downlink Packet Access (HSDPA). The throughput of the HSDPA sub-system of UMTS usually suffers significantly from multiple access interference in the wireless channel. An LMMSE equalizer at the receiver allows achieving higher throughput than a conventional RAKE receiver, at the cost of higher complexity. In this chapter we study the implementation of the Krylov subspace method for this specific system. Slow variations of the channel can be exploited to allow acceleration of the Krylov subspace algorithm by using suitable initialization methods.

As a counterpart to linear detection, we discuss non-linear methods in **Part III**. Focus is made on maximum likelihood detection, and its low-complexity implementation using sphere decoding.

Chapter 6 gives an introduction to maximum likelihood detection and sphere decoding. The basic idea and the algorithm are described, and a brief classification of various search models is presented.

In **Chapter 7** we modify the conventional sphere decoder algorithm using a basis expansion channel model. We define a reduced-rank sphere decoder for time-varying channels, making use of the basis expansion to avoid full computations for each new realization of the channel. Here, the most expensive part of the computations are performed on a per OFDM block basis. A modified tree search algorithm using the reduced-rank representation of the channel is described. Finally, application of the reduced-rank sphere decoder to compute log-likelihood ratios providing soft outputs is detailed.

Simulation results and discussion on computational complexity are gathered in **Chapter 8**. Bit error rate performances of the sphere decoder providing hard or soft outputs are compared, and computational complexity using either the conventional or the reduced-rank approach is discussed. For relevancy, comparison to the detection using LMMSE as described in Chapter 2 is made as well.

Finally, **Chapter 9** concludes this thesis and summarizes the most important results of this work.

Part I

Multi-User MIMO System Description

Chapter 2
Multi-User MIMO-OFDM System

Throughout this thesis, we focus on time-varying wireless channels, and more specifically on the uplink of a multi-user system based on multi-carrier (MC) code-division multiple access (CDMA). With the exception of Chapter 5 which will deal with a different application, the system described in this chapter is the one used in all simulations.

2.1 Propagation Model

The time variation of the channel is due to user mobility and changing environment (i.e. for example scatterers' mobility), which is also assumed to introduce multi-path propagation.

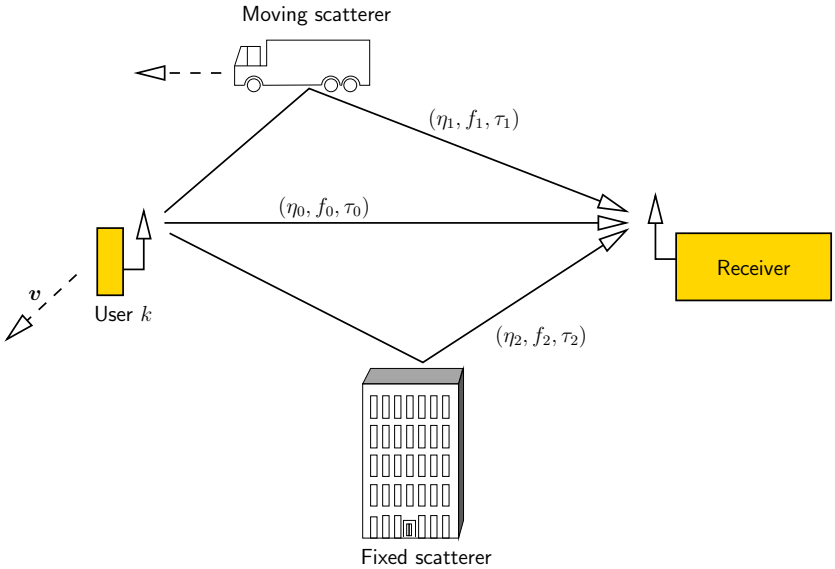


Figure 2.1: Time-varying multi-path propagation environment for one user.

Figure 2.1 shows the time-varying multi-path propagation environment [Jak74, VA03, Mol05]. Each path ℓ' is described through its attenuation $\eta_{\ell'}$, its Doppler shift $f_{\ell'}$ and its time delay $\tau_{\ell'}$. The users are moving at speed v. The time-varying channel impulse response considering only specular scattering is then given by the sum of contributions from L' different paths,

$$h(t,\tau) = \sum_{\ell'=0}^{L'-1} \eta_{\ell'} \mathrm{e}^{j2\pi f_{\ell'} t} \delta(\tau - \tau_{\ell'}) \,. \tag{2.1}$$

Throughout this thesis we consider the channel sampled in time and frequency obtained using

$$h[m,\ell] = h(mPT_{\mathrm{C}}, \ell T_{\mathrm{C}}) \,, \tag{2.2}$$

where P is the ratio of chip rate T_{C} to symbol rate T_{S}

$$P = \frac{T_{\mathrm{S}}}{T_{\mathrm{C}}} \,. \tag{2.3}$$

The realizations of the time-varying frequency-selective channel $h[m,\ell]$ follow an exponentially decaying power delay profile

$$\begin{aligned} \eta^2[\ell] &\triangleq \mathrm{E}\{|h[m,\ell]|^2\} \\ &= \mathrm{e}^{-\frac{\ell}{L_{\mathrm{D}}}} / \sum_{\ell'=0}^{L-1} \mathrm{e}^{-\frac{\ell'}{L_{\mathrm{D}}}} \,, \end{aligned} \tag{2.4}$$

where L_{D} denotes the root mean square delay spread normalized to the sample rate $1/T_{\mathrm{C}}$

$$L_{\mathrm{D}} = \frac{T_{\mathrm{D}}}{T_{\mathrm{C}}} \,. \tag{2.5}$$

The fading of one single tap $h[n,\ell]$ is assumed to follow Clarke's model [Cla68] under the wide sense stationary uncorrelated scattering (WSS-US) assumption, with correlation function

$$\begin{aligned} R_h[k,\ell] &\triangleq \mathrm{E}\{h[m,\ell]h^*[m+k,\ell]\} \\ &= \eta^2[\ell]\mathrm{J}_0(2\pi\nu_{\mathrm{D}}k) \,, \end{aligned} \tag{2.6}$$

where J_0 is the zeroth order Bessel function of the first kind. The normalized Doppler bandwidth ν_{D} depends on the velocity v of the user, the carrier frequency f_{C}, the speed of light c_0 and the symbol duration T_{S} according to

$$\nu_{\mathrm{D}} = \frac{v f_{\mathrm{C}}}{c_0} T_{\mathrm{S}} \,. \tag{2.7}$$

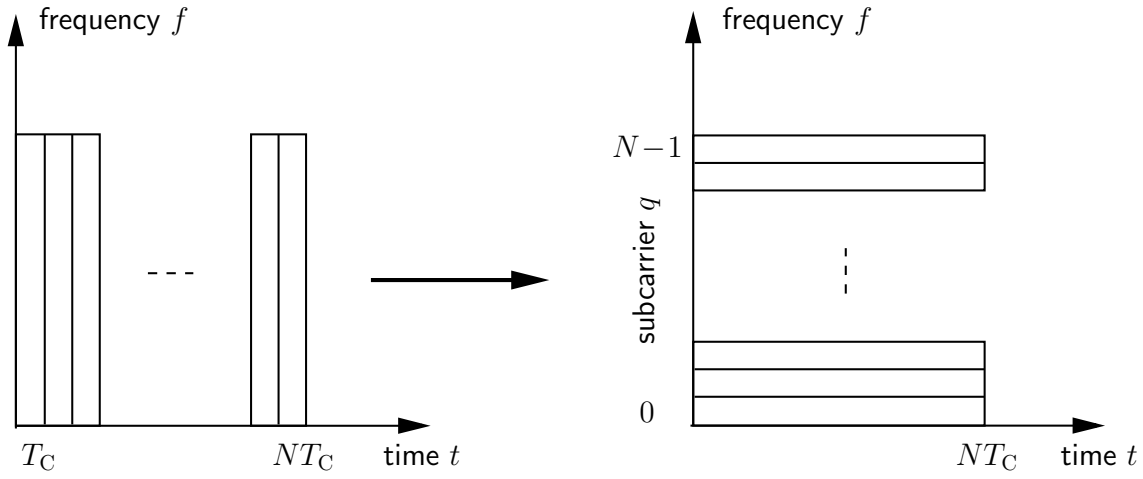


Figure 2.2: Multi-carrier vs. single-carrier system.

The correlation function (2.6) results in the Clarke's Doppler spectrum

$$S_{hh}[\nu,\ell] = \begin{cases} \frac{\eta^2[\ell]}{\pi\nu_{\mathrm{D}}\sqrt{1-\left(\frac{\nu}{\nu_{\mathrm{D}}}\right)^2}} & \text{for } |\nu| < \nu_{\mathrm{D}}\,, \\ 0 & \text{otherwise.} \end{cases} \tag{2.8}$$

2.2 Multi-User MIMO-OFDM

We present the general model based on orthogonal frequency division multiplexing (OFDM) of our multi-user multiple input multiple output (MIMO) system in this section.

The transmitted signal arriving at the receiver via multiple paths with different delays may experience inter symbol interferences (ISI). This ISI may increase with the number of paths L, which itself increases with the sample rate $1/T_{\mathrm{C}}$ [Zem04]. A possibility to avoid ISI without considerable increase in complexity is to perform OFDM. Instead of transmitting N symbols over the whole bandwidth and each symbol at a time, the same N symbols are transmitted simultaneously, each over one subcarrier, with an N-times increased symbol duration [WE71], as shown in Figure 2.2. This way the data rate is not affected. Further suppression of the ISI is performed using a cyclic prefix, being the tail of a symbol inserted before the symbol itself [Bin90].

The time-varying channel impulse response with L resolvable paths is defined as

$$\boldsymbol{h}[m] = \begin{bmatrix} h[m,0] \\ \vdots \\ h[m,L-1] \end{bmatrix} \in \mathbb{C}^{\,L} \tag{2.9}$$

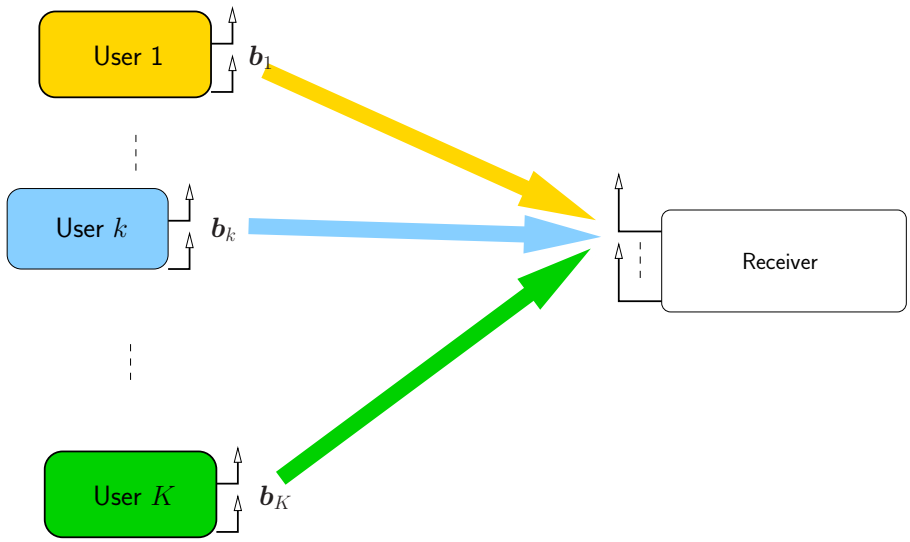


Figure 2.3: Multi-user MIMO uplink. User k sends the symbol vector $\boldsymbol{b}_k = [b_{(k,1)}, \ldots, b_{(k,T)}]^{\mathrm{T}}$ over its T transmit antennas.

and yields the time-varying frequency response according to

$$\boldsymbol{g}[m] = \sqrt{N}\boldsymbol{F}_{N\times L}\boldsymbol{h}[m]\,. \tag{2.10}$$

Making use of multiple antennas, both at the transmitter and receiver sides, MIMO systems allow increasing the data rate for a given signal-to-noise ratio (SNR), or, in other words, maintaining the data rate while lowering the SNR [PNG03]. Multiple antennas at the transmitter side allow sending independent streams over each transmit antenna, hence increasing the data rate. Transmit antenna diversity can be exploited by suitable design of the transmitted signals, making use for example of space-time codes [SW94, Ala98, TSC98]. At the receiver side, multiple antennas allow exploiting the correlation between all independent received signals. The global signal resulting from suitable combination of the received signals shows increase in the signal power proportionally to the number of receive antennas (array gain) and reduced amplitude variability (receive antenna diversity) [PNG03].

Obviously, the increasing parameters (for example number of antennas, users, subcarriers) enhance the complexity of the system, making the OFDM-MIMO multi-user system very complex to process at the receiver. In time-invariant channels, most of the tasks of the receiver (channel estimation, design of the multi-user detectors) can be pre-processed before treating each received symbol. In time-varying channels, each received symbol corresponds to a new channel realization and low-complexity processing

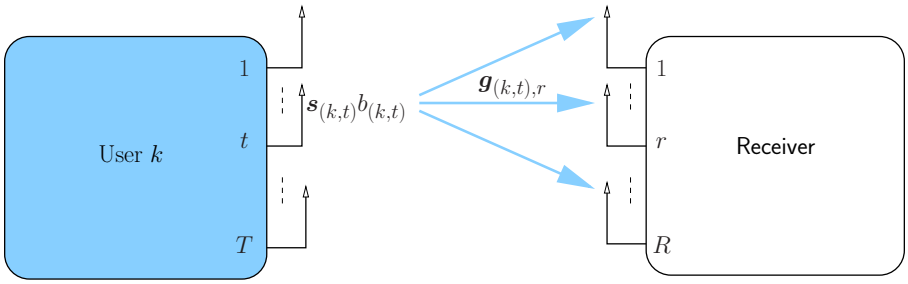


Figure 2.4: OFDM MIMO Channel for user k. Transmit antenna (k,t) sends the symbol $b_{(k,t)}$ spread over N subcarriers with the sequence $\boldsymbol{s}_{(k,t)}$. The channel coefficients between transmit antenna (k,t) and receive antenna r for N subcarriers are given in $\boldsymbol{g}_{(k,t),r}$.

is required to manage real-time feasibility. Low-complexity algorithms usually require approximations inducing a certain loss in performance, however trade-offs between reduced complexity and performance loss have to be made.

Throughout this thesis, we consider the uplink of a multi-user system, i.e. the transmission from $K \geq 1$ users to a base station. As shown in Figure 2.3, the receiver collects the K signals coming from the K users.

Each user is equipped with $T \geq 1$ transmit antennas, while the base station has $R \geq 1$ receive antennas. Thus the channel from one user to the base station is indeed a MIMO channel, as detailed in Figure 2.4. The transmit antenna t of user k, denoted through the transmit antenna (k,t) in the following, sends the symbol $b_{(k,t)}$ spread over the N subcarriers using the spreading sequence $\boldsymbol{s}_{(k,t)} \in \mathbb{C}^N$.

For sake of clarity, we briefly summarize the meaning and range of the main indices used in the following in Table 2.1.

Index	Range	Meaning
k	$\{1,\ldots,K\}$	user
t	$\{1,\ldots,T\}$	transmit antenna
r	$\{1,\ldots,R\}$	receive antenna
m	$\{0,\ldots,M-1\}$	symbol (=time)
q	$\{0,\ldots,N-1\}$	subcarrier

Table 2.1: Main indices used throughout the thesis.

The time-varying channels are frequency selective, however, using OFDM over N subcarriers allows considering time-varying frequency flat channels over each subcarrier.

The channels are assumed to be time-varying over a block of M OFDM symbols, but the duration T_{S} of one OFDM symbol $m \in \{1, \ldots, M\}$ is supposed small enough to allow considering constant channels for the duration of this symbol. We denote through

$$\boldsymbol{g}_{(k,t),r}[m] \in \mathbb{C}^N \tag{2.11}$$

with elements $g_{(k,t),r}[m, q]$ the vector containing the channel coefficients for symbol m, over all N subcarriers, between transmit antenna (k, t) and receive antenna r, and defined as in (2.10). For an illustration, see Figure 2.4.

In a single antenna system, the load of the system

$$\zeta = \frac{K}{N} \tag{2.12}$$

is defined as the ratio between the number of users to the numbers of subcarriers.

In a MIMO system, every transmit antenna can be considered as an independent virtual user, from a receiver point-of-view. Furthermore, it is known that a MIMO system with R receive antennas and spreading factor N behaves as a single-antenna system with spreading factor NR [HT01]. This allows defining a virtual number of users KT as well as a virtual spreading factor NR, and we extend the definition (2.12) to the load of a MIMO system

$$\zeta = \frac{KT}{NR}\,. \tag{2.13}$$

In the remainder of this chapter, we focus on how the multiple antenna transceivers are designed. We detail the signal processing steps at the receiver, and the following chapters of the thesis will describe in more detail different possibilities to reduce complexity at the receiver while limiting the loss in performance.

2.3 Multiple Antenna Transmitter

Transmit antenna $(k, t) \in \{1, \ldots, K\} \times \{1, \ldots, T\}$ transmits symbols $b_{(k,t)}[m]$ with symbol rate $1/T_{\mathrm{S}}$, where m denotes symbol time and T_{S} the symbol duration. Each symbol $b_{(k,t)}[m]$ is spread over all N subcarriers by the random spreading sequence $\boldsymbol{s}_{(k,t)} \in \mathbb{C}^N$ with independent identically distributed elements from the set $\{\pm 1 \pm \mathrm{j}\}/\sqrt{2N}$. The number of subcarriers N is also the spreading sequence length of the system.

Allowing room for insertion of J pilot symbols, the $M - J$ data symbols $b_{(k,t)}[m]$ result from a binary information sequence of length $2(M - J)R_C$ by convolutional encoding, random interleaving and QPSK modulation with Gray labeling [ALSO04, Gra53]. We

consider a non-systematic, non-recursive, 4 state, rate $R_C = 1/2$ code with generator polynomial $(5, 7)_8$.

The $M - J$ data symbols are distributed over a block of length M fulfilling

$$\begin{aligned} b_{(k,t)}[m] \in \{\pm 1 \pm \mathrm{j}\}/\sqrt{2} \quad \text{for} \quad m \notin \mathcal{P}\,, \\ b_{(k,t)}[m] = 0 \quad \text{for} \quad m \in \mathcal{P}\,, \end{aligned} \tag{2.14}$$

where $\mathcal{P}$ denotes the pilot index set.

At each transmitter (i.e. user), we consider two spatial modes of encoding, as described below:

• **Independent Encoding:** The independent encoding transmitter structure is shown in Figure 2.5. The T data blocks of one user are encoded, interleaved, mapped to the QPSK constellation, spread and transmitted independently of each other. This means that each transmit antenna (k, t) can be considered a single antenna transmitter.

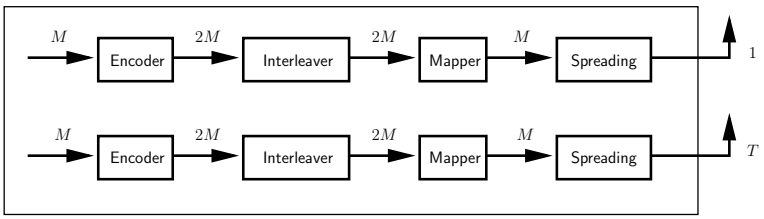


Figure 2.5: Independent encoding for T transmit antennas.

• **Joint Encoding:** The joint encoding transmitter structure is shown in Figure 2.6. The T data blocks of one user are jointly encoded, interleaved and mapped. Then, the resulting $T(M-J)$ coded symbols are split into coded symbol blocks of duration $M-J$ that are finally independently spread to be transmitted over their corresponding antenna.

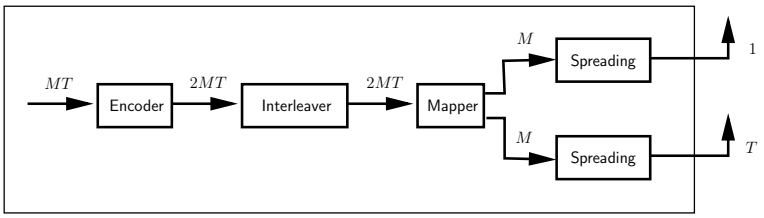


Figure 2.6: Joint encoding for T transmit antennas.

In the following, we will restrict ourselves to the case where the encoding scheme is the same for all users. Regardless of the encoding scheme used, the pilot symbols

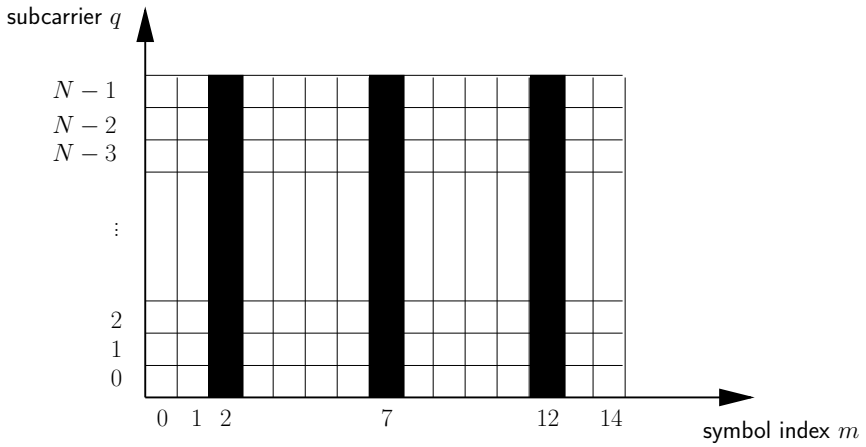


Figure 2.7: Example of pilot placement for the pattern $\mathcal{P} = \{2, 7, 12\}$ for N subcarriers, block length $M = 15$ and $J = 3$ pilots.

$\boldsymbol{p}_{(k,t)}[m] \in \mathbb{C}^N$ are added after spreading so that the transmitted data reads

$$\boldsymbol{d}_{(k,t)}[m] = \boldsymbol{s}_{(k,t)} b_{(k,t)}[m] + \boldsymbol{p}_{(k,t)}[m] \,. \tag{2.15}$$

In one block of length M, $M - J$ symbols are dedicated to data sent by the user while the remaining J symbols are used for pilot symbols, allowing for channel estimation at the receiver. The J pilot symbols are distributed over a block in a regular manner, using the pilot index set given by

$$\mathcal{P} = \left\{ \left\lfloor i\frac{M}{J} + \frac{M}{2J} \right\rfloor \;\middle|\; i \in \{0, \ldots, J-1\} \right\} . \tag{2.16}$$

This set is identical for all subcarriers, i.e. at time instant $m \in \mathcal{P}$, pilots are sent over all subcarriers. Figure 2.7 gives an illustration of the pilot placement.

For $m \in \mathcal{P}$ and $q \in \{0, \ldots, N-1\}$, the elements $p_{(k,t)}[m, q]$ of the pilot symbol vector $\boldsymbol{p}_{(k,t)}[m]$ are randomly chosen from the QPSK symbol set

$$\frac{\{\pm 1 \pm \mathrm{j}\}}{\sqrt{2N}} \,. \tag{2.17}$$

Otherwise $\boldsymbol{p}_{(k,t)}[m] = \mathbf{0}_N$ for $m \notin \mathcal{P}$.

At each transmit antenna an N-point inverse discrete Fourier transform (DFT) is performed on $\boldsymbol{d}_{(k,t)}[m]$ and a cyclic prefix of length G is inserted [TV05]. A single OFDM symbol together with the cyclic prefix has length $P = N + G$ chips. After parallel to serial conversion the chip stream with chip rate $1/T_{\mathrm{C}} = P/T_{\mathrm{S}}$ is transmitted over a time-varying multi-path fading channel with L resolvable paths, see Section 2.1.

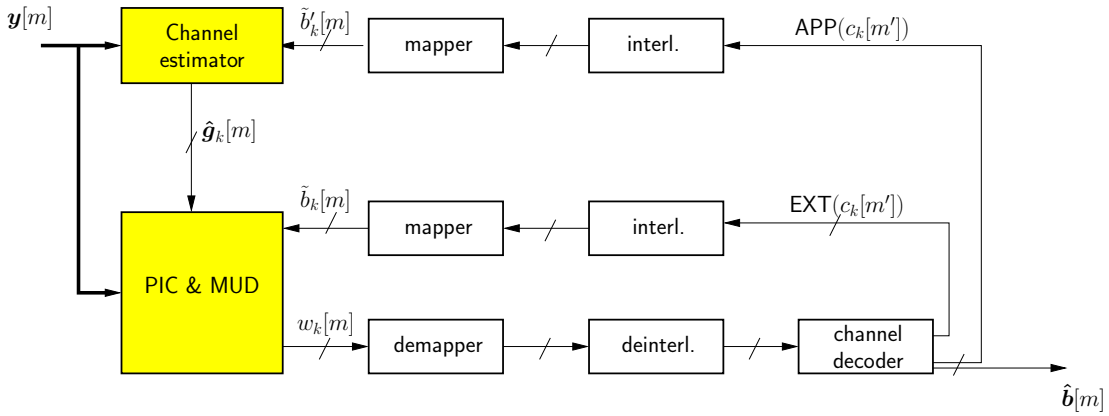


Figure 2.8: MC-CDMA receiver. The receiver performs iterative time-varying channel estimation jointly for all users and independently per receive antenna as well as multi-user detection.

2.4 Multiple Antenna Receiver

The *maximum a posteriori* (MAP) detector [DeG70, Sor80] is prohibitively complex to solve in real systems, and further complicated by channel uncertainty [TB05]. A suboptimal approach is an iterative detector performing jointly channel estimation, data detection and decoding. The channel estimator and the multi-user detector support each other by exchanging soft information, based on the Turbo principle [ARAS99, TKS02, KST04]. This allows for a system supporting high loads at acceptable complexity.

This kind of receivers has been extensively analyzed in different scenarios, for example for single [Weh05] or multiple [LMW+04] antenna transceivers, over frequency-flat [KBC01] or frequency-selective [OT04] channels.

In most cases, soft feedback information is used to improve the quality of the channel state information compared to pilot-aided only channel estimation, by properly weighting reliable or unreliable symbols [ZMWM06, KBC01, TOS02, NFS07]. What kind of soft information to use from the channel estimator to the multi-user detector and back is a rather unclear question, although the authors in [HLR+08] try to develop a formal optimization framework for handling soft-symbol processing. In their analysis, the iterative receiver presented in [WM06], used in this thesis, is shown to be second best, with very slight loss in performance at high SNR compared to what the authors in [HLR+08] propose.

Our receiver depicted in Figure 2.8 performs channel estimation using *a priori* probabilities (APP) from the decoding stage and pilot symbols. The channel estimates are used to perform multi-user detection after parallel interference cancelation, using the *extrinsic*

probabilities (EXT) from the decoding stage of the previous iteration. The processing at the receiver is described in more detail in the following.

At each antenna $r \in \{1, \ldots, R\}$ of the receiver, the contributions of all KT transmit antennas add up. This way the received signal at the output of the OFDM demodulator at antenna r can be written as

$$y_r[m,q] = \sum_{k=1}^{K} \sum_{t=1}^{T} g_{(k,t),r}[m,q] d_{(k,t)}[m,q] + n_r[m,q]\,, \tag{2.18}$$

where $n_r[m,q]$ is additive white Gaussian noise with variance σ_n^2, $d_{(k,t)}[m,q]$ are the elements of $\boldsymbol{d}_{(k,t),r}[m]$ defined in (2.15) and $g_{(k,t),r}[m,q]$ represents the channel between one specific transmit antenna (k,t) and one specific receive antenna r at frequency $q \in \{0, \ldots, N-1\}$.

2.5 Channel Decoder

To compute soft output values of the received code symbols we use a BCJR decoder [BCJR74]. Given the detected coded bits $c_{(k,t)}[m']$, the BCJR decoder provides *extrinsic* and *a posteriori* probabilities, denoted through $\Pr^{(\mathrm{EXT})}\left\{c_{(k,t)}[m']\right\}$ and $\Pr^{(\mathrm{APP})}\left\{c_{(k,t)}[m']\right\}$, respectively.

The detected coded bits $c_{(k,t)}[m']$ are obtained after QPSK demapping and deinterleaving of the detected symbols provided by the multi-user detector.

To perform channel estimation, soft symbol estimates are needed, that are computed from the *a posteriori* probabilities supplied by the decoding stage. The multi-user detector makes use of the *extrinsic* probabilities to perform interference cancelation. We use the notation EXT and APP to differentiate *extrinsic* and *a posteriori* soft symbols.

The *extrinsic* soft symbol resulting from QPSK modulation with Gray labeling is given by

$$\tilde{b}_{(k,t)}[m] = \frac{1}{\sqrt{2}}\left(2\mathrm{EXT}(c_{(k,t)}[2m]) - 1\right) + \mathrm{j}\frac{1}{\sqrt{2}}\left(2\mathrm{EXT}(c_{(k,t)}[2m+1]) - 1\right)\,, \tag{2.19}$$

where

$$\begin{aligned} \mathrm{EXT}(c_{(k,t)}[m]) &= \Pr^{(\mathrm{EXT})}\left\{c_{(k,t)}[m] = +1\right\} - \Pr^{(\mathrm{EXT})}\left\{c_{(k,t)}[m] = -1\right\} \\ &= 2\Pr^{(\mathrm{EXT})}\left\{c_{(k,t)}[m] = +1\right\} - 1\,. \end{aligned} \tag{2.20}$$

In a similar way, we compute the *a posteriori* soft symbols needed for parallel interference cancelation using

$$\tilde{b}'_{(k,t)}[m] = \frac{1}{\sqrt{2}}\left(2\mathrm{APP}(c_{(k,t)}[2m]) - 1\right) + \mathrm{j}\frac{1}{\sqrt{2}}\left(2\mathrm{APP}(c_{(k,t)}[2m+1]) - 1\right)\,, \tag{2.21}$$

where $\mathrm{APP}(c_{(k,t)}[m])$ is given by

$$\begin{aligned}\mathrm{APP}(c_{(k,t)}[m]) &= \Pr^{(\mathrm{APP})}\left\{c_{(k,t)}[m]=+1\right\} - \Pr^{(\mathrm{APP})}\left\{c_{(k,t)}[m]=-1\right\} \\ &= 2\Pr^{(\mathrm{APP})}\left\{c_{(k,t)}[m]=+1\right\} - 1\,.\end{aligned} \tag{2.22}$$

From iteration to iteration, the *extrinsic* and *a posteriori* probabilities get better, yielding more accurate channel estimation and almost perfect parallel interference cancelation. When interference cancelation is perfect, the multi-user detector reduces to a bunch of single-user detectors.

2.6 Iterative Time-Varying Channel Estimation

The transmission of KT symbols at time instant m is done over K independent $T \times R$ MIMO OFDM channels. These KTR channels are assumed uncorrelated and can be estimated at each receive antenna independently without loss of information. Hence, each receive antenna has its own channel estimator. For this reason, we drop the receive antenna index r in the remainder of this section.

As a first step, the cyclic prefix is removed and a DFT is performed. The performance of the iterative receiver depends on the channel estimates for the time-varying frequency response $\boldsymbol{g}_{(k,t),r}[m]$ since the effective spreading sequence $\tilde{\boldsymbol{s}}_{(k,t),r}$ directly depends on the actual channel realization.

2.6.1 Basis Expansion Model for Low-Complexity Channel Estimation

The maximum variation in time of the wireless channel is upper bounded by the maximum normalized one-sided Doppler bandwidth

$$\nu_{\mathrm{Dmax}} = \frac{v_{\max} f_{\mathrm{C}}}{c_0} T_{\mathrm{S}}\,, \tag{2.23}$$

where $v_{\max}$ is the maximum user velocity, T_{S} is the OFDM symbol duration, f_{C} is the carrier frequency and c_0 the speed of light. Time-limited snapshots of the band-limited fading process belong to a subspace with very small dimensionality. This subspace is spanned by discrete prolate spheroidal (DPS) sequences [Sle78, ZM05].

The DPS sequences $\{u_i[m]\}$ are defined as the real solutions of the eigenproblem

$$\lambda_i u_i[m] = \sum_{\ell=0}^{M-1} \frac{\sin(2\pi\nu_{\mathrm{Dmax}}(\ell-m))}{\pi(\ell-m)} u_i[\ell]\,, \tag{2.24}$$

for $i \in \mathcal{I}_M \triangleq \{0, \ldots, M-1\}$ and $m \in \{-\infty, \ldots, \infty\} = \mathbb{Z}$. The eigenvalues λ_i are ordered such that $\lambda_1 > \lambda_2 \ldots > \lambda_M$.

The sequences $\{u_i[m]\}$ have the following properties: they are doubly orthogonal over the infinite set $\mathbb{Z}$ and the finite set $\mathcal{I}_M$, i.e.

$$\sum_{m=0}^{M-1} u_i[m]u_\ell[m] = \sum_{m=-\infty}^{\infty} u_i[m]u_\ell[m] = \delta_{i\ell}\,, \tag{2.25}$$

band-limited with bandwidth ν_{Dmax} and maximally energy-concentrated on $\mathcal{I}_M$.

We are interested in describing the time-varying frequency selective channel $\boldsymbol{g}_{(k,t)}[m]$ for the duration of a single data block $\mathcal{I}_M$ and use for that purpose a Slepian basis expansion [ZM05]. The Slepian basis functions are defined by the DPS sequences

$$\boldsymbol{u}_i = [u_i[0], \ldots, u_i[M-1]]^{\mathrm{T}}\,, \tag{2.26}$$

for $i \in \{0, \ldots, D-1\}$, where D denotes the number of sequences used (i.e. dimension of the projection subspace). Following (2.24), the Slepian basis functions are the eigenvectors of the matrix $\boldsymbol{C} \in \mathbb{R}^{M\times M}$ with elements

$$[\boldsymbol{C}]_{i,\ell} = \frac{\sin\left(2\pi(i-\ell)\nu_{\mathrm{Dmax}}\right)}{\pi(i-\ell)}\,. \tag{2.27}$$

Finally, if h denote a time-varying function and $\tilde{h}$ its approximation using a basis expansion model over the time set $\mathcal{I}_M$, the bias is defined as the mean square error (MSE) in the absence of noise [Sch91] given by

$$\mathrm{MSE} = \frac{1}{M}\sum_{m=0}^{M-1} \mathrm{E}\left\{\left|h[m] - \tilde{h}[m]\right|^2\right\}\,. \tag{2.28}$$

The bias depends on the basis expansion used and is proven to be at least one magnitude smaller using a Slepian basis expansion than with a Fourier basis expansion [Zem04].

The time-varying frequency selective channel $\boldsymbol{g}_{(k,t)}[m]$ for the duration of a single data block $\mathcal{I}_M$ is projected onto the subspace spanned by the first D Slepian sequences, and is thus approximated as

$$g_{(k,t)}[m,q] \approx \tilde{g}_{(k,t)}[m,q] = \sum_{i=0}^{D-1} u_i[m]\psi_{(k,t)}[i,q]\,, \tag{2.29}$$

where $\boldsymbol{\psi}_{(k,t)}[q]$ contains the projection coefficients.

The above expression can be rewritten in a simpler way using matrix notation

$$\boldsymbol{g}_{(k,t)}[m] \approx \boldsymbol{\psi}_{(k,t)} \boldsymbol{f}[m]\,, \tag{2.30}$$

where

$$\boldsymbol{\psi}_{(k,t)} = \begin{bmatrix} \psi_{(k,t)}[0,0] & \cdots & \psi_{(k,t)}[D-1,0] \\ \vdots & \psi_{(k,t)}[i,q] & \vdots \\ \psi_{(k,t)}[0,N-1] & \cdots & \psi_{(k,t)}[D-1,N-1] \end{bmatrix} \in \mathbb{C}^{N \times D} \tag{2.31}$$

and

$$\boldsymbol{f}[m] = [u_0[m], \ldots, u_{D-1}[m]]^{\mathrm{T}} \in \mathbb{R}^{D} \tag{2.32}$$

contains the basis expansion at time m.

The dimension D of this basis expansion is chosen as

$$\lceil 2\nu_{\mathrm{Dmax}} M \rceil + 1 \leq D \ll M - 1\,. \tag{2.33}$$

Substituting the basis expansion (2.29) for the time-varying subcarrier coefficients $g_{(k,t)}[m,q]$ into the received signal (2.18), we finally obtain

$$y[m,q] = \sum_{k=1}^{K} \sum_{t=1}^{T} \sum_{i=0}^{D-1} u_i[m] \psi_{(k,t)}[i,q] d_{(k,t)}[m,q] + n[m,q]\,. \tag{2.34}$$

2.6.2 Estimation of the Projection Coefficients

For channel estimation, the $M-J$ unknown symbols are replaced by soft symbols, that are calculated from the *a posteriori* probabilities using (2.21). At the first iteration of the receiver, where no feedback is available yet, they are simply set to zero. This enables us to obtain refined channel estimates when the soft symbols get more reliable from iteration to iteration.

Since channel estimation is performed for every subchannel $q \in \{1, \ldots, N\}$ independently, we drop the index q in the following in order to simplify the equations. At each receive antenna, the subcarrier coefficients $\psi_{(k,t)}[i]$ can be obtained jointly for all KT virtual users. We define the vectors

$$\boldsymbol{\psi}_d = [\psi_{(1,1)}[d], \ldots, \psi_{(1,T)}[d], \ldots, \psi_{(K,1)}[d], \ldots, \psi_{(K,T)}[d]]^{\mathrm{T}} \in \mathbb{C}^{KT}\,, \tag{2.35}$$

for $d \in \{0, \ldots, D-1\}$ containing all coefficients for one DPS sequence and

$$\boldsymbol{\phi} = [\boldsymbol{\psi}_0^{\mathrm{T}}, \ldots, \boldsymbol{\psi}_{D-1}^{\mathrm{T}}]^{\mathrm{T}} \in \mathbb{C}^{KTD}\,, \tag{2.36}$$

containing the basis expansion coefficients of all KT virtual users. The received symbol sequence of each single data block is given by

$$\boldsymbol{y} = [y[0], \ldots, y[M-1]]^{\mathrm{T}} \in \mathbb{C}^{M} . \tag{2.37}$$

Using these definitions we rewrite (2.34) as

$$\boldsymbol{y} = \tilde{\boldsymbol{\mathcal{D}}}\boldsymbol{\phi} + \boldsymbol{z} , \tag{2.38}$$

where

$$\tilde{\boldsymbol{\mathcal{D}}} = \left[\mathrm{diag}\left(\boldsymbol{u}_0\right)\tilde{\boldsymbol{D}}, \ldots, \mathrm{diag}\left(\boldsymbol{u}_{D-1}\right)\tilde{\boldsymbol{D}}\right] \in \mathbb{C}^{M\times KTD} . \tag{2.39}$$

The matrix $\tilde{\boldsymbol{D}} \in \mathbb{C}^{M\times KT}$ contains all the transmitted symbols at all time instants $m \in \mathcal{I}_M$

$$\tilde{\boldsymbol{D}} = \begin{bmatrix} \tilde{d}'_{(1,1)}[0] & \ldots & \tilde{d}'_{(k,t)}[0] & \ldots & \tilde{d}'_{(K,T)}[0] \\ \vdots & & \vdots & & \vdots \\ \tilde{d}'_{(1,1)}[M-1] & \ldots & \tilde{d}'_{(k,t)}[M-1] & \ldots & \tilde{d}'_{(K,T)}[M-1] \end{bmatrix} . \tag{2.40}$$

Here,

$$\tilde{d}'_{(k,t)}[m] = s_{(k,t)}\tilde{b}'_{(k,t)}[m] + p_{(k,t)}[m] \tag{2.41}$$

is computed using the *a posteriori* soft symbols provided by the decoding stage (2.21).

