Alexander Onic

Receiver Concepts for Unique Word OFDM

DISSERTATION

Zur Erlangung des akademischen Grades
Doktor der Technischen Wissenschaften (Dr.-techn.)

11/2013

Alpen-Adria-Universität Klagenfurt
Fakultät für Technische Wissenschaften
Institute of Networked and Embedded Systems

FAKULTÄT FÜR TECHNISCHE WISSENSCHAFTEN

Gutachter: Prof. Dr. techn. **Mario Huemer**
Prof. Dr.-Ing. Dr.-Ing. habil. **Johannes Huber**

Ehrenwörtliche Erklärung

Ich erkläre ehrenwörtlich, dass ich die vorliegende wissenschaftliche Arbeit selbstständig angefertigt und die mit ihr unmittelbar verbundenen Tätigkeiten selbst erbracht habe. Ich erkläre weiters, dass ich keine anderen als die angegebenen Hilfsmittel benutzt habe. Alle ausgedruckten, ungedruckten oder dem Internet im Wortlaut oder im wesentlichen Inhalt übernommenen Formulierungen und Konzepte sind gemäß den Regeln für wissenschaftliche Arbeiten zitiert und durch Fußnoten bzw. durch andere genaue Quellenangaben gekennzeichnet.

Die während des Arbeitsvorganges gewährte Unterstützung einschließlich signifikanter Betreuungshinweise ist vollständig angegeben.

Die wissenschaftliche Arbeit ist noch keiner anderen Prüfungsbehörde vorgelegt worden. Diese Arbeit wurde in gedruckter und elektronischer Form abgegeben. Ich bestätige, dass der Inhalt der digitalen Version vollständig mit dem der gedruckten Version übereinstimmt.

Ich bin mir bewusst, dass eine falsche Erklärung rechtliche Folgen haben wird.

(Unterschrift) (Ort, Datum)

Abstract

Unique Word OFDM (orthogonal frequency division multiplexing) is an attractive alternative to OFDM with cyclic prefix, which is adopted for data transmission in standards like DSL, LTE, DVB and IEEE 802.11 (WLAN). In this signaling concept, a deterministic sequence, a "unique word" (UW), is inserted into the transmit stream, instead of a cyclic copy of the data. Furthermore, this UW is part of the IDFT (inverse discrete Fourier transform) interval. This property distinguishes UW-OFDM from most other OFDM variants, while it offers the same advantages as the conventional OFDM (free of intersymbol interference, diagonalization of the channel matrix). By defining of a sequence in time domain, some capacity has to be allocated for redundancy in frequency domain. This redundancy solely depends on the transmit data (and defined system parameters) and can be utilized for a reliable recovery of the data. In order to exploit this redundancy, sophisticated receiver structures need to be employed, which is topic of this work. The achieved gain can be used for a higher data rate, range, reliability, capacity or battery lifespan.

Methods to generate valid UW-OFDM symbols are introduced in two variants: The systematic generation of UW-OFDM symbols, which can be done directly or in two steps, and the non-systematic generation. An analysis of the mean transmit energy of all generation methods sheds light on their suitability for communication systems and reveals possibilities for optimization.

The main part of this work is about suited receivers for UW-OFDM that are able to reconstruct the data reliably, after transmission over a dispersive channel. Besides the estimated transmit symbols, all these receivers need to provide reliability information, which enables a channel decoder to achieve better decoding results. All receivers are investigated regarding their bit error performance with and without channel coding, in the AWGN (additive white Gaussian noise) channel as well as in a multipath environment.

Besides two rather intuitively derived, two more optimum linear receivers are discussed, which emerge from classical as well as Bayesian estimation theory: The BLUE (best linear unbiased estimator) and the LMMSE (linear minimum mean square error) estimator. The computational complexity of all these receivers is analyzed for both OFDM symbol generation approaches and compared numerically. When using real transmit symbol constellations, the LMMSE estimator can be outperformed by the WLMMSE (widely LMMSE) estimator. Furthermore, a symbol scaling effect can be identified for these Bayesian receivers. This turns out to be harmful for the detection quality with higher order constellations, such as 16-QAM or 4-ASK. Symbol scaling compensated versions of the LMMSE and WLMMSE estimators are introduced and their performance documented.