To summarize, $\boldsymbol{\phi}$ contains all projection coefficients for all K users, all M symbols $m \in \mathcal{I}_M$ and all D DPS sequences. A linear minimum mean square error (LMMSE) estimator of these coefficients, allowing for channel estimation on a per subcarrier basis, is expressed as

$$\hat{\boldsymbol{\phi}} = \left(\tilde{\boldsymbol{\mathcal{D}}}^{\mathrm{H}}\boldsymbol{\Delta}^{-1}\tilde{\boldsymbol{\mathcal{D}}} + \boldsymbol{\mathcal{C}}_{\phi}^{-1}\right)^{-1}\tilde{\boldsymbol{\mathcal{D}}}^{\mathrm{H}}\boldsymbol{\Delta}^{-1}\boldsymbol{y} , \tag{2.42}$$

where

$$\boldsymbol{\Delta} \triangleq \boldsymbol{\Lambda} + \sigma_n^2\boldsymbol{I}_M \in \mathbb{C}^{M\times M} \tag{2.43}$$

is diagonal. The elements of $\boldsymbol{\Lambda} \in \mathbb{C}^{M\times M}$ are defined as

$$[\boldsymbol{\Lambda}]_{mm} = \frac{1}{N}\sum_{k=1}^{K}\sum_{t=1}^{T}\sum_{i=0}^{D-1}\lambda_i u_i^2[m](1 - |\tilde{b}'_{(k,t)}[m]|^2) . \tag{2.44}$$

The diagonal covariance matrix $\boldsymbol{C}_{\phi}$ for $\boldsymbol{\phi_q}$ is given by

$$\boldsymbol{C}_{\phi} = \frac{1}{2\nu_{\mathrm{Dmax}}}\boldsymbol{I}_{KT} \otimes \mathrm{diag}([\lambda_0, \ldots, \lambda_{D-1}]) , \tag{2.45}$$

where $\otimes$ denotes the Kronecker product [Sch66].

Using the estimates $\hat{\phi}_q$ given by (2.42), we obtain channel estimates $\hat{\boldsymbol{g}}'_{(k,t)}[m]$ according to (2.30). Exploiting the correlation between the subcarriers to suppress further noise, we obtain an estimate for the channel between transmit antenna (k,t) and receive antenna r

$$\hat{\boldsymbol{g}}_{(k,t)}[m] = \boldsymbol{F}_{N\times L}\boldsymbol{F}^{\mathrm{H}}_{N\times L}\hat{\boldsymbol{g}}'_{(k,t)}[m]\,. \tag{2.46}$$

More details on the channel estimator can be found in [Zem04].

2.7 Multi-User Detection

For multi-user detection we need to distinguish between the receive antennas. For this purpose we re-introduce the receive antenna index r in the equations.

Now assuming we have channel estimates from (2.46), we can rewrite the received signal (2.18) as

$$\boldsymbol{y}_r[m] = \sum_{k=1}^{K}\sum_{t=1}^{T} \operatorname{diag}\left(\hat{\boldsymbol{g}}_{(k,t),r}[m]\right)\boldsymbol{d}_{(k,t)}[m] + \boldsymbol{n}_r[m]\,, \tag{2.47}$$

where $\boldsymbol{n}_r[m]$ is complex additive white Gaussian noise with zero mean and covariance $\sigma_n^2\boldsymbol{I}_N$. The estimated effective spreading sequence at time m for the channel $\boldsymbol{g}_{(k,t),r}$ is

$$\tilde{\boldsymbol{s}}_{(k,t),r}[m] = \operatorname{diag}\left(\hat{\boldsymbol{g}}_{(k,t),r}[m]\right)\boldsymbol{s}_{(k,t)} \in \mathbb{C}^{N}\,. \tag{2.48}$$

Unless necessary, we will omit the time-index m for sake of clarity. The time-varying effective spreading matrix containing the KT spreading sequences at receive antenna r is given by concatenation of all effective spreading sequences (2.48) according to

$$\tilde{\boldsymbol{S}}_r = [\underbrace{\tilde{\boldsymbol{s}}_{(1,1),r},\ldots,\tilde{\boldsymbol{s}}_{(1,T),r}}_{\text{user }1},\ldots,\tilde{\boldsymbol{s}}_{(k,t),r},\ldots,\underbrace{\tilde{\boldsymbol{s}}_{(K,1),r},\ldots,\tilde{\boldsymbol{s}}_{(K,T),r}}_{\text{user }K}]\,. \tag{2.49}$$

Using these definitions the signal (2.47) writes for $r \in \{1,\ldots,R\}$

$$\boldsymbol{y}_r = \tilde{\boldsymbol{S}}_r\boldsymbol{b} + \boldsymbol{n}_r\,, \tag{2.50}$$

where

$$\boldsymbol{b} = \left[b_{(1,1)},\ldots,b_{(1,T)},\ldots,b_{(K,1)},\ldots,b_{(K,T)}\right]^{\mathrm{T}} \in \mathbb{C}^{KT} \tag{2.51}$$

contains the data symbols for the KT transmit antennas.

Multi-user detection is performed using the soft symbol estimates $\tilde{b}_{(k,t)}$, computed from the *extrinsic* probabilities as in (2.19). We define

$$\tilde{\boldsymbol{b}} = \left[\tilde{b}_{(1,1)}, \ldots, \tilde{b}_{(1,T)}, \ldots, \tilde{b}_{(K,1)}, \ldots, \tilde{b}_{(K,T)}\right]^{\mathrm{T}} \in \mathbb{C}^{KT}, \tag{2.52}$$

and the error covariance matrix of the soft symbols

$$\boldsymbol{V} = \mathrm{E}\{(\boldsymbol{b} - \tilde{\boldsymbol{b}})(\boldsymbol{b} - \tilde{\boldsymbol{b}})^{\mathrm{H}}\} \tag{2.53}$$

with diagonal elements

$$V_{(k,t)} = \mathrm{E}\{1 - |\tilde{b}_{(k,t)}|^2\} \tag{2.54}$$

that are constant during one iteration of the receiver. The expectation operator in (2.54) is implemented as empirical mean according to

$$\mathrm{E}\{1 - |\tilde{b}_{(k,t)}|^2\} = \frac{1}{M} \sum_{m=0}^{M-1} (1 - |\tilde{b}_{(k,t)}|^2)\,. \tag{2.55}$$

The symbols $\boldsymbol{b}$ and $\tilde{\boldsymbol{b}}$ are supposed to be independent and the other elements of $\boldsymbol{V}$ are assumed to be zeros.

In the receiver, the transmission encoding scheme (i.e. independent or joint, see Section 2.3) is taken into account by appropriate demapping, deinterleaving and decoding. For multi-user detection, we compare two different architectures, as described in the following.

2.7.1 Single Antenna Detection

For single antenna detection (SAD) the received signal at each receive antenna is processed independently. This is equivalent to having multiple independent receivers, as depicted in Figure 2.9. The R outputs of these sub-receivers are combined using maximum ratio combining (MRC) providing the final symbol estimates.

We perform soft parallel interference cancelation (PIC) [VA90] for transmit antenna t of user k at receive antenna r, by removing the contribution of all other transmit antennas using the soft symbols (2.19)

$$\tilde{\boldsymbol{y}}_{(k,t),r} = \boldsymbol{y}_r + \tilde{\boldsymbol{s}}_{(k,t),r}\tilde{b}_{(k,t)} - \tilde{\boldsymbol{S}}_r\tilde{\boldsymbol{b}}\,. \tag{2.56}$$

To obtain improved soft symbols, we use the unbiased conditional LMMSE filter given by [CMT04, Hay05]

$$\boldsymbol{f}_{(k,t),r} = \frac{\left(\sigma_n^2\boldsymbol{I}_N + \tilde{\boldsymbol{S}}_r\boldsymbol{V}\tilde{\boldsymbol{S}}_r^{\mathrm{H}}\right)^{-1}\tilde{\boldsymbol{s}}_{(k,t),r}}{\tilde{\boldsymbol{s}}_{,r}{}^{\mathrm{H}}\left(\sigma_n^2\boldsymbol{I}_N + \tilde{\boldsymbol{S}}_r\boldsymbol{V}\tilde{\boldsymbol{S}}_r^{\mathrm{H}}\right)^{-1}\tilde{\boldsymbol{s}}_{(k,t),r}}\,. \tag{2.57}$$

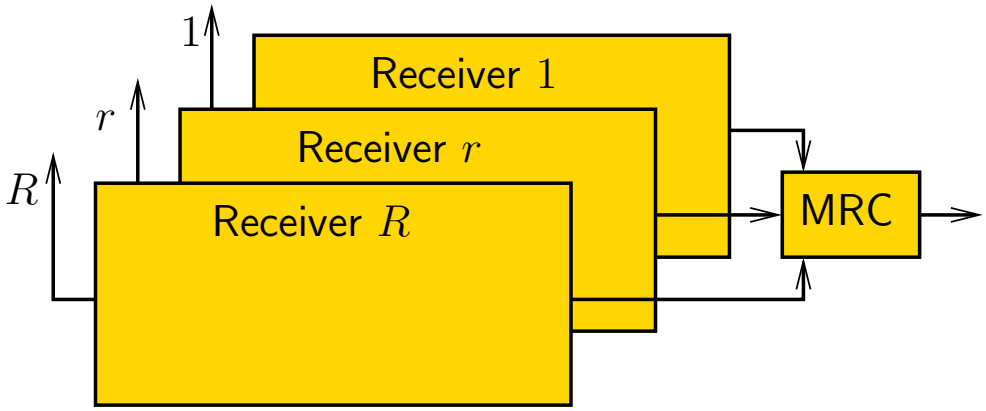


Figure 2.9: Single antenna detection receiver.

The output of the filter is an estimate of $b_{(k,t)}$ for receive antenna r expressed by

$$\hat{w}_{(k,t),r} = \boldsymbol{f}^{\mathrm{H}}_{(k,t),r} \tilde{\boldsymbol{y}}_{(k,t),r} \,. \tag{2.58}$$

After independent processing at each receive antenna, the R estimates $\hat{w}_{(k,t),r}$ of $b_{(k,t)}$ are combined using maximum ratio combining, leading to the final estimate

$$w_{(k,t)} = \frac{\sum_{r=1}^{R} ||\hat{\boldsymbol{g}}_{(k,t),r}||^2 \hat{w}_{(k,t),r}}{\sum_{r=1}^{R} ||\hat{\boldsymbol{g}}_{(k,t),r}||^2} \,. \tag{2.59}$$

The $w_{(k,t)}$ are decoded by a BCJR decoder after demapping and deinterleaving, providing *a priori* and *extrinsic* probabilities for the following receiver iteration.

Single antenna detection corresponds to the simplest multi-antenna receiver that can be built from a single antenna receiver. However it does not take into account correlation between the R received signals coming from the same sources. Exploiting the multi-antenna structure of the receiver is done by combining first all received signals before any processing, as is presented in the next Section.

2.7.2 Joint Antenna Detection

For joint antenna detection (JAD), all received signals are combined into one global signal allowing for joint processing. It is well known that a CDMA system with spreading factor N and R receive antennas may behave like a single antenna system with spreading factor NR (*resource pooling* result [HT01]). Hence, maximal advantage of the multi-antenna structure of the receiver is taken, allowing virtual increase of the spreading

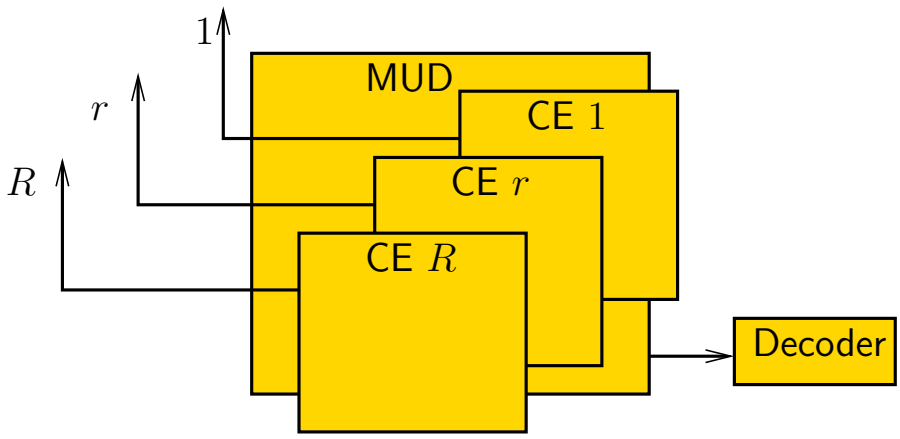


Figure 2.10: Joint antenna detection receiver with one channel estimator (CE) per antenna and a common multi-user detector (MUD).

factor [HT01, DZ06b]. However, it usually results in systems with high dimensionality and complex processing at the receiver.

The R received signals (2.50) are concatenated in one global received signal

$$\boldsymbol{y} = \begin{bmatrix} \boldsymbol{y}_1 \\ \vdots \\ \boldsymbol{y}_r \\ \vdots \\ \boldsymbol{y}_R \end{bmatrix} = \begin{bmatrix} \tilde{\boldsymbol{S}}_1\boldsymbol{b} + \boldsymbol{n}_1 \\ \vdots \\ \tilde{\boldsymbol{S}}_r\boldsymbol{b} + \boldsymbol{n}_1 \\ \vdots \\ \tilde{\boldsymbol{S}}_R\boldsymbol{b} + \boldsymbol{n}_1 \end{bmatrix} = \begin{bmatrix} \tilde{\boldsymbol{S}}_1 \\ \vdots \\ \tilde{\boldsymbol{S}}_r \\ \vdots \\ \tilde{\boldsymbol{S}}_R \end{bmatrix} \boldsymbol{b} + \begin{bmatrix} \boldsymbol{n}_1 \\ \vdots \\ \boldsymbol{n}_r \\ \vdots \\ \boldsymbol{n}_R \end{bmatrix} \in \mathbb{C}^{NR}. \tag{2.60}$$

After definition of the global effective spreading matrix

$$\tilde{\boldsymbol{S}} = [\tilde{\boldsymbol{S}}_1^{\mathrm{T}}, \ldots, \tilde{\boldsymbol{S}}_R^{\mathrm{T}}]^{\mathrm{T}} \in \mathbb{C}^{NR \times KT} \tag{2.61}$$

and noise vector

$$\boldsymbol{n} = [\boldsymbol{n}_1^{\mathrm{T}}, \ldots, \boldsymbol{n}_R^{\mathrm{T}}]^{\mathrm{T}} \in \mathbb{C}^{NR}, \tag{2.62}$$

we can rewrite the received signal (2.60) as

$$\boldsymbol{y} = \tilde{\boldsymbol{S}}\boldsymbol{b} + \boldsymbol{n}. \tag{2.63}$$

The column $T(k-1)+t$ of $\tilde{\boldsymbol{S}}$, denoted through $\tilde{\boldsymbol{s}}_{(k,t)}$, corresponds to the joint effective spreading sequence of user (k,t) and contains the R effective spreading sequences of this user

$$\tilde{\boldsymbol{s}}_{(k,t)} = [\tilde{\boldsymbol{s}}_{(k,t),1}^{\mathrm{T}}, \ldots, \tilde{\boldsymbol{s}}_{(k,t),R}^{\mathrm{T}}]^{\mathrm{T}}. \tag{2.64}$$

In a similar way as for single antenna detection, we perform parallel interference cancelation for transmit antenna (k,t)

$$\tilde{\boldsymbol{y}}_{(k,t)} = \boldsymbol{y} + \tilde{\boldsymbol{s}}_{(k,t)}\tilde{b}_{(k,t)} - \tilde{\boldsymbol{S}}\tilde{\boldsymbol{b}}\,. \tag{2.65}$$

The unbiased LMMSE filter writes very similarly as in (2.57)

$$\boldsymbol{f}_{(k,t)} = \frac{\left(\sigma_n^2\boldsymbol{I}_{NR} + \tilde{\boldsymbol{S}}\boldsymbol{V}\tilde{\boldsymbol{S}}^{\mathrm{H}}\right)^{-1}\tilde{\boldsymbol{s}}_{(k,t)}}{\tilde{\boldsymbol{s}}_{(k,t)}^{\mathrm{H}}\left(\sigma_n^2\boldsymbol{I}_{NR} + \tilde{\boldsymbol{S}}\boldsymbol{V}\tilde{\boldsymbol{S}}^{\mathrm{H}}\right)^{-1}\tilde{\boldsymbol{s}}_{(k,t)}}\,. \tag{2.66}$$

Estimates of the transmitted symbols $b_{(k,t)}$ are finally given by

$$w_{(k,t)} = \boldsymbol{f}_{(k,t)}^{\mathrm{H}}\tilde{\boldsymbol{y}}_{(k,t)} \tag{2.67}$$

and decoded by a BCJR decoder, after demapping and deinterleaving, providing *a priori* and *extrinsic* probabilities for the following receiver iteration.

2.7.3 Joint Antenna Detection with PIC in User Space

So far interference cancelation has been performed in the *chip space*, i.e. the space of the signal received in the frequency domain, without despreading. After looking at the LMMSE filters (2.57) and (2.66) for single and joint antenna detection, respectively, we can see that as many LMMSE filters as there are transmit antennas need to be computed. We note also that the KT LMMSE filters have a common matrix inverse. Thus, exact computation of these KT filters requires one matrix inversion only, while an alternative approximation method to compute each of them would have to run KT times in parallel.

This way, eventual complexity reduction allowed by an approximation method (such as the Krylov subspace method, see Chapter 3), might be annihilated by the multiplicative factor KT introduced. Parallelization of the computations is achieved but the total computational complexity remains of the same order.

In what follows, we try to solve this complexity issue by derivating a unique LMMSE filter that would allow detection of all KT symbols at once. For this purpose, we apply a matched filter on the received signal (2.63) in order to switch to the *user space*. Each element of the user-space received signal is assigned to the corresponding transmit antenna and contributions of the other transmit antennas is removed by performing interference cancelation.

In Section 4 we will analyze in more details the performance and overall complexity of the different schemes and discuss advantages and disadvantages for each of them.

Let us consider the received signal (2.63). We can apply a matched filter without loss of information [Ver98], leading to the *user space* received signal

$$\boldsymbol{x} = \tilde{\boldsymbol{S}}^{\mathrm{H}}\boldsymbol{y} = \tilde{\boldsymbol{S}}^{\mathrm{H}}\tilde{\boldsymbol{S}}\boldsymbol{b} + \tilde{\boldsymbol{S}}^{\mathrm{H}}\boldsymbol{n}\,. \tag{2.68}$$

We introduce the covariance matrix $\boldsymbol{R} = \tilde{\boldsymbol{S}}^{\mathrm{H}}\tilde{\boldsymbol{S}}$ such as (2.68) reads

$$\boldsymbol{x} = \boldsymbol{R}\boldsymbol{b} + \tilde{\boldsymbol{S}}^{\mathrm{H}}\boldsymbol{n}\,. \tag{2.69}$$

Performing parallel interference cancelation for user (k,t) in a mathematically exactly identical way as in (2.65) gives

$$\tilde{\boldsymbol{x}}_{(k,t)} = \boldsymbol{x} - \boldsymbol{R}\tilde{\boldsymbol{b}} + \tilde{\boldsymbol{S}}^{\mathrm{H}}\tilde{\boldsymbol{s}}_{(k,t)}\tilde{b}_{(k,t)}\,. \tag{2.70}$$

Let us define the unit vector $\boldsymbol{e}_{(k,t)} \in \mathbb{R}^{NR}$ having all zeros except for a one at position $T(k-1)+t$. In Equation (2.70), the element

$$\tilde{x}_{(k,t)} = \boldsymbol{e}_{(k,t)}^{\mathrm{T}}\tilde{\boldsymbol{x}}_{(k,t)} \tag{2.71}$$

of $\tilde{\boldsymbol{x}}_{(k,t)}$ contains most information on the specific user (k,t). In all other elements of $\tilde{\boldsymbol{x}}_{(k,t)}$, the information about user (k,t) consists of interference which is mostly canceled using parallel interference cancelation. From now on, we set these correcting terms to zero. This way the received signal after parallel interference cancelation for user (k,t) is simplified to

$$\begin{aligned}\hat{x}_{(k,t)} &= x_{(k,t)} - \boldsymbol{e}_{(k,t)}^{\mathrm{T}}\boldsymbol{R}\tilde{\boldsymbol{b}} + \tilde{\boldsymbol{s}}_{(k,t)}^{\mathrm{H}}\tilde{\boldsymbol{s}}_{(k,t)}\tilde{b}_{(k,t)} \\ &= \boldsymbol{e}_{(k,t)}^{\mathrm{T}}\left(\boldsymbol{x} - \boldsymbol{R}\tilde{\boldsymbol{b}} + \mathrm{diag}(\boldsymbol{R})\tilde{\boldsymbol{b}}\right)\,.\end{aligned} \tag{2.72}$$

Combining these KT expressions in a matrix form yields

$$\hat{\boldsymbol{x}} = \boldsymbol{x} - (\boldsymbol{R} - \boldsymbol{D})\tilde{\boldsymbol{b}}\,, \tag{2.73}$$

where $\boldsymbol{D} \triangleq \mathrm{diag}(\boldsymbol{R}) \in \mathbb{C}^{KT\times KT}$ contains the diagonal elements of $\boldsymbol{R}$.

For this scheme, the LMMSE filter $\boldsymbol{F}$ defined by

$$\boldsymbol{F}^{\mathrm{H}} = \underset{\boldsymbol{F}}{\mathrm{argmin}}\,\mathrm{E}\{\|\boldsymbol{F}^{\mathrm{H}}\hat{\boldsymbol{x}} - \boldsymbol{b}\|^2\} \tag{2.74}$$

can be expressed as

$$\begin{aligned} \boldsymbol{F}^{\mathrm{H}} &= \mathrm{E}\left\{\boldsymbol{b}\hat{\boldsymbol{x}}^{\mathrm{H}}\right\}\left(\mathrm{E}\left\{\hat{\boldsymbol{x}}\hat{\boldsymbol{x}}^{\mathrm{H}}\right\}\right)^{-1} \\ &= (\boldsymbol{V}\boldsymbol{R}-\boldsymbol{V}\boldsymbol{D}+\boldsymbol{D})\left(\boldsymbol{R}\boldsymbol{V}\boldsymbol{R}+\sigma_n^2\boldsymbol{R}+\boldsymbol{D}(\boldsymbol{I}_{KT}-\boldsymbol{V})\boldsymbol{D}\right)^{-1}, \end{aligned} \tag{2.75}$$

as shown in Appendix A.

Estimates $w_{(k,t)}$ of the transmitted symbols $b_{(k,t)}$ are then given in

$$\boldsymbol{w} = \boldsymbol{F}^{\mathrm{H}}\hat{\boldsymbol{x}} \in \mathbb{C}^{KT} \tag{2.76}$$

for every symbol $m \notin \mathcal{P}$. These $KT(M-J)$ estimates are demapped, deinterleaved and decoded by a BCJR decoder, providing *a priori* and *extrinsic* probabilities for the following receiver iteration.

Performing parallel interference cancelation in user-space as described in (2.73) allows joint detection of all users *using one filter only*. It is expected that some information will get lost since we have set some terms to zero, and as a consequence performance will slightly degrade.

Part II

Linear Detection: The Krylov Subspace Method

Chapter 3
The Krylov Subspace Method

In a time-varying communication system, where the channel changes rapidly, solving a new high-dimensional linear system for each new realization of the channel might get too complex to be feasible. Approximation methods such as Krylov subspace methods [Saa03] allow trading accuracy for complexity.

Krylov subspace methods are iterative projection methods, i.e. a starting value is iteratively projected onto a subspace of increasing dimension, until a given maximum dimension S or a given error tolerance is reached. This method is presented here for the case of a symmetric matrix, and its application to LMMSE filters is analyzed in this chapter.

3.1 Basic Idea

In this section we describe the Krylov subspace method based on [Saa03]. Let us consider a linear system $\boldsymbol{A}\boldsymbol{x} = \boldsymbol{a}$ where $\boldsymbol{A}$ is a known symmetric matrix with size $a \times a$ and $\boldsymbol{a}$ is a known vector with size $a \times 1$. Based on the Cayley-Hamilton theorem [Ati94] for an invertible matrix $\boldsymbol{A}$, we can state that there is a minimum polynomial $\mathcal{R}$ of degree $b \leq a$ such that $\{\boldsymbol{I}_a, \boldsymbol{A}, \ldots, \boldsymbol{A}^{b-1}\}$ are linearly independent and $\mathcal{R}(\boldsymbol{A}) = \boldsymbol{0}$. This leads to

$$\boldsymbol{A}^{-1} = \sum_{i=1}^{b} \alpha_i \boldsymbol{A}^{i-1}\,, \tag{3.1}$$

where $\alpha_1, \cdots, \alpha_{b-1} \in \mathbb{C}$ are computed from $\mathcal{R}$. It follows

$$\boldsymbol{x} = \boldsymbol{A}^{-1}\boldsymbol{a} = \sum_{i=1}^{b} \alpha_i \boldsymbol{A}^{i-1}\boldsymbol{a} \in \mathbb{K}_b(\boldsymbol{A}, \boldsymbol{a})\,, \tag{3.2}$$

where

$$\mathbb{K}_b(\boldsymbol{A}, \boldsymbol{a}) \triangleq \text{span}\left\{\boldsymbol{a}, \boldsymbol{A}\boldsymbol{a}, \ldots, \boldsymbol{A}^{b-1}\boldsymbol{a}\right\} \tag{3.3}$$

is the Krylov subspace formed by $\boldsymbol{A}$ and $\boldsymbol{a}$ of dimension b. The idea behind Krylov subspace methods is to iteratively compute an approximation $\boldsymbol{x}_s$ of $\boldsymbol{x}$ for s incrementing up to b, using projections of the corresponding residual vector $\boldsymbol{r}_s = \boldsymbol{a} - \boldsymbol{A}\boldsymbol{x}_s$ onto the Krylov subspace $\mathbb{K}_s$.

As it can be noticed, these kinds of iterative methods do not compute the matrix $\boldsymbol{A}^{-1}$ explicitly, but give an approximation of the solution $\boldsymbol{A}^{-1}\boldsymbol{a}$ of the linear system for a given $\boldsymbol{a}$. Thus, and we emphasize this important detail for discussion on computational complexity in Chapter 4, if we need to solve several linear systems with the same matrix but different vectors $\boldsymbol{a}$, then the Krylov subspace method will have to be performed for each $\boldsymbol{a}$.

To limit the complexity of such a method, which is proportional to the number of iterations performed by the algorithm, a given maximum dimension S is fixed, such that the output is

$$\boldsymbol{x} \approx \sum_{i=1}^{S} \alpha_i \boldsymbol{A}^{i-1}\boldsymbol{a} \in \mathbb{K}_S(\boldsymbol{A}, \boldsymbol{a})\,. \tag{3.4}$$

3.2 The Algorithm

The iterative process begins with some initial value $\boldsymbol{x}_0$. At step s, the approximation of $\boldsymbol{x}$ is given by $\boldsymbol{x}_s$. We define for each step the residual vector $\boldsymbol{r}_s$ according to

$$\boldsymbol{r}_s = \boldsymbol{a} - \boldsymbol{A}\boldsymbol{x}_s\,. \tag{3.5}$$

The iteration is done until $s = S$ by constructing $\boldsymbol{x}_s$ following

$$\begin{aligned} &\boldsymbol{x}_s \in \boldsymbol{x}_0 + \underbrace{\operatorname{span}\left\{\boldsymbol{r}_0, \boldsymbol{A}\boldsymbol{r}_0, \ldots, \boldsymbol{A}^{s-1}\boldsymbol{r}_0\right\}}_{\mathbb{K}_s(\boldsymbol{A}, \boldsymbol{r}_0) \triangleq \mathbb{K}_s} \quad \text{and} \\ &\boldsymbol{r}_s = \boldsymbol{a} - \boldsymbol{A}\boldsymbol{x}_s \perp \mathbb{K}_s\,. \end{aligned} \tag{3.6}$$

In the previous equation, the notation $\boldsymbol{x}_0 + \mathbb{S}$ denotes the subspace $\mathbb{S}$ shifted by $\boldsymbol{x}_0$. Thus, $\boldsymbol{x}_s \in \boldsymbol{x}_0 + \mathbb{S}$ is equivalent to $(\boldsymbol{x}_s - \boldsymbol{x}_0) \in \mathbb{S}$.

It can be easily proven that the error $\boldsymbol{\epsilon}_s = \boldsymbol{x} - \boldsymbol{x}_s$ follows

$$\boldsymbol{A}\boldsymbol{\epsilon}_s = \boldsymbol{r}_s\,. \tag{3.7}$$

The above orthogonality condition on $\boldsymbol{r}_s$ (3.6) is the basis for the so-called Ritz-Galerkin approach [vdV03]. Another possible condition might be for example a minimum residual

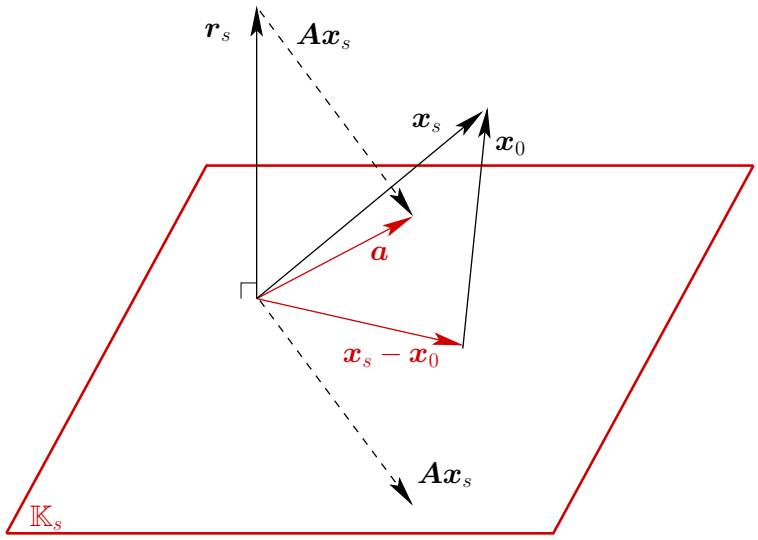


Figure 3.1: The Ritz-Galerkin condition, seen geometrically: find $\boldsymbol{x}_s \in \mathbb{K}_s$ such that $\boldsymbol{r}_s = \boldsymbol{a} - \boldsymbol{A}\boldsymbol{x}_s$ is orthogonal to $\mathbb{K}_s$.

norm approach, computing $\boldsymbol{x}_s$ such that the ℓ_2 norm of $\boldsymbol{r}_s$ is minimized. We will not consider this second approach in this work. However, we will try to define a suitable value of $\boldsymbol{x_0}$ that allows acceleration of the Krylov algorithm, see Section 3.4.

The Ritz-Galerkin condition is illustrated on Figure 3.1: at each step s, we need to find $\boldsymbol{x}_s$ such that $\boldsymbol{x}_s - \boldsymbol{x}_0 \in \mathbb{K}_s$ and the residual vector $\boldsymbol{r}_s$ is orthogonal to $\mathbb{K}_s$.

Note that for $s \leq S$, $\boldsymbol{x}_s - \boldsymbol{x}_0$ is an element of $\mathbb{K}_s$ and can as such be written

$$\boldsymbol{x}_s - \boldsymbol{x}_0 = \boldsymbol{W}_s \boldsymbol{v}_s \,, \tag{3.8}$$

for any basis

$$\boldsymbol{W}_s = [\boldsymbol{w}_1, \ldots, \boldsymbol{w}_s] \tag{3.9}$$

of $\mathbb{K}_s$ and $\boldsymbol{v}_s \in \mathbb{C}^s$. An orthonormal basis $\boldsymbol{W}_s$ can be computed by applying the Gram-Schmidt method [GL96] on the Krylov basis $\boldsymbol{B}_s = [\boldsymbol{r}_0, \boldsymbol{A}\boldsymbol{r}_0, \ldots, \boldsymbol{A}^{s-1}\boldsymbol{r}_0]$.

The Ritz-Galerkin condition $\boldsymbol{r}_s \perp \mathbb{K}_s$ writes

$$\boldsymbol{W}_s^{\mathrm{H}} \boldsymbol{r}_s = 0 \Leftrightarrow \boldsymbol{W}_s^{\mathrm{H}} \boldsymbol{r}_0 = \boldsymbol{W}_s^{\mathrm{H}} \boldsymbol{A} \boldsymbol{W}_s \boldsymbol{v}_s \,. \tag{3.10}$$

Furthermore, the vectors $\boldsymbol{w}_i$ for $i \in \{1, \ldots, s\}$ are such that $\boldsymbol{A}\boldsymbol{w}_i \in \mathbb{K}_{i+1}$, leading by construction to the property

$$\boldsymbol{w}_\ell^{\mathrm{H}} \boldsymbol{A} \boldsymbol{w}_i = 0 \quad \text{if} \quad s \geq \ell > i+1 \,. \tag{3.11}$$

1	input $\boldsymbol{A}$, $\boldsymbol{a}$, $\boldsymbol{x}_0$, S	11	$\boldsymbol{u} = \boldsymbol{A}\boldsymbol{w}_s$
2	$\boldsymbol{r}_0 = \boldsymbol{a} - \boldsymbol{A}\boldsymbol{x}_0$	12	$\mu = \boldsymbol{w}_s^{\mathrm{H}}\boldsymbol{u}$
3	$\boldsymbol{w}_1 = \boldsymbol{r}_0 \| \boldsymbol{r}_0 \|$	13	$\beta = \mu - \nu^2 \boldsymbol{c}_{\mathrm{last}}[s-1]$
4	$\boldsymbol{u} = \boldsymbol{A}\boldsymbol{w}_1$	14	$\boldsymbol{c} = \beta^{-1} \left[-\nu \boldsymbol{c}_{\mathrm{last}}^{\mathrm{T}}, 1 \right]^{\mathrm{T}}$
5	$\mu = \boldsymbol{w}_1^{\mathrm{H}}\boldsymbol{u}$	15	$\boldsymbol{c}_{\mathrm{first}} = \left[\boldsymbol{c}_{\mathrm{first}}^{\mathrm{T}}, 0 \right]^{\mathrm{T}} - c_{\mathrm{last}}[1]^* \nu \boldsymbol{c}$
6	$\boldsymbol{c}_{\mathrm{first}} = \boldsymbol{c}_{\mathrm{last}} = 1/\mu$	16	$\boldsymbol{c}_{\mathrm{last}} = \boldsymbol{c}$
7	$\boldsymbol{w} = \boldsymbol{u} - \mu \boldsymbol{w}_1$	17	$\boldsymbol{w} = \boldsymbol{u} - \mu \boldsymbol{w}_s - \nu \boldsymbol{w}_{s-1}$
8	for $s = 2, \ldots, S$	18	end
9	$\nu = \| \boldsymbol{w} \|$	19	$\boldsymbol{W}_S = [\boldsymbol{w}_1, \ldots, \boldsymbol{w}_S]$
10	$\boldsymbol{w}_s = \boldsymbol{w}/\nu$	20	output $\boldsymbol{x}_S = \| \boldsymbol{r}_0 \| \boldsymbol{W}_S \boldsymbol{c}_{\mathrm{first}} + \boldsymbol{x}_0$

Algorithm 3.1: Krylov subspace algorithm for a Hermitian matrix.

The matrix $\boldsymbol{T}_s = \boldsymbol{W}_s^{\mathrm{H}} \boldsymbol{A} \boldsymbol{W}_s$ has elements $[\boldsymbol{T}_s]_{i,\ell} = \boldsymbol{w}_\ell^{\mathrm{H}} \boldsymbol{A} \boldsymbol{w}_i$. Thus, we can state that it is an upper Hessenberg matrix (i.e. that the element below the first sub-diagonal are zeros). For $\boldsymbol{A}$ being symmetric, $\boldsymbol{T}_s$ will be tridiagonal symmetric (i.e. all elements outside the three main diagonals are zeros), and we denote its elements on the main diagonal as $\mu_i \in \mathbb{C}$ and on the secondary diagonals as $\nu_i \in (0; +\infty)$. Here $(\cdot;\cdot)$ denotes an open interval. Finally, we know by construction of the orthonormal basis $\boldsymbol{W}_s$ that $\boldsymbol{r}_0 = \| \boldsymbol{r}_0 \| \boldsymbol{w}_1$.

Inserting these results into (3.10), we now need to solve

$$\boldsymbol{v}_s = \boldsymbol{T}_s^{-1} \| \boldsymbol{r}_0 \| \boldsymbol{e}_s \,, \tag{3.12}$$

where $\boldsymbol{e}_s = [1, 0, \ldots, 0]^{\mathrm{T}}$ has length s. To compute $\boldsymbol{v}_s$, the first column of $\boldsymbol{T}_s^{-1}$ is needed only. We apply the matrix inversion lemma for partitioned matrices [MS00] to obtain the iterative relation

$$\boldsymbol{T}_s = \begin{bmatrix} \boldsymbol{T}_{s-1} & \nu_s \tilde{\boldsymbol{e}}_{s-1} \\ \nu_s \tilde{\boldsymbol{e}}_{s-1}^{\mathrm{T}} & \mu_s \end{bmatrix} , \tag{3.13}$$

where $\tilde{\boldsymbol{e}}_s = [0, \ldots, 0, 1]^{\mathrm{T}}$ has length s for $s \in \{1, \ldots, S\}$. This gives the following set of iterative equations

$$\begin{aligned} \boldsymbol{c}_{\mathrm{first}}^{(s)} &= \begin{bmatrix} \boldsymbol{c}_{\mathrm{first}}^{(s-1)} \\ 0 \end{bmatrix} + \beta_s^{-1} \boldsymbol{c}_{\mathrm{last}}^{(s-1)}[1]^* \begin{bmatrix} \nu_s^2 \boldsymbol{c}_{\mathrm{last}}^{(s-1)} \\ -\nu_s \end{bmatrix} \\ \boldsymbol{c}_{\mathrm{last}}^{(s)} &= \beta_s^{-1} \begin{bmatrix} -\nu_s \boldsymbol{c}_{\mathrm{last}}^{(s-1)} \\ 1 \end{bmatrix} , \end{aligned} \tag{3.14}$$

```
1  for l = 1 : a − 1
2     A(l + 1 : a, l) = A(l + 1 : a, l)/[A]_{l,l}
3     A(l + 1 : a, l + 1 : a) = A(l + 1 : a, l + 1 : a) − A(l + 1 : a, l)A(l, l + 1 : a)
4  end
```

Algorithm 3.2: Gaussian Elimination for $\boldsymbol{A}$ of size $a \times a$

where $\boldsymbol{c}_{\text{first}}^{(s)}$ and $\boldsymbol{c}_{\text{last}}^{(s)}$ denote respectively the first and last columns of $\boldsymbol{T}_s^{-1}$, and

$$\beta_s = \mu_s - \nu_s^2 \boldsymbol{c}_{\text{last}}^{(s-1)}[s-1] \tag{3.15}$$

is a scalar.

The algorithm is detailed in Algorithm 3.1. Note that by using an appropriate initial value $\boldsymbol{x}_0$ the Krylov subspace algorithm may converge faster. In other words, a smaller subspace dimension s would be needed to achieve a given error ϵ

$$\left\| \boldsymbol{A}^{-1}\boldsymbol{a} - \boldsymbol{x}_s \right\| \leq \epsilon\,. \tag{3.16}$$

3.3 Complexity for Computing an LMMSE Filter

We discuss here the computational complexity using the Krylov subspace method for calculating an LMMSE filter. Let us here define a *flop* as a floating point operation, as given in [GL96]. A *flop* is either an addition, subtraction, multiplication, division or square root operation in the *real* domain. Thus, one complex multiplication (CM) requires 4 real multiplications and 2 additions, leading to 6 *flops*. Similarly, one complex addition (CA) corresponds to 2 *flops*.

A general structure of an LMMSE filter can be written as (see Chapter 2)

$$\boldsymbol{f} = \underbrace{\left(\boldsymbol{D}_1 + \boldsymbol{M}\boldsymbol{D}_2\boldsymbol{M}^{\mathrm{H}}\right)}_{\boldsymbol{A}}{}^{-1} \underbrace{\boldsymbol{M}\boldsymbol{v}}_{\boldsymbol{a}}, \tag{3.17}$$

where $\boldsymbol{M} \in \mathbb{C}^{a\times b}$, $\boldsymbol{v} \in \mathbb{C}^{b}$, and $\boldsymbol{D}_1 \in \mathbb{C}^{a\times a}$and $\boldsymbol{D}_2 \in \mathbb{C}^{b\times b}$ are two diagonal matrices.

3.3.1 Case $a \leq b$

Let us assume first the dimensions are such that $a \leq b$. Exactly computing $\boldsymbol{f}$ in (3.17) requires the following operations:

- The computation of $\boldsymbol{A} = \boldsymbol{D}_1 + \boldsymbol{M}\boldsymbol{D}_2\boldsymbol{M}^{\mathrm{H}}$, i.e.

$$ab\,\mathrm{CM} + a^2(b\,\mathrm{CM} + (b-1)\,\mathrm{CA}) + a\,\mathrm{CA} = 8a^2b + 6ab - 2a^2 + 2a\;flops\,;$$

- The inversion of $\boldsymbol{A}$:
  The Gaussian elimination algorithm [GL96, Algorithm3.2.1] to invert a complex matrix $\boldsymbol{A}$ of size $a \times a$ is given in Algorithm 3.2. For each step l of the Gaussian elimination, $a-l$ multiplications are needed at line 2 and $(a-l)^2$ multiplications as well as $(a-l)^2$ additions at line 3. This leads to the total computational complexity in a complex case (recalling one complex multiplication corresponds to 6 $flops$ and one complex addition requires 2 $flops$):

$$\begin{aligned} C_{\mathrm{GE}} &= \sum_{l=1}^{a-1} \left(2(a-l) + 8(a-l)^2\right) \\ &= \sum_{l=1}^{a-1} \left(2l + 8l^2\right) \\ &= \tfrac{1}{3}a(a-1)(8a-1)\,. \end{aligned}$$

  Thus the complexity for inverting $\boldsymbol{A}$ is given by $\frac{1}{3}a(a-1)(8a-1)\;flops$;

- The computation of $\boldsymbol{A}^{-1}\boldsymbol{a}$ with $\boldsymbol{a} = \boldsymbol{M}\boldsymbol{v}$, i.e. $4a(2a+2b-1)\;flops$.

This leads to the approximate complexity for computing the exact LMMSE filter of

$$C^1_{\mathrm{LMMSE}} \approx 8a^2\left(\frac{a}{3} + b\right)\;flops\,. \tag{3.18}$$

Using the Krylov approximation, the main computations required to approximate the filter $\boldsymbol{f}$ are:

- The product $\boldsymbol{A}\boldsymbol{w}_s = \boldsymbol{D}_1\boldsymbol{v}_s + \boldsymbol{M}(\boldsymbol{D}_2(\boldsymbol{M}^{\mathrm{H}}\boldsymbol{v}_s))$ for $s \in \{1,\ldots,S\}$, in lines 4 and 11 of the algorithm in Algorithm 3.1, as well as $\boldsymbol{A}\boldsymbol{x}_0$ on line 2, i.e. $(16ab + 6a + 4b)(S+1)\;flops$;

- Two inner products or norms for S steps, in lines 3, 5 and lines 9 and 12, i.e. $2(8a-2)S\;flops$.

The exact step-by-step complexity is given in Table 3.1, using the same step numbering as in Algorithm 3.1.

Adding all these steps, the total computational complexity using the Krylov subspace method becomes approximately

$$C^1_{\mathrm{K}} \approx (16ab + 42a)(S+1) + 8ab\;flops\,. \tag{3.19}$$

1	$2a(4b-1)$	11	$16ab+6a+4b$
2	$16ab+8a+4b$	12	$8a-2$
3	$10a-1$	13	$2a+2$
4	$16ab+6a+4b$	14	$6(s+1)$
5	$8a-2$	15	$8(s+1)$
6	1	16	
7	$8a$	17	$8a$
8		18	
9	$8a-1$	19	
10	$2a$	20	$2a(4S+1)$

Table 3.1: Step-by-step complexity using the LMMSE filter as in Section 3.3.

3.3.2 Case $a \geq b$

If $a \geq b$, the computational complexity can be reduced by first applying the matrix inversion lemma [MS00] to (3.17), leading to

$$\boldsymbol{f} = \boldsymbol{D}_1^{-1}\boldsymbol{M}(\boldsymbol{I}_b + \boldsymbol{D}_2\boldsymbol{M}^{\mathrm{H}}\boldsymbol{D}_1^{-1}\boldsymbol{M})^{-1}\boldsymbol{v}\,. \tag{3.20}$$

The proof of (3.20) is given in Appendix B.

Similar computational complexity expressions as (3.18) and (3.19) can be obtained for the new LMMSE filter and its Krylov approximation

$$\begin{aligned} C^2_{\mathrm{LMMSE}} &\approx 8b^2(\tfrac{b}{3}+a)\,flops\,, \\ C^2_{\mathrm{K}} &\approx (8ab+14b+12a)(2S+3)+12b\,flops\,. \end{aligned} \tag{3.21}$$

3.3.3 General Case

Finally we note that, if a and b are large enough, as it will be the case in our application, the terms in C^1_{K} and C^2_{K} that are not linear in ab might be ignored, leading to

$$\begin{aligned} C_{\mathrm{LMMSE}} &= \min\{C^1_{\mathrm{LMMSE}}, C^2_{\mathrm{LMMSE}}\} \approx 8ab\min\{a,b\} + \tfrac{8}{3}\min\{a,b\}^3\,flops\,, \\ C_{\mathrm{K}} &= \min\{C^1_{\mathrm{K}}, C^2_{\mathrm{K}}\} \approx 8ab(2S+3)\,flops\,. \end{aligned} \tag{3.22}$$

In (3.22), a and b are interchangeable, so we can set $\min\{a,b\} = a$ without loss of generality. The exact LMMSE filter has a complexity of order $\mathcal{O}(8a^2(\frac{a}{3}+b))$ and the ratio defined by

$$\gamma = \frac{C_{\mathrm{K}}}{C_{\mathrm{LMMSE}}} \tag{3.23}$$

is of order

$$\mathcal{O}\left(\frac{2S+3}{a\left(\frac{a}{3b}+1\right)}\right) \leq \mathcal{O}\left(\frac{2S+3}{a}\right). \tag{3.24}$$

Assuming $S \ll a$, the computational complexity reduction by the Krylov subspace method is substantial. The degradation due to the approximation $\boldsymbol{x} \approx \boldsymbol{x}_s$ will strongly depend on the context of the application of the Krylov method. We will discuss the performance loss in case of various LMMSE filters in Chapter 4.

3.4 Integration of the Krylov Subspace Method in the Iterative Receiver

The error $\|\boldsymbol{r}_S\|_{\boldsymbol{A}}$, where $\|\|_{\boldsymbol{A}}$ denotes the $\boldsymbol{A}$-norm, i.e.

$$\|\boldsymbol{v}\|_{\boldsymbol{A}} = \boldsymbol{v}^{\mathrm{H}}\boldsymbol{A}\boldsymbol{v} \tag{3.25}$$

for any vector $\boldsymbol{v}$, resulting from the Krylov subspace method to approximate $\boldsymbol{A}^{-1}\boldsymbol{a}$, is bounded by [KS99]

$$\|\boldsymbol{r}_S\|_{\boldsymbol{A}} \leq 2\|\boldsymbol{r}_0\|_{\boldsymbol{A}}\left(\frac{\sqrt{k_A}-1}{\sqrt{k_A}+1}\right)^S. \tag{3.26}$$

The condition number of $\boldsymbol{A}$, denoted through $k_A > 1$, is the ratio of largest and smallest eigenvalues. Convergence of the Krylov subspace method is assured since the ratio

$$\frac{\sqrt{k_A}-1}{\sqrt{k_A}+1} < 1, \tag{3.27}$$

yields

$$\|\boldsymbol{r}_S\|_{\boldsymbol{A}} \to 0 \quad \text{when} \quad S \to \infty. \tag{3.28}$$

However, the convergence speed depends strongly on the one hand on the matrix $\boldsymbol{A}$, on the other hand on the initial guess $\boldsymbol{x}_0$. Hence, it is necessary to appropriately choose the parameters S and $\boldsymbol{x}_0$.

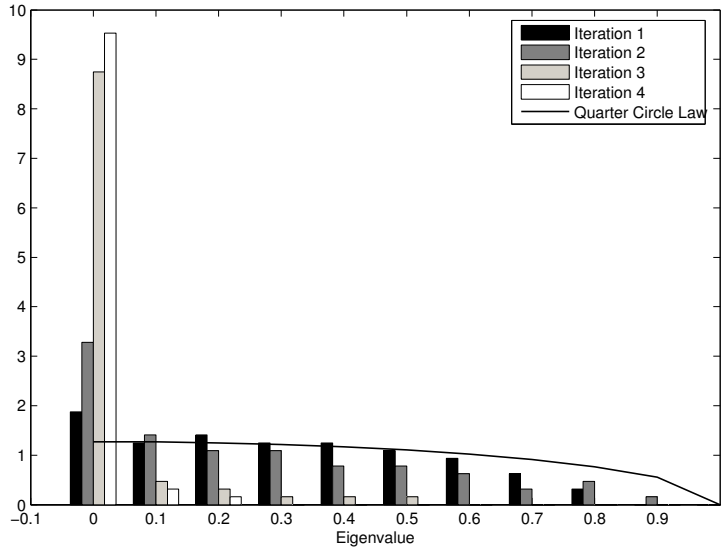


Figure 3.2: Eigenvalue distribution of $(\tilde{\boldsymbol{S}}\boldsymbol{V}\tilde{\boldsymbol{S}}^{\mathrm{H}})^{1/2}$ for receiver iteration $i \in \{1,2,3,4\}$, averaged over 100 data blocks.

3.4.1 On the Condition Number

Considering the LMMSE filter given for example by (2.57), we can write

$$\boldsymbol{A} = \sigma_n^2 \boldsymbol{I}_N + \tilde{\boldsymbol{S}}\boldsymbol{V}\tilde{\boldsymbol{S}}^{\mathrm{H}} \,. \tag{3.29}$$

In this case, and assuming that the smallest eigenvalue of $\tilde{\boldsymbol{S}}\boldsymbol{V}\tilde{\boldsymbol{S}}^{\mathrm{H}}$ is zero, the condition number of $\boldsymbol{A}$ can be written

$$k_{\boldsymbol{A}} = 1 + \rho/\sigma_n^2 \,, \tag{3.30}$$

where the spectral radius ρ denotes the highest eigenvalue of $\tilde{\boldsymbol{S}}\boldsymbol{V}\tilde{\boldsymbol{S}}^{\mathrm{H}}$.

We plot the empirical eigenvalue distribution of $(\tilde{\boldsymbol{S}}\boldsymbol{V}\tilde{\boldsymbol{S}}^{\mathrm{H}})^{1/2}$, obtained from simulations, for receiver iterations $i \in \{1,\ldots,4\}$ in Figure 3.2. At the first iteration, we know the diagonal covariance matrix $\boldsymbol{V} = \boldsymbol{I}_N$ since no soft symbols are available. We observe that the distribution of the eigenvalues $0 \leq \lambda \leq 1$ for the first receiver iterations follows a quarter circle distribution.

This accords the random matrix theory [Mül04], that states that if $\tilde{\boldsymbol{S}} \in \mathbb{C}^{N\times K}$ has i.i.d. elements with zero mean and variance $1/2N$, as the spreading matrix $\boldsymbol{S}$ for a single input single output (SISO) system, then the eigenvalue distribution of $(\tilde{\boldsymbol{S}}\tilde{\boldsymbol{S}}^{\mathrm{H}})^{1/2}$

converges to a density represented by a quarter circle, i.e.

$$p(\lambda) = \begin{cases} \frac{1}{\pi}\sqrt{1-\lambda^2} & \text{if} \quad 0 \leq \lambda \leq 1\,, \\ 0 & \text{elsewhere}\,, \end{cases} \tag{3.31}$$

when N grows to infinity.

With increasing receiver iteration i of the receiver, the symbol estimates get more accurate. Hence, the diagonal covariance matrix $\boldsymbol{V}$ in (2.53) will converge towards zero, and so will the eigenvalue distribution of $(\tilde{\boldsymbol{S}}\boldsymbol{V}\tilde{\boldsymbol{S}}^{\mathrm{H}})^{1/2}$. We can see this effect for increasing number of receiver iterations in Figure 3.2. For the practical implementation these results tell us that the eigenvalue distribution of the matrix $(\tilde{\boldsymbol{S}}\boldsymbol{V}\tilde{\boldsymbol{S}}^{\mathrm{H}})^{1/2}$ moving towards zero implies that the condition number decreases and thus the Krylov algorithm should converge faster. Hence, the necessary Krylov subspace dimension s should decrease with increasing iteration count i. However, this has not been investigated in the work presented in this thesis, where we chose to fix the Krylov subspace dimension for all receiver iterations.

Nevertheless, we exploit the information available from the receiver structure, in order to improve the choice of the initial value $\boldsymbol{x}_0$ and allow possible faster convergence of the Krylov subspace method.

3.4.2 Initialization Methods

As detailed in Chapter 2, the iterative receiver needs two LMMSE filters for channel estimation and multi-user detection. These filters are computed at every iteration i of the receiver. We describe below how to make use of the receiver structure to use the result of previous iterations, or the slow time variations of the channel to start the Krylov algorithm with a better initial value. We give details for every LMMSE filter encountered in case of parallel interference in chip space.

Illustration of the various initialization methods are shown in Figure 3.3.

3.4.2.1 Channel Estimation

Let us recall the LMMSE estimator for channel coefficients (2.42)

$$\hat{\boldsymbol{\phi}} = \left(\tilde{\boldsymbol{\mathcal{D}}}^{\mathrm{H}}\boldsymbol{\Delta}^{-1}\tilde{\boldsymbol{\mathcal{D}}} + \boldsymbol{\mathcal{C}}_{\phi}^{-1}\right)^{-1}\tilde{\boldsymbol{\mathcal{D}}}^{\mathrm{H}}\boldsymbol{\Delta}^{-1}\boldsymbol{y}\,, \tag{3.32}$$

where $\tilde{\boldsymbol{\mathcal{D}}}$ contains the channel coefficients for all users and symbols within one block, see Section 2.6.2. This filter is computed per subcarrier q, receive antenna r and iteration of the receiver i.

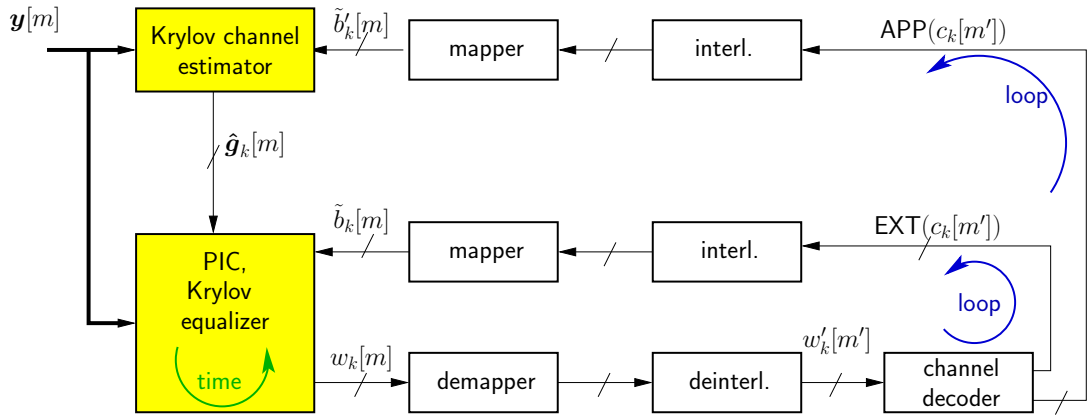


Figure 3.3: Initialization methods for the Krylov subspace method.

First, a basic initialization method which is used when no information is known about the LMMSE filter is Zeros initialization: the initial value is always

$$\boldsymbol{x}_0[q,r]^{(i)} = \mathbf{0}_{KTD}\,, \tag{3.33}$$

for all subcarrier q, receive antenna r and iteration i of the receiver. This is the simplest initialization method, most spread in literature, also used in [DZ05]. Nevertheless, it does not take into account any possible connection between two LMMSE filters.

Since the iterative receivers provides with more accurate channel estimates at each iteration, we suppose the channel estimates at iteration $(i-1)$ provide a good initial value for estimating the channel coefficients using the Krylov subspace method at the following iteration i. This allows defining Loop initialization at iteration i according to

$$\boldsymbol{x}_0[q,r]^{(i)} = \tilde{\boldsymbol{\phi}}_{q,r}^{(i-1)} \in \mathbb{C}^{KTD}\,, \tag{3.34}$$

where $\tilde{\boldsymbol{\phi}}_{q,r}^{(i)}$ denotes the estimate of $\hat{\boldsymbol{\phi}}$ (3.32) obtained with the Krylov subspace method at iteration i, for subcarrier q and receive antenna r. We state

$$\boldsymbol{x}_0^{(0)}[q,r] = \mathbf{0}_{KTD}\,. \tag{3.35}$$

This initialization provides an iteration-to-iteration update of the initial value, which means the vectors $\tilde{\boldsymbol{\phi}}_{q,r}^{(i)}$ need to be stored during the whole iteration to be used for the next one. Thus storage of NR vectors of size KTD per iteration is required.

3.4.2.2 Single Antenna Detection

Similarly as for channel estimation, we develop various initializations for the single antenna detection case. Let us recall the LMMSE estimator for single antenna detection

(2.57)

$$\boldsymbol{f}_{(k,t),r} = \frac{\left(\sigma_n^2 \boldsymbol{I}_N + \tilde{\boldsymbol{S}}_r \boldsymbol{V} \tilde{\boldsymbol{S}}_r^{\mathrm{H}}\right)^{-1} \tilde{\boldsymbol{s}}_{(k,t),r}}{\tilde{\boldsymbol{s}}_{,r}{}^{\mathrm{H}} \left(\sigma_n^2 \boldsymbol{I}_N + \tilde{\boldsymbol{S}}_r \boldsymbol{V} \tilde{\boldsymbol{S}}_r^{\mathrm{H}}\right)^{-1} \tilde{\boldsymbol{s}}_{(k,t),r}}\,, \tag{3.36}$$

that needs to be computed for every transmit antenna (k,t), receive antenna r, symbol m and receiver iteration i.

As in the previous section, the Zeros initialization using

$$\boldsymbol{x}_0[m,(k,t),r]^{(i)} = \boldsymbol{0}_{KTD}\,, \tag{3.37}$$

as well as Loop initialization according to

$$\text{Loop}: \begin{cases} \boldsymbol{x}_0^{(i)}[m,(k,t),r] = \tilde{\boldsymbol{f}}_0^{(i-1)}[m,(k,t),r] \in \mathbb{C}^N\,, \\ \\ \boldsymbol{x}_0^{(0)}[m,(k,t),r] = \boldsymbol{0}_N \end{cases} \tag{3.38}$$

can be defined for $m \notin \mathcal{P}$, $k \in \{1,\ldots,K\}$, $t \in \{1,\ldots,T\}$, $r \in \{1,\ldots,R\}$ and for all receiver iteration i. Here, $\tilde{\boldsymbol{f}}_0^{(i)}[m,(k,t),r]$ denotes the estimate of $\boldsymbol{f}_{(k,t),r}$ in (3.36) obtained using the Krylov subspace method at iteration i. Again, these estimates need to be stored for being used at the following iteration and storage of $(M-J)KTR$ vectors of size N is required.

Furthermore, we assume that the variation of the channel's frequency response $\boldsymbol{g}_k[m]$ in time is band-limited by the maximum Doppler frequency [ZM05]. This is also true for the effective spreading sequence (2.48). Hence we can reasonably assume that both the effective spreading sequence and the LMMSE filter will vary slowly in time. This allows defining the Time initialization according to

$$\text{Time}: \begin{cases} \boldsymbol{x}_0^{(i)}[m,(k,t),r] = \tilde{\boldsymbol{f}}_{(k,t),r}^{(i)}[m-1,(k,t),r] \in \mathbb{C}^N\,, \\ \\ \boldsymbol{x}_0^{(i)}[0,(k,t),r] = \boldsymbol{0}_N\,, \end{cases} \tag{3.39}$$

as an update from time instant to time instant, for a given iteration. Here computing (3.36) is done independently per transmit and receive antenna. Thus, the LMMSE filter (3.36) at time m is immediately reused for computing the following filter at time $m+1$, and no storage is required. The channel being estimated for the whole block of length M at once, this method is not applicable to the channel estimation LMMSE filter.

3.4.2.3 Joint Antenna Detection

The same initializations as for single antenna detection apply for joint antenna detection, with adapted indexing. We recall the LMMSE estimator for joint antenna detection (2.66)

$$\boldsymbol{f}_{(k,t)} = \frac{\left(\sigma_n^2 \boldsymbol{I}_{NR} + \tilde{\boldsymbol{S}}\boldsymbol{V}\tilde{\boldsymbol{S}}^{\mathrm{H}}\right)^{-1} \tilde{\boldsymbol{s}}_{(k,t)}}{\tilde{\boldsymbol{s}}_{(k,t)}^{\mathrm{H}} \left(\sigma_n^2 \boldsymbol{I}_{NR} + \tilde{\boldsymbol{S}}\boldsymbol{V}\tilde{\boldsymbol{S}}^{\mathrm{H}}\right)^{-1} \tilde{\boldsymbol{s}}_{(k,t)}}, \tag{3.40}$$

which is computed jointly for all receive antennas r, but individually for all transmit antenna (k,t), symbol m and at each receiver iteration i. The same initializations as for single antenna detection are possible, namely the Zeros initialization

$$\boldsymbol{x}_0^{(i)}[m,(k,t)] = \mathbf{0}_{NR}\,, \tag{3.41}$$

the Loop initialization using

$$\text{Loop}: \begin{cases} \boldsymbol{x}_0^{(i)}[m,(k,t)] = \tilde{\boldsymbol{f}}_0^{(i-1)}[m,(k,t)] \in \mathbb{C}^{NR}\,, \\ \\ \boldsymbol{x}_0^{(0)}[m,(k,t)] = \mathbf{0}_{NR}\,, \end{cases} \tag{3.42}$$

and the Time initialization according to

$$\text{Time}: \begin{cases} \boldsymbol{x}_0^{(i)}[m,(k,t)] = \tilde{\boldsymbol{f}}_0^{(i)}[m-1,(k,t)] \in \mathbb{C}^{NR}\,, \\ \\ \boldsymbol{x}_0^{(i)}[0,(k,t)] = \mathbf{0}_{NR}\,, \end{cases} \tag{3.43}$$

for all symbol $m \notin \mathcal{P}$, user $k \in \{,\ldots,K\}$, transmit antenna $t \in \{1,\ldots,T\}$, and receiver iteration i. Here, $\tilde{\boldsymbol{f}}_0^{(i)}[m,(k,t)]$ denotes the estimate of $\boldsymbol{f}_{(k,t)}$ (3.40) obtained using the Krylov subspace method at iteration i, for transmit antenna (k,t) and symbol m.

For joint antenna detection too, storage of $(M-J)KT$ vectors of size NR is required during one receiver iteration when using the Loop initialization, while the Time initialization does not need long-term storage.

The Loop and Time initializations are depicted in Figure 3.3 for the iterative receiver using Krylov channel estimation and Krylov multi-user detection.

Chapter 4
Performance and Complexity

In this chapter, we implement all LMMSE filters described in Chapter 2, for channel estimation or multi-user detection, using their approximation by means of the Krylov subspace method (see Chapter 3). We present the simulation setup in Section 4.1, followed by simulation results for a single antenna system in Section 4.2 and a multiple antenna system in Section 4.3. The computational complexity of all these filters and their Krylov equivalent given the simulation results is finally discussed in Section 4.4 and 4.5. Memory requirements are briefly mentioned in Section 4.6.

4.1 Simulation Setup

4.1.1 System Parameters

We consider either a SISO system, i.e. $T = R = 1$, or a MIMO square system with $T = R = 4$ antennas at each transceiver. The number of users is $K \in \{32, 48, 64\}$ depending on the simulations, always smaller than the spreading factor for MC-CDMA $N = 64$. N is also the number of subcarriers of the OFDM system. The transmission is done over a block of $M = 256$ symbols, of which $J = 60$ symbols are pilots used for channel estimation. We work at carrier frequency $f_C = 2\,\text{GHz}$. The OFDM symbol has length $P = G + N = 79$ with cyclic prefix of length $G = 15$.

The load of the system (2.12) or (2.13) with these parameters is

$$\zeta = \frac{K}{N} = \frac{KT}{NR} \leq 1\,. \tag{4.1}$$

All simulations results are obtained by averaging over 100 independent channel realizations. Unless specified otherwise, all results are shown after 4 turbo iterations of the receiver.

For data transmission, a convolutional, non-systematic, non-recursive, 4 state, rate $R_C = 1/2$ code with code generators $[101]$ and $[111]$ denoted $(5, 7)_8$ [HLY02], is used.

The symbols stem from a QPSK constellation with energy normalized to 1 and we define the signal-to-noise ratio (SNR) at the receiver

$$E_\mathrm{b}/N_0 = \frac{1}{2R_C\sigma_n^2}\frac{P}{N}\frac{M}{M-J} \tag{4.2}$$

taking into account the loss due to coding, pilots and cyclic prefix. The noise variance σ_n^2 is assumed to be known at the receiver and is varied for the simulations.

4.1.2 Channel Model Parameters

The channel is sampled at rate $1/T_\mathrm{C} = 3.84$MHz [Cor01]. The root mean square delay spread follows $T_\mathrm{D} = L_\mathrm{D}T_\mathrm{C} = 1\mu$s with $L_\mathrm{D} = 4$. We assume $L = 15$ resolvable paths.

The maximum variation in time of the wireless channel is given by the maximum normalized Doppler bandwidth that depends on the maximum velocity of the transmitter $v_\mathrm{max} = 102.5\,\mathrm{kmh}^{-1}$, the carrier frequency f_C, the speed of light c_0 and the symbol duration T_S with $1/T_\mathrm{S} = 48\cdot 10^3\mathrm{s}^{-1}$ according to

$$\nu_\mathrm{Dmax} = \frac{v_\mathrm{max}f_\mathrm{C}}{c_0}T_\mathrm{S} = 3.9\cdot 10^{-3}\,. \tag{4.3}$$

The actual realization of the time-varying flat-fading channel $h[m,q]$ at time m and subcarrier q is calculated according to the simulation model from [ZX03] with correction for low-velocity as described in [ZM05]. This ensures a Rayleigh distribution of $g_{(k,t),r}[m,q]$.

For the simulations, the users are all moving at the same velocity

$$v = 70\mathrm{kmh}^{-1} \leq v_\mathrm{max}\,, \tag{4.4}$$

which gives a Doppler bandwidth of $B_\mathrm{D} = 126$Hz and a normalized Doppler bandwidth $\nu_\mathrm{D} = 2.6\cdot 10^{-3}$. The configuration of the system designed for maximum user velocity v_max results in a dimension $D = 3$ for the Slepian basis expansion, according to (2.33).

Note at this point that we focus on comparing the iterative receiver with channel estimation and multi-user detection as presented in [ZMWM06] to its approximation using the Krylov subspace method [DZ05, DZ06a, DZ08]. Although the performance of the receiver should clearly depend on parameters like the velocity of the user or the delay spread, we believe that these parameters will have only small influence on the comparative behavior in term of performance and complexity that we are interested in.

The channel taps are normalized so that

$$\mathrm{E}\left\{\sum_{r=1}^{R}\sum_{t=1}^{T}\sum_{\ell=0}^{L-1}|h[m,\ell]|^2\right\} = 1\,, \tag{4.5}$$

in order to analyze the diversity gain of the receiver only. No antenna gain is present due to this normalization.

4.1.3 Krylov Parameters

For some simulations, the several initialization methods presented in Section 3.4 will be compared. We recall them briefly here

- The Zeros initialization does not take any potentially available information into account and fixes the starting vector for the Krylov iterations at zero.
- The Loop initialization allows adaptation from receiver iteration to receiver iteration. The starting vector at iteration $(i+1)$ is the result obtained at the previous iteration (i).
- The Time initialization allows adaptation from symbol to symbol. The starting vector at symbol $m+1$ is the result obtained at the previous symbol m. This method is available for multi-user detection only.

The Krylov subspace dimension is varied in order to obtain a best possible performance. We denote the Krylov subspace dimension for multi-user detection and channel estimation through S and S' respectively.

In the simulations, curves will be labelled as, for example, 'Time $S = 2$' to show the curve using the Time initialization with Krylov subspace dimension $S = 2$, unless no confusion can be made.

4.2 Performance of a SISO System

As a first step let us look at a SISO system ($R = T = 1$). We want to analyze the use of the Krylov subspace method for multi-user detection and channel estimation in the corresponding system. The Krylov subspace method is first used for channel estimation only and then for both channel estimation and multi-user detection.

Figure 4.1 shows the performance for a half or fully loaded system (i.e. $K \in \{32, 64\}$ users), using the Krylov subspace method in its basic form with the Zeros initialization method. We depict the bit error rate (BER) versus E_b/N_0 after 4 iterations for the iterative receiver. Figure 4.1.a illustrates the performance with iterative time-varying channel estimation based on the Slepian basis expansion combined with the Krylov subspace method. Here multi-user detection is performed using exact LMMSE filtering in

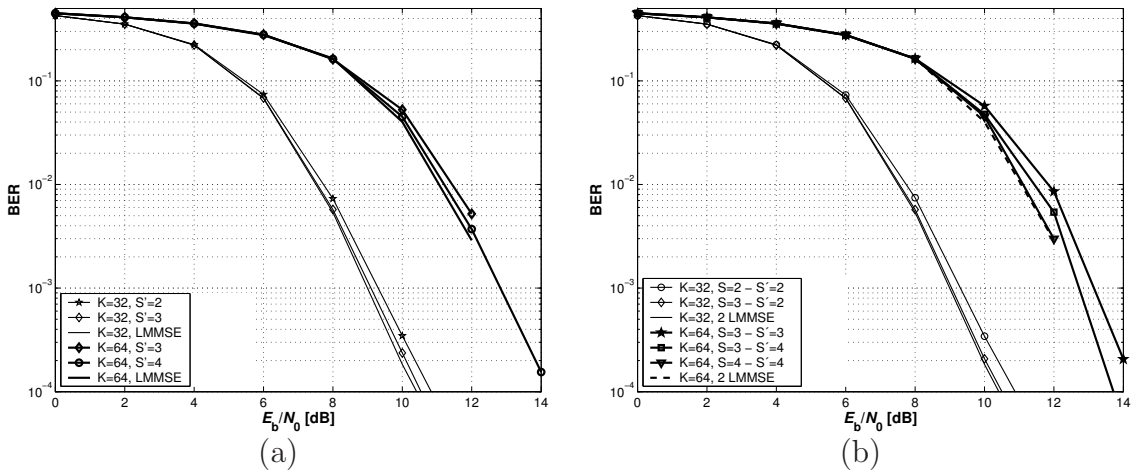


Figure 4.1: Krylov subspace method for a SISO system. Data detection uses either exact LMMSE (a) or a Krylov subspace dimension $S \in \{2, 3, 4\}$ (b). Channel estimation uses dimension $S' \in \{2, 3, 4\}$. Both consider the Zeros initialization only. The system has $K \in \{32, 64\}$ users

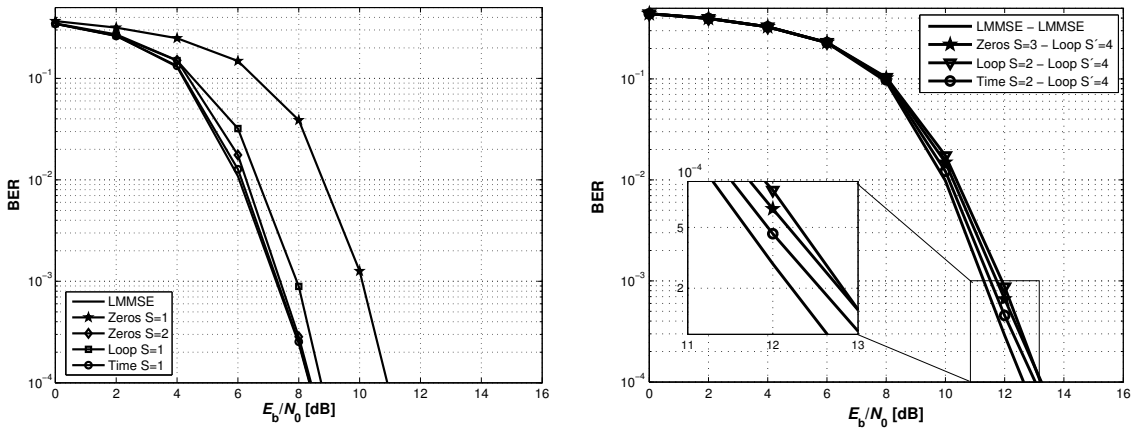
.

order to observe the influence of the Krylov subspace dimension S for channel estimation. We can observe that a dimension $3 \leq S \leq 4$ is sufficient to reach close-to-LMMSE performance. In Figure 4.1.b, simulation results for double Krylov subspace methods (i.e. for channel estimation applied to the filter (2.42) and for multi-user detection applied to the filter (2.57) simultaneously) are shown. Krylov subspace dimensions $2 \leq S, S' \leq 4$ are sufficient for both applications to reach close-to-LMMSE performance.

In the next step, we focus on comparing the initialization methods, first with perfect channel knowledge and then with the double Krylov subspace method, for channel estimation and multi-user detection. The initialization method varies as well as the Krylov subspaces dimensions S and S', in order to find out the method that reaches close-to-LMMSE performance with the smallest possible Krylov subspace dimension. Since the computational complexity order is proportional to the subspace dimension, we aim at reducing complexity as much as possible. However, we need to keep in mind the following aspects of the fed back initialization:

- One Krylov iteration for channel estimation costs less computational complexity than one for multi-user detection.

- The Loop initialization for the channel estimation need a storage of $KTDN$ complex elements, see Section 3.4, while the Loop initialization for the multi-user



(a) (b)

Figure 4.2: Double Krylov method for SISO channels. Data detection uses either exact LMMSE or a Krylov subspace method with Zeros, Loop or Time initialization and dimension $S \in \{1, 2, 3\}$. Channel estimation is perfect (a) or uses either exact LMMSE or a Krylov subspace method with Loop initialization and dimension $S' = 4$ (b). The system has $K = 48$ users.

detector needs saving an even larger quantity $(M - J)KTN$ of complex elements. This might deprecate this method for a hardware implementation of the multi-user detector.

The corresponding simulation results are shown in Figure 4.2. For Figure 4.2.a, we assume first perfect channel knowledge at the receiver, thus channel estimation is not needed. We show the performance in terms of BER versus E_b/N_0 for the MC-CDMA uplink with $K = 48$ users. We can conclude the following from these results:

- At same Krylov subspace dimension, a considerable gain in performance is to be observed using feed-back information (i.e. Loop or Time methods), and the Time method performs best.
- Using the Time method allows lowering the subspace dimension for close-to-LMMSE performance, thus reducing complexity.

In Figure 4.2.b, similar results are shown for an estimated channel. Now we use a double Krylov subspace method, for channel estimation and multi-user detection, and vary in both cases the Krylov subspace dimensions S and S' as well as the initialization method. We observe first that having no knowledge of the channel at the receiver implies a need of a higher Krylov subspace dimension for the multi-user detector. An optimal

combination for the Krylov subspace method could be using a dimension $S = 4$ with the Loop initialization at channel estimation, and a dimension $S' = 2$ with the Time initialization for multi-user detection. Of course a trade-off between performance and complexity has to be found, since a smaller dimension implies less complexity but a slight loss in performance.

4.3 Performance of a MIMO System

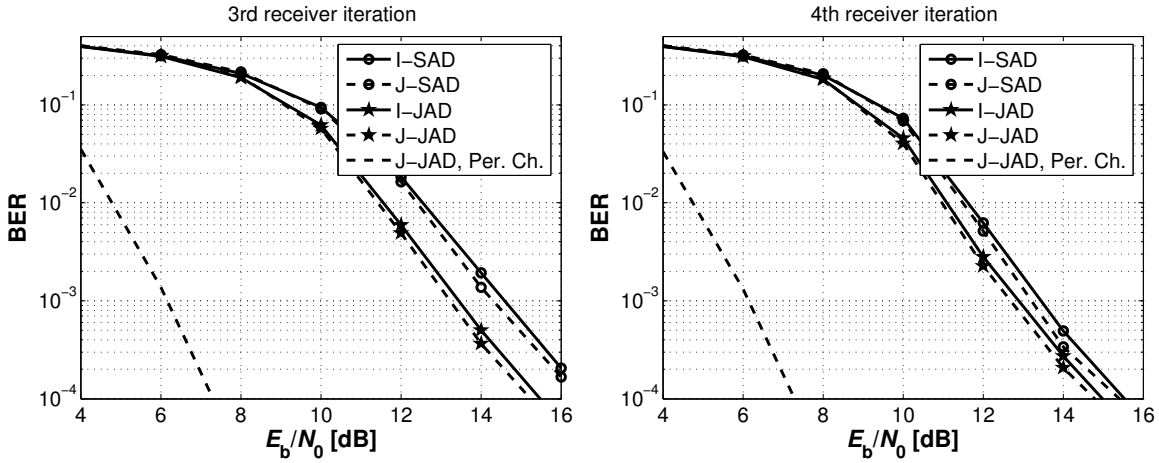
The following simulations are performed for 4×4 channels. First we compare the different scenarios at the transmitters (i.e. independent or joint encoding) and at the receiver (i.e. single or joint antenna detection). Taking the best performing of these schemes, we then apply the Krylov subspace method. In these simulations, $T = R = 4$ and the system is half loaded with $K = 32$ users.

4.3.1 Single vs. Joint Antenna Detection

We compare the performance of independent or joint encoding at the transmitter and single or joint antenna detection at the receiver, after the 3rd and 4th iteration of the receiver, in Figures 4.3.a and 4.3.b respectively. For the moment all LMMSE filters are computed exactly, the channel being not known and estimated. For comparison we also show the LMMSE performance when the channel is perfectly known. Following conclusion can be drawn:

- Joint encoding at the transmitter slightly outperforms independent encoding for a given iteration count at the receiver;
- Joint antenna detection expectedly outperforms single antenna detection;
- Using joint antenna detection at the third iteration of the receiver or single antenna detection at the fourth iteration of the receiver leads to the same performance; thus, assuming this performance is satisfying, the choice of joint antenna detection allows reducing by $1/4$ the global computational complexity using the Krylov subspace method, by iterating less often.

A best choice not regarding to complexity is obviously to perform joint encoding at the transmitter and joint antenna detection at the receiver. The question if three receiver iterations are enough is left for the discussion on complexity. In the following, we will focus on this "best" combination and apply the Krylov subspace method.



(a) (b)

Figure 4.3: Encoding and antenna detection schemes for a MIMO system. BER using independent (I) or joint (J) encoding at the transmitter with single (SAD) or joint (JAD) antenna detection after 3 (a) and 4 (b) iterations of the receiver. Both channel estimation and multi-user detection use an LMMSE filter. The system has $K = 32$ users.

4.3.2 Krylov Detector in the Chip Space

Now the channel estimator (2.42) and the multi-user detector (2.66) have been replaced by their equivalent using the Krylov subspace method. Based on the results for the SISO case, we decided to keep only the best performing initialization methods, i.e. the Loop method for channel estimation and the Time method for multi-user detection. In Figure 4.4, the BER versus E_b/N_0 performance is shown for varying Krylov subspace dimension $S' \in \{4, 6, 8\}$ for channel estimation and a fix (minimal) subspace dimension $S = 1$ for multi-user detection.

Sufficient parameters using a Krylov based receiver are

- for multi-user detection: Time initialization with a Krylov subspace of dimension $S = 1$,
- for channel estimation: Loop initialization with a Krylov subspace of dimension $S' = 8$.

We notice that the dimension S' for channel estimation is considerably increased compared to the SISO case, due to the higher dimensions of the system. However, no change can be observed for multi-user detection. We can trade performance against computational complexity by reducing the Krylov subspace dimension.

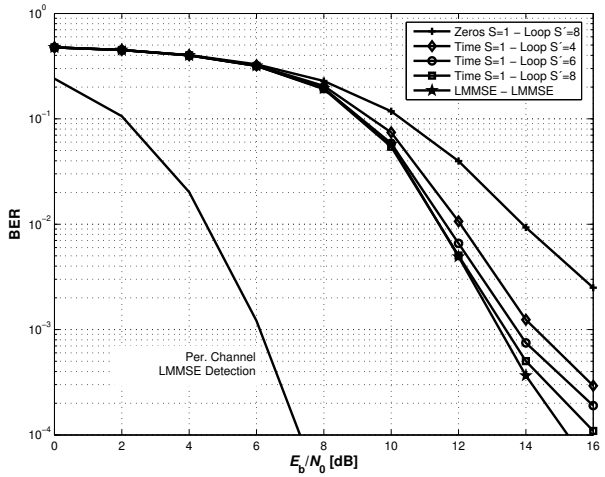


Figure 4.4: Double Krylov subspace method for MIMO channels. Multi-user detection uses either exact LMMSE or the Krylov subspace method with Zeros or Time initialization and dimension $S = 1$. Channel estimation uses either exact LMMSE or the Krylov subspace method with Loop initialization and dimension $S' \in \{4, 6, 8\}$. The system has $K = 32$ users.