As another main topic of this work, a few nonlinear receivers are discussed, starting with two decision directed concepts. First, a method for noise interpolation is introduced, which exploits the correlation of the data symbols after an LMMSE estimation, in order

to obtain improved estimates. It turns out that the selection of the samples which are used for estimation is decisive for the performance of this receiver.

In decision feedback equalization, the influence of detected data symbols on the receive signal is subtracted iteratively, in order to allow for a more reliable decision of the remaining symbols. Here, the order of detection is crucial for the decision quality.

For the derived linear UW-OFDM system model, a maximum-likelihood sequence estimation (MLSE) yields the best estimates possible. However, due to its computational complexity, it is unsuitable for practical application. As a practical realization of the MLSE, sphere decoding is presented, which obtains the same results with acceptable effort. For the determination of reliability information, however, a mathematical approximation and a limitation of parameter dynamics has to be applied, to keep the complexity in adequate limits, which destroys the optimality of the method.

An investigation of QR decomposition, as it is directly used for sphere decoding and in a version of decision feedback equalization, shows that the way, how the QR decomposition is computed, has significant impact on runtime or detection performance, respectively. An overall performance investigation reveals that the nonlinear receivers clearly outperform the LMMSE estimator in uncoded transmission. If channel coding is used, they are still able to achieve a small gain over the best performing linear estimator. However, the LMMSE estimator constitutes a highly reasonable compromise when performance and complexity are taken into account.

Contents

1. Introduction

1.1. Communication with High Data Rate over Dispersive Channels

With the demand of higher data rates and the technological possibilities to build communication systems transmitting many Megabits per second, the time interval to transmit one symbol has become very small. For this reason the dispersive properties of the propagation channel dominate the transmission quality. For wireless single-carrier communication scenarios, where one data symbol is transmitted after another, multipath propagation (see Figure 1.1) becomes a tough issue. In order to recover the transmitted data, it needs to be dealt with reflections, diffusions, diffractions and scattering of the electromagnetic wave at walls, buildings, landmarks and basically any obstacle, as this causes initially independent data symbols to interfere with each other. A simple computation shows that, considering a symbol rate of 10 million symbols per second, which is close to the fastest mode in IEEE 802.11g [IEE06] or LTE [Tec09], an indirect propagation path adding a detour of 30 m already corresponds to the duration of one symbol, thus causing strong inter-symbol interference. It is obvious that for cellular scenarios many hundreds of meters in addition to the shortest path are common. For DSL transmission, which is based on copper wire, the symbol rate is so high that a transmitted symbol can have an effect on a receive symbol received more than 100 symbols later. An important task for a high data rate receiver is to cope with these dispersive propagation effects.

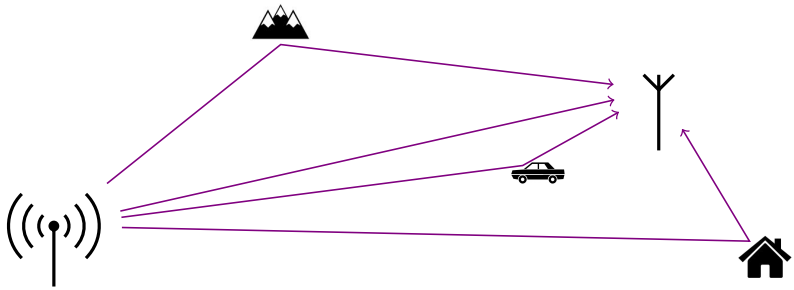


Figure 1.1.: Schematic visualization of multipath propagation.

The high symbol rate is a reason for inter-symbol interference (ISI) in the first place. Beyond that it puts a lot of pressure on all the countermeasures a receiver employs in order to deal with these effects regarding complexity and computation time. In the usual block-based real-time communication scenario, the receiver