4.3.3 Parallel Interference Cancelation in User Space

In this section, we consider the basic implementation of the Krylov subspace method only, i.e. using the Zeros initialization method. Simulations are performed in three steps. Firstly, we compare parallel interference cancelation in chip space and in user space in terms of bit error rate versus E_b/N_0. All filters are exact LMMSE filters (see (2.66) for chip space, (2.75) for user space), and the receiver performs 4 iterations. In Figure 4.5, we see that when parallel interference cancelation is performed in the user space, a slight increase in BER can be observed. However, we will see in the discussion on the computational complexity in this chapter, that regarding the considerable complexity reduction this method allows, this slight loss becomes negligible.

Secondly, we focus on the joint antenna detector with parallel interference cancelation in user space: at this point, the channel is either perfectly known or LMMSE channel estimates are used. The multi-user detector utilizes the Krylov subspace method. Figure 4.6.a shows the BER curves for varying Krylov subspace dimension S. As lower bound we plot the BER curve with the exact LMMSE filter for multi-user detection. When the channel is perfectly known, $S = 2$ is sufficient to reach LMMSE multi-user detection performance. When LMMSE channel estimates are used, some loss in performance appears, and a higher subspace dimension is required ($S = 5$ leads to a loss of approx-

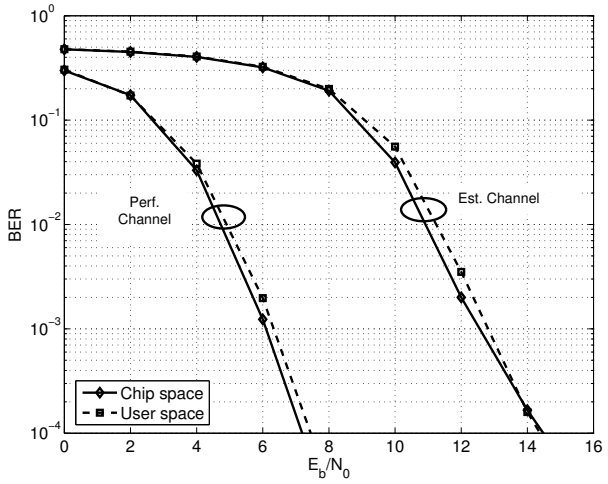


Figure 4.5: Parallel interference cancelation in chip and user space. BER versus E_{b}/N_0 for 4×4 MIMO channels with $K = 32$ users, with perfect ('Perf.') or LMMSE estimated ('Est.') channels.

imatively 0.25 dB). The expected computational complexity reduction involved allows trading accuracy for efficiency.

Finally, we keep $S = 5$ constant for joint antenna detection with PIC in user space, and vary the Krylov subspace dimension S' for channel estimation. These results are shown in Figure 4.6.b. Again, a slight loss is inevitable but a trade-off has to be made between computation complexity and performance. A dimension $S' = 12$ introduces a loss of about 0.5dB compared to the double LMMSE receiver.

4.4 Computational Complexity Expressions

The results in the complexity section are given for multiple antenna systems, since the single antenna case can be easily derived using $T = R = 1$. The model presented in Section 3.3 can be used to compute the computational complexity for most of the LMMSE filters and their corresponding approximations presented so far. More details are given below.

We recall here the typical LMMSE filter

$$\boldsymbol{f} = \underbrace{(\boldsymbol{D}_1 + \boldsymbol{M}\boldsymbol{D}_2\boldsymbol{M}^{\mathrm{H}})}_{\boldsymbol{A}}^{-1} \underbrace{\boldsymbol{M}\boldsymbol{v}}_{\boldsymbol{a}}, \tag{4.6}$$

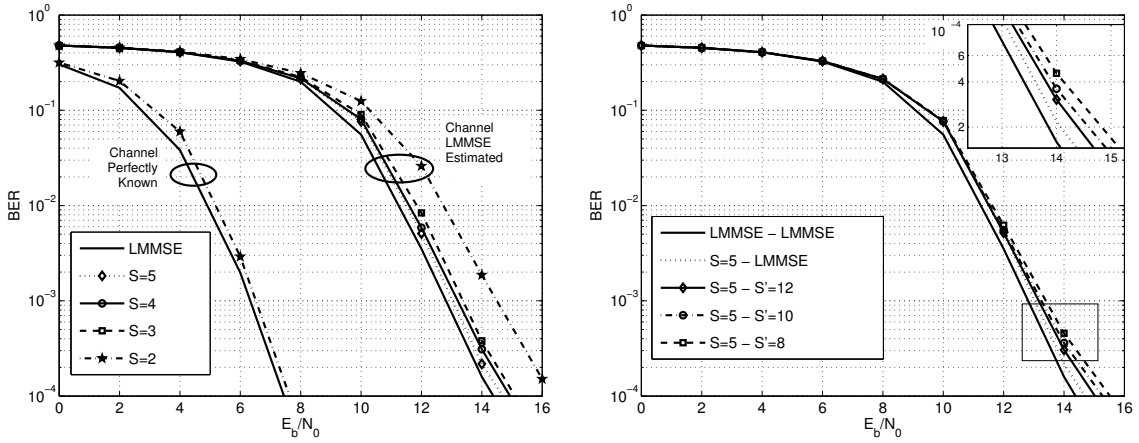


Figure 4.6: Double Krylov method for PIC in user space. (a) Multi-user detection uses either exact LMMSE or the Krylov subspace method with Zeros initialization and dimension $S \in \{2, 3, 4, 5\}$, while channel is either prefect or uses LMMSE estimates. (b) Multi-user detection is fixed using the Krylov subspace method with Zeros initialization and dimension $S = 5$, and channel estimation uses the Krylov subspace method with Zeros initialization and dimension $S' \in \{8, 10, 12\}$. The system has 4×4 channels with $K = 32$ users.

and the computational complexity results from Section 3.3

$$\begin{aligned} C_{\text{LMMSE}} &\approx 8ab\min\{a,b\} + \tfrac{8}{3}\min\{a,b\}^3 \; flops \\ C_{\text{K}} &\approx 8ab(2S+3) \; flops\,, \end{aligned} \tag{4.7}$$

where a and b are the dimensions of $\boldsymbol{M}$.

4.4.1 Time-varying Channel Estimation

Implementing the LMMSE filter (2.42) using the Krylov subspace method leads to the following computational complexity expressions, taking the model (4.6). The corresponding quantities are given by

$$\begin{aligned} &\boldsymbol{D}_1 = \boldsymbol{\mathcal{C}}_{\phi}^{-1}\,, \\ &\boldsymbol{D}_2 = \boldsymbol{\Delta}^{-1}\,, \\ &\boldsymbol{M} = \tilde{\boldsymbol{\mathcal{D}}}_q^{\text{H}}\,, \\ &\boldsymbol{v} = \boldsymbol{\Delta}^{-1}\boldsymbol{y}_q\,, \\ &a = KTD \quad \text{and} \quad b = M\,. \end{aligned} \tag{4.8}$$

Here the approximation is computed individually for each subcarrier q but jointly for all virtual users (k,t), which has no influence for the comparison LMMSE versus Krylov subspace method. Furthermore, channel estimation is done at each receive antenna.

In the SISO case, we have $KTD = KD < M$, whereas for a MIMO case, a becomes larger. For discussion on the complexity, we focus on the MIMO case. Computations can easily be extended to the SISO case according to (4.7). In the MIMO case, $KTD > M$ implies $a > b$, although a and b are of the same order. Thus we obtain the following expressions of the complexity per subcarrier ('CE' stands for channel estimation)

$$\begin{aligned} C_{\mathrm{LMMSE}}^{\mathrm{CE}} &\approx 8M^2(\tfrac{M}{3} + KTD)RN\,, \\ C_{\mathrm{K}}^{\mathrm{CE}} &\approx 8MKTD(2S+3)RN\,. \end{aligned} \tag{4.9}$$

The multiplicative factor RN comes from the fact that channel estimation is done per antenna and subcarrier. The ratio

$$\gamma^{\mathrm{CE}} = \frac{C_{\mathrm{K}}^{\mathrm{CE}}}{C_{\mathrm{LMMSE}}^{\mathrm{CE}}} \tag{4.10}$$

is of order $\mathcal{O}\left(\frac{2(S+3)}{M}\right)$. According to the simulations results, $S \leq 12$ implies $\gamma^{\mathrm{CE}} \leq 11.7\%$, meaning considerable computational complexity reduction.

We detail in the following the computational complexity for the various multi-user detectors presented in Section 2.7.

4.4.2 Single Antenna Detection

The LMMSE filter for single antenna detection is given by (2.57), and we apply the Krylov subspace method with the following quantities

$$\begin{aligned} &\boldsymbol{D}_1 = \sigma_n^2 \boldsymbol{I}_N\,, \\ &\boldsymbol{D}_2 = \boldsymbol{V}\,, \\ &\boldsymbol{M} = \tilde{\boldsymbol{S}}_r\,, \\ &\boldsymbol{v} = \boldsymbol{e}_{(k,t)}\,, \\ &a = N \quad \text{and} \quad b = KT\,. \end{aligned} \tag{4.11}$$

We have again a situation where $a > b$ in the SISO case and $a < b$ in the MIMO case. We focus on the latter, computations for the former case being straightforward. In this situation, each virtual user (k,t) requires its own filter, while the KT filters (2.57) have a common matrix inverse. This adds a multiplicative factor KT in the global computational

complexity using the Krylov subspace method, finally leading to ('SAD' stands for single antenna detection)

$$\begin{aligned} \mathcal{C}_{\text{LMMSE}}^{\text{SAD}} &= 8N^2(N + \tfrac{KT}{3})R\,, \\ \mathcal{C}_{\text{Krylov}}^{\text{SAD}} &= 8KTN(2S+3)RKT\,. \end{aligned} \tag{4.12}$$

The ratio

$$\gamma^{\text{SAD}} = \frac{C_{\text{K}}^{\text{SAD}}}{C_{\text{LMMSE}}^{\text{SAD}}} = \frac{2S+3}{\left(\frac{N}{KT}\right)^2 + \frac{N}{3KT}} \tag{4.13}$$

satisfies

$$\gamma^{\text{SAD}} \geq \frac{3}{4}(2S+3) \geq 3.75\,. \tag{4.14}$$

Obviously in this case, no computational complexity reduction is achieved.

4.4.3 Joint Antenna Detection

Once again, we apply the model (4.6) with the following quantities

$$\begin{aligned} &\boldsymbol{D}_1 = \sigma_n^2 \boldsymbol{I}_{NR}\,, \\ &\boldsymbol{D}_2 = \boldsymbol{V}\,, \\ &\boldsymbol{M} = \tilde{\boldsymbol{S}}\,, \\ &\boldsymbol{v} = \boldsymbol{e}_{(k,t)}\,, \\ &a = NR \quad \text{and} \quad b = KT\,. \end{aligned} \tag{4.15}$$

We assume a non overloaded system, thus $b \leq a$. As for the single antenna detection case, we have one filter (2.66) per virtual user (k,t) with a common matrix. However, since we perform joint antenna detection, the global multiplicative factor R present in (4.12) disappears, being now included in the dimension $a = NR$. We obtain ('JAD' stands for joint antenna detection)

$$\begin{aligned} \mathcal{C}_{\text{LMMSE}}^{\text{JAD}} &\approx 8(KT)^2(NR + \tfrac{KT}{3})\,, \\ \mathcal{C}_{\text{Krylov}}^{\text{JAD}} &\approx 8NRKT(2S+3)KT\,. \end{aligned} \tag{4.16}$$

For our non-overloaded system, the ratio

$$\gamma^{\text{JAD}} = \frac{C_{\text{K}}^{\text{JAD}}}{C_{\text{LMMSE}}^{\text{JAD}}} = \frac{2S+3}{1 + \frac{KT}{3NR}} \tag{4.17}$$

satisfies

$$\frac{3}{4}(2S+3) \leq \gamma^{\text{JAD}} \leq 2S+3\,. \tag{4.18}$$

Thus γ^{SAD} is of order $\mathcal{O}(2S+3)$. The complexity reduction expected by using the Krylov subspace method is neutralized by the multiplicative factor KT and no computational complexity is achieved. However parallelization of the computations in KT branches is possible, since the KT LMMSE filters can be computed in parallel for a similar overall complexity [DZ06b]. This allows dividing latency time by a factor $\frac{KT}{2S+3}$.

4.4.4 Detection in User Space

The LMMSE filter in this case is given by (2.75). Although more complex than the one in chip space, the product $\boldsymbol{F}^{\mathrm{H}}\boldsymbol{x}$ needs to be computed only once to detect all users simultaneously, allowing getting rid of the multiplicative factor KT in the computational complexity for the Krylov subspace method (see (4.12) and (4.16)).

The model (4.6) we used in the previous applications is not valid here anymore. However, similar computations as detailed in Section 3.3 can be done by analyzing the algorithm in Table 3.1 for this new LMMSE filter. Basic changes consist of replacing the complexity of the computation of $\boldsymbol{a}$, $\boldsymbol{A}$ and the multiplication of $\boldsymbol{A}$ with a vector, which leads to the following expressions ('User' denotes detection after parallel interference cancelation in user space)

$$\begin{aligned} C_{\mathrm{LMMSE}}^{\mathrm{User}} &\approx 8(KT)^2(NR+\tfrac{4}{3}KT)\,, \\ C_{\mathrm{K}}^{\mathrm{User}} &\approx 24KNTRS(2S+3)\,. \end{aligned} \tag{4.19}$$

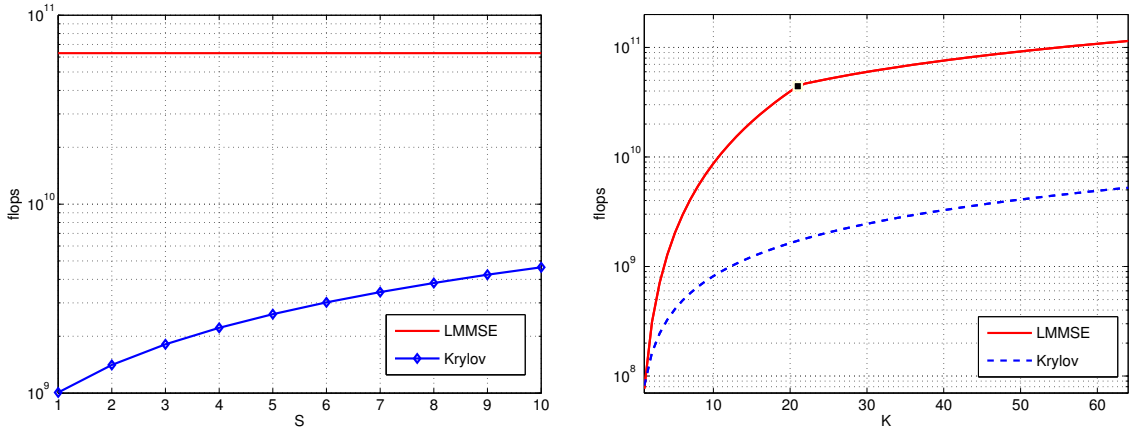
Note that in this case, no simple expression could be obtained for the filter (2.75) using a matrix inversion lemma, thus an eventual computational complexity reduction can not be achieved this way. The ratio

$$\gamma^{\mathrm{User}} = \frac{C_{\mathrm{K}}^{\mathrm{User}}}{C_{\mathrm{LMMSE}}^{\mathrm{User}}} = \frac{6S+9}{KT\left(1+\frac{4KT}{3NR}\right)} \leq \frac{6S+9}{KT}\,. \tag{4.20}$$

According to the simulation results, $S \leq 5$ is sufficient which implies $\gamma^{\mathrm{User}} \leq 30.5\%$, yielding considerable computational complexity reduction.

4.5 Computational Complexity Comparison

From the expressions for computational complexity presented in the previous section and the simulation results, we make the following observations:



(a) (b)

Figure 4.7: Computational complexity for channel estimation. Evolution of the complexity versus the subspace dimension S for $K = 32$ users (a), and versus the number of users K for a dimension $S = 5$ (b).

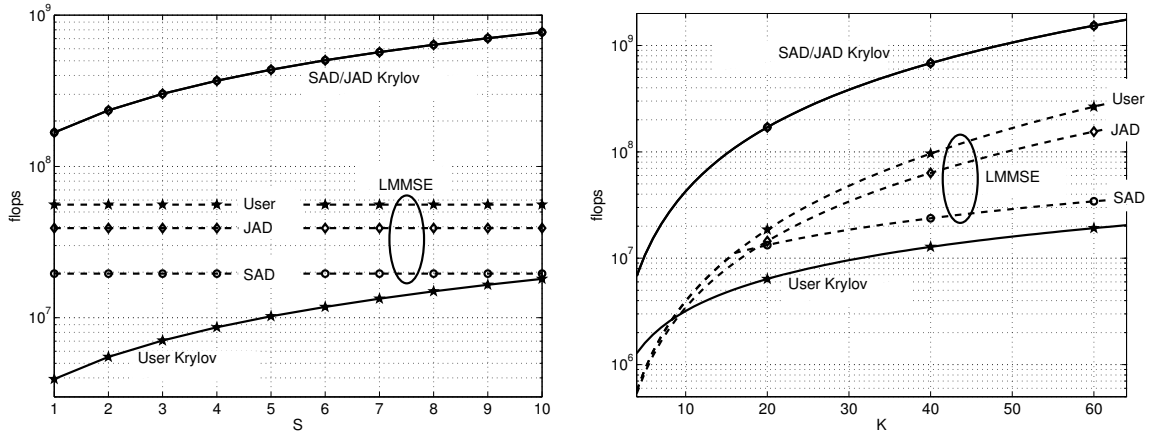
- The computational complexity using the Krylov subspace method is identical for single and joint antenna detection, although their corresponding LMMSE complexity differ by a ratio $\frac{C_{\text{LMMSE}}^{\text{SAD}}}{C_{\text{LMMSE}}^{\text{JAD}}} \approx 0.35$ (see (4.12) and (4.16)).

- All complexities using the Krylov subspace method for multi-user detection are multiple of $8NRKT(2S+3)$, with a multiplicative factor of KT for single or joint antenna detection, and three for PIC in user space, which makes PIC in user space clearly less complex.

- We have seen in Figure 4.3 that the receiver using joint antenna detection requires less receiver iterations than the receiver with single antenna detection for the same performance, which will have some impact on the global computational complexity.

- For parallel interference cancelation in chip space, considerable storage requirement reduction is achieved as well as parallelization of the multi-user detection computations into KT branches, which is highly beneficial for a low latency hardware implementation.

Figure 4.7 shows the computational complexity in $flops$ for channel estimation, versus the Krylov subspace dimension (Figure 4.7.a), and versus the number of users (4.7.b), according to (4.9). The parameters are as in the simulations, see Section 4.1.



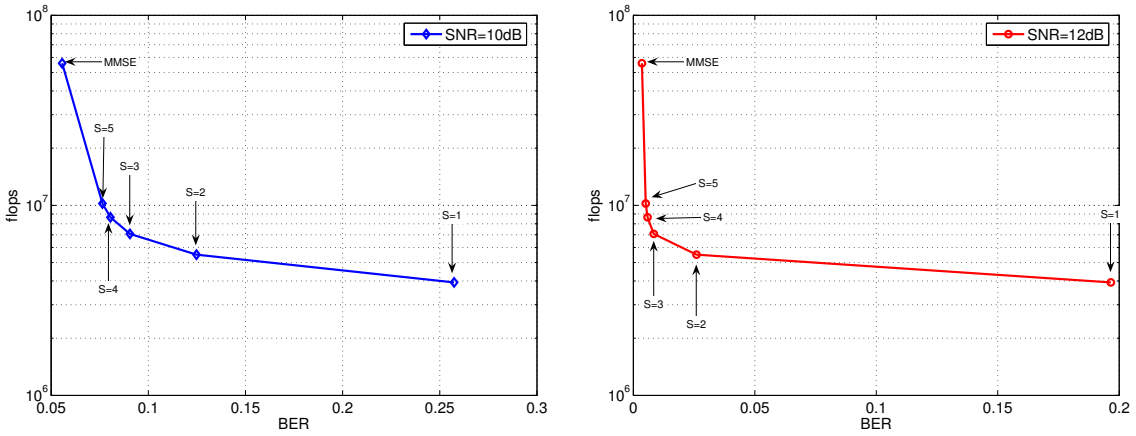
(a) (b)

Figure 4.8: Computational complexity for multi-user detection. Evolution of the complexity versus the subspace dimension S for $K = 32$ users (a), or versus the number of users K for a dimension $S = 5$ (b). Both curves for exact LMMSE and Krylov approximation are shown.

- In Figure 4.7.a, the system has $K = 32$ users and the Krylov subspace dimension varies as $S \in \{1, \ldots, 10\}$.
- In Figure 4.7.b, the Krylov subspace dimension is set to $S = 5$ and the number of users vary as $K \in \{1, \ldots, 64\}$ (the upper limit 64 corresponding to a non overloaded system as defined in Section 4.1). Since the complexity for the LMMSE filter is the minimum of two expressions depending on the system parameters, there is a break point where the change between the two expressions occurs. We observe this point at $K = 21$, corresponding to the point $KDT = M$ where $a = b$.

There is more than one order of magnitude gain in computational complexity using the Krylov subspace method. Thus, implementing channel estimation with the Krylov subspace method will certainly introduce a slight loss in performance, as seen in Figure 4.4, but allows considerable computational complexity reduction.

Figure 4.8 shows the computational complexity in $flops$ for multi-user detection versus the Krylov subspace dimension S for $K = 32$ users (left), and the number of users K for a Krylov dimension $S = 5$ (right). Computational complexities using single antenna detection (denoted through 'SAD' in the figures), joint antenna detection (denoted through 'JAD') or detection after parallel interference cancelation in the user space (denoted through 'User') are shown. It is important here to recall that the complexity using the Krylov subspace method for single or joint antenna detection has the same



(a) (b)

Figure 4.9: Computational complexity for multi-user detection after PIC in the user space vs bit error rate for (a) SNR=10dB and (b) SNR=12dB. We vary the Krylov subspace dimension $S \in \{1, 2, 3, 4, 5\}$. For comparison we also show the complexity using an exact LMMSE filter.

order (see (4.12) and (4.16)). The conclusions we can draw from these results are as follows

- The complexities using exact LMMSE are of similar order for all three detection methods.
- Using the Krylov subspace method for either single or joint antenna detection has much higher complexity than using exact LMMSE filtering. This is due to the fact that only one matrix inverse is needed using LMMSE, whereas we need as many calls of the Krylov algorithm as there are virtual users KT.
- Using the Krylov subspace method for detection after parallel interference cancelation in the user space allows for a considerable complexity reduction of up to one order of magnitude. This method would be the most complex if using exact LMMSE computations, LMMSE computations are least complex for lower number of users $K \leq 10$.

For a better overview, we show the computational complexity in *flops* versus the bit error rate in Figure 4.9. Two values of the SNR $\in \{10, 12\}$ dB are chosen from Figure 4.6.a as the values where most variations can be observed. Multi-user detection is performed in the user space and the channel is estimated using exact LMMSE filtering.

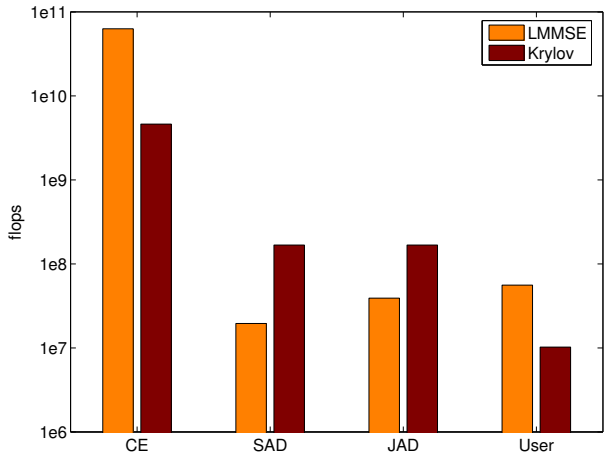


Figure 4.10: Computational complexity per user according to the simulation results from Chapter 4. The Krylov subspace dimensions are $S' = 12$ for channel estimation, and $S = 5$ for multi-user detection. The system has 4×4 channels, $N = 64$ subcarriers, and block length $M = 196$. Channel estimation uses $D = 3$ Slepian basis.

The Krylov subspace dimension is varied $S \in \{1, 2, 3, 4, 5\}$, and for each value of S we show the corresponding BER and computational complexity. From these two figures we can see that

- The Krylov subspace method converges fast, it reaches close to LMMSE performance within a few steps;
- Convergence is faster at higher SNR, with no noticeable performance loss when using $S = 5$ steps at 12dB;
- Sustainable computational complexity reduction is reached when replacing the LMMSE filter by the Krylov subspace method.

Figure 4.10 summarizes all computational complexities according to simulation results in one graph, for channel estimation and various multi-user detectors. We set the parameters as in the simulation results in Chapter 4, i.e. $S' = 12$ for channel estimation and $S = 5$ for multi-user detection. Other parameters are as described in the simulation setup: $K = 32$ users, $T = R = 4$ receive and transmit antennas per user, $N = 64$ subcarriers yielding a load $\zeta = 1/2$; $M = 256$ OFDM symbols from which $J = 60$ are pilots; the basis expansion used for channel estimation has a dimension $D = 3$.

An important fact to note is that the most complex part in the receiver is for channel estimation, no matter which kind of detection is used. There are 2 to 3 orders of magnitude difference between channel estimation and multi-user detection.

4.6 Memory Requirements

We have seen that although not necessarily less complex than exact LMMSE, using the Krylov subspace method might be interesting from a computational point of view. For example, using the Krylov subspace method for detection in the chip space increases the complexity. In turn, the computations can be parallelized into KT branches. Furthermore, as detailed in Section 3.4, less storage is required with the Krylov subspace method: the computation of the matrix $\boldsymbol{A}$ and its inverse are not required, and only some vectors are locally stored for each of the KT branches, until the filter output is computed.

The different initialization methods allow reducing computational complexity by having smaller Krylov subspace dimensions. However, they imply storage of the values that are needed as initial values.

- The Loop method needs to store the values of the filters (2.66) during the whole iteration to use as starting value for the next iteration, which corresponds to a matrix of size $KT \times \min\{KT, NR\}$ for each OFDM symbol $m \in \{0, \ldots, M-1\}$.
- The Time method requires local saving of each filter at time m to use for the next time instance $m+1$, but no long-term storage.

Chapter 5
Krylov Subspace Method for HSDPA

This chapter focuses on another application of the Krylov subspace method (see Chapter 3), namely to equalization of a High Speed Downlink Packet Access (HSDPA) system. HSPDA is a sub-system of Universal Mobile Telecommunication System (UMTS), a mobile communication standard of the 3rd generation that has been released in 1999 by the 3rd Generation Partnership Project (3GPP) [3GP, Mem04b, Mem05b, Mem05c, Mem05d]. It offers multimedia services (e.g. video telephony) and improves existing data services.

UMTS features moderate data rates of up to 384 kbps for packet data services. Efficient, fast and flexible assignment of radio resources to packet users is highly desirable from both a user's and an operator's point of view. Release 5 addresses future packet services over links with fluctuating quality via its HSDPA sub-system [Mem05a].

A linear minimum mean square error (LMMSE) equalizer at the receiver achieves interference reduction and thus higher throughput than a conventional Rake receiver, however at the cost of higher complexity. In this chapter, we briefly present the system and the receiver with LMMSE equalizer in Sections 5.1 and 5.2, respectively. The implementation of the equalizer using the Krylov subspace method is described in Section 5.3. Simulation results are presented in Section 5.4.

5.1 HSDPA System Description

HSDPA introduces three new physical channels into UMTS Release 5, as depicted in Figure 5.1. The high speed physical downlink shared channel (HS-PDSCH) is the data channel shared by all HSDPA users of a single cell in time and code domain. It consists of up to 15 sub-channels corresponding to 15 Walsh-Hadamard channelization codes with spreading factor (SF) 16. The transmission time interval (TTI) is 2ms, which is also called a subframe. A fast scheduler which resides in the base station (Node B) is responsible for selecting packets to be transmitted in each subframe. Up to four high speed shared control channels (HS-SCCH) inform the selected user about the used modulation and

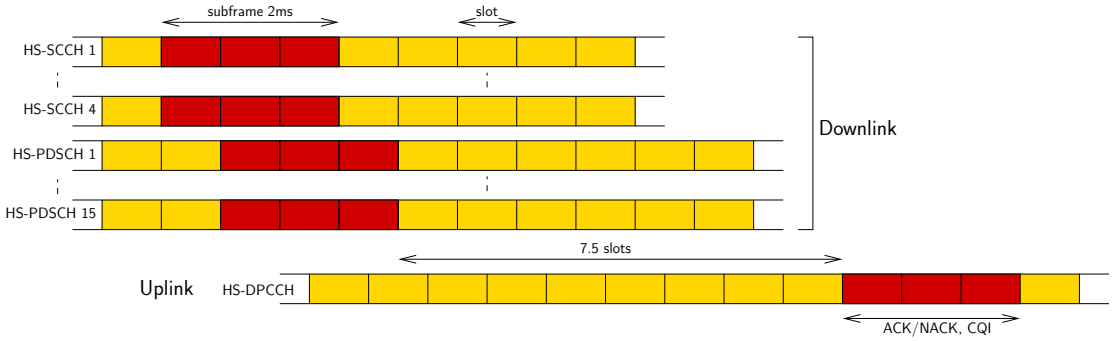


Figure 5.1: HSDPA Channel Structure.

coding scheme, the current hybrid automatic repeat request (HARQ) process, and the redundancy and constellation version (RV) of the retransmission to be used.

The high speed dedicated physical control channel (HS-DPCCH) is an uplink signaling channel assigned to each user. It carries acknowledgment messages (ACK), respectively negative acknowledgment (NACK) messages, as well as the channel quality indicator (CQI). This information is transmitted at most 7.5 slots after the corresponding frame was transmitted.

Three closely coupled procedures govern the performance of HSDPA, namely adaptive modulation and coding, fast HARQ and fast packet scheduling.

• *Adaptive Modulation and Coding*: Instead of compensating the varying downlink radio conditions on the HS-PDSCH by means of fast power control at a fixed data rate, the data rate is adjusted depending on measured channel quality in each subframe [PEM+03]. The data rate adjustment is achieved by puncturing and repetition ("rate matching") of the rate $1/3$ turbo-coded data stream and by selecting either QPSK or 16QAM modulation.

• *Fast HARQ*: In HSDPA, it is foreseen that the user equipment stores the data from the previous transmission to enable joint decoding of retransmission with incremental redundancy (IR). The IR versions are generated by rate matching and constellation rearrangements in case of 16QAM. The user equipment needs internal memory to store the original data packet which is combined with the retransmitted packet [HBB03]. This technique increases the probability of successful decoding for retransmission significantly. Retransmissions are requested until correct decoding or a maximum number of attempts is exceeded [FK04].

• *Fast Packet Scheduling*: This is a key component of HSDPA which is located in the base station (also called Node B). For each TTI, the scheduling algorithm controls the allocation of channelization codes on the HS-PDSCH to the users. The scheduling policy

itself is not standardized in UMTS and various trade-offs between simplicity, throughput and user fairness can be defined.

A simulation environment specifically tailored to HSDPA for investigating HSDPA receiver requirements was developed in [KFP+05, FKG+05]. In the following we shortly describe the transmitter. More details can be found in [Kal07].

For each subframe the HSDPA transmitter generates the HS-PDSCHs for the scheduled users, the synchronization channel (SCH), the primary common control physical channel (P-CCPCH) and the primary common pilot channel (P-CPICH) according to [Mem05b]. To achieve total transmit power $I_{or} = 0$dB, further interfering channels are generated by the so-called orthogonal channel noise simulator (OCNS) and added to the chip stream [Mem04b]. The OCNS channels have a fixed SF 128 and simulate the users or control signals on the other orthogonal channels of the downlink. The HS-SCCHs are not transmitted but signaled error-free to the receiver.

All channels except for the SCH are modulated, spread by orthogonal Walsh-Hadamard sequences and subsequently scrambled by the cell-specific Gold sequence [Gol67]. They are weighted and added to a single chip stream $s[i]$ according to [Mem05d]. The weighting factors are calculated from the relative power ratios of the channels to the total transmit power spectral density (E_c/I_{or}).

We assume a frequency selective, time-varying channel model where each tap of the impulse response has a Clarke Doppler spectrum [Cla68] with a maximum Doppler frequency given by the user equipment (UE) speed. The channel is implemented as a time-varying finite impulse response filter with sample spaced taps. We use oversampling factor 2 and root raised cosine filters at the transmitter and receiver. The sample spaced filter coefficients are generated from the International Telecommunication Union (ITU) channel models by a low-pass interpolation. The received signal after down-sampling is

$$r[i] = \sum_{l=0}^{L_h-1} h_l[i]s[i-l] + n[i]\,, \tag{5.1}$$

where $s[i-l]$ is the transmitted signal delayed by l, $h_l[i]$ are the down-sampled and filtered channel coefficients, L_h denotes the delay spread in chips and $n[i]$ is i.i.d. zero-mean additive white Gaussian noise with variance σ_n^2.

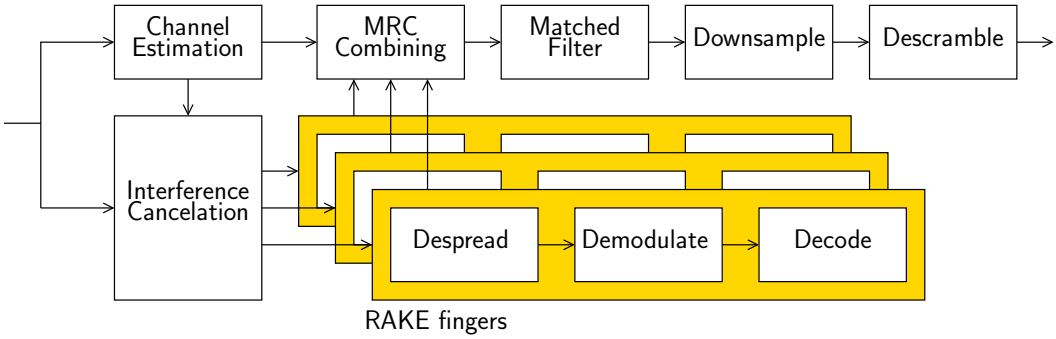


Figure 5.2: Structure of the Rake receiver with interference cancelation.

5.2 Receiver for HSDPA

5.2.1 The Rake receiver

A receiver for UMTS HSDPA consists of two parts. The front end consists of a matched filter, down-sampling and descrambling while the second part consists of despreading, demodulation and decoding of the received signal. Because of the wireless nature of the radio propagation channels, the signal contains echoes arriving at different time instants, creating interference at the receiver. A Rake receiver consists of multiple, time-shifted correlation receivers (called Rake fingers) and allows identification and combination of these echoes. Each of these fingers is allocated a delay (corresponding to a significant channel tap) and performs channel estimation. The outputs of the fingers are then combined using maximum ratio combining.

Multiple access interference as well as the interference from signaling channels like the SCH and the common pilot channel (CPICH) significantly degrade the performance of an HSDPA system [KFP+05] when a Rake receiver is used. A Rake receiver with interference cancelation, as depicted in Figure 5.2, can be used to suppress interference [Kal07]. Another way to fight interference is to use an LMMSE equalizer. In [GMR05], the authors show that an LMMSE equalizer with filter length L_f allows considerable amelioration of the bit error rate. However, the inversion of a matrix of size $L_f \times L_f$ is required, which is rather complex for implementation in real time, especially in the case of time-varying channels.

5.2.2 LMMSE Equalizer

Every subframe, the LMMSE equalizer $\boldsymbol{f}$ minimizes the mean square error between the equalized received signal $\boldsymbol{f}^{\mathrm{H}}\boldsymbol{r}_i$ and the transmitted signal $s[i-d]$:

$$\boldsymbol{f} = \underset{\boldsymbol{f}}{\operatorname{argmin}} \left\{ \mathrm{E}\{ \mid \boldsymbol{f}^{\mathrm{H}}\boldsymbol{r}_i - s[i-d] \mid^2 \} \right\} , \tag{5.2}$$

where d is the delay caused by the system and

$$\boldsymbol{r}_i = [r[i], \ldots, r[i-L_f+1]]^{\mathrm{T}} \tag{5.3}$$

contains the last L_f received samples. The solution is given by

$$\boldsymbol{f} = (\boldsymbol{H}\boldsymbol{H}^{\mathrm{H}} + \sigma^2 \boldsymbol{I}_{L_f})^{-1} \boldsymbol{H}_d \, , \tag{5.4}$$

where σ^2 denotes the ratio σ_s^2/σ_n^2 between transmitted signal power and noise variance and $\boldsymbol{H}_d$ is the d-th column of $\boldsymbol{H}$. The channel matrix $\boldsymbol{H}$ has Toeplitz structure and size $L_f \times (L_h + L_f - 1)$ corresponding to the convolution with the channel impulse response

$$\boldsymbol{H} = \begin{bmatrix} h_0 & \cdots & h_{L_h-1} & & \mathbf{0} \\ & \ddots & \ddots & \ddots & \\ \mathbf{0} & & h_0 & \cdots & h_{L_h-1} \end{bmatrix} . \tag{5.5}$$

5.3 Krylov Equalizer for HSDPA

The LMMSE equalizer (5.4) can be approximated by the Krylov subspace method. We compare the computational complexity of the LMMSE and the approximate Krylov solutions. The same model (3.17) could be used to obtain an expression of the computational complexity, however, it would not take into account the Toeplitz structure of $\boldsymbol{H}$ (5.5). For this reason we will detail in the following the exact computational complexity for computing (3.17).

Results are given in $flops$ as in Chapter 4. We recall that multiplication of vector or matrix entities require as many complex multiplication (CM) as complex additions (CA), and the corresponding complexity in $flops$ is given by multiplying by eight the number of CM.

5.3.1 Complexity Comparison

Using the exact LMMSE filter (5.4), we have to compute the matrix

$$\boldsymbol{A} = \boldsymbol{H}\boldsymbol{H}^{\mathrm{H}} + \sigma^2 \boldsymbol{I}_{L_f} \, . \tag{5.6}$$

Given the structure of $\boldsymbol{H}$, this corresponds to about $0.5L_h^2$ CM. Its inverse using [KM87] and the product $\boldsymbol{A}^{-1}\boldsymbol{b}$ cost about $1.25L_f^2 + 7.25L_f$ CM and L_f^2 CM respectively. This leads to a total computational complexity

$$\mathcal{C}_{\mathrm{LMMSE}} \approx 4L_h^2 + 2L_f(5L_f + 29)\, flops\,. \tag{5.7}$$

Using the Krylov based algorithm, two matrix-vector products in

$$\boldsymbol{H}(\boldsymbol{H}^{\mathrm{H}}\boldsymbol{v}_s) + \sigma^2\boldsymbol{v}_s \tag{5.8}$$

(equivalent to $L_h(L_h + L_f)$ CM) and two inner products to compute α and β ($2L_f$ CM) are required at every step s. The total computational complexity after S iterations is then

$$\mathcal{C}_{\mathrm{Krylov}} \approx 8S(L_fL_h + 2L_f + L_h^2)\, flops\,. \tag{5.9}$$

Furthermore, using the Krylov subspace method allows storage savings: instead of storing $\boldsymbol{A}$, only $\boldsymbol{v}_s$ for $s \in \{1, \ldots, S\}$ is stored.

The computational complexity as well as the equalizer performance increase with L_f, thus a trade-off needs to be found. A reasonable choice is $L_f \approx 3L_h$ [GMR05], where the delay spread L_h is given by the channel model used. A comparison of the computational complexity is shown in Figure 5.3 for $L_f = 48$ and $L_h = 15$ (Vehicular A channels in our simulations). The approximate number of $flops$ is shown versus the Krylov subspace dimension (or iteration number) S.

5.3.2 Choice of Initialization

As already mentioned in Section 3.4, the error $\boldsymbol{r}_S$ resulting from the Krylov subspace method is bounded by [KS99]

$$\|\boldsymbol{r}_S\|_A \leq 2\|\boldsymbol{r}_0\|_A \left(\frac{\sqrt{k_A} - 1}{\sqrt{k_A} + 1}\right)^S, \tag{5.10}$$

where $k_A > 1$ is the condition number of $\boldsymbol{A}$ (ratio of largest and smallest eigenvalues). Convergence is thus assured, but the convergence speed depends strongly on the matrix $\boldsymbol{A}$ and on the initial guess $\boldsymbol{x}_0$. It is necessary to appropriately choose the parameters S and $\boldsymbol{x}_0$.

If no information on $\boldsymbol{f}$ is available at the receiver, we choose $\boldsymbol{x}_0 = [0, \ldots, 0]^{\mathrm{T}}$, also called Zeros method in Section 3.4. However, the LMMSE equalizer (5.4) depends only on the channel estimate $\boldsymbol{H}$. When $\boldsymbol{H}$ is varying slowly, the equalizers from one subframe

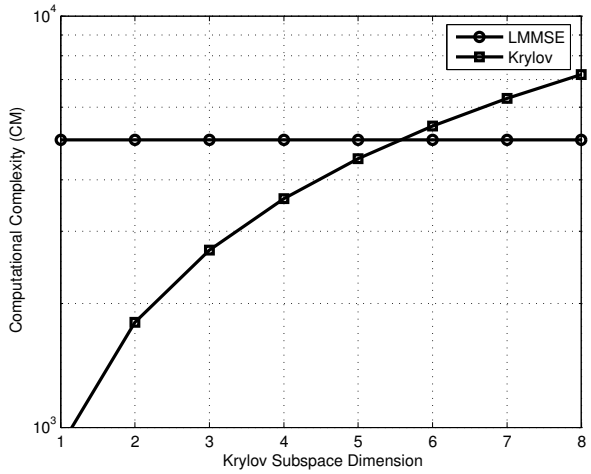


Figure 5.3: Computational complexity for the exact LMMSE equalizer and the Krylov based equalizer, versus the Krylov subspace dimension $S \in \{1, \ldots, 8\}$.

to another are assumed to be strongly related. Thus, a more suitable choice for initial guess is to chose

$$\boldsymbol{x}_0^{(sub+1)} = \boldsymbol{f}^{(sub)}\,, \tag{5.11}$$

where $^{(sub)}$ denotes the subframe index. This could be seen as a Time initialization method. We consider these two approaches in the following.

5.4 Simulation Setup and Results

Throughput simulations for an HSDPA receiver with "UE capability 6" [Mem04a] were carried out for 16QAM modulation. At the transmitter, the fixed reference channel H-Set 3 as defined in [Mem04b] is generated. For all simulated physical channels, the relative power ratios E_c/I_{or} to the total transmit power spectral density I_{or} are set in compliance to the HSDPA test cases [Mem04b].

The E_c/I_{or} of the HS-PDSCH is varied, while it is constant for the P-CPICH, SCH and P-CCPCH as in Table 5.1. No pathloss is assumed ($\hat{I}_{or} = I_{or}$). The interference from other cells and the noise is modeled as additive white Gaussian noise (AWGN) with variance $\sigma_n^2 = I_{oc}$. Table 5.1 summarizes the simulation setup. The simulations include retransmissions as required by the H-Set 3 testing.

We simulated the frequency-selective Rayleigh fading channels ITU Pedestrian A (PA3), B (PB3) and Vehicular A (VA30 and VA120) [Mem04b]. The UE speed is 3 kmh^{-1}, 30 kmh^{-1} and 120 kmh^{-1} for PB3/PA3, VA30 and VA120 respectively. The

FIXED REFERENCED CHANNEL H-SET 3		
Modulation	**QPSK**	**16QAM**
Nominal Avg. Inf. Bit Rate [kbps]	1601	2332
Inter-TTI Distance	1	1
No. of HARQ Processes	6	6
Coding Rate	0.67	0.61
No. of Physical Channel Codes	5	4
SIMULATION PARAMETERS		
RV coding sequence	$\{0, 2, 5, 6\}$	$\{6, 2, 1, 5\}$
UE capability class	6	
Combining	soft	
P-CPICH E_c/I_{or}	$-10\,\mathrm{dB}$	
SCH E_c/I_{or}	$-12\,\mathrm{dB}$	
P-CCPCH E_c/I_{or}	$-12\,\mathrm{dB}$	
OCNS	on	
$\hat{I}_{or}/I_{oc}$	$10\,\mathrm{dB}$	
Equalizer length L_f	48	
Channel coefficient estimation	least squares	
Turbo decoding	max-log-MAP - 8 iterations	

Table 5.1: Simulation setup.

symbols in Figure 5.4 and 5.5 at -3dB and -6dB show the minimum requirements given by the UMTS standard [Mem04b].

We show the throughput for the frequency selective channels PA3 and PB3 in Figure 5.4.a, and for the channels VA30 and VA120 in Figure 5.4.b. We compare the throughput for the exact LMMSE equalizer, and the Zeros and Time methods for the Krylov equalizers, with varying subspace dimension.

We observe that the Krylov subspace method using the Time initialization converges faster to the LMMSE performance than the Zeros one since the slow-variations of the channels are taken into account. However, while increasing the UE speed, the Zeros equalizer performance gets closer to the Time one (for VA30) until it outperforms it (for VA120).

These results show that, when the channel changes slowly, the LMMSE equalizer will show small variations and thus time coherence from one subframe to another can be exploited. However, when the UE is moving fast, this information is no longer benefi-

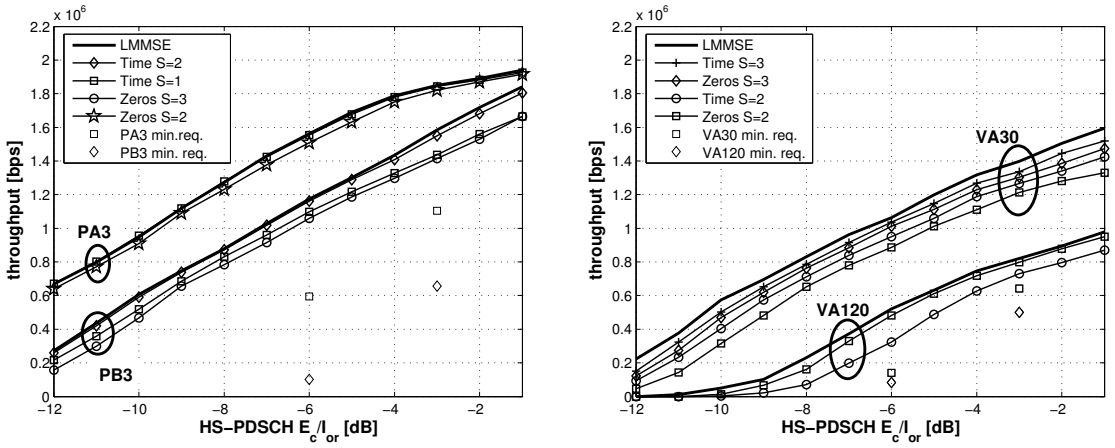


Figure 5.4: Throughput using the Krylov subspace method. We show the ITU-PA3 and PB3 (a), and VA30 and VA120 (b) models with 16QAM modulation and OCNS, using Zeros or Time method with $S \in \{1, 2, 3\}$ iterations. We show also the minimum requirements for all models.

cial. For the investigated channels, a Krylov subspace dimension $S \leq 3$ is sufficient to attain the LMMSE throughput. Referring to Figure 5.3, we see that the computational complexity is reduced by about 50%.

To have an element for comparison, we implemented a low-complexity least mean square (LMS) equalizer as well, see [GMR05]. Throughput simulations for LMS are shown in Figure 5.5. Although the LMS equalizer performs sufficiently when no OCNS are present, it obviously suffers from the interference caused by the OCNS and provides very poor results in that case: a loss up to 9dB can be observed. Hence we can conclude that the Krylov subspace method to approximate an LMMSE equalizer is more robust than the LMS equalizer.

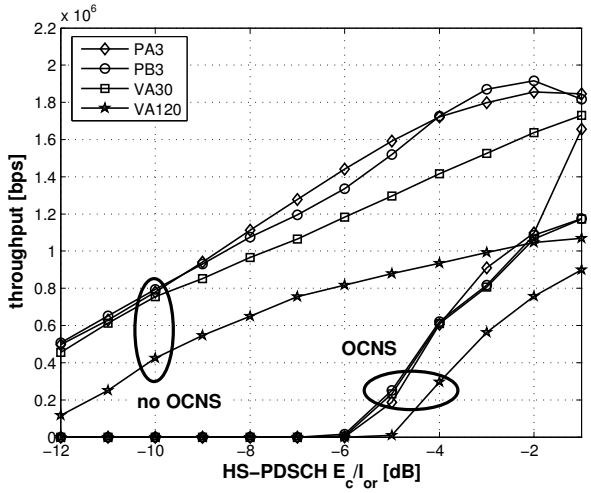


Figure 5.5: Throughput using the LMS equalizer. We show the ITU-PA3, PB3, VA30 and VA120 models with 16-QAM modulation. We give both results with and without OCNS channel.

Part III

Non-Linear Detection: Sphere Decoding

Chapter 6
Sphere Decoder

In Part III we aim at replacing the LMMSE multi-user detector described in Section 2.7 by a sphere decoder. Chapter 6 gives an introduction to sphere decoding. Its application to the iterative receiver will be described in Chapter 7.

6.1 The Maximum Likelihood Equation

Let us consider the signal model of a MIMO system defined by

$$\boldsymbol{y} = \boldsymbol{H}\boldsymbol{b} + \boldsymbol{n}\,, \tag{6.1}$$

where $\boldsymbol{b} = [b_1, \ldots, b_T]^{\mathrm{T}}$ contains the T inputs, $\boldsymbol{H} \in \mathbb{C}^{R\times T}$ is the MIMO channel, $\boldsymbol{y} \in \mathbb{C}^R$ is the received signal and $\boldsymbol{n} \in \mathbb{C}^R$ is additive complex white Gaussian noise (AWGN) with zero-mean and variance $\sigma_n^2 \boldsymbol{I}_R$. An illustration of the system with T inputs and R outputs is shown in Figure 6.1.

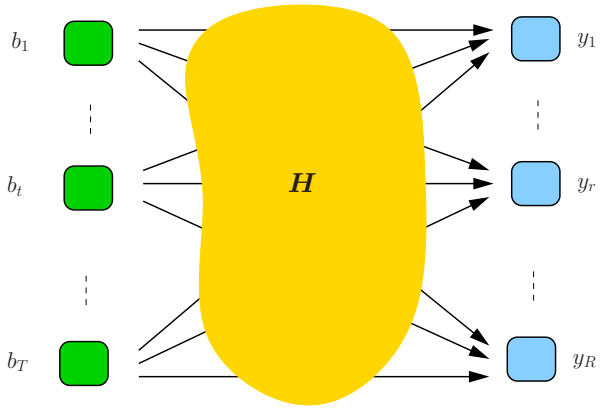


Figure 6.1: MIMO Channel $\boldsymbol{H}$ with T inputs and R outputs.

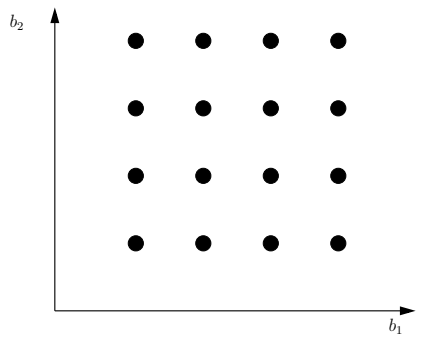


Figure 6.2: Lattice of $\boldsymbol{b} = [b_1, b_2]^{\mathrm{T}}$ for $|\mathcal{A}| = 4$ and $T = 2$.

Depending on the system considered, the T inputs can represent the number of antennas of a single transmitter in a multiple antenna system, the number of users in a multi-user system, or a product of both in a multiuser MIMO system. Similarly, R can be the number of antennas at the receiver side of a multiple antenna system, of subcarriers in an OFDM system, or the product of both as in Chapter 2. The channel $\boldsymbol{H}$ might contain channel coefficients only, but also any additional pre-processing (pre-coding matrix, spreading...). Here we keep a general definition, but details on application to the system from Chapter 2 will be given in the following chapter.

We denote by $\mathcal{A}$ the alphabet of the values taken by the inputs b_t for $t \in \{1, \ldots, T\}$, i.e. $\boldsymbol{b} \in \mathcal{A}^T$. The *maximum likelihood* (ML) detector searches for the vector $\boldsymbol{b}$ in the discrete alphabet $\mathcal{A}^T$ which minimizes the distance $\|\boldsymbol{y} - \boldsymbol{H}\boldsymbol{b}\|^2$, i.e.

$$\hat{\boldsymbol{b}} = \underset{\boldsymbol{b} \in \mathcal{A}^T}{\operatorname{argmin}} \{\|\boldsymbol{y} - \boldsymbol{H}\boldsymbol{b}\|^2\} \,. \tag{6.2}$$

The brute force implementation of a ML detector needs to search over $|\mathcal{A}|^T$ elements ($|\cdot|$ denoting cardinality) and its complexity increases exponentially with T [JO05a]. This might become extremely complex if considering a large alphabet or an increasing number of transmit antennas. Thus, ML detection is rather difficult to implement in a real-time system. Sphere decoders have been introduced in order to reduce the number of possible candidate vectors for the search in (6.2). In the following, we give a short description and history of sphere decoder algorithms.

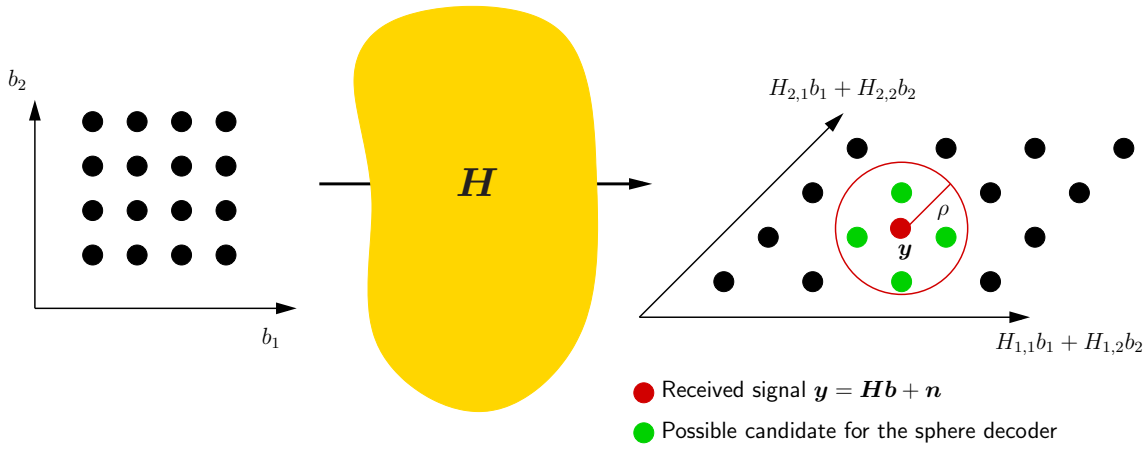


Figure 6.3: Lattice of $\boldsymbol{Hb}$ and Sphere Decoder.

6.2 Sphere Decoder

6.2.1 Definition

Originally developed to search for vectors within a lattice [Poh81, FP85], sphere decoders aim at performing ML detection without requiring exhaustive search [AEVZ02, HV05a, HV05b].

Assuming all b_t for $t \in \{1, \ldots, T\}$ can take all the values in $\mathcal{A}$, the complete alphabet $\mathcal{A}^T$ can be represented by a finite lattice of points, each of them representing a possible vector $\boldsymbol{b}$. A simple example for an alphabet of cardinality $|\mathcal{A}| = 4$ and two dimensions (i.e. $T = 2$) is given in Figure 6.2.

After being transmitted through the channel $\boldsymbol{H}$, the elements given by $\boldsymbol{Hb}$ belong to a new lattice that is not necessarily orthogonal anymore. A sphere decoder searches only the elements $\boldsymbol{b}$ whose image $\boldsymbol{Hb}$ lies within a sphere centered around the received signal $\boldsymbol{y}$, with a predefined radius ρ. An illustration of the image lattice obtained through the channel

$$\boldsymbol{H} = \begin{bmatrix} H_{1,1} & H_{1,2} \\ H_{2,1} & H_{2,2} \end{bmatrix} \tag{6.3}$$

and of the sphere centered on $\boldsymbol{y}$ with radius ρ is given in Figure 6.3.

Since the ML solution is the closest point to the received signal $\boldsymbol{y}$, it is assured to be in any sphere centered on $\boldsymbol{y}$, under the condition that the sphere is not empty, i.e. the radius is not too small to reach the closest element. This means the value of ρ needs to be determined in a way that at least one element of the received lattice belongs to the

sphere. In our simulations, we will use a sub-optimal solution to choose ρ, i.e.

$$\rho = \|\boldsymbol{y} - \boldsymbol{H}\boldsymbol{b}_{\mathrm{ZF}}\| \,, \tag{6.4}$$

where the zero-forcing solution $\boldsymbol{b}_{\mathrm{ZF}}$ is given by

$$\boldsymbol{b}_{\mathrm{ZF}} = \left(\boldsymbol{H}^{\mathrm{H}}\boldsymbol{H}\right)^{-1}\boldsymbol{H}^{\mathrm{H}}\boldsymbol{y} \,. \tag{6.5}$$

Computing $\boldsymbol{b}_{\mathrm{ZF}}$ introduces complexity in the decoding stage, however this is not relevant for comparing different sphere decoder implementations, as we do in Chapter 7.

For a given radius ρ, the sphere constraint

$$\|\boldsymbol{y} - \boldsymbol{H}\boldsymbol{b}\|^2 < \rho^2 \tag{6.6}$$

allows to rewrtite the ML equation (6.2) as

$$\hat{\boldsymbol{b}} = \underset{\boldsymbol{b}\in\mathcal{A}^T | \|\boldsymbol{y}-\boldsymbol{H}\boldsymbol{b}\|^2<\rho^2}{\operatorname{argmin}} \{\|\boldsymbol{y} - \boldsymbol{H}\boldsymbol{b}\|^2\} \,. \tag{6.7}$$

6.2.2 Implementation

Let us consider the *thin* QR factorization of the matrix $\boldsymbol{H}$, as defined in [GL96], such as $\boldsymbol{H} = \boldsymbol{Q}\boldsymbol{R}$, where $\boldsymbol{Q} \in \mathbb{C}^{R\times T}$ is a unitary matrix (i.e. $\boldsymbol{Q}^{\mathrm{H}}\boldsymbol{Q} = \boldsymbol{I}_T$) and $\boldsymbol{R} \in \mathbb{C}^{T\times T}$ is an upper triangular matrix. This factorization is known to be unique [GL96].

Since the matrix $\boldsymbol{Q}$ is unitary, the sphere constraint (6.6) is equivalent to

$$\|\boldsymbol{z} - \boldsymbol{R}\boldsymbol{b}\|^2 < \rho^2 \,, \tag{6.8}$$

where $\boldsymbol{z} = \boldsymbol{Q}^{\mathrm{H}}\boldsymbol{y}$. The ML solution minimizes the error

$$\boldsymbol{\epsilon} = \boldsymbol{z} - \boldsymbol{R}\boldsymbol{b} \,. \tag{6.9}$$

At this point, let us give some definitions useful to describe the sphere decoder algorithm.

- The **partial vectors** $\in \mathbb{C}^{(T-t+1)}$ for $t \in \{1, \ldots, T\}$, containing elements t to T are given by

$$\boldsymbol{z}^{(t)} = \begin{bmatrix} z_t \\ \vdots \\ z_T \end{bmatrix}, \quad \boldsymbol{b}^{(t)} = \begin{bmatrix} b_t \\ \vdots \\ b_T \end{bmatrix}, \quad \text{and} \quad \boldsymbol{\epsilon}^{(t)} = \begin{bmatrix} \epsilon_t \\ \vdots \\ \epsilon_T \end{bmatrix}; \tag{6.10}$$

- The **partial matrix** $\in \mathbb{C}^{(T-t+1)\times(T-t+1)}$, for $t \in \{1,\ldots,T\}$

$$\boldsymbol{R}^{(t)} = \begin{bmatrix} R_{t,t} & \cdots & R_{t,T} \\ 0 & \ddots & \vdots \\ 0 & 0 & R_{T,T} \end{bmatrix} \tag{6.11}$$

  corresponds to the lower-right corner of $\boldsymbol{R}$.

- The **partial squared distance** d_t^2 for $t \in \{1,\ldots,T\}$ is defined as the squared norm of the partial error vector $\boldsymbol{\epsilon}^{(t)}$ according to

$$d_t^2 = \|\boldsymbol{\epsilon}^{(t)}\|^2 . \tag{6.12}$$

The matrix $\boldsymbol{R}$ being upper triangular, it allows to write $\boldsymbol{\epsilon}^{(t)}$, $t \in \{1,\ldots,T\}$, as

$$\begin{bmatrix} \epsilon_t \\ \vdots \\ \epsilon_T \end{bmatrix} = \begin{bmatrix} z_t \\ \vdots \\ z_T \end{bmatrix} - \begin{bmatrix} R_{t,t} & \cdots & R_{t,T} \\ 0 & \ddots & \vdots \\ 0 & 0 & R_{T,T} \end{bmatrix} \begin{bmatrix} b_t \\ \vdots \\ b_T \end{bmatrix} \tag{6.13}$$

$$\boldsymbol{\epsilon}^{(t)} = \boldsymbol{z}^{(t)} - \boldsymbol{R}^{(t)} \boldsymbol{b}^{(t)} ,$$

as function of the partial vectors and matrix only. This yields the recursive expression for the partial squared distance

$$d_t^2 = \sum_{i=t}^{T} |\epsilon_i|^2 = d_{t+1}^2 + |\epsilon_t|^2 . \tag{6.14}$$

Using (6.14), the squared distance

$$\|\boldsymbol{\epsilon}\|^2 = d_1^2 \tag{6.15}$$

can be computed starting from the initial value

$$d_T^2 = |\epsilon_T|^2 = |z_T - R_{T,T} b_T|^2 , \tag{6.16}$$

and adding the value of

$$|\epsilon_t|^2 = \left| z_t - [R_{t,t},\ldots,R_{t,T}]\boldsymbol{b}^{(t)} \right|^2 , \tag{6.17}$$

0	Set $\mathcal{V}_{T+1} = \emptyset$
1	For t from T to 1
2	for all $\boldsymbol{b}^{(t+1)} \in \mathcal{V}_{t+1}$
3	for all $b_t \in \mathcal{A}$
4	compute d_t^2 from d_{t+1}^2 using (6.14)
5	if $d_t^2 \leq \rho^2$, store $\boldsymbol{b}^{(t)} \in \mathcal{V}_t$ and d_t^2
6	end

Algorithm 6.1: The T steps of the sphere decoder

for t decreasing from $T-1$ to 1. Clearly, the partial squared distance is a monotonically increasing function for decreasing t. It is obvious that if the partial squared distance exceeds the squared radius, i.e.

$$d_{t_0}^2 > \rho^2 \tag{6.18}$$

for some $t = t_0$, the same is true for the total squared distance

$$\|\boldsymbol{\epsilon}\|^2 > \rho^2 \,. \tag{6.19}$$

Hence, all vectors $\boldsymbol{b}$ corresponding to $\boldsymbol{b}^{(t_0)}$ lie outside the sphere. As a further consequence, all vectors in $\mathcal{A}^T$ that have the same partial vector $\boldsymbol{b}^{(t_0)}$ at step t_0 can be discarded from the search as well, since they have the same partial squared distance $d_{t_0}^2 > \rho^2$ and lie outside the sphere as well.

As a result, we obtain an efficient algorithm to search the elements $\boldsymbol{b} \in \mathcal{A}^T$ and discard the ones lying outside the sphere, shown in Algorithm 6.1.

After each step t, a set $\mathcal{V}_t$ with cardinality q_t of valid partial vectors $\boldsymbol{b}^{(t)}$ is available and helps determining the next set $\mathcal{V}_{t-1}$. After the last step, the sphere decoding algorithm terminates and provides the set $\mathcal{V}_1$ containing the vector $\boldsymbol{b}$ with the smallest squared distance d_1^2, provided ρ^2 is not too small. This output is the ML solution (6.2).

Note that the radius ρ helps reducing the search space $\mathcal{A}^T$ to a smaller set defined as a sphere centered on the received signal. If it is set to $\rho = \infty$, all candidates are admissible since $d_t^2 \leq \infty$ and the set of valid candidates after step t is constant

$$\mathcal{V}_t = \mathcal{A}^{T-t+1} \,. \tag{6.20}$$

In this case, sphere decoding simply allows efficient implementation of an exhaustive search, without reducing the search space.

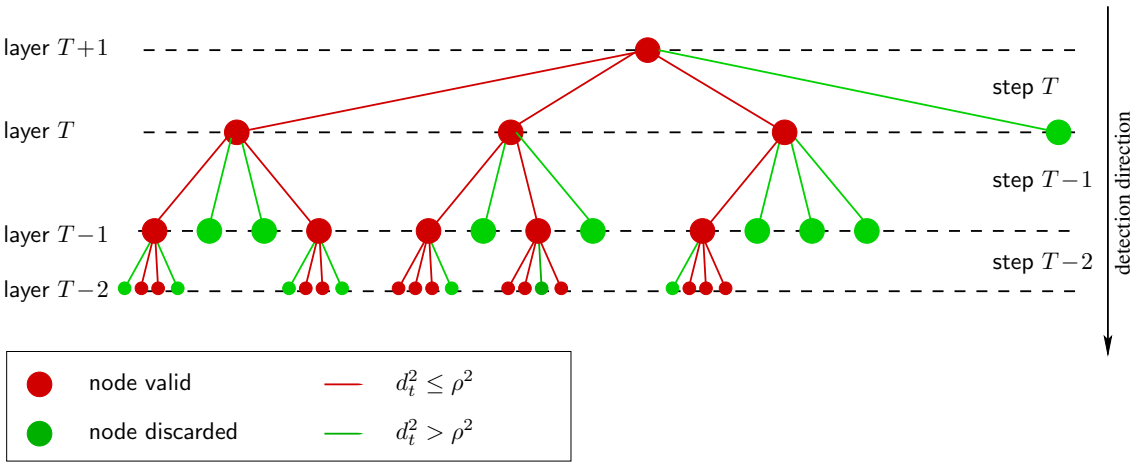


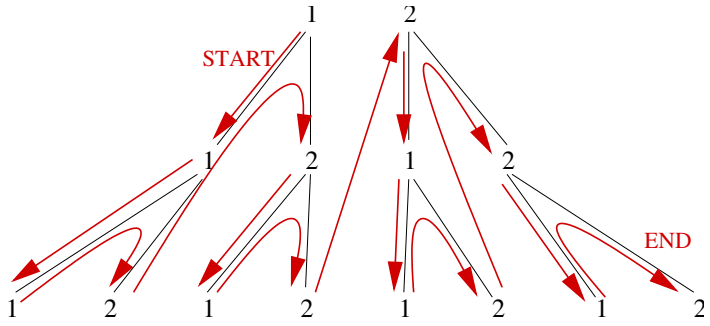
Figure 6.4: Depth-first tree search for an alphabet with $|\mathcal{A}| = 4$ elements and $T = 2$ inputs.

6.2.3 Tree Search

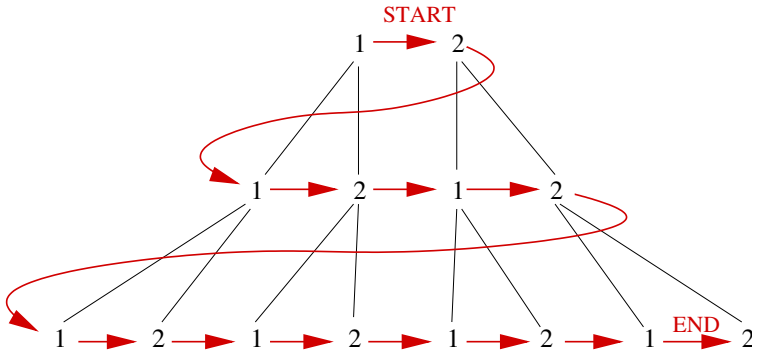
The sphere decoding algorithm described in Algorithm 6.1 can be seen as a pruning algorithm on a tree, whose branches correspond to the elements of the symbol alphabet.

Sphere decoding starts the search from the root of the tree and works its way down the branches. Each layer $t \in \{1, \dots, T\}$ of the tree corresponds to a new element b_t of the symbol vector $\boldsymbol{b}$, and the nodes of this layer to possible values of $b_t \in \mathcal{A}$. At one node of layer $T - t + 1$, the squared partial distance d_t^2 in (6.14) is computed. This way, working down the tree, the algorithm computes the squared distance of all possible symbol vectors, until the sphere constraint is violated, which occurs when d_t^2 exceeds ρ^2. In this case, the corresponding branch is pruned and its children nodes are discarded. Then, the algorithm proceeds through a different branch until it reaches a valid leaf (i.e. such that $d_1^2 \leq \rho^2$) which becomes a valid candidate. If several candidates are found, the ML solution is the one with the lowest squared distance d_1^2.

An example of tree search for an alphabet with four elements, $T = 3$ is given in Figure 6.4. At one node of layer $T - t + 1$, If $d_t^2 \leq \rho$, the node is kept as a valid candidate and search continues; If $d_t^2 > \rho$, the node and its children lie outside the sphere and are discarded.



(a) Depth-first tree search



(b) Breadth-first tree search

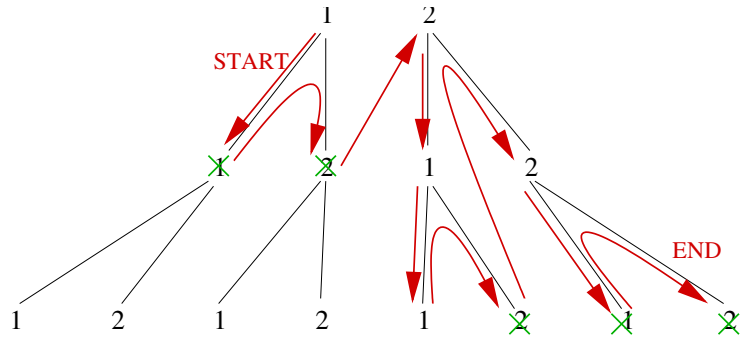
Figure 6.5: Depth- and breadth-first tree search for a binary alphabet (with elements denoted using '1' and '2').

6.3 Classification of Sphere Decoders

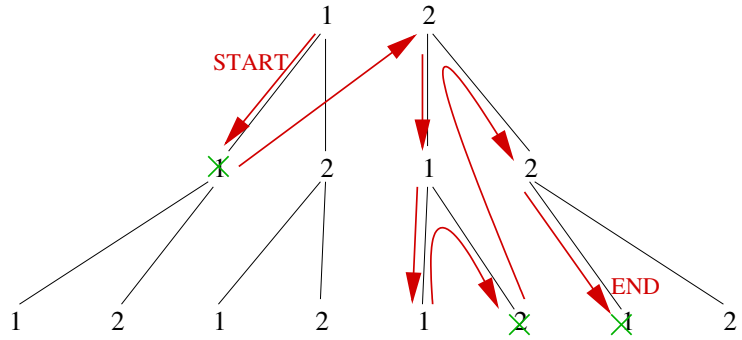
When implementing a sphere decoder, two questions need to be answered: how is ρ determined? And how to define the ordering of the nodes to be searched? Both radius and ordering have an influence on the number of nodes visited during the search and thus on the overall complexity.

Obviously, the "best" radius ρ is just large enough in order for the sphere to contain exactly one solution. However, if chosen too large, it might contain the whole alphabet $\mathcal{A}$, leading back to exhaustive search and high complexity. A possibility to bypass this problem is for example by using an adaptive radius [DEGC03]. As soon as a leaf node is reached, radius ρ is replaced by the norm d_1^2 corresponding to this leaf.

This section provides a brief classification of the various existing sphere decoder algorithms. Firstly, we distinguish two kinds of prioritization for the tree search, see Figure 6.5 for an example with three layers and a binary alphabet.



(a) Effect of the Sphere Constraint



(b) Effect of the ordering of elements

Figure 6.6: Sphere constraint (a) and elements ordering (b) for depth-first search.

Depth-First Tree Search is shown in Figure 6.5.a. Priority is given to the vertical direction. The search goes through the first branch down to the first leaf. If a node is discarded or if a leaf is reached, the search goes one layer up and continues.

Breadth-First Tree Search is shown in Figure 6.5.b. The search prioritizes the horizontal levels and goes down to the next layer only when search at the previous level is completed. If the search is complete, i.e. if all nodes are visited, both Breadth- and Depth-First Tree Search have the same complexity. A possibility to decrease the number of visited nodes is by using horizontal ordering of the nodes. At each level, the nodes are ordered in some way, for example with increasing partial distance, which may allow to stop searching this level earlier. In Figure 6.6.a we show the effect of the sphere constraint. The crossed nodes are nodes checked and discarded because they violate the sphere constraint. Their children are not checked at all. In Figure 6.6.b, we add horizontal ordering. The decoder has further knowledge of the ordering of the elements (namely

'1' $>$ '2' according to some metric). Thus, if a node with bit '1' is discarded, so will be the following node with bit '2' as well as its children. This allows visiting two nodes less in our example. We assume both schemes show exactly the same tree (i.e. there is also ordering in Figure 6.6.a, but the sphere decoder has no knowledge about it.

Finally, we present a short overview of some well-known sphere decoder variants and their characteristics.

• **Fincke-Pohst Algorithm:** first published in 1985 [Poh81, FP85], this is a depth-first search algorithm. The nodes are sorted according to increasing values (or absolute values). Sorting is independent of the search and absolute. This strategy has been introduced to digital communications in [VB93, VB99]. A geometrical approach of the Fincke-Pohst algorithm is given in [CL02].

• **Schnorr-Euchner Algorithm:** originally presented in [SE94] in 1994, it is adapted from the Fincke-Pohst algorithm, with a different ordering scheme. Nodes are sorted according to increasing partial distances. Since it depends on the channel realization, this ordering is new for every search, which might increase complexity. On the other hand, when a node is discarded, so are the following nodes. This strategy has been used in [AEVZ02].

• K**-Best Search:** this is a breadth-first search using the following metric: at each layer t, only K nodes corresponding to the K smallest partial distances are stored, the others are discarded. It allows fixed complexity search [WTCM02, BT06] (with constant number of visited nodes), however its solution is not necessarily the ML solution.

• **Best First Search:** each newly computed node is added to the set of already computed nodes, in such a manner that all nodes in this set are sorted with increasing cost function [PPWL04]. As soon as one leaf is reached, it is added to the set and the search is stopped. This is guaranteed to be the exact ML solution.

• **Adaptive Radius:** to avoid an empty sphere, one can start with an infinite radius. Every leaf node reached allows updating the radius with a smaller value while ensuring that the sphere will not be empty [DEGC03,XWZW04]. Updating the radius has different effects depending on the tree search used. For example, the first leaf reached using the Schnorr-Euchner algorithm is the zero-forcing solution (also called Babai estimate [Bab86]). When used with breadth-first search, the first leaf is immediately the ML solution.

Chapter 7
Reduced-Rank Sphere Decoder

In this chapter, we apply a sphere decoder as presented in Chapter 6 to the OFDM-CDMA iterative receiver described in Chapter 2. First, a straightforward application to detection on a per user basis is presented. A better implementation, making use of the channel model and more suitable to the considered system, is developed afterwards. Finally, application to computation of log-likelihood ratios (LLR) is detailed.

7.1 Sphere Decoder Applied to OFDM-CDMA

The global received signal in a joint antenna detection scheme (2.63)

$$\boldsymbol{y} = \tilde{\boldsymbol{S}}\boldsymbol{b} + \boldsymbol{n} \tag{7.1}$$

is obtained by stacking the received signals $\boldsymbol{y}_r$ for each receive antenna $r \in \{1, \ldots, R\}$:

$$\boldsymbol{y} = [\boldsymbol{y}_1^{\mathrm{T}}, \ldots, \boldsymbol{y}_r^{\mathrm{T}}, \ldots, \boldsymbol{y}_R^{\mathrm{T}}]^{\mathrm{T}} . \tag{7.2}$$

In (7.1), the contribution of user k stemming from the symbols

$$\boldsymbol{b}^{(k)} = [b_{(k,1)}, \ldots, b_{(k,t)}, \ldots, b_{(k,T)}]^{\mathrm{T}} \tag{7.3}$$

is given by

$$\boldsymbol{y}^{(k)} = \tilde{\boldsymbol{S}}^{(k)}\boldsymbol{b}^{(k)} , \tag{7.4}$$

where the effective spreading matrix

$$\tilde{\boldsymbol{S}}^{(k)} = \begin{bmatrix} \tilde{\boldsymbol{s}}_{1,(k,1)} & \cdots & \tilde{\boldsymbol{s}}_{1,(k,T)} \\ \vdots & \tilde{\boldsymbol{s}}_{r,(k,t)} & \vdots \\ \tilde{\boldsymbol{s}}_{R,(k,1)} & \cdots & \tilde{\boldsymbol{s}}_{R,(k,T)} \end{bmatrix} \in \mathbb{C}^{NR \times T} \tag{7.5}$$

contains the effective spreading sequences from all transmit antennas of user k to all receive antennas. Hence, the global received signal (7.1) can also be written as

$$\boldsymbol{y} = \sum_{k=1}^{K} \boldsymbol{y}^{(k)} + \boldsymbol{n}\,. \tag{7.6}$$

After comparing the generic model (6.1) to our signal model (7.1), a straightforward extension of sphere decoding to OFDM-CDMA is done by setting

$$\boldsymbol{H} = \tilde{\boldsymbol{S}}\,. \tag{7.7}$$

However, due to the large dimensions of the system ($\boldsymbol{b} \in \mathbb{C}^{KT}$), even a very tight radius yields prohibitively complex sphere decoding. Arranging the KT symbols $b_{(k,t)}$ into sub-vectors of smaller dimensions can solve this complexity issue. Under the assumption that all channels are independent and uncorrelated, the grouping of transmit antennas based on their physical co-location is not a prerequisite and any partitioning would do. Nevertheless, for intuitive reasons and eventual extension to correlated co-located antennas, we prefer partitioning on a per user basis, as described below.

We aim at performing detection on a per user basis, i.e. applying sphere decoding on the signal (7.4) for user k, for $k \in \{1, \ldots, K\}$. However, only the global receive signal (7.1) is available at the receiver. An estimation of the contribution of one user k in (7.4) can be obtained using parallel interference cancelation. This consists in removing the estimated contribution of the other users according to

$$\boldsymbol{y}^{(k)} = \boldsymbol{y} - \sum_{k' \neq k} \boldsymbol{y}^{(k')}\,. \tag{7.8}$$

Using soft estimates $\tilde{\boldsymbol{b}}^{(k)}$ computed from the *extrinsic* probabilities as given in (2.21), we define the received signal after parallel interference cancelation for user k by

$$\tilde{\boldsymbol{y}}^{(k)} \triangleq \boldsymbol{y} - \left(\sum_{k' \neq k} \tilde{\boldsymbol{S}}^{(k')} \tilde{\boldsymbol{b}}^{(k')} \right) = \boldsymbol{y} - \tilde{\boldsymbol{S}}\tilde{\boldsymbol{b}} + \tilde{\boldsymbol{S}}^{(k)} \tilde{\boldsymbol{b}}^{(k)}\,. \tag{7.9}$$

Assuming perfect interference cancelation, there is equality between (7.8) and (7.9) and we can apply sphere decoding.

Comparing the generic model (6.1) to the signal received from user k (7.9), we now set

$$\boldsymbol{b} = \boldsymbol{b}^{(k)}, \quad \boldsymbol{H} = \tilde{\boldsymbol{S}}^{(k)}, \quad \text{and} \quad \boldsymbol{y} = \tilde{\boldsymbol{y}}^{(k)}\,. \tag{7.10}$$

The sphere constraint becomes

$$\|\tilde{\boldsymbol{y}}^{(k)} - \tilde{\boldsymbol{S}}^{(k)}\boldsymbol{b}^{(k)}\|^2 < \rho^2\,, \tag{7.11}$$

and the ML equation finally writes

$$\hat{\boldsymbol{b}}^{(k)} = \underset{\boldsymbol{b}\in\mathcal{A}^T | \|\tilde{\boldsymbol{y}}^{(k)} - \tilde{\boldsymbol{S}}^{(k)}\boldsymbol{b}\|^2 < \rho^2}{\operatorname{argmin}} \{\|\tilde{\boldsymbol{y}}^{(k)} - \tilde{\boldsymbol{S}}^{(k)}\boldsymbol{b}\|^2\}\,. \tag{7.12}$$

Although the use of a sphere decoder allows implementing a maximum likelihood detector with relatively low-complexity, there are always possibilities to further reduce complexity. As shown in Chapter 6, before the actual detection the need for a QR factorization of the channel arises. This is a rather expensive part of the sphere decoder in terms of computational complexity, especially in time-varying channels where the channel realization changes for every OFDM symbol within one data block.

In the following section, we describe how to make use of the channel model described in Section 2.6.1 in order to allow efficient implementation of the sphere decoder. We show how the channel can be factorized in a product of one time-invariant and one time-varying sparse block diagonal matrix. Using this structure, one QR factorization needs to be performed per block of length M only.

As described previously, sphere decoding is now applied on a per user basis, i.e. on the received signal after parallel interference cancelation, in parallel for all K users. From now on, we describe sphere decoding for one specific user and drop the index k for convenience.

7.2 Reduced-Rank Sphere Decoder

In this section, we aim at using reduced-rank basis expansion to model the channel as described in Section 2.6.1 for efficient implementation of the sphere decoder. Hence, we refer to this specific implementation of a sphere decoder as a reduced-rank sphere decoder.

The sphere decoder as presented in Chapter 6 and applied to CDMA in Section 7.1 requires a QR factorization of the matrix $\boldsymbol{H}$ (or more precisely in our case the effective spreading sequence matrix $\tilde{\boldsymbol{S}}$). In time-varying channels, this means a QR factorization for every OFDM symbol in one data block.

7.2.1 Channel Basis Expansion

As we have seen in Section 2.6.1, the time-varying channel between transmit antenna t and receive antenna r can be approximated as linear superposition of D basis functions

band-limited within the time interval $m \in \mathcal{I}_M = \{0, \ldots, M-1\}$ using (2.30)

$$\tilde{\boldsymbol{g}}_{r,t}[m] = \boldsymbol{\Gamma}_{r,t} \boldsymbol{f}[m] \,, \tag{7.13}$$

where $\boldsymbol{\Gamma}_{r,t}$ contains the constant projection coefficients, and

$$\boldsymbol{f}[m] = \begin{bmatrix} u_0[m] \\ \vdots \\ u_{D-1}[m] \end{bmatrix} \in \mathbb{C}^{\,D} \tag{7.14}$$

is defined as in (2.32). The $\boldsymbol{u}_i = [u_i[0], \ldots, u_i[M-1]]^{\mathrm{T}}$ for $i \in \{0, \ldots, D-1\}$ are the time-varying basis vectors. Note that we use the Slepian basis expansion in this thesis and for simulations; however, the approach described in the following could be applied to any basis expansion.

After estimation of the coefficients $\boldsymbol{\Gamma}_{r,t}$ as discussed in Section 2.6.2, the effective spreading sequence (2.48)

$$\tilde{\boldsymbol{s}}_{r,t}[m] = \mathrm{diag}(\boldsymbol{s}_t) \boldsymbol{\Gamma}_{r,t} \boldsymbol{f}[m] \tag{7.15}$$

for the time-varying channel $\tilde{\boldsymbol{g}}_{r,t}$ allows rewriting the signal received at antenna r as

$$\boldsymbol{y}_r[m] = \sum_{t=1}^{T} \mathrm{diag}(\boldsymbol{s}_t) \boldsymbol{\Gamma}_{r,t} \boldsymbol{f}[m] b_t[m] + \boldsymbol{n}_r[m] \tag{7.16}$$

as well as the global received signal

$$\begin{aligned} \boldsymbol{y}[m] &= [\boldsymbol{y}_1[m]^{\mathrm{T}}, \ldots, \boldsymbol{y}_R[m]^{\mathrm{T}}]^{\mathrm{T}} \\ &= \boldsymbol{\Gamma} \boldsymbol{F}[m] \boldsymbol{b}[m] + \boldsymbol{n}[m] \,. \end{aligned} \tag{7.17}$$

In (7.16) and (7.17), $\boldsymbol{n}_r$ and $\boldsymbol{n}$ denote AWGN with zero mean and variance $\sigma^2 \boldsymbol{I}_N$ and $\sigma^2 \boldsymbol{I}_{NR}$, respectively. The time-invariant matrix

$$\boldsymbol{\Gamma} = \begin{bmatrix} \mathrm{diag}(\boldsymbol{s}_1) \boldsymbol{\Gamma}_{1,1} & \cdots & \mathrm{diag}(\boldsymbol{s}_T) \boldsymbol{\Gamma}_{1,T} \\ \vdots & \ddots & \vdots \\ \mathrm{diag}(\boldsymbol{s}_1) \boldsymbol{\Gamma}_{R,1} & \cdots & \mathrm{diag}(\boldsymbol{s}_T) \boldsymbol{\Gamma}_{R,T} \end{bmatrix} \in \mathbb{C}^{\,N \times DT} \tag{7.18}$$

contains both the spreading sequences and the projection coefficients, whereas

$$\boldsymbol{F}[m] = \begin{bmatrix} \boldsymbol{f}[m] & \boldsymbol{0} & \boldsymbol{0} \\ \boldsymbol{0} & \ddots & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{f}[m] \end{bmatrix} \in \mathbb{R}^{\,DT \times T} \tag{7.19}$$

is a time-varying sparse block diagonal matrix containing a repetition of the basis expansion (7.14) at time m.

Using these notations, the signal after parallel interference cancelation (7.9) becomes

$$\tilde{\boldsymbol{y}}[m] \approx \boldsymbol{\Gamma}\boldsymbol{F}[m]\boldsymbol{b}[m] + \boldsymbol{n}[m]\,. \tag{7.20}$$

We make use of this new structure (7.20) of the received signal (after interference cancelation) to develop efficient implementation of a sphere decoder applied to our system.

7.2.2 Implementation

Let the unitary matrix $\boldsymbol{Q} \in \mathbb{C}^{NR\times DT}$ and the upper triangular matrix $\boldsymbol{R} \in \mathbb{C}^{DT\times DT}$ be the *thin* QR factorization [GL96] of $\boldsymbol{\Gamma} = \boldsymbol{Q}\boldsymbol{R}$. Multiplying (7.20) from the left side with $\boldsymbol{Q}^{\mathrm{H}}$ leads to

$$\boldsymbol{z}[m] = \boldsymbol{Q}^{\mathrm{H}}\tilde{\boldsymbol{y}}[m] = \boldsymbol{R}\boldsymbol{F}[m]\boldsymbol{b}[m] + \boldsymbol{Q}^{\mathrm{H}}\boldsymbol{n}[m]\,, \tag{7.21}$$

where $\boldsymbol{Q}^{\mathrm{H}}\boldsymbol{n}$ is AWGN with zero mean and covariance matrix $\sigma_n^2\boldsymbol{I}_{DT}$. The maximum likelihood detector after QR factorization of $\boldsymbol{\Gamma}$ is

$$\hat{\boldsymbol{b}}[m] = \underset{\boldsymbol{b}[m]\in\mathcal{A}^T}{\operatorname{argmin}}\{\|\boldsymbol{z}[m] - \boldsymbol{R}\boldsymbol{F}[m]\boldsymbol{b}[m]\|^2\}\,. \tag{7.22}$$

To make use of the block diagonal structure of $\boldsymbol{F}[m]$, the matrix $\boldsymbol{R}$ is decomposed into blocks of size $D \times D$

$$\boldsymbol{R} = \begin{bmatrix} \boldsymbol{\Delta}_{1,1} & \boldsymbol{\Delta}_{1,2} & \cdots & \boldsymbol{\Delta}_{1,T} \\ \boldsymbol{0} & \boldsymbol{\Delta}_{2,2} & \cdots & \boldsymbol{\Delta}_{2,T} \\ \vdots & \ddots & \ddots & \vdots \\ \boldsymbol{0} & \cdots & \boldsymbol{0} & \boldsymbol{\Delta}_{T,T} \end{bmatrix}, \tag{7.23}$$

where the matrices $\boldsymbol{\Delta}_{t,t} \in \mathbb{C}^{D\times D}$ for $t \in \{1,\ldots,T\}$ in the diagonal are upper triangular. The sphere constraint (7.11) becomes $\|\boldsymbol{\epsilon}[m]\|^2 < \rho^2$, where the error reads

$$\boldsymbol{\epsilon}[m] = \boldsymbol{z}[m] - \boldsymbol{R}\boldsymbol{F}[m]\boldsymbol{b}[m]\,. \tag{7.24}$$

Here we need to redefine the partial vectors, matrices and squared distance, similarly as in Section 6.2 but in a block manner, for $t \in \{1,\ldots,T\}$.

- Given $\boldsymbol{z}[m] \in \mathbb{C}^{DT}$ and $\boldsymbol{\epsilon}[m] \in \mathbb{C}^{DT}$, the **partial vectors**

$$\begin{aligned} \hat{\boldsymbol{z}}^{(t)}[m] &= [z_{D(t-1)+1}[m], \ldots, z_{Dt}[m]] \in \mathbb{C}^{D}, \\ \hat{\boldsymbol{\epsilon}}^{(t)}[m] &= [\epsilon_{D(t-1)+1}[m], \ldots, \epsilon_{Dt}[m]] \in \mathbb{C}^{D} \end{aligned} \tag{7.25}$$

  and

$$\begin{aligned} \boldsymbol{z}^{(t)}[m] &= [z_{D(t-1)+1}[m], \ldots, z_{DT}[m]]^{\mathrm{T}} \in \mathbb{C}^{D(T-t+1)}, \\ \boldsymbol{\epsilon}^{(t)}[m] &= [\epsilon_{D(t-1)+1}[m], \ldots, \epsilon_{DT}[m]]^{\mathrm{T}} \in \mathbb{C}^{D(T-t+1)} \end{aligned} \tag{7.26}$$

  are defined block-wise such that

$$\begin{aligned} \boldsymbol{z}^{(t)}[m] &= [\hat{\boldsymbol{z}}^{(t)}[m], \ldots, \hat{\boldsymbol{z}}^{(T)}[m]]^{\mathrm{T}} \quad \text{and} \\ \boldsymbol{\epsilon}^{(t)}[m] &= [\hat{\boldsymbol{\epsilon}}^{(t)}[m], \ldots, \hat{\boldsymbol{\epsilon}}^{(T)}[m]]^{\mathrm{T}}. \end{aligned} \tag{7.27}$$

  We also define the partial vector

$$\boldsymbol{b}^{(t)}[m] = [b_t[m], \ldots, b_T[m]]^{\mathrm{T}}. \tag{7.28}$$

- The block-wise **partial matrices**

$$\boldsymbol{R}^{(t)} = \begin{bmatrix} \boldsymbol{\Delta}_{t,t} & \cdots & \boldsymbol{\Delta}_{t,T} \\ \mathbf{0} & \ddots & \vdots \\ \mathbf{0} & \mathbf{0} & \boldsymbol{\Delta}_{T,T} \end{bmatrix} \quad \text{and} \quad \boldsymbol{F}^{(t)}[m] = \begin{bmatrix} \boldsymbol{f}[m] & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \ddots & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \boldsymbol{f}[m] \end{bmatrix} \tag{7.29}$$

  contain the last $(T-t+1)$ blocks of $\boldsymbol{R}$ and $\boldsymbol{F}[m]$, respectively.

- We also denote the first row of $\boldsymbol{R}^{(t)}$ by

$$\hat{\boldsymbol{\Delta}}^{(t)} = \left[\boldsymbol{\Delta}_{t,t}, \cdots, \boldsymbol{\Delta}_{t,T}\right] \in \mathbb{C}^{D(T-t+1)}. \tag{7.30}$$

- Finally, the **partial squared distance** d_t^2 at time m

$$\begin{aligned} d_t^2[m] &= \|\boldsymbol{\epsilon}^{(t)}[m]\|^2 \\ &= \|\boldsymbol{z}^{(t)}[m] - \boldsymbol{R}^{(t)}\boldsymbol{F}^{(t)}[m]\boldsymbol{b}^{(t)}[m]\|^2 \\ &= \sum_{i=t}^{T} \|\hat{\boldsymbol{z}}^{(i)}[m] - \boldsymbol{\Delta}^{(i)}\boldsymbol{F}^{(i)}[m]\boldsymbol{b}^{(i)}[m]\|^2 \end{aligned} \tag{7.31}$$

verifies the following recursive expression

$$\begin{aligned} d_t^2[m] &= d_{t+1}[m]^2 + \|\hat{\boldsymbol{z}}^{(t)}[m] - \boldsymbol{\Delta}^{(t)}\boldsymbol{F}^{(t)}[m]\boldsymbol{b}^{(t)}[m]\|^2 \\ &= d_{t+1}[m]^2 + \|\hat{\boldsymbol{\epsilon}}^{(t)}[m]\|^2 \,. \end{aligned} \tag{7.32}$$

For a better understanding of the notations and the partial vectors and matrices, the underlying partitioning is visualized in Table 7.1.

Similarly as for the conventional sphere decoder as described in Section 6.2, the squared norm of the error $\boldsymbol{\epsilon}$ can be computed for a given $\boldsymbol{b}$, starting with

$$d_T^2[m] = \|\hat{\boldsymbol{z}}^{(T)}[m] - \boldsymbol{\Delta}_{T,T}\boldsymbol{f}[m]b_T[m]\|^2 \tag{7.33}$$

and computing $d_1[m]^2$ in a recursive way using (7.32).

By writing down the T steps of the reduced-rank sphere decoder, we notice they are the same as for a conventional sphere decoder with new definitions of the partial vectors, matrices and distances. The algorithm is thus identical to the one in Algorithm 6.1, except that we use the new recursive relation (7.32) instead of (6.14) at line four. Furthermore, the tree search is again identical, however computing the partial squared distances using (7.32) instead of (6.14) at each node.

We presented here how to use the channel basis expansion model to develop a reduced-rank sphere decoder. Mathematically, conventional and reduced rank sphere decoders give the same solution. Thus, the performance of the iterative receiver using one or the other will be identical. The main difference resides in the complexity of the sphere decoder itself, if implemented in a conventional way (one QR factorization and one tree search are both performed per OFDM symbol) or using the reduced-rank channel model (one pre-processed QR factorization, followed by tree search for each symbol). Furthermore, wherever a sphere decoder might be used, the reduced-rank implementation can also be considered. As an example, in the following section we derive expressions of log-likelihood ratios, that can be computed using conventional or reduced-rank sphere decoding.

Table 7.1 Notation for the Reduced-Rank Sphere Decoder

$$\|\boldsymbol{\epsilon}[m]\|^2 = \left\| \begin{bmatrix} z_1[m] \\ \vdots \\ z_{D(t-1)+1}[m] \\ \vdots \\ z_{DT}[m] \end{bmatrix} - \begin{bmatrix} \boldsymbol{\Delta}_{1,1} & \boldsymbol{\Delta}_{1,2} & \cdots & \cdots & \boldsymbol{\Delta}_{1,T} \\ \ddots & \ddots & \ddots & & \vdots \\ \mathbf{0} & \ddots & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & \boldsymbol{\Delta}_{t,T} \\ \mathbf{0} & \cdots & \cdots & \mathbf{0} & \boldsymbol{\Delta}_{T,T} \end{bmatrix} \begin{bmatrix} \boldsymbol{f}[m] & \mathbf{0} & \cdots & \cdots & \mathbf{0} \\ \ddots & \ddots & \ddots & & \vdots \\ \mathbf{0} & \ddots & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & \mathbf{0} \\ \mathbf{0} & \cdots & \cdots & \mathbf{0} & \boldsymbol{f}[m] \end{bmatrix} \begin{bmatrix} b_1[m] \\ \vdots \\ b_t[m] \\ \vdots \\ b_T[m] \end{bmatrix} \right\|^2$$

$\boldsymbol{z}[m]$ $\boldsymbol{R}$ $\boldsymbol{F}[m]$ $\boldsymbol{b}[m]$

$$d_t^2[m] = \left\| \begin{bmatrix} z_{D(t-1)+1}[m] \\ \vdots \\ z_{Dt}[m] \\ \vdots \\ z_{DT}[m] \end{bmatrix} - \begin{bmatrix} \boldsymbol{\Delta}_{t,t} & \cdots & \boldsymbol{\Delta}_{t,T} \\ \ddots & \ddots & \vdots \\ \mathbf{0} & \cdots & \boldsymbol{\Delta}_{T,T} \end{bmatrix} \begin{bmatrix} \boldsymbol{f}[m] & \cdots & \mathbf{0} \\ \ddots & \ddots & \vdots \\ \mathbf{0} & \cdots & \boldsymbol{f}[m] \end{bmatrix} \begin{bmatrix} b_t[m] \\ \vdots \\ b_T[m] \end{bmatrix} \right\|^2$$

$\boldsymbol{z}^{(t)}[m]$ $\boldsymbol{R}^{(t)}$ $\boldsymbol{F}^{(t)}[m]$ $\boldsymbol{b}^{(t)}[m]$

$$d_t^2[m] = d_{t+1}[m]^2 + \left\| \begin{bmatrix} z_{D(t-1)+1}[m] \\ \vdots \\ z_{Dt}[m] \end{bmatrix} - \begin{bmatrix} \boldsymbol{\Delta}_{t,t} & \cdots & \boldsymbol{\Delta}_{t,T} \end{bmatrix} \begin{bmatrix} \boldsymbol{f}[m] & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \ddots & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \boldsymbol{f}[m] \end{bmatrix} \begin{bmatrix} b_t[m] \\ \vdots \\ b_T[m] \end{bmatrix} \right\|^2$$

$\hat{\boldsymbol{z}}^{(t)}[m]$ $\hat{\boldsymbol{\Delta}}^{(t)}$ $\boldsymbol{F}^{(t)}[m]$ $\boldsymbol{b}^{(t)}[m]$

7.3 Log-Likelihood Ratios using Rank-Reduction Sphere Decoder

As will be discussed in Chapter 8, sphere decoding providing hard outputs as described so far, has indeed low-complexity but reduced performance compared to LMMSE detection. This is due to the fact that for complexity reasons, sphere decoding is performed on a per user basis, thus assuming perfect interference cancelation, whereas LMMSE detection considers remaining interference. Another source of performance degradation are the hard inputs fed to the BCJR decoder by the sphere decoder, while the LMMSE detector provides soft inputs.

In this section we aim at computing log-likelihood ratios, making use of sphere decoding.

7.3.1 From Imaginary to Real Space

So far all quantities involved have been complex valued. The transmitted symbols $\boldsymbol{b}$ stem from a QPSK symbol alphabet

$$\mathcal{A} = \{\frac{\pm 1 \pm \mathrm{j}}{\sqrt{2}}\}\,. \tag{7.34}$$

The computation of log-likelihood ratios using the sphere decoder is somewhat simpler to perform in the real domain.

In order to convert the signal model into the real domain, we use the superscripts $(\mathbf{r})$ and $(\mathbf{i})$, respectively denoting the real and imaginary part of a complex vector or matrix. Furthermore, we scale the input vector such that we consider entries for $\hat{\boldsymbol{b}}$ in $\{\pm 1\}$ only. Using the notation from Section 7.2, we define the real-valued quantities

$$\hat{\boldsymbol{b}}[m] = \sqrt{2}\begin{bmatrix} \boldsymbol{b}^{(\mathbf{r})}[m] \\ \boldsymbol{b}^{(\mathbf{i})}[m] \end{bmatrix}, \quad \hat{\boldsymbol{y}}[m] = \begin{bmatrix} \tilde{\boldsymbol{y}}^{(\mathbf{r})}[m] \\ \tilde{\boldsymbol{y}}^{(\mathbf{i})}[m] \end{bmatrix}, \quad \hat{\boldsymbol{n}}[m] = \begin{bmatrix} \boldsymbol{n}^{(\mathbf{r})}[m] \\ \boldsymbol{n}^{(\mathbf{i})}[m] \end{bmatrix}, \tag{7.35}$$

$$\hat{\boldsymbol{\Gamma}} = \begin{bmatrix} \boldsymbol{\Gamma}^{(\mathbf{r})} & -\boldsymbol{\Gamma}^{(\mathbf{i})} \\ \boldsymbol{\Gamma}^{(\mathbf{i})} & \boldsymbol{\Gamma}^{(\mathbf{r})} \end{bmatrix} \quad \text{and} \quad \hat{\boldsymbol{F}}[m] = \begin{bmatrix} \boldsymbol{F}[m] & \mathbf{0} \\ \mathbf{0} & \boldsymbol{F}[m] \end{bmatrix}. \tag{7.36}$$

For notational convenience, we drop the symbol $\hat{\cdot}$, in the remainder of this Section. It is important to keep in mind the fact that now we work in the real domain and only the matrix $\boldsymbol{\Gamma}$ (and its QR-factors $\boldsymbol{Q}$ and $\boldsymbol{R}$) are time-invariant. The received signal is given as before by

$$\boldsymbol{y}[m] = \frac{1}{\sqrt{2}}\boldsymbol{\Gamma}\boldsymbol{F}[m]\boldsymbol{b}[m] + \boldsymbol{n}[m]\,. \tag{7.37}$$

Let

$$\boldsymbol{\Gamma} = \boldsymbol{Q}\boldsymbol{R} \tag{7.38}$$

be the QR factorization of the $\boldsymbol{\Gamma}$. After multiplication with $\sqrt{2}\boldsymbol{Q}^{\mathrm{H}}$, (7.37) becomes

$$\boldsymbol{z}[m] = \sqrt{2}\boldsymbol{Q}^{\mathrm{H}}\boldsymbol{y}[m] = \boldsymbol{R}\boldsymbol{F}[m]\boldsymbol{b}[m] + \sqrt{2}\boldsymbol{Q}^{\mathrm{H}}\boldsymbol{n}[m]\,. \tag{7.39}$$

The authors in [WG04] present a method to compute the log-likelihood ratios using sphere decoding. We recall this method in the following.

7.3.2 Log-Likelihood Ratios

Let us define the *a priori* LLR

$$\lambda_{\mathrm{PRIOR}}(b_t) = \ln\left(\frac{p(b_t = +1)}{p(b_t = -1)}\right)\,, \tag{7.40}$$

the *a posteriori* LLR

$$\lambda_{\mathrm{POST}}(b_t) = \ln\left(\frac{p(b_t = +1|\boldsymbol{z})}{p(b_t = -1|\boldsymbol{z})}\right)\,, \tag{7.41}$$

and the *extrinsic* LLR

$$\lambda_{\mathrm{EXT}}(b_t) = \lambda_{\mathrm{POST}}(b_t) - \lambda_{\mathrm{PRIOR}}(b_t)\,, \tag{7.42}$$

for b_t, $t \in \{1, \ldots, 2T\}$.

Let $\mathbb{B}_t^+$ and $\mathbb{B}_t^-$ be the subsets of $\mathbb{B} = \{-1, +1\}^{2T}$ such that the t-th element $b_t = +1$ or $b_t = -1$, respectively. After computations and simplifications as given in Appendix C, the *a posteriori* LLR $\lambda_{\mathrm{POST}}(b_t)$ can be written as

$$\begin{aligned}\lambda_{\mathrm{POST}}(b_t) \approx \; & \max_{\boldsymbol{b}\in\mathbb{B}_t^+}\left\{-\tfrac{1}{\sigma^2}\|\boldsymbol{z} - \boldsymbol{R}\boldsymbol{F}\boldsymbol{b}\|^2 + \frac{\boldsymbol{b}^{\mathrm{T}}\boldsymbol{\lambda}_{\mathrm{PRIOR}}}{2}\right\} \\ & - \max_{\boldsymbol{b}\in\mathbb{B}_t^-}\left\{-\tfrac{1}{\sigma^2}\|\boldsymbol{z} - \boldsymbol{R}\boldsymbol{F}\boldsymbol{b}\|^2 + \frac{\boldsymbol{b}^{\mathrm{T}}\boldsymbol{\lambda}_{\mathrm{PRIOR}}}{2}\right\}\,.\end{aligned} \tag{7.43}$$

We denote by $\mathcal{E}(\boldsymbol{b})$ the expression in curly braces in (7.43), i.e. up to the sign

$$\mathcal{E}(\boldsymbol{b}) \triangleq \frac{1}{\sigma^2}\|\boldsymbol{z} - \boldsymbol{R}\boldsymbol{F}\boldsymbol{b}\|^2 - \frac{\boldsymbol{b}^{\mathrm{T}}\boldsymbol{\lambda}_{\mathrm{PRIOR}}}{2}\,, \tag{7.44}$$

such that

$$\lambda_{\mathrm{POST}}(b_t) \approx \min_{\boldsymbol{b}\in\mathbb{B}_t^-}\{\mathcal{E}(\boldsymbol{b})\} - \min_{\boldsymbol{b}\in\mathbb{B}_t^+}\{\mathcal{E}(\boldsymbol{b})\} \tag{7.45}$$

needs to minimize $\mathcal{E}$ independently over the two subsets $\mathbb{B}_t^+$ and $\mathbb{B}_t^-$.

Let us find a vector $\boldsymbol{\Lambda}$ such that

$$4(\boldsymbol{RF})^{\mathrm{T}}\boldsymbol{\Lambda} = \sigma^2 \boldsymbol{\lambda}_{\mathrm{PRIOR}}\,. \tag{7.46}$$

For instance, let

$$\boldsymbol{\Lambda} = \frac{\sigma^2}{4u_{D-1}}\Big(\boldsymbol{R}^{\mathrm{T}}\Big)^{-1}\boldsymbol{q}\,, \tag{7.47}$$

where

$$\boldsymbol{q} = [\mathbf{0}_{D-1}^{\mathrm{T}}, \lambda_{\mathrm{PRIOR}}(1), \ldots, \mathbf{0}_{D-1}^{\mathrm{T}}, \lambda_{\mathrm{PRIOR}}(2T)]^{\mathrm{T}} \in \mathbb{R}^{2TD}\,. \tag{7.48}$$

Vector $\boldsymbol{q}$ satisfies

$$\boldsymbol{F}^{\mathrm{T}}\boldsymbol{q} = \begin{bmatrix} \boldsymbol{f}^{\mathrm{T}} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \ddots & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \boldsymbol{f}^{\mathrm{T}} \end{bmatrix} \begin{bmatrix} \mathbf{0}_{D-1} \\ \lambda_{\mathrm{PRIOR}}(1) \\ \vdots \\ \mathbf{0}_{D-1} \\ \lambda_{\mathrm{PRIOR}}(2T) \end{bmatrix} = u_{D-1}\boldsymbol{\lambda}_{\mathrm{PRIOR}}\,, \tag{7.49}$$

yielding (7.46).

This allows writing

$$\boldsymbol{b}^{\mathrm{T}}\boldsymbol{\lambda}_{\mathrm{PRIOR}} = \frac{4}{\sigma^2}(\boldsymbol{RFb})^{\mathrm{T}}\boldsymbol{\Lambda} \tag{7.50}$$

and the expression $\mathcal{E}(\boldsymbol{b})$ in (7.44) becomes

$$\mathcal{E}(\boldsymbol{b}) = \frac{1}{\sigma^2}\|\boldsymbol{z} + \boldsymbol{\Lambda} - \boldsymbol{RFb}\|^2 - \mathrm{c}\,, \tag{7.51}$$

where

$$\mathrm{c} = \frac{2\boldsymbol{\Lambda}^{\mathrm{T}}\boldsymbol{z} + \|\boldsymbol{\Lambda}\|^2}{\sigma^2} \tag{7.52}$$

is a constant independent of $\boldsymbol{b}$, that disappears in the final expression

$$\lambda_{\mathrm{POST}}(b_t) \approx \min_{\boldsymbol{b}\in\mathbb{B}_t^-}\left\{\frac{1}{\sigma^2}\|\boldsymbol{z} + \boldsymbol{\Lambda} - \boldsymbol{RFb}\|^2\right\} - \min_{\boldsymbol{b}\in\mathbb{B}_t^+}\left\{\frac{1}{\sigma^2}\|\boldsymbol{z} + \boldsymbol{\Lambda} - \boldsymbol{RFb}\|^2\right\}\,. \tag{7.53}$$

7.3.3 Sphere Decoder for LLRs

Recalling the expression

$$\mathcal{E}(\boldsymbol{b}) = \frac{1}{\sigma^2}\|\boldsymbol{z} + \boldsymbol{\Lambda} - \boldsymbol{RFb}\|^2 \tag{7.54}$$

which needs to be minimized over the two subsets $\mathbb{B}_t^+$ and $\mathbb{B}_t^-$ to compute the *a posteriori* LLR (7.53)

$$\lambda_{\mathrm{POST}}(b_t) \approx \min_{\boldsymbol{b}\in\mathbb{B}_t^-} \{\mathcal{E}(\boldsymbol{b})\} - \min_{\boldsymbol{b}\in\mathbb{B}_t^+} \{\mathcal{E}(\boldsymbol{b})\} \;, \tag{7.55}$$

it is straightforward to notice that computing $\lambda_{\mathrm{POST}}(b_t)$ requires two solutions of the ML equation $\operatorname{argmin}\{\mathcal{E}(\boldsymbol{b})\}$ over the two subsets $\mathbb{B}_t^-$ and $\mathbb{B}_t^+$, respectively. Let us define

$$\boldsymbol{b}^* = \operatorname*{argmin}_{\boldsymbol{b}\in\mathbb{B}} \{\mathcal{E}(\boldsymbol{b})\} \tag{7.56}$$

the ML solution over the whole alphabet $\mathbb{B}$. For $t \in \{1,\ldots,2T\}$, we denote by s_t the sign of $b^*(t)$. Obviously, the set $\mathbb{B}_t^{s_t}$ is per definition

$$\mathbb{B}_t^{s_t} \triangleq \{\boldsymbol{b}\in\mathbb{B} \mid b_t = b_t^*\} \;, \tag{7.57}$$

yielding

$$\boldsymbol{b}^* \in \mathbb{B}_t^{s_t} \quad \text{for} \quad t \in \{1,\ldots,2T\}\,. \tag{7.58}$$

Hence, the ML solution $\boldsymbol{b}^*$, minimizing $\mathcal{E}(\boldsymbol{b})$ over $\mathbb{B}$, is also the solution minimizing $\mathcal{E}(\boldsymbol{b})$ over $\mathbb{B}_t^{s_t}$

$$\boldsymbol{b}^* = \operatorname*{argmin}_{\boldsymbol{b}\in\mathbb{B}_t^{s_t}} \{\mathcal{E}(\boldsymbol{b})\} \quad \text{for} \quad t \in \{1,\ldots,2T\}\,, \tag{7.59}$$

and can be found using sphere decoding. Given $\boldsymbol{b}^*$, Equation (7.55) rewrites

$$\lambda_{\mathrm{POST}}(b_t) \approx -b_t^*\mathcal{E}(\boldsymbol{b}^*) + b_t^* \min_{\boldsymbol{b}\in\mathbb{B}_t^{-s_t}} \{\mathcal{E}(\boldsymbol{b})\} \;. \tag{7.60}$$

For $t \in \{1,\ldots,2T\}$, we finally define

$$\boldsymbol{b}_{(t)}^* = \operatorname*{argmin}_{\boldsymbol{b}\in\mathbb{B}_t^{-s_t}} \{\mathcal{E}(\boldsymbol{b})\} \tag{7.61}$$

the second solution needed to compute the *a posteriori* LLR, i.e. the solution over the set $\mathbb{B}_t^{-s_t}$. Note that $\boldsymbol{b}_{(t)}^*$ is t-dependent and can also be computed using sphere decoding over the subset $\mathbb{B}_t^{-s_t}$.

Using this notation, the *a posteriori* LLR of b_t (7.53) finally becomes

$$\lambda_{\mathrm{POST}}(b_t) \approx -\frac{b_t^*}{\sigma^2}\mathcal{E}(\boldsymbol{b}^*) + \frac{b_t^*}{\sigma^2}\mathcal{E}(\boldsymbol{b}_{(t)}^*)\,. \tag{7.62}$$

To summarize, computing log-likelihood ratios using sphere decoding means finding $2T+1$ ML solutions, denoted through $\boldsymbol{b}^*$ and $\boldsymbol{b}_t^*$ for $t \in \{1,\ldots,2T\}$. These $2T+1$ solutions need to be computed for every channel realization, and a reduced-rank sphere

decoder may be used in order to reduce complexity. A possibility to further reduce complexity is to perform single tree search [SBB08], i.e. all the $2T + 1$ solutions are searched during the same tree search. However, we do not consider this option in this work.

The use of the max-log approximation (see Proof 7.1) and a sphere decoder allow computing log-likelihood ratios with low complexity. Obviously the overall complexity will be higher than for the hard sphere decoder since more ML solutions have to be computed. Nevertheless, a considerable gain in term of performance is expected.

Chapter 8
Performance and Complexity

This chapter presents a comparison of the various possibilities to incorporate a sphere decoder in the iterative receiver described in Chapter 2. We proceed in two steps. First we compare the performance of hard and soft-output sphere decoding implemented in the iterative receiver, in terms of bit error rate (BER). Then we consider the computational complexity in three cases: hard-output detection using either the conventional implementation of a sphere decoder, or using the reduced-rank sphere decoder and a soft-output detection using the reduced-rank sphere decoder. For more relevance, we also compare these schemes to the same iterative receiver using an LMMSE detector, as described in Chapter 2.

8.1 Simulation Setup

The setup used in the simulations is as described in Section 4.1. For practical implementation of the sphere decoder, we consider a conventional Fincke-Pohst strategy. The nodes are ordered in a universal way from the QPSK constellation, i.e. the ordering is random but constant during the detection process.

For definition of the radius, we use the zero-forcing solution as defined in (6.5)

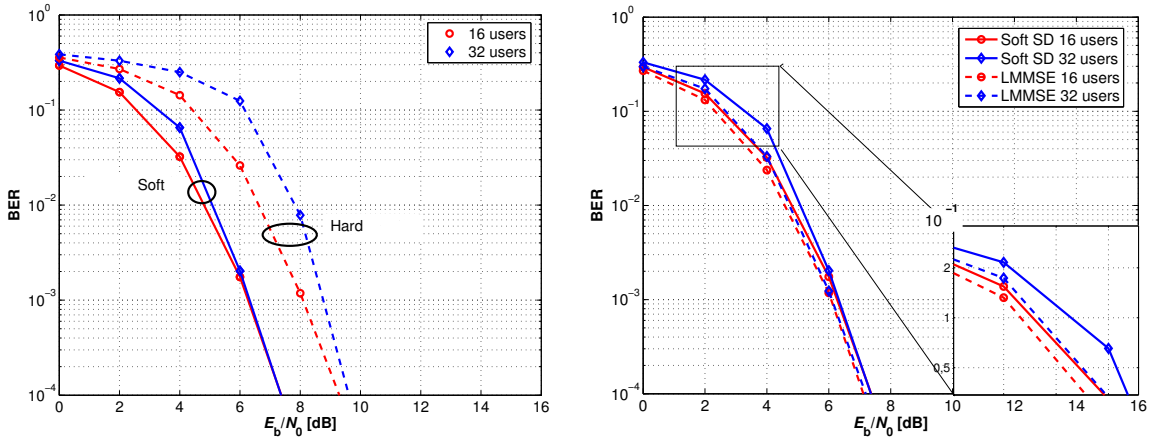
$$\boldsymbol{b}_{\mathrm{ZF}} = \left(\boldsymbol{H}^{\mathrm{H}}\boldsymbol{H}\right)^{-1}\boldsymbol{H}^{\mathrm{H}}\boldsymbol{y} \tag{8.1}$$

and

$$\rho^2 = \|\boldsymbol{y} - \boldsymbol{b}_{\mathrm{ZF}}\|^2 . \tag{8.2}$$

This way the radius is computed for every new search.

For efficient comparison we need to recall that in all cases parallel interference cancelation is performed. Sphere decoding is performed on a per user basis (assuming all other users have been perfectly canceled) while LMMSE detection takes into account remaining interference.



(a) (b)

Figure 8.1: BER versus SNR with perfect channel knowledge. After 4 receiver iterations and for $K \in \{16, 32\}$ users, we compare (a) hard and soft sphere decoder; (b) soft sphere decoder and LMMSE.

8.2 Performance of the Sphere Decoders

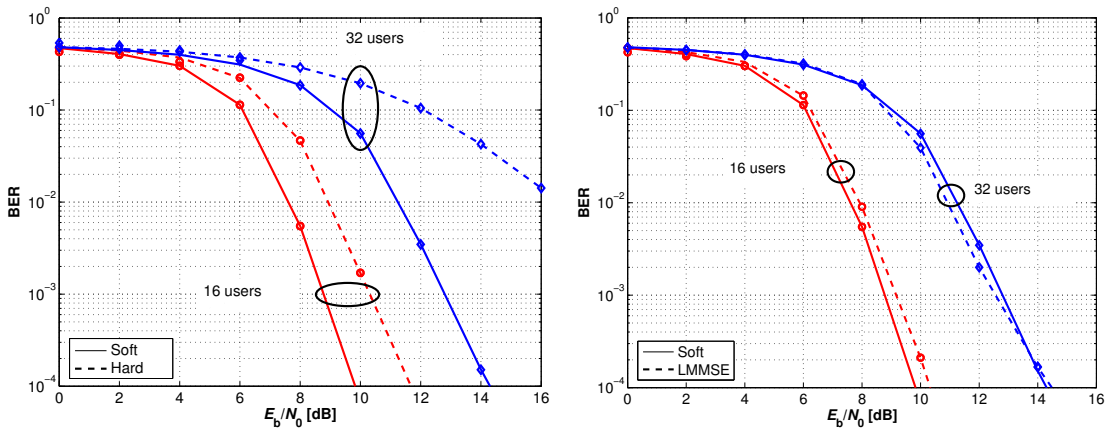
In this section we show a performance comparison of the iterative receiver described in Section 2.4 using either a hard or soft-output sphere decoder or an LMMSE filter for multi-user detection. For clarity of the pictures we proceed in two steps: firstly, hard and soft-output sphere decoding are compared, secondly soft-output sphere decoding is compared to LMMSE detection. The BER versus the signal to noise ratio E_b/N_0 is shown for $K = 16$ and $K = 32$ users.

In Figure 8.1 we show the performance of the receiver when the channel is assumed perfectly known. Similar results for estimated channel are given in Figure 8.2 (the channel is estimated using LMMSE filtering as in Section 2.6).

In both figures, the solid lines always represent the soft-output sphere decoder while the dashed lines show either the hard sphere decoder (Figure 8.1.a and 8.2.a) or the receiver using LMMSE detection (Figure 8.1.b and and 8.2.b).

From these figures, the following conclusions can be drawn:

- Channel estimation results in a loss of at least 2dB, no matter which detection method is used.
- The gap between hard and soft sphere decoding increases with imperfect channel knowledge, although LMMSE and soft sphere decoding remain very close. Hence,



(a) (b)

Figure 8.2: BER versus SNR with estimated channel. After 4 receiver iterations and for $K \in \{16, 32\}$ users, we compare (a) hard and soft sphere decoder; (b) soft sphere decoder and LMMSE.

the hard sphere decoder is more sensitive to channel errors than soft sphere decoding or LMMSE detection.

- In case of a perfectly known channel, using a sphere decoder for computing soft information allows a gain of up to 3dB over the hard-output sphere decoder.

- When the channel is not known but estimated, the use of soft outputs becomes necessary when the system load increases (i.e. more users are decoded). For $K = 32$ users, the hard-output sphere decoder is not able to provide sustainable performance.

- The soft-output sphere decoder reaches close to LMMSE performance. In Figures 8.1.b and 8.2.b, we can notice a very slight loss at low SNR using soft-output sphere decoding. At higher SNR, this loss tends to disappear and with channel estimation, the soft-output sphere decoders can outperform the LMMSE receiver.

In Figure 8.3 we compare the performance of the LMMSE detector and the soft sphere decoder for different number of iterations at the receiver. On the same figure we show the case where the channel is perfectly known and when it is estimated. We observe that the receiver using LMMSE detectors converges faster and has in general better performance than when soft sphere decoding is used. Nevertheless, as seen in Figures 8.1.b and 8.2.b, after 4 iterations they show very similar performance. Hence, we see

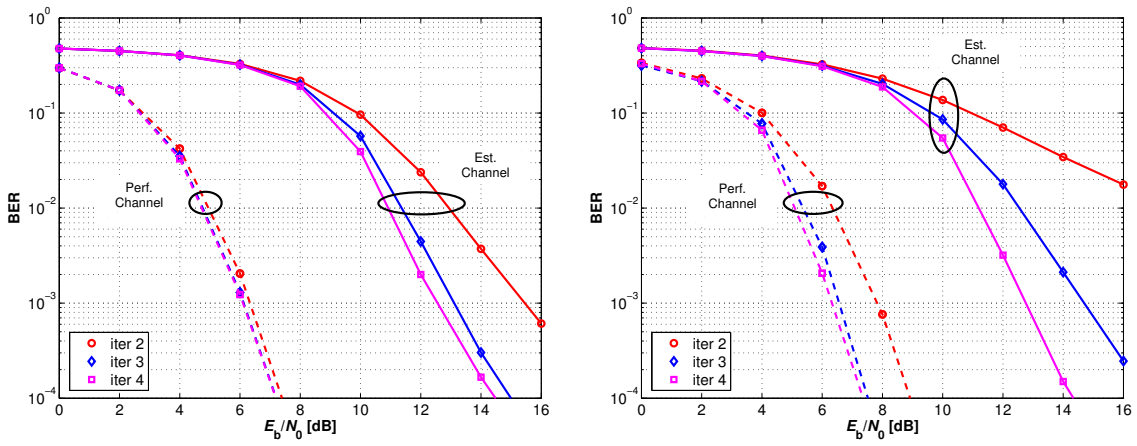


Figure 8.3: BER vs SNR for receiver iterations 2 to 4. We show the performance for perfect channel knowledge ('Perf. Channel') or estimated channel ('Est. Channel') using (a) an LMMSE detector and (b) a soft sphere decoder.

that increasing the number of iterations yields more improvement when using soft sphere decoding.

To summarize, one can say that soft-output sphere decoding performs as well as LMMSE detection with four iteration of the receiver, and it profits more from an increasing number of receiver iteration.

8.3 Computational Complexity Expressions

We recall that one complex multiplication (CM) requires 6 *flops*, while one complex addition (CA) requires 2 *flops*.

From Chapter 7, we have q_t denoting the number of candidates in the set $\mathcal{C}_t$ after step t. Both q_t and $\mathcal{C}_t$ are random variables, since they depend on the current realization of the symbols, channel and noise. Q is the size of the alphabet $\mathcal{A}$.

8.3.1 QR factorization

Efficient implementation of a sphere decoder, either in the conventional or the reduced-rank version, requires a QR factorization of the effective spreading sequence matrix. We extend the result for the complexity of the QR factorization given in [GL96] for real matrices to the complex case. Except for computing soft outputs, we consider complex

matrices. The number of multiplications and additions in the computation is about the same order, thus, we multiply the complexity for real matrices with 8 (as one CM and one CA) to obtain the complexity for complex matrices.

As a very first step we discuss the need of QR factorization by comparing exhaustive search. Indeed, the QR factorization usually requires high computational complexity. Thus, it is not straightforward that performing QR factorization before exhaustive search is of benefit. We compare the computational complexity for exhaustive search (i.e. for computing $\boldsymbol{Hb}$ for all $\boldsymbol{b}$ in the alphabet), either using the full matrix $\boldsymbol{H}$ or its triangulated matrix $\boldsymbol{R}$ obtained by a QR factorization.

- For exhaustive search (denoted by 'ES') without QR factorization, $\boldsymbol{H}$ is a matrix of size $NR \times T$. Thus, for all possible vectors $\boldsymbol{b} \in \mathcal{A}^T$, the computation of $\boldsymbol{Hb}$ requires NRT CM and $NR(T-1)$ CA, leading to

$$\mathcal{C}_{\mathrm{ES}}^{\mathrm{noQR}} = 2Q^T NR(4T-1)\, flops\,. \tag{8.3}$$

- Using exhaustive search with QR factorization, we need to perform one QR factorization with complexity of

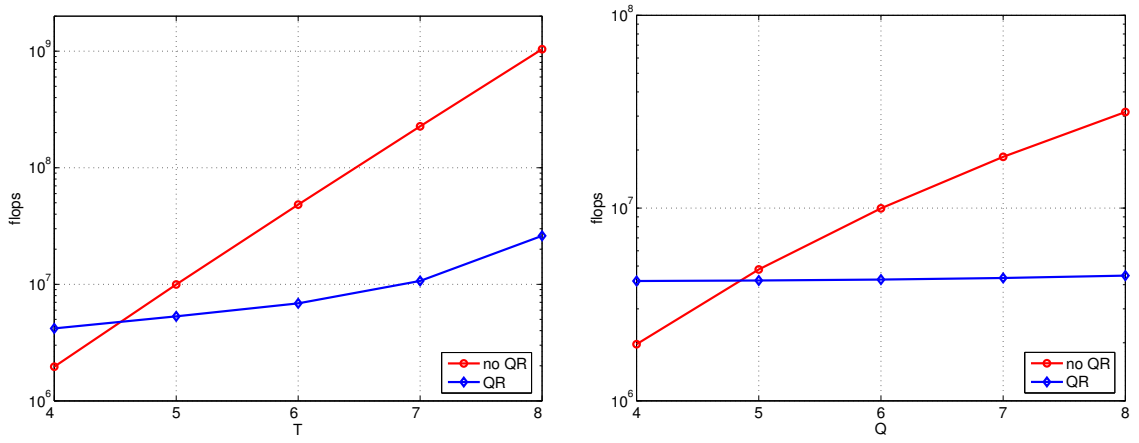
$$8T\left(4(NR)^2 - 2NRT + \frac{2T^2}{3}\right)\, flops\,. \tag{8.4}$$

  Then, $\boldsymbol{R}$ is triangular and has size $T \times T$ and for all possible vectors $\boldsymbol{b} \in \mathcal{A}^T$, the computation of $\boldsymbol{Rb}$ requires $\frac{T(T+1)}{2}$ CM and $\frac{T(T-1)}{2}$ CA, yielding

$$\mathcal{C}_{\mathrm{ES}}^{\mathrm{QR}} = 4T\left(4(NR)^2 - 2NRT + \frac{2T^2}{3}\right) + 2T(2T+1)Q^T\, flops\,. \tag{8.5}$$

Notation 'QR' and 'noQR' denote exhaustive search using first a QR factorization or not, with complexity given by $\mathcal{C}_{\mathrm{ES}}^{\mathrm{QR}}$ and $\mathcal{C}_{\mathrm{ES}}^{\mathrm{noQR}}$, respectively. Figure 8.4 shows the computational complexity of exhaustive search with varying number of transmit antennas T in Figure 8.4.a or varying alphabet size Q in Figure 8.4.b.

As either T or Q is increasing, it becomes necessary to perform QR factorization to save complexity. This is not the case for the parameters used in our simulations ($Q = T = 4$), though we assume more complex systems in general and thus always perform QR factorization first. Furthermore, only the triangularity of the channel allows efficient implementation of a sphere decoder as described so far. In the following, we work with triangular channels for sphere decoding.



(a) (b)

Figure 8.4: Complexity of exhaustive search. Comparison of the complexity using first QR factorization or not, with $Q = 4$ and varying number of transmit antennas T (a) or $T = 4$ and varying alphabet size Q (b). Other parameters are $R = 4$, $K = 32$ and $N = 64$.

8.3.2 Conventional Sphere Decoder

The algorithm for a conventional sphere decoder computing hard outputs (see Section 7.1) is given in Algorithm 8.1 with corresponding complexity.

With the conventional sphere decoder (SD), the following operations are needed per symbol $m \notin \mathcal{P}$:

- One *thin* QR factorization of size $NR \times T$, see Section 8.3.1, with complexity

$$c_{\mathrm{QR}} = 8T\left(2(NR)^2 - NRT + \frac{T^2}{3}\right) \quad \text{and} \tag{8.6}$$

- one call of the sphere decoder, with complexity according to Algorithm 8.1

$$c_{\mathrm{SD}}[m] = 6QT + 2\sum_{t=1}^{T-1}(4T - 4t + Q - 1)q_{t+1}[m]\,. \tag{8.7}$$

An upper bound of $c_{\mathrm{SD}}[m]$ can be computed using the worst-case scenario, i.e. $q_t[m] \leq Q^{(T-t+1)}$ for $t \in \{1, \ldots, T\}$, which corresponds to an exhaustive search using a sphere decoder implementation with an infinite radius. Observing that

$$\sum_{t=1}^{T-1}(Q-1)Q^{T-t} = Q^T - Q \tag{8.8}$$

	Perform	with complexity
1	Set $\mathcal{C}_{T+1} = \emptyset$ and $q_{T+1} = 0$	
2	for t from T to 1	
3	for all Q values of $b_t \in \mathcal{A}$	
4	compute $c(t) = R_{t,t} b_t$	$6Q$
5	for all q_{t+1} vectors $\boldsymbol{b}^{(t+1)} \in C_{t+1}$	
6	compute $\left[R_{t,t+1}, \ldots, R_{t,T}\right] \boldsymbol{b}^{(t+1)}$	$(8(T-t)-2)\, q_{t+1}$
7	and add it to all Q values $c(t)$	$2Qq_{t+1}$
8	end	

Algorithm 8.1: The conventional sphere decoder algorithm.

and

$$\sum_{t=1}^{T-1} (T-t) Q^{T-t} = \sum_{t=1}^{T-1} t Q^t \,, \tag{8.9}$$

we obtain using (8.7)

$$c_{\mathrm{SD}}[m] \leq 2Q(3T + Q^{T-1} - 1) + 8 \sum_{t=1}^{T-1} t Q^t \,. \tag{8.10}$$

Finally, the complete complexity of the conventional sphere decoder computing hard outputs for a data block of size $M' = M - J$ is

$$C_{\mathrm{SD}} = M' \cdot c_{\mathrm{QR}} + \sum_{m \notin \mathcal{P}} c_{\mathrm{CSD}}[m] \,, \tag{8.11}$$

and after approximations considering the parameters of the system, (8.11) can be upper bounded as follows

$$\boxed{C_{\mathrm{SD}} \leq 2M' \left(8T(NR)^2 - 4NRT^2 + \tfrac{4T^3}{3} + Q^T + 4 \sum_{t=1}^{T-1} t Q^t \right) .} \tag{8.12}$$

8.3.3 Reduced-Rank Sphere Decoder

We detail here the computations needed to obtain hard outputs using the reduced-rank sphere decoder (RR-SD), as presented in Section 7.2. Similarly as in the previous Section, we detail the computations and the step-by-step complexity in Algorithm 8.2.

The computational complexity for a block of length M' using the reduced-rank sphere decoding requires:

	Perform	with complexity
1	Set $\mathcal{C}_{T+1} = \emptyset$ and $q_{T+1} = 0$	
2	for t from T to 1	
3	for all Q values of $\boldsymbol{\beta} = \boldsymbol{f}[m]b_t$ with $b_t \in \mathcal{A}$	$6DQ$
4	compute $c(t) = \boldsymbol{\Delta}_{t,t}\boldsymbol{\beta}$	$2DQ(2D+1)$
5	for all q_{t+1} vectors $\boldsymbol{b}^{(t+1)} \in C_{t+1}$	
6	compute $\left[\boldsymbol{\Delta}_{t,t+1}, \ldots, \boldsymbol{\Delta}_{t,T}\right] \boldsymbol{F}^{(t+1)}[m]\boldsymbol{b}^{(t+1)}$	$8D^2(T-t)q_{t+1}$
7	and add it to all Q values $c(t)$	$2DQq_{t+1}$
8	end	

Algorithm 8.2: The reduced-rank sphere decoder algorithm.

- One *thin* complex QR factorization of size $NR \times DT$, with computational complexity according to Section 8.3.1,

$$c_{\mathrm{QR}} = 8DT\left(2(NR)^2 - NRDT + \frac{(DT)^2}{3}\right) \tag{8.13}$$

  valid for the whole block, and

- M' calls of the reduced-rank sphere decoder with complexity as in Algorithm 8.2

$$c_{\mathrm{RR-SD}}[m] = 4DTQ(D+2) + 2D\sum_{t=1}^{T}\left(4D(T-t)+Q\right)q_{t+1}[m]\,. \tag{8.14}$$

Using the worst-case $q_t[m] \leq Q^{(T-t+1)}$ for $t \in \{1, \ldots, T\}$ equivalent to exhaustive search, as well as (8.8) and (8.9) leads to the following upper bound for (8.14)

$$c_{\mathrm{RR-SD}}[m] \leq 4DTQ(D+2) + 2D\sum_{t=1}^{T} Q^t(4Dt+Q)\,. \tag{8.15}$$

The total computational complexity

$$C_{\mathrm{RR-SD}} = c_{\mathrm{QR}} + \sum_{m \notin \mathcal{P}} c_{\mathrm{SD}}[m]\,, \tag{8.16}$$

for a single data block can be upper bounded by

$$\boxed{\begin{aligned} C_{\mathrm{RR-SD}} \;\leq\; & 2DM'\left(2TQ(D+2) + \sum_{t=1}^{T} Q^t(4Dt+Q)\right) \\ & +8DT\left(2(NR)^2 - NRDT + \tfrac{(DT)^2}{3}\right). \end{aligned}} \tag{8.17}$$

8.3.4 Soft-Output Sphere Decoder

For soft-output sphere decoding, we consider the reduced-rank implementation only, as it will be shown later that it achieves considerable computational complexity reduction. The algorithm and the corresponding step-by-step complexity is detailed in Algorithm 8.3 below.

	Perform	with complexity
1	Set $\mathcal{C}_{2T+1} = \emptyset$ and $q_{2T+1} = 0$	
2	for t from $2T$ to 1	
3	for all $Q/2$ values of $\boldsymbol{\beta} = \boldsymbol{f}[m]b_t$ with $b_t \in \mathcal{A}'$	DQ
4	compute $c(t) = \boldsymbol{\Delta}_{t,t}\boldsymbol{\beta}$	$QD^2/2$
5	for all q_{t+1} vectors $\boldsymbol{b}^{(t+1)} \in C_{t+1}$	
6	compute $\left[\boldsymbol{\Delta}_{t,t+1}, \ldots, \boldsymbol{\Delta}_{t,2T}\right] \boldsymbol{F}^{(t+1)}[m]\boldsymbol{b}^{(t+1)}$	$2D^2(2T-t)q_{t+1}$
7	and add it to all $Q/2$ values $c(t)$	$DQq_{t+1}/2$
8	end	

Algorithm 8.3: The soft sphere decoder algorithm.

Hard and soft-output sphere decoders differ in two aspects. When computing soft outputs, we work in the real domain. Thus, matrices have higher dimensions. This does not necessarily mean higher complexity, since one computation is one $flop$ only. Furthermore, at each time instant, several calls of the sphere decoder are needed:

- One of size $2T$ for computing the solution of the ML equation (7.56),
- $(2T-1)$ of size $(2T-1)$ for computing the side vectors $\boldsymbol{b}_{\mathrm{SD},t}$ as defined in Section 7.3.2.

For simplicity we consider all sphere decoder searches over the full dimension $2T$. Taking these elements into account, the computations needed for soft sphere decoding are as follow:

- One *thin* **real** QR factorization of size $2NR \times 2DT$, with computational complexity

$$c_{\mathrm{QR}} = 16DT\left(2(NR)^2 - NRDT + \frac{(DT)^2}{3}\right), \tag{8.18}$$

- $2M'T$ calls of the reduced-rank sphere decoder with complexity according to Algorithm 8.3

$$c_{\text{soft}}[m] = (2T-1)DQ(1+\frac{D}{2}) + \sum_{t=1}^{2T-1}\left(2D^2(2T-t)+\frac{DQ}{2}\right)q_{t+1}[m]\,. \quad (8.19)$$

Again, using the worst case $q_t[m] \leq Q^{(2T-t+1)}$ for $t \in \{1,\ldots,2T\}$ equivalent to exhaustive search as well as (8.8) and (8.9) yields the following upper bound for (8.19)

$$c_{\text{soft}}[m] \leq (2T-1)DQ(1+\frac{D}{2}) + \frac{D}{2}\sum_{t=1}^{2T-1} Q^t(4Dt+Q)\,. \quad (8.20)$$

The total computational complexity is as in (8.16), and upper bounded by

$$\boxed{\begin{aligned} C_{\text{SSD}} \;\leq\; & 2M'T\left((2T-1)DQ(1+\tfrac{D}{2}) + \tfrac{D}{2}\sum_{t=1}^{2T-1} Q^t(4Dt+Q)\right) \\ & +16DT\left(2(NR)^2 - NRDT + \tfrac{(DT)^2}{3}\right)\,. \end{aligned}} \quad (8.21)$$

Further reduction of complexity can be achieved by performing single tree search: The $2T+1$ solutions are searched simultaneously, which implies less complexity (i.e. the multiplicative factor $2T$ in (8.21) disappears) but more storage is required.

8.3.5 LMMSE Detector

For comparative purposes, we recall the computational complexity using an LMMSE detector, given in (4.16)

$$\mathcal{C}_{\text{LMMSE}}^{\text{JAD}} \approx 8(KT)^2\left(NR+\frac{KT}{3}\right)\,. \quad (8.22)$$

The computation of the KT linear filters (2.66) adds to $\mathcal{C}_{\text{LMMSE}}^{\text{JAD}}$, with complexity $8(KT)^2$ each. Finally, the global complexity for LMMSE detection for a data block of length M' and per user becomes

$$\boxed{C_{\text{LMMSE}} \approx 8M'KT^2\left(NR+\tfrac{4KT}{3}\right)\,.} \quad (8.23)$$

8.4 Computational Complexity Comparison

In Figure 8.5 we show the complexity for conventional or reduced-rank hard-output sphere decoder, soft-output sphere decoder and LMMSE detection, according to Equations (8.12), (8.17), (8.21) and (8.23), respectively. We vary the number of transmit antennas T in Figure 8.5.a and the size of the alphabet Q in Figure 8.5.b.

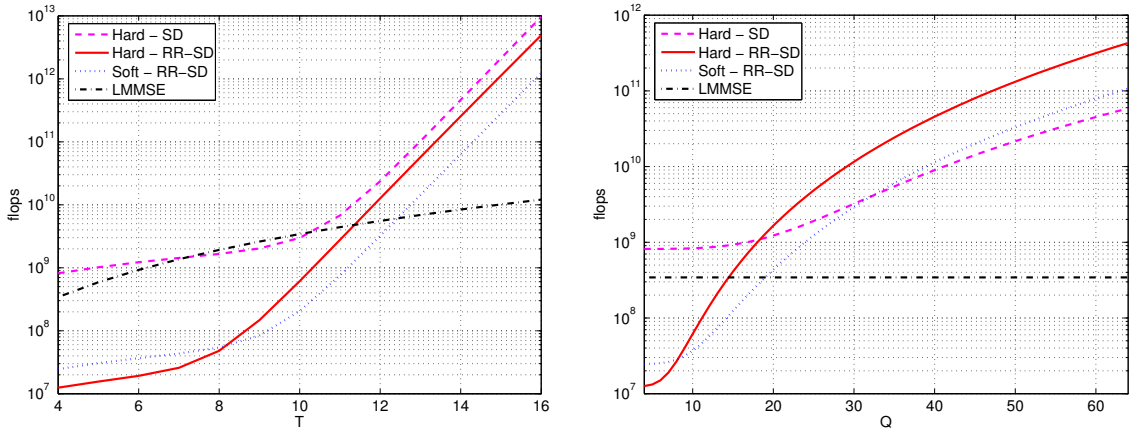
The following conclusions can be drawn for the low-dimensions $T \leq 10$ and $Q \leq 16$:

- The computational complexity of the conventional hard-output sphere decoder is at best of the same order as the one for LMMSE, and also the highest of all schemes. Referring to Figures 8.1 and 8.2, we can thus discard this method as both too complex and poorly performing.
- The use of the reduced-rank implementation for hard-output sphere decoding leads to complexity reduction of up to two orders of magnitude. This is very promising for time-varying channels, where computations are performed for every symbol.
- The use of soft-output sphere decoding implies a slight increase in computational complexity. Related to the considerable gain in performance over hard-output sphere decoding (see Figures 8.1 and 8.2), the increase in complexity is negligible enough to prefer this method. For higher T or Q, the soft-output sphere decoder using reduced-rank implementation even becomes the least complex approach.
- Using single tree search would further reduce the complexity of the soft-output sphere decoder.

As soon as dimensions increase more, i.e. $T \geq 10$ or $Q \geq 16$, we observe that

- only the LMMSE detector remains in a reasonable complexity range, with in some cases more than two orders of magnitude less complexity compared to any variant of sphere decoding;
- all sphere decoder variants seem to converge to a similar order of complexity;

Common knowledge is that a sphere decoder performs better than an LMMSE detector, while requiring more effort. This does not apply in our comparison, where the sphere decoder performs on a per user basis, for feasibility reasons, while LMMSE decodes all symbols at once. On one hand, the performance of the sphere decoder on a per user basis will be at best similar to the LMMSE performance. On the other hand, at higher dimensions (T and Q), a sphere decoder has complexity increasing exponentially [JO05a], while LMMSE has polynomial complexity.



(a) (b)

Figure 8.5: Complexity comparison. We show the complexity in $flops$ per block and per user. In Figure 8.5.a we plot $flops$ $vs.$ the number of transmit antennas T. In Figure 8.5.b we vary the alphabet size Q. Other parameters are $R = 4$, $K = 32$ and $N = 64$.

This makes the sphere decoders less favorable for increasing matrix dimensions and LMMSE becomes a better option, both in terms of complexity and performance.

Although $T = 12$ is not very realistic as a number of transmit antennas for a mobile unit, it also represents the number of symbols decoded with one sphere decoder: a sphere decoder with $T = 12$ could as well be seen as for example decoding three users with four transmit antennas each. In terms of complexity, it is thus more interesting to decode as few symbols as possible. This will evidently have a negative effect on the performance, since it requires interference cancelation, and a compromise between low complexity and good performance needs to be found depending on the parameters.

8.5 Empirical Computational Complexity

In the previous section we have seen that computational complexity for a sphere decoder can not be exhaustively computed, but only upper-bounded, since the number of visited nodes in the tree depends on the realization of the channel. Here we show some empirical cumulative distribution functions (cdf) of the number of nodes that were visited during the simulations.

The empirical cdf of the sphere decoder computational complexity can be computed using the actual number of nodes visited for each step in the simulations, as well as the

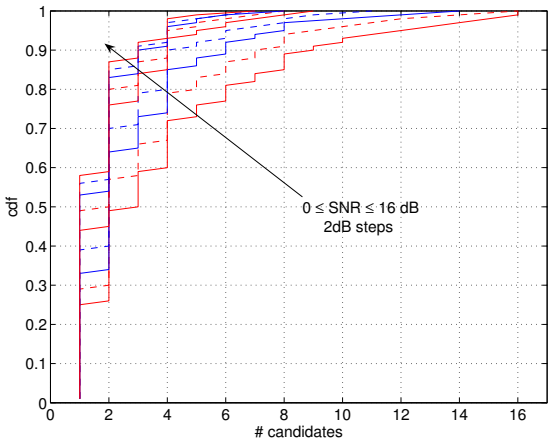


Figure 8.6: Hard-output sphere decoder: number of visited nodes in a $K = 24$ multi-user system and 4×4 MIMO channels, for various SNR $\in \{0, 2, \ldots, 16\}$ dB. The arrow shows increasing SNR.

complexity expressions from Section 8.3.

Figure 8.6 shows the influence of the signal to noise ratio on the cdf of the number of candidates q_t, for hard-output sphere decoder. Here, the system has $K = 24$ user, 4×4 channels and LMMSE channels estimates are used. We show the number of candidates after step $s = T - 1 = 3$, which is upper bounded by $q_s \leq Q^{T-s+1} = 16$. The various curves stand for different values of the signal to noise ratio, with the SNR as a parameter varying as in the simulations in 2dB steps from 0 to 16 dB. The arrow shows increasing SNR. As intuitively expected, the number of visited nodes tends to get higher with decreasing SNR.

Figure 8.7 shows the cdf of the computational complexity for a hard-output sphere decoder using either a conventional (Figure 8.7.a) or a reduced-rank (Figure 8.7.b) implementation. Complexity is computed according to (8.7) and (8.15), and averaged over all values of SNR. Various curves show different numbers of users $K \in \{16, 24\}$ and both cases of perfect channel knowledge or estimated channel. For comparison, the upper bound corresponding to exhaustive search (i.e. for a constant radius $\rho = \infty$) is shown.

Some conclusions follow:

- Firstly, the use of the sphere decoder over a low-complexity exhaustive search (i.e. a sphere decoder with infinite radius) allows more than one order of magnitude complexity reduction in almost 20% of the cases. For 90% of the cases, the complexity is reduced by a factor of five, for 95% by a factor of three.

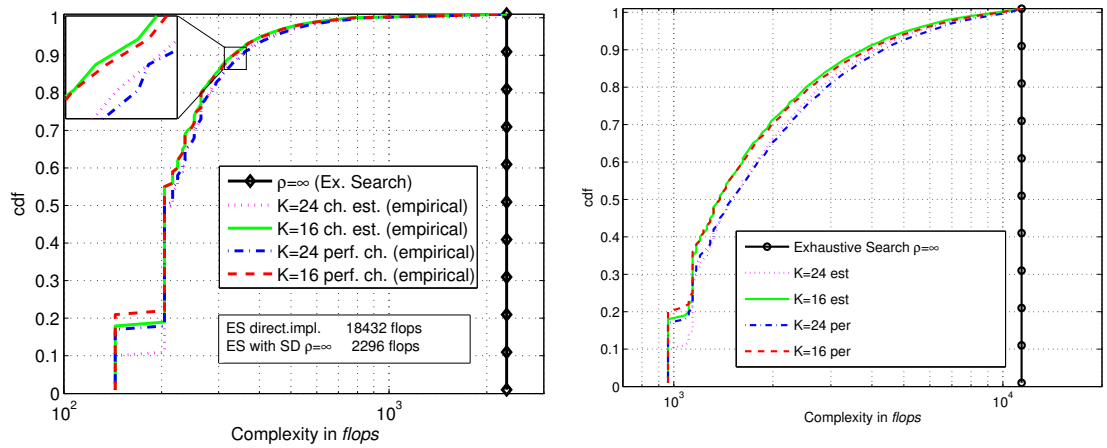


Figure 8.7: Hard-output sphere decoder: complexity using perfect channel knowledge ('per') or estimated channel ('est'), and $K \in \{16, 24\}$ users. We show (a) the conventional and (b) the reduced-rank implementation of a hard-output sphere decoder.

- Secondly, as expected, the number of users or the fact that the channel is estimated (and thus erroneous) or not has very little influence on the total complexity.
- Finally, the search using the reduced-rank sphere decoder is more complex than the search using the conventional implementation, however, this is largely compensated by the complexity of the QR factorization.

We have to keep in mind that in these pictures, the complexity is shown per user, thus the global complexity would be proportional to the number of users.

Chapter 9
Conclusion

We considered an iterative receiver for multi-carrier (MC) code division multiple access (CDMA) sub-optimally implementing a *maximum a posteriori* (MAP) detector. Such a receiver has a block structure including a channel estimator, a multi-user detector following parallel interference cancelation (PIC) and a series of single-user BCJR decoders. The iterative process consists in feeding the channel estimators and the multi-user detector with refined soft information producing more accurate estimates of the transmitted symbols.

As the system grows complex with the number of users, the multiple input multiple output (MIMO) channels, the number of subcarriers, or the fast variations of the channels, signal processing becomes prohibitively complex.

We aimed at reducing complexity of the iterative receiver, and more specifically of the channel estimator and the multi-user detector, originally performing *linear minimum mean square error* (LMMSE) detection. This thesis is split into three parts. In the first part we described in detail how to design transmitters and receivers to take maximum profit of the multiple antennas, regardless of the complexity. We presented various scenarios that were used in the remainder of the thesis. In the second part we focused on linear detection and the use of the Krylov subspace method to approximate the LMMSE filters. In the third part we replaced the LMMSE multi-user detector by a non-linear *maximum likelihood* (ML) detector in its low-complexity implementation based on sphere decoding. In both parts II and III, complexity and performance of various scenarios was discussed.

Key findings and conclusions are listed in the following.

Multiple Antenna Transceivers: Various scenarios to combine multiple antennas at the transmitters and receivers are presented, regardless of their complexity or feasibility. Joint encoding at the transmitter and joint antenna detection at the receiver appeared to perform best, despite being the most complex scenario.

Krylov Subspace Method: The Krylov subspace method is described in details as well as its application to a generic LMMSE filter. Commonly, using the Krylov subspace method is shown to enable computational complexity reduction.

Parallel Interference Cancelation in Chip Space: In the iterative receiver performing PIC in chip space, as many LMMSE filters as there were transmit antennas needed to be computed, despite a common matrix inverse. As a consequence, using the Krylov subspace method for each of these filters did not allow complexity reduction compared to exact LMMSE detection, nevertheless computations could be implemented in a parallel manner and storage requirements are reduced.

Parallel Interference Cancelation in User Space: Performing matched filtering prior to PIC leads to a signal in a user space. A new LMMSE filter in the user space was derived that leads to a unique equalizer detecting all transmitted symbols at once. Previous complexity issues due to numerous LMMSE filters were avoided, and use of the Krylov subspace method allowed considerable complexity reduction. A very slight loss in performance is observed, however negligible regarding to the complexity and storage savings allowed.

Adaptive Krylov Methods: Various methods have been investigated for initializing the iterative algorithm for Krylov subspace based linear detection. The following possibilities have been defined and compared:

• The Zeros initialization is used by default, when no specific information is available. In all cases, the initial value is set to zero.

•The Time initialization makes use of the time variations of the channel. Results obtained at time instant are used as initial value in the following instant.

• The Loop initialization makes use of the iterative structure of the receiver. The results obtained at one receiver iteration, getting more accurate from iteration to iteration, are used as initial values in the following iteration.

When used in an appropriate way, they appeared to accelerate convergence of the Krylov subspace method. For example, the Time initialization was shown to be more efficient for slower variations of the channel.

Low-Complexity Equalizer for HSDPA: The LMMSE equalizer in the HSDPA receiver was replaced by its approximation using the Krylov subspace method. Performance comparison with a least mean square (LMS) equalizer was performed, and the Krylov subspace method was shown to be more robust to the presence of orthogonal channel noise simulator (OCNS). The Zeros and Time-adaptive initialization methods were used for various velocities of the mobile units. The Zeros initialization turned out to be more adequate for fast-varying channels while the Time-adaptive initialization accelerated convergence in slow-varying channels.

Sphere Decoder: We replaced the LMMSE equalizer of our iterative receiver by a sphere decoder allowing low-complexity ML detection on a per user basis. Comparison with LMMSE filtering showed considerable gain in complexity but performance degradation, mainly due to hard outputs feeding the BCJR decoder downwards the multi-user detector.

Reduced-Rank Sphere Decoder: A large amount of complexity in the sphere decoder appeared to be in the QR factorization needed to triangulate the channel, which has to be computed with every new realization of the channel. We made use of the basis expansion channel model to develop a reduced-rank implementation of the sphere decoder using a block structure. This implementation is especially sustainable for time-varying channels, allowing considerable reduction in complexity.

Soft Sphere Decoder: To compensate the performance degradation induced by sphere decoding, we computed soft outputs using log-likelihood ratios. These soft symbols were computed by means of the reduced-rank sphere decoder, thus combining both low-complexity and suitable performance.

Computational Complexity: Exhaustive expressions for the computational complexity of an LMMSE filter, its approximation using the Krylov subspace method, and the various sphere decoders have been derived throughout this thesis.

Appendix A LMMSE Filter (2.75)

We compute the LMMSE $\boldsymbol{F}$ verifying [Ver98]

$$\boldsymbol{F}^{\mathrm{H}} = \underset{\boldsymbol{F}}{\operatorname{argmin}}\, \mathrm{E}\{\|\boldsymbol{F}^{\mathrm{H}}\hat{\boldsymbol{x}} - \boldsymbol{b}\|^2\} = \mathrm{E}\left\{\boldsymbol{b}\hat{\boldsymbol{x}}^{\mathrm{H}}\right\}(\mathrm{E}\left\{\hat{\boldsymbol{x}}\hat{\boldsymbol{x}}^{\mathrm{H}}\right\})^{-1},$$

with $\hat{\boldsymbol{x}} = \tilde{\boldsymbol{S}}^{\mathrm{H}}\boldsymbol{y} - (\boldsymbol{R}-\boldsymbol{D})\tilde{\boldsymbol{b}}$.

We recall that $\boldsymbol{R} = \tilde{\boldsymbol{S}}^{\mathrm{H}}\tilde{\boldsymbol{S}}$ and

$$\begin{aligned}
\mathrm{E}\{\boldsymbol{b}\boldsymbol{b}^{\mathrm{H}}\} &= \boldsymbol{I}_{KT} \\
\mathrm{E}\{\tilde{\boldsymbol{b}}\boldsymbol{b}^{\mathrm{H}}\} &= \mathrm{E}\{\boldsymbol{b}\tilde{\boldsymbol{b}}^{\mathrm{H}}\} = \mathrm{E}\{\tilde{\boldsymbol{b}}\tilde{\boldsymbol{b}}^{\mathrm{H}}\} = \boldsymbol{I}_{KT} - \boldsymbol{V} \\
\mathrm{E}\{\boldsymbol{y}\boldsymbol{y}^{\mathrm{H}}\} &= \mathrm{E}\left\{\left(\tilde{\boldsymbol{S}}\boldsymbol{b} + \boldsymbol{z}\right)\left(\boldsymbol{b}^{\mathrm{H}}\hat{\boldsymbol{S}}^{\mathrm{H}} + \boldsymbol{z}^{\mathrm{H}}\right)\right\} = \tilde{\boldsymbol{S}}\tilde{\boldsymbol{S}}^{\mathrm{H}} + \sigma_n^2\boldsymbol{I}_N \\
\mathrm{E}\{\boldsymbol{y}\tilde{\boldsymbol{b}}^{\mathrm{H}}\} &= \mathrm{E}\left\{\left(\tilde{\boldsymbol{S}}\boldsymbol{b} + \boldsymbol{z}\right)\tilde{\boldsymbol{b}}^{\mathrm{H}}\right\} = \tilde{\boldsymbol{S}}(\boldsymbol{I}_{KT} - \boldsymbol{V})
\end{aligned}.$$

Let us consider the elements $\mathrm{E}\left\{\boldsymbol{b}\hat{\boldsymbol{x}}^{\mathrm{H}}\right\}$ and $\mathrm{E}\left\{\hat{\boldsymbol{x}}\hat{\boldsymbol{x}}^{\mathrm{H}}\right\}$ separately. We have on the one hand

$$\begin{aligned}
\boldsymbol{b}\hat{\boldsymbol{x}}^{\mathrm{H}} &= \boldsymbol{b}\left(\boldsymbol{y}^{\mathrm{H}}\tilde{\boldsymbol{S}} - \tilde{\boldsymbol{b}}^{\mathrm{H}}(\boldsymbol{R}-\boldsymbol{D})\right) \\
&= \boldsymbol{b}\left(\boldsymbol{b}^{\mathrm{H}}\boldsymbol{R} + \boldsymbol{z}^{\mathrm{H}}\tilde{\boldsymbol{S}} - \tilde{\boldsymbol{b}}^{\mathrm{H}}(\boldsymbol{R}-\boldsymbol{D})\right)
\end{aligned}$$

and taking the expectation yields

$$(a) \quad \mathrm{E}\{\boldsymbol{b}\hat{\boldsymbol{x}}^{\mathrm{H}}\} = (\boldsymbol{V}\boldsymbol{R} - \boldsymbol{V}\boldsymbol{D} + \boldsymbol{D}).$$

On the other hand, we get

$$\hat{\boldsymbol{x}}\hat{\boldsymbol{x}}^{\mathrm{H}} = \left(\tilde{\boldsymbol{S}}^{\mathrm{H}}\boldsymbol{y}\boldsymbol{y}^{\mathrm{H}}\tilde{\boldsymbol{S}} + (\boldsymbol{R}-\boldsymbol{D})\tilde{\boldsymbol{b}}\tilde{\boldsymbol{b}}^{\mathrm{H}}(\boldsymbol{R}-\boldsymbol{D}) - (\boldsymbol{R}-\boldsymbol{D})\tilde{\boldsymbol{b}}\boldsymbol{y}^{\mathrm{H}}\tilde{\boldsymbol{S}} - \tilde{\boldsymbol{S}}^{\mathrm{H}}\boldsymbol{y}\tilde{\boldsymbol{b}}^{\mathrm{H}}(\boldsymbol{R}-\boldsymbol{D})\right)$$

with expectation

$$\begin{aligned}
(b) \quad \mathrm{E}\{\hat{\boldsymbol{x}}\hat{\boldsymbol{x}}^{\mathrm{H}}\} &= \big(\boldsymbol{R}^2 + \sigma_n^2\boldsymbol{R} - (\boldsymbol{R}-\boldsymbol{D})(\boldsymbol{I}_{KT} - \boldsymbol{V})\boldsymbol{R} - \boldsymbol{R}(\boldsymbol{I}_{KT} - \boldsymbol{V})(\boldsymbol{R}-\boldsymbol{D}) \\
&\qquad + (\boldsymbol{R}-\boldsymbol{D})(\boldsymbol{I}_{KT} - \boldsymbol{V})(\boldsymbol{R}-\boldsymbol{D})\big) \\
&= \left(\boldsymbol{R}\boldsymbol{V}\boldsymbol{R} + \sigma_n^2\boldsymbol{R} + \boldsymbol{D}(\boldsymbol{I}_{KT} - \boldsymbol{V})\boldsymbol{D}\right).
\end{aligned}$$

we finally obtain the LMMSE filter (2.75)

$$\boldsymbol{F}^{\mathrm{H}} = (\boldsymbol{V}\boldsymbol{R} - \boldsymbol{V}\boldsymbol{D} + \boldsymbol{D})\left(\boldsymbol{R}\boldsymbol{V}\boldsymbol{R} + \sigma_n^2\boldsymbol{R} + \boldsymbol{D}(\boldsymbol{I}_{KT} - \boldsymbol{V})\boldsymbol{D}\right)^{-1}$$

□

Appendix B Matrix Inversion Lemma for (3.17)

Using the matrix inversion lemma [MS00] we can write

$$\begin{aligned}(\boldsymbol{D}_1 + \boldsymbol{M}\boldsymbol{D}_2\boldsymbol{M}^{\mathrm{H}})^{-1} = \\ = \boldsymbol{D}_1^{-1} - \boldsymbol{D}_1^{-1}\boldsymbol{M}(\boldsymbol{D}_2^{-1} + \boldsymbol{M}^{\mathrm{H}}\boldsymbol{D}_1^{-1}\boldsymbol{M})^{-1}\boldsymbol{M}^{\mathrm{H}}\boldsymbol{D}_1^{-1}\end{aligned}$$

$$\begin{aligned}(\boldsymbol{D}_1 + \boldsymbol{M}\boldsymbol{D}_2\boldsymbol{M}^{\mathrm{H}})^{-1}\boldsymbol{M} = \\ = \boldsymbol{D}_1^{-1}\boldsymbol{M} - \boldsymbol{D}_1^{-1}\boldsymbol{M}(\boldsymbol{D}_2^{-1} + \boldsymbol{M}^{\mathrm{H}}\boldsymbol{D}_1^{-1}\boldsymbol{M})^{-1}\boldsymbol{M}^{\mathrm{H}}\boldsymbol{D}_1^{-1}\boldsymbol{M} \\ = \boldsymbol{D}_1^{-1}\boldsymbol{M}(\boldsymbol{I}_b - (\boldsymbol{D}_2^{-1} + \boldsymbol{M}^{\mathrm{H}}\boldsymbol{D}_1^{-1}\boldsymbol{M})^{-1}\boldsymbol{M}^{\mathrm{H}}\boldsymbol{D}_1^{-1}\boldsymbol{M}) \\ = \boldsymbol{D}_1^{-1}\boldsymbol{M}(\boldsymbol{D}_2^{-1} + \boldsymbol{M}^{\mathrm{H}}\boldsymbol{D}_1^{-1}\boldsymbol{M})^{-1}\boldsymbol{D}_2^{-1} \\ = \boldsymbol{D}_1^{-1}\boldsymbol{M}(\boldsymbol{I}_b + \boldsymbol{D}_2\boldsymbol{M}^{\mathrm{H}}\boldsymbol{D}_1^{-1}\boldsymbol{M})^{-1}\end{aligned}$$

Finally, we obtain

$$(\boldsymbol{D}_1 + \boldsymbol{M}\boldsymbol{D}_2\boldsymbol{M}^{\mathrm{H}})^{-1}\boldsymbol{M} = \boldsymbol{D}_1^{-1}\boldsymbol{M}(\boldsymbol{I}_b + \boldsymbol{D}_2\boldsymbol{M}^{\mathrm{H}}\boldsymbol{D}_1^{-1}\boldsymbol{M})^{-1}$$

and the LMMSE filter (3.20). □

Appendix C A Posteriori LLR (7.43)

Using Bayes' theorem [Bay63] we can write

$$p(b_t = \pm 1|\boldsymbol{z}) = \sum_{\boldsymbol{b}\in\mathbb{B}_t^{\pm}} p(\boldsymbol{b}|\boldsymbol{z}) = \sum_{\boldsymbol{b}\in\mathbb{B}_t^{\pm}} p(\boldsymbol{z}|\boldsymbol{b})\frac{p(\boldsymbol{b})}{p(\boldsymbol{z})}\,.$$

Knowing $p(\boldsymbol{b}) = p(b_t = \pm 1)\prod\limits_{t'\neq t} p(b_{t'})$, this becomes

$$p(b_t = \pm 1|\boldsymbol{z}) = \frac{p(b_t = \pm 1)}{p(\boldsymbol{z})}\sum_{\boldsymbol{b}\in\mathbb{B}_t^{\pm}} p(\boldsymbol{z}|\boldsymbol{b})\prod_{t'\neq t} p(b_{t'})\,,$$

The *a posteriori* LLR can thus be written

$$\begin{aligned}\lambda_{\mathrm{POST}}(b_t) &= \ln\left(\frac{p(b_t = +1|\boldsymbol{z})}{p(b_t = -1|\boldsymbol{z})}\right)\\ &= \lambda_{\mathrm{PRIOR}}(b_t) + \ln\left(\frac{\sum\limits_{\boldsymbol{b}\in\mathbb{B}_t^{+}} p(\boldsymbol{z}|\boldsymbol{b})\cdot\prod\limits_{t'\neq t} p(b_{t'})}{\sum\limits_{\boldsymbol{b}\in\mathbb{B}_t^{-}} p(\boldsymbol{z}|\boldsymbol{b})\cdot\prod\limits_{t'\neq t} p(b_{t'})}\right)\,.\end{aligned}$$

We further know from the definition of the *a priori* LLR that

$$\begin{aligned}p(b_{t'} = \pm 1) &= \frac{\exp\left(\pm\lambda_{\mathrm{PRIOR}}(b_{t'})\right)}{1+\exp(\pm\lambda_{\mathrm{PRIOR}}(b_{t'}))}\\ &= \underbrace{\frac{\exp\left(-\lambda_{\mathrm{PRIOR}}(b_{t'})/2\right)}{1+\exp(-\lambda_{\mathrm{PRIOR}}(b_{t'}))}}_{A_{t'}}\cdot\exp\left(\pm\tfrac{\lambda_{\mathrm{PRIOR}}(b_{t'})}{2}\right)\\ &= A_{t'}\exp\left(b_{t'}\tfrac{\lambda_{\mathrm{PRIOR}}(b_{t'})}{2}\right)\end{aligned}$$

which allows writing

$$\begin{aligned}\prod_{t'\neq t} p(b_{t'}) &= \underbrace{\left(\prod_{t'\neq t} A_{t'}\right)}_{\mathcal{A}_t} \cdot \prod_{t'\neq t} \exp\left(b_{t'}\frac{\lambda_{\mathrm{PRIOR}}(b_{t'})}{2}\right) \\ &= \mathcal{A}_t \exp\left(\sum_{t'\neq t} \frac{b_{t'}\lambda_{\mathrm{PRIOR}}(b_{t'})}{2}\right) .\end{aligned}$$

We define $\boldsymbol{b}_t \in \mathbb{C}^{2T-1}$ containing $\boldsymbol{b}$ except b_t, and $\boldsymbol{\lambda}_{\mathrm{PRIOR},t}$ containing the *a priori* probabilities of the elements of $\boldsymbol{b}_t$. The above expression becomes

$$\prod_{t'\neq t} p(b_{t'}) = \mathcal{A}_t \exp\left(\frac{\boldsymbol{b}_t^{\mathrm{T}}\boldsymbol{\lambda}_{\mathrm{PRIOR},t}}{2}\right)$$

and the *extrinsic* LLR given by $\lambda_{\mathrm{EXT}}(b_t) = \lambda_{\mathrm{POST}}(b_t) - \lambda_{\mathrm{PRIOR}}(b_t)$ becomes

$$\lambda_{\mathrm{EXT}}(b_t) = \ln\left(\frac{\displaystyle\sum_{\boldsymbol{b}\in\mathbb{B}_t^+} p(\boldsymbol{z}|\boldsymbol{b}) \cdot \exp\left(\frac{\boldsymbol{b}_t^{\mathrm{T}}\boldsymbol{\lambda}_{\mathrm{PRIOR},t}}{2}\right)}{\displaystyle\sum_{\boldsymbol{b}\in\mathbb{B}_t^-} p(\boldsymbol{z}|\boldsymbol{b}) \cdot \exp\left(\frac{\boldsymbol{b}_t^{\mathrm{T}}\boldsymbol{\lambda}_{\mathrm{PRIOR},t}}{2}\right)}\right) .$$

Using on one hand the fact that $\boldsymbol{b}^{\mathrm{T}}\boldsymbol{\lambda}_{\mathrm{PRIOR}} = \boldsymbol{b}_t^{\mathrm{T}}\boldsymbol{\lambda}_{\mathrm{PRIOR},t} - b_t\lambda_{\mathrm{PRIOR}}(b_t)$ and on the other hand that the elements of the noise vector $\boldsymbol{n}$ follow a Gaussian distribution with zero mean and variance σ^2

$$p(\boldsymbol{z}|\boldsymbol{b}) \propto \exp\left(-\frac{1}{\sigma^2}\|\boldsymbol{z} - \boldsymbol{R}\boldsymbol{F}\boldsymbol{b}\|^2\right) ,$$

we obtain the final expression of the *extrinsic* LLR after max-log approximation [HtB03]

$$\begin{aligned}\lambda_{\mathrm{EXT}}(b_t) \;\approx\; & \max_{\boldsymbol{b}\in\mathbb{B}_t^+}\left\{-\tfrac{1}{\sigma^2}\|\boldsymbol{z} - \boldsymbol{R}\boldsymbol{F}\boldsymbol{b}\|^2 + \frac{\boldsymbol{b}^{\mathrm{T}}\boldsymbol{\lambda}_{\mathrm{PRIOR}}}{2}\right\} \\ & - \max_{\boldsymbol{b}\in\mathbb{B}_t^-}\left\{-\tfrac{1}{\sigma^2}\|\boldsymbol{z} - \boldsymbol{R}\boldsymbol{F}\boldsymbol{b}\|^2 + \frac{\boldsymbol{b}^{\mathrm{T}}\boldsymbol{\lambda}_{\mathrm{PRIOR}}}{2}\right\} \\ & - \lambda_{\mathrm{PRIOR}}(b_t) .\end{aligned}$$

It follows

$$\begin{aligned}\lambda_{\mathrm{POST}}(b_t) \;\approx\; & \max_{\boldsymbol{b}\in\mathbb{B}_t^+}\left\{-\tfrac{1}{\sigma^2}\|\boldsymbol{z} - \boldsymbol{R}\boldsymbol{F}\boldsymbol{b}\|^2 + \frac{\boldsymbol{b}^{\mathrm{T}}\boldsymbol{\lambda}_{\mathrm{PRIOR}}}{2}\right\} \\ & - \max_{\boldsymbol{b}\in\mathbb{B}_t^-}\left\{-\tfrac{1}{\sigma^2}\|\boldsymbol{z} - \boldsymbol{R}\boldsymbol{F}\boldsymbol{b}\|^2 + \frac{\boldsymbol{b}^{\mathrm{T}}\boldsymbol{\lambda}_{\mathrm{PRIOR}}}{2}\right\} .\end{aligned}$$

□

Appendix D Notation

We use the notation presented in Table D.1.

Symbol	Description
$f[m]$	function of a discrete variable
$\boldsymbol{a}$	column vector
$a[i]$	i-th element of $\boldsymbol{a}$
$\boldsymbol{A}$	matrix
$[\boldsymbol{A}]_{i,\ell}$ or $A_{i,\ell}$	i,ℓ-th element of $\boldsymbol{A}$
$\boldsymbol{A}_{P\times Q}$	upper left part of $\boldsymbol{A}$ with dimension $P \times Q$
$\boldsymbol{A}^{\mathrm{T}}$	transpose of $\boldsymbol{A}$
$\boldsymbol{A}^{\mathrm{H}}$	conjugate transpose of $\boldsymbol{A}$
$\mathrm{diag}(\boldsymbol{a})$	diagonal matrix with entries $a[i]$
$\boldsymbol{I}_Q$	$Q \times Q$ identity matrix
$\boldsymbol{F}_Q$	$Q \times Q$ unitary Fourier matrix
$\mathbf{1}_Q$	$Q \times 1$ column vector with all ones
$\mathbf{0}_Q$	$Q \times 1$ column vector with all zeros
a^*	complex conjugate of a
$\lfloor a \rfloor$	largest integer, lower or equal than $a \in \mathbb{R}$
$\lceil a \rceil$	smallest integer, greater or equal than $a \in \mathbb{R}$
$\lvert a \rvert$	absolute value of a
$\lVert \boldsymbol{a} \rVert$	ℓ_2 norm of vector $\boldsymbol{a}$
$\lVert \boldsymbol{A} \rVert_{\mathrm{F}}$	Frobenius norm of matrix $\boldsymbol{A}$
$\mathrm{vec}(\boldsymbol{A})$	stacks all columns of matrix $\boldsymbol{A}$ in a single vector
j	$\sqrt{-1}$
δ_{ij}	1 for $i = j$, 0 otherwise

Table D.1: Notation used throughout this thesis.

Functions

All functions used throughout this thesis are listed in Table D.2.

Function	Description
$\underset{b}{\operatorname{argmax}}\{\mathcal{E}\}$	b such as the expression $\mathcal{E}$ is maximized
$\underset{b}{\operatorname{argmin}}\{\mathcal{E}\}$	b such as the expression $\mathcal{E}$ is minimized
$\operatorname{diag}(\boldsymbol{e})$	creates a diagonal matrix with the elements of $\boldsymbol{e}$
$\max\{a, b\}$	maximum of a and b
$\min\{a, b\}$	minimum of a and b
span	subspace spanned by

Table D.2: Functions used throughout this thesis.

Variables

All variables used throughout this thesis are listed below, the Greek Symbols in Table D.3, the lower case symbols in Table D.4 and the upper case symbols in Table D.5.

Symbol	Description
α, β	general coefficients
γ	ratio
δ	distance as the ℓ_2 norm
$\boldsymbol{\Delta}$, $\boldsymbol{D}$	diagonal matrix
ϵ, $\boldsymbol{\epsilon}$	error
η	path attenuation
ζ	load
λ	eigenvalue or log-likelihood ratio
μ, ν	general matrix elements
ψ, Ψ	projection coefficients
$\hat{\psi}$, $\hat{\Psi}$	estimated projection coefficients
ν_{D}	normalized Doppler frequency
ρ	sphere radius
σ^2	variance

Table D.3: Greek symbols used throughout this thesis.

Symbol	Description
b, $\boldsymbol{b}$	data symbol
$\tilde{b}$, $\tilde{\boldsymbol{b}}$	soft symbols using extrinsic probabilities
$\tilde{b}'$, $\tilde{\boldsymbol{b}}'$	soft symbols using a posteriori probabilities
c	code bit
c_0	speed of light
d, $\boldsymbol{d}$	transmitted symbol
f, $\boldsymbol{f}$	filter
f_C	carrier frequency
g, $\boldsymbol{g}$	channel frequency response
$\tilde{g}$, $\tilde{\boldsymbol{g}}$	channel projected onto basis expansion
$\hat{g}$, $\hat{\boldsymbol{g}}$	estimated channel frequency response
h, $\boldsymbol{h}$	channel impulse response
i, ℓ	general Index
$.^{(i)}$	iteration index
$.^{(\mathrm{i})}$	imaginary part
j	$\sqrt{-1}$
k	user index
$k_{\boldsymbol{A}}$	condition number of matrix $\boldsymbol{A}$
m	discrete time index
n, $\boldsymbol{n}$	noise
p	probability distribution
$\boldsymbol{p}$	pilot symbols
q	subcarrier index
r	receive antenna index
$.^{(\mathrm{r})}$	real part
$\boldsymbol{s}$	spreading sequence
$\tilde{\boldsymbol{s}}$	effective spreading sequence
t	transmit antenna index
u, $\boldsymbol{u}$	basis function
v	velocity
w, $\boldsymbol{w}$	estimated transmitted symbols (after detection)
y, $\boldsymbol{y}$	received signal after cyclic prefix removal and DFT
$\tilde{y}$, $\tilde{\boldsymbol{y}}$	signal after parallel interference cancelation
$\boldsymbol{z}$	pseudo-received signal

Table D.4: Lower case symbols used throughout this thesis.

Symbol	Description
A	symbol alphabet size
$\mathcal{A}$	symbol alphabet
B_{d}	one sided Doppler bandwidth
$\mathcal{C}$	computational complexity
D	dimension of basis expansion
$\mathcal{E}$	mathematical expression
G	length of cyclic prefix
$\boldsymbol{H}$	channel matrix
J	number of pilot symbols
K	number of users
$\mathbb{K}$, $\mathbb{S}$	subspace of $\mathbb{C}^a$
$\mathbb{K}_s$, $\mathbb{K}_s(\boldsymbol{A}, \boldsymbol{a})$	Krylov subspace of dimension s derived from $\boldsymbol{A}$ and $\boldsymbol{a}$
L	essential support of the channel response
L_f	filter length
L_h	delay spread
M	block length
N	number of subcarriers
N_0	noise power spectral density
P	length of an OFDM symbol with cyclic prefix
$\mathcal{P}$	pilot placement index set
$\boldsymbol{Q}$, $\boldsymbol{R}$	output of the QR factorization of a matrix
R	number of receive antennas at the receiver
R_{C}	code rate of the convolutional encoder
R_{S}	code rate of the symbol mapper
s, S	Krylov subspace dimension
$\boldsymbol{S}$	spreading matrix
$\tilde{\boldsymbol{S}}$	effective spreading matrix
T	number of transmit antennas per transmitter
T_{C}	chip duration
T_{D}	root mean square delay spread
T_{S}	symbol duration
$\boldsymbol{V}$	error covariance matrix
$\mathcal{V}$	set of valid/candidate vectors

Table D.5: Upper case symbols used throughout this thesis.

Abbreviations

All abbreviations are listed in two Tables D.6 and D.7.

Abbreviation	Description
3GPP	3rd Generation Partnership Project
ACK/NACK	positive/negative Acknowledgement
APP	A Posteriori Probability
AWGN	Additive White Gaussian noise
BER	Bit Error Rate
CA	Complex Addition
cdf	Cumulative Distribution Function
CDMA	Code Division Multiple Access
CE	Channel Estimation
CM	Complex Multiplication
CPICH	Common Pilot Channel
CQI	Channel Quality Indicator
DFT	Discrete Fourier Transform
DPS	Discrete Prolate Spheroidal
ES	Exhaustive Search
EXT	Extrinsic Probabilities
flop	Floating Point Operation
HARQ	Hybrid Automatic Repeat Request
HSDPA	High Speed Downlink Packet Access
HS-DPCCH	High Speed Dedicated Physical Control Channel
HS-PDSCH	High Speed Physical Downlink Shared Control Channel
HS-SCCH	High Speed Shared Control Channel
i.i.d.	independent identical distributed
IR	Incremental Redundancy

Table D.6: Abbreviations 0-I used throughout this thesis.

Abbreviation	Description
ISI	Inter-Symbol Interference
ITU	International Telecommunication Union
JAD	Joint Antenna Detection
LLR	Log Likelihood Ratio
LMMSE	Linear Minimum Mean Square Error
LMS	Least Mean Square
MAI	Multi-Access Interference
MAP	Maximum A Posteriori
MC-CDMA	Multi-Carrier Code Division Multiple Access
MIMO	Multiple Input Multiple Output
ML	Maximum Likelihood
MRC	Maximum Ratio Combining
MSE	Mean Square Error
MUD	Multi-User Detection
OCNS	Orthogonal Channel Noise Simulator
OFDM	Orthogonal Frequency Division Multiplexing
P-CCPCH	Primary Common Control Physical Channel
P-CPICH	Primary Common Pilot Channel
PIC	Parallel Interference Cancelation
QAM	Quadrature Amplitude Modulation
QPSK	Quadrature Phase Shift Keying
RR-SD	Reduced-Rank Sphere Decoder
RV	Redundancy and constellation Version
SAD	Single Antenna Detection
SCH	Synchronization Channel
SD	Sphere Decoder
SF	Spreading Factor
SISO	Single Input Single Output
SNR	Signal to Noise Ratio
TTI	Transmission Time Interval
UE	User Equipment
UMTS	Universal Mobile Telecommunications System
WSS-US	Wide-Sense Stationary Uncorrelated Scattering
ZF	Zero Forcing

Table D.7: Abbreviations I-Z used throughout this thesis.

Bibliography

[3GP] 3GPP. Third Generation Partnership Project (3GPP), http://www.3gpp.org/.

[AEVZ02] E. Agrell, T. Eriksson, A. Vardy, and K. Zeger. Closest point search in lattices. *IEEE Transactions on Information Theory*, 48(8):2201–2214, August 2002.

[Ala98] S. M. Alamouti. A simple transmitter diversity scheme for wireless communications. *IEEE Journal on Selected Areas in Communications*, 16(8):1451–1458, October 1998.

[ALSO04] E. Agrell, J. Lassing, E. G. Strom, and T. Ottosson. On the optimality of the binary reflected Gray code. *IEEE Transactions on Information Theory*, 50(12):3170–3182, December 2004.

[ARAS99] P. D. Alexander, M. C. Reed, J. A. Asentorfer, and C. B. Schlegel. Iterative multi-user interference reduction: Turbo CDMA. *IEEE Transactions on Communications*, 47(7):1008–1014, July 1999.

[Ati94] M. Atiyah. *Introduction To Commutative Algebra*. Westview Press, Boulder, CO, USA, February 1994.

[Bab86] L. Babai. On lovász lattice reduction and the nearest lattice point problem. *Combinatorica*, 6(1):1–13, March 1986.

[Bay63] T. Bayes. An essay towards solving a problem in the doctrine of chances. *Letter from Richard Price to John Canton*, December 1763.

[BB03] L. Brunel and J. J. Boutros. Lattice decoding for joint detection in direct-sequence CDMA systems. *IEEE Transactions on Information Theory*, 49(4):1030–1037, April 2003.

[BCJR74] L. R. Bahl, J. Cocke, F. Jelinek, and J. Raviv. Optimal decoding of linear codes for minimizing symbol error rate. *IEEE Transactions on Information Theory*, 20(2):284–287, March 1974.

[BGBF03] J. Boutros, N. Gresset, L. Brunel, and M. Fossorier. Soft-input soft-output lattice sphere decoder for linear channels. In *IEEE Global Telecommunications Conference (GLOBECOM)*, volume 3, pages 1583–1587, San Francisco, CA, USA, December 2003.

[Bin90] J. A. C. Bingham. Multicarrier modulation for data transmission: An idea whose time has come. *IEEE Transactions on Communications*, 28(5):5–14, May 1990.

[BRC08] L. G. Barbero, T. Ratnarajah, and C. Cowan. On the complexity of the sphere decoder for frequency-selective MIMO channels. *IEEE Transactions on Signal Processing*, 56(12):6031–6043, December 2008.

[Bru01] L. Brunel. Optimum multi-user detection for MC-CDMA systems using sphere decoding. In *Proc. 12th IEEE International Symposium on Personal, Indoor and Mobile Radio Communication (PIMRC)*, San Diego, CA, USA, October 2001.

[Bru04] L. Brunel. Multi-user detection techniques using maximum likelihood sphere decoding in multicarrier CDMA systems. *IEEE Transactions on Wireless Communications*, 3(3):949–957, May 2004.

[BT06] L. G. Barbero and J. S. Thompson. A fixed-complexity MIMO detector based on the complex sphere decoder. In *IEEE 7th Workshop on Signal Processing Advances in Wireless Communications (SPAWC)*, Cannes, France, July 2006.

[CL02] A. M. Chan and I. Lee. A new reduced-complexity sphere decoder for multiple antenna systems. In *Proc. IEEE International Conference on Communications (ICC)*, New York, NY, USA, April 2002.

[Cla68] R. H. Clarke. A statistical theory of mobile-radio reception. *The Bell System Technical Journal*, 47:957–1000, July/August 1968.

[CMD06] L. Cottatellucci, R. R. Müller, and M. Debbah. Linear detectors for multi-user systems with correlated spatial diversity. In *Proc. 14th European Signal Processing Conference (EUSIPCO)*, Florence, Italy, September 2006.

[CMT04] G. Caire, R. R. Müller, and T. Tanaka. Iterative multi-user joint decoding: Optimal power allocation and low-complexity implementation. *IEEE Transactions on Information Theory*, 50(9):1950–1973, September 2004.

[Cor01] L. M. Correia. *Wireless Flexible Personalised Communications*. Wiley, New York, NY, USA, 2001.

[DeG70] M. H. DeGroot. *Optimal Statistical Decisions*. McGraw-Hill, New York, NY, USA, 1970.

[DEGC03] M. O. Damen, H. El Gamal, and G. Caire. On maximum-likelihood detection and the search for the closest lattice point. *IEEE Transactions on Information Theory*, 49(10):2389–2402, October 2003.

[DU07] G. Dietl and W. Utschick. Complexity reduction of iterative receivers using low-rank equalization. *IEEE Transactions on Signal Processing*, 55(3):1035–1046, March 2007.

[DZ05] C. Dumard and T. Zemen. Double Krylov subspace approximation for low-complexity iterative multi-user decoding and time-variant channel estimation. In *Proc. IEEE Workshop on Signal Processing Advances in Wireless Communications (SPAWC)*, New York, NY, USA, June 2005.

[DZ06a] C. Dumard and T. Zemen. Integration of the Krylov subspace method in an iterative multi-user detector for time-variant channels. In *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Toulouse, France, May 2006.

[DZ06b] C. Dumard and T. Zemen. Krylov subspace method based low-complexity MIMO multi-user receiver for time-variant channels. In *Proc. 17th IEEE International Symposium on Personal, Indoor and Mobile Radio Communication (PIMRC)*, Helsinki, Finland, September 2006.

[DZ08] C. Dumard and T. Zemen. Low-complexity MIMO multi-user receiver: A joint antenna detection scheme for time-varying channels. *IEEE Transactions on Signal Processing*, 56(7):2931–2940, July 2008.

[FG98] G. J. Foschini and M. J. Gans. On limits of wireless communications in a fading environment when using multiple antennas. *Wireless Personal Communications*, 6:311–335, March 1998.

[FK04] I. Forkel and H. Klenner. High speed downlink packet access (HSDPA) - A means of increasing downlink capacity in WCDMA cellular networks? In *Proc. of the 5th European Wireless Conference 2004*, Barcelona, Spain, February 2004.

[FKG+05] K. Freudenthaler, F. Kaltenberger, S. Geirhofer, S. Paul, J. Wehinger, C. F. Mecklenbräuker, A. Springer, and J. Berkmann. Throughput simulations for a UMTS high speed downlink packet access LMMSE equalizer. In *Proc. IST Mobile Summit*, Dresden, Germany, June 2005.

[FP85] U. Fincke and M. Pohst. Improved methods for calculating vectors of short length in a lattice, including a complex analysis. *Mathematics of Computation*, 44(169-170):463–471, April 1985.

[GJ79] M. R. Garey and D. S. Johnson. *Computers and Intractability: A Guide to the Theory of NP-Completeness*. W. H. Freeman, New York, NY, USA, 1979.

[GL96] G. H. Golub and C. F. Van Loan. *Matrix Computations*. Johns Hopkins University Press, Baltimore, MD, USA, 3rd edition, 1996.

[GMR05] S. Geirhofer, C. Mehlführer, and M. Rupp. Design and real-time measurement of HSDPA equalizers. In *Proc. IEEE Workshop on Signal Processing Advances in Wireless Communications (SPAWC)*, New York, NY, USA, June 2005.

[Gol67] R. Gold. Optimal binary sequences for spread spectrum multiplexing. *IEEE Transactions on Information Theory*, 13(4):619–621, October 1967.

[Gra53] F. Gray. Pulse Code Communication. US Patent Application 2632058, March 1953.

[Hay05] S. Haykin. *Adaptive Filter Theory*. Prentice-Hall, Upper Saddle River, New Jersey, 2005.

[HBB03] B. Haberland, S. Bloch, and V. Braun. 3G evolution towards high speed downling packet access. In *Alcatel Telecommunications Review*, number 4, January 2003.

[HLR+08] B. Hu, I. Land, L. Rasmussen, R. Piton, and B. H. Fleury. A divergence minimization approach to joint multi-user decoding for coded CDMA. *IEEE Journal on Selected Areas in Communications*, 26(3):432–445, April 2008.

[HLY02] L. Hanzo, T. H. Liew, and B. L. Yeap. *Turbo Coding, Turbo Equalization and Space-Time Coding for Transmission over Fading Channels*. Wiley, New York, NY, USA, 2002.

[HT01] S. V. Hanly and D. N. C. Tse. Resource pooling and effective bandwidths in CDMA networks with multi-user receivers and spatial diversity. *IEEE Transactions on Information Theory*, 47(4):1328–1351, May 2001.

[HtB03] B. M. Hochwald and S. ten Brink. Achieving near-capacity on a multiple antenna channel. *IEEE Transactions on Communications*, 51(3):389–399, March 2003.

[HV05a] B. Hassibi and H. Vikalo. On the sphere-decoding algorithm I: expected complexity. *IEEE Transactions on Signal Processing*, 53(8):2806–2818, August 2005.

[HV05b] B. Hassibi and H. Vikalo. On the sphere-decoding algorithm II: generalizations, second-order statistics, and applications to communications. *IEEE Transactions on Signal Processing*, 53(8):2819–2834, August 2005.

[HWS04] M. L. Honig, G. K. Woodward, and Y. Sun. Adaptive iterative multi-user decision feedback detection. *IEEE Transactions on Wireless Communications*, 3(2):477–485, March 2004.

[Jak74] W. C. Jakes. *Microwave Mobile Communications*. John Wiley & Sons, New York, USA, 1974.

[JO05a] J. Jalden and B. Ottersten. On the complexity of sphere decoding in digital communications. *IEEE Transactions on Signal Processing*, 53(4):1474–1484, April 2005.

[JO05b] J. Jalden and B. Ottersten. Parallel implementation of a soft-output sphere decoder. In *39th Asilomar Conference on Signals, Systems and Computers*, Pacific Grove, CA, USA, October 2005.

[Kal07] F. Kaltenberger. *Low-Complexity Real-Time Signal Processing for Wireless Communications*. PhD thesis, Technical University of Vienna, May 2007.

[KBC01] M. Kobayashi, J. Boutros, and G. Caire. Successive interference cancelation with SISO decoding and EM channel estimation. *IEEE Journal on Selected Areas in Communications*, 19(8):1450 – 1460, August 2001.

[KFP+05] F. Kaltenberger, K. Freudenthaler, S. Paul, J. Wehinger, C. F. Mecklenbräuker, and A. Springer. Throughput enhancement by cancelation of

synchronization and pilot channel for UMTS high speed downlink packet access. In *Proc. IEEE Workshop on Signal Processing Advances in Wireless Communications (SPAWC)*, New York, NY, USA, June 2005.

[KH06] I. P. Kirsteins and G. Hongya. Performance analysis of Krylov space adaptive beamformers. In *Proc. IEEE Workshop on Sensor Array and Multichannel Signal*, volume 3, pages 16–20, July 2006.

[KM87] H. Krishna and S. D. Morgera. The Levinson recurrence and fast algorithms for solving Toeplitz systems of linear equations. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 35(6):839–848, June 1987.

[KS99] T. Kailath and A. H. Sayed. *Fast Reliable Algorithms for Matrices with Structure*. SIAM, Philadelphia, PA, USA, 1999.

[KST04] R. Koetter, A. C. Singer, and M. Tuchler. Turbo equalization. *IEEE Signal Processing Magazine*, 21(1):67–80, January 2004.

[KT04] B. Kecicioglu and M. Torlak. Reduced-rank beamforming methods for SDMA/OFDM communications. In *Proc. 38th Asilomar Conference on Signals, Systems and Computers*, volume 2, pages 1973–1977, Pacific Grove, CA, USA, July 2004.

[LMW+04] M. Lončar, R.R. Müller, J. Wehinger, C.F. Mecklenbräuker, and T. Abe. Iterative channel estimation and data detection in frequency selective fading MIMO channels. *European Transcations on Telecommunications*, 15(5):459–470, September 2004.

[Mem04a] Members of 3GPP. Technical specification group radio access network; UE radio access capabilities definition (3GPP TS 25.306 version 6.3.0). Technical report, 3GPP, March 2004.

[Mem04b] Members of 3GPP. Technical specification group radio access network; User Equipment (UE) radio transmission and reception (FDD) (3GPP TS 25.101 version 6.4.0). Technical report, 3GPP, March 2004.

[Mem05a] Members of 3GPP. Technical specification group radio access network; High speed downlink packet access: UE radio transmission and reception (FDD) (3GPP TS 25.890 version 1.0.0). Technical report, 3GPP, May 2005.

[Mem05b] Members of 3GPP. Technical specification group radio access network; Multiplexing and channel coding (FDD) (3GPP TS 25.212 version 6.5.0). Technical report, 3GPP, June 2005.

[Mem05c] Members of 3GPP. Technical specification group radio access network; Physical channel and mapping of transport channels onto physical channels (FDD) (3GPP TS 25.211 version 6.5.0). Technical report, 3GPP, June 2005.

[Mem05d] Members of 3GPP. Technical specification group radio access network; Spreading and modulation (FDD) (3GPP TS 25.213 version 6.3.0). Technical report, 3GPP, June 2005.

[MGDC06] A. D. Murugan, H. E. Gamal, M. O. Damen, and G. Caire. A unified framework for tree search decoding: rediscovering the sequential decoder. *IEEE Transactions on Information Theory*, 52(3):933–953, March 2006.

[Mol05] A. Molisch. *Wireless Communications*. Wiley-IEEE Press, New York, NY, USA, November 2005.

[Mow94] W. H. Mow. Maximum likelihood sequence estimation from the lattice viewpoint. *IEEE Transactions on Information Theory*, 40(5):1591–1600, September 1994.

[MS00] T. K. Moon and W. C. Stirling. *Mathematical Methods and Algorithms*. Prentice Hall, Upper Saddle River, NJ, USA, 2000.

[Mül04] R. R. Müller. Random matrices, free probability and the replica method. In *EURASIP European Signal Processing Conference (EUSIPCO)*, Vienna, Austria, September 2004.

[NFS07] M. Nicoli, S. Ferrara, and U. Spagnolini. Soft-iterative channel estimation: Methods and performance analysis. *IEEE Transactions on Signal Processing*, 55(6):2993–3006, June 2007.

[OT04] R. Otnes and M. Tuchler. Iterative channel estimation for turbo equalization of time-varying frequency-selective channels. *IEEE Transactions on Wireless Communications*, 3(6):1918–1923, November 2004.

[PEM+03] S. Parkvall, E. Englund, P. Malm, T. Hedberg, M. Persson, and J. Peisa. WCDMA evolved - high speed packet data services. In *Ericsson Review*, number 2, pages 56–65, 2003.

[PNG03] A. Paulraj, R. Nabar, and D. Gore. *Introduction to Space-Time Wireless Communications*. Cambridge University Press, Cambridge, UK, May 2003.

[Poh81] M. Pohst. On the computation of lattice vectors of minimal length, successive minima and reduced bases with applications. *SIGSAM Bulletin*, 15(1):37–44, 1981.

[PPWL04] D. Pham, K. R. Pattipati, P. K. Willett, and J. Luo. An improved complex sphere decoder for V-BLAST systems. *IEEE Signal Processing Letters*, 11(9):748–751, September 2004.

[Saa03] Y. Saad. *Iterative Methods for Sparse Linear Systems*. SIAM, Philadelphia, PA, USA, 2nd edition, 2003.

[SB08] D. Seethaler and H. Bölcskei. Infinity-norm sphere-decoding. In *IEEE International Symposium on Information Theory (ISIT)*, Toronto, Canada, July 2008.

[SBB08] C. Studer, A. Burg, and H. Bölcskei. Soft-output sphere decoding: Algorithms and VLSI implementation. *IEEE Journal on Selected Areas in Communications*, 26(2):290–300, February 2008.

[Sch66] R. D. Schafer. *An Introduction to Nonassociative Algebras*. Academic Press, New York, NY, USA, January 1966.

[Sch91] Louis L. Scharf. *Statistical Signal Processing: Detection, Estimation, and Time Series Analysis*. Addison-Wesley Publishing Company, Inc., Reading, MA, USA, 1991.

[SE94] C. P. Schnorr and M. Euchner. Lattice basis reduction: improved practical algorithms and solving subset sum problems. *Mathematical Programming*, 66(2):181–199, April 1994.

[Sle78] D. Slepian. Prolate spheroidal wave functions, Fourier analysis, and uncertainty - V: The discrete case. *The Bell System Technical Journal*, 57(5):1371–1430, May-June 1978.

[SMH06] D. Seethaler, G. Matz, and F. Hlawatsch. Efficient soft demodulation in MIMO-OFDM systems with BICM and constant modulus alphabets. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Toulouse, France, May 2006.

[Sor80] H. W. Sorenson. *Parameter Estimation: Principles and Problems*. Marcel Dekker Inc, New York, NY, USA, 1980.

[SW94] N. Seshadri and J. H. Winters. Two signaling schemes for improving the error performance of frequency-division-duplex (FDD) transmission systems with transmitter diversity. *International Journal on Wireless Information Networks*, 1(1), 1994.

[TB05] G. Taricco and E. Biglieri. Space-time decoding with imperfect channel estimation. *IEEE Transactions on Wireless Communications*, 4(4):1874–1888, July 2005.

[TKS02] M. Tüchler, R. Koetter, and A. Singer. Turbo equalization: Principles and new results. *IEEE Transactions on Communications*, 50(5):754–767, May 2002.

[TOS02] M. Tüchler, R. Otnes, and A. Schmidbauer. Performance of soft iterative channel estimation in turbo equalization. In *IEEE International Conference on Communications (ICC)*, volume 3, pages 1858–1862, New York, NY, USA, 2002.

[TSC98] V. Tarokh, N. Seshadri, and A. R. Calderbank. Space-time codes for high data rate wireless communication: Performance criterion and code construction. *IEEE Transactions on Information Theory*, 44(2):744–765, March 1998.

[TV05] D. Tse and P. Viswanath. *Fundamentals of Wireless Communications*. Cambridge University Press, Cambridge, UK, 2005.

[VA90] M. K. Varanasi and B. Aazhang. Multi-stage detection in asynchronous code-division multiple-access communications. *IEEE Transactions on Communications*, 38(4):509–519, April 1990.

[VA03] R. Vaughan and J. B. Andersen. *Channels, Propagation and Antennas for Mobile Communications*. Institution of Engineering and Technology, February 2003.

[VB93] E. Viterbo and E. Biglieri. A universal lattice decoder. In *14è Colloque du Groupe de Recherche et d'Études du Traitement du Signal et des Images (GRETSI)*, Juan les Pins, France, September 1993.

[VB99] E. Viterbo and J. Boutros. A universal lattice code decoder for fading channels. *IEEE Transactions on Information Theory*, 45(5):1639–1642, July 1999.

[vdV03] H. van der Vorst. *Iterative Krylov Methods for Large Linear Systems*. Cambridge University Press, Cambridge, UK, 2003.

[Ver98] S. Verdú. *Multi-user Detection*. Cambridge University Press, New York, USA, 1998.

[VH02] H. Vikalo and B. Hassibi. Maximum-likelihood sequence detection of multiple antenna systems over dispersive channels via sphere decoding. *EURASIP Journal on Applied Signal Processing*, 2002(5):525–531, May 2002.

[WE71] S. B. Weinstein and P. M. Ebert. Data transmission by frequency-divison multiplexing using the discrete fourier transform. *IEEE Transactions on Communications*, 19(5):628–634, October 1971.

[Weh05] J. Wehinger. *Iterative Multi-User Receivers for CDMA Systems*. PhD thesis, Vienna University of Technology, Vienna, Austria, July 2005.

[WG04] R. Wang and G. B. Giannakis. Approaching MIMO channel capacity with reduced-complexity soft sphere decoding. In *Proc. Wireless Communications and Networking Conference (WCNC)*, Atlanta, Georgia, USA, March 2004.

[WM06] J. Wehinger and C. F. Mecklenbrauker. Iterative CDMA multi-user receiver with soft decision-directed channel estimation. *IEEE Transactions on Signal Processing*, 54(10):3922–3934, October 2006.

[WTCM02] K.-W. Wong, C.-Y. Tsiu, R. S.-K. Cheng, and W.-H. Mow. A VLSI architecture of a K-best lattice decoding algorithm for MIMO channels. In *IEEE International Symposiumon Circuits and Systems (ISCAS)*, Scottsdale, AZ, USA, May 2002.

[XWZW04] W. Xu, Y. Wang, Z. Zhou, and J. Wang. A computationally efficient exact ML sphere decoder. In *IEEE Global Telecommunications Conference (GLOBECOM)*, volume 4, pages 2594–2598, Dallas, TX, USA, November 2004.

[Zem04] T. Zemen. *OFDM Multi-User Communication Over Time-Variant Channels*. PhD thesis, Vienna University of Technology, Vienna, Austria, July 2004.

[ZM05] T. Zemen and C. F. Mecklenbräuker. Time-variant channel estimation using discrete prolate spheroidal sequences. *IEEE Transactions on Signal Processing*, 53(9):3597–3607, September 2005.

[ZMWM06] T. Zemen, C. F. Mecklenbräuker, J. Wehinger, and R. R. Müller. Iterative joint time-variant channel estimation and multi-user detection for MC-CDMA. *IEEE Transactions on Wireless Communications*, 5(6):1469–1478, June 2006.

[ZX03] Y. R. Zheng and C. Xiao. Simulation models with correct statistical properties for Rayleigh fading channels. *IEEE Transactions on Communications*, 51(6):920–928, June 2003.

Biography

Charlotte Dumard was born in Paris, France, in May 1979. She received a double Master of Science degree in electrical engineering from Supélec - Ecole Supérieure d'Électricité, Gif-sur-Yvette, France, in December 2002 and from KTH - Kungl Techniska Högskolan (Royal Institute of Technology), Stockholm, Sweden, in March 2003.

She joined Vienna University of Technology as a doctoral student in March 2006 and completed her PhD in April 2009 right on time before turning 30.

She joined FTW - Forschungszentrum Telekommunikation Wien (Telecommunication Research Center Vienna), Vienna, Austria, in September 2004, where she worked as a junior researcher in two projects funded by the WWTF - Wiener Wissenschafts-, Forschungs- und Technologiefonds (Vienna Science and Technology Fund), under the "Mathematics *and*..." thematic program. First she worked in the project Math+MIMO (2004-2007) which aimed at developing new mathematical methods in order to model and analyze multi-antenna communications systems. She focused on low-complexity receiver algorithms. After completion of this project she worked in COCOMINT - Cooperative Communications for Traffic Telematics (2008-2010), in which she focused on developing distributed receiver algorithms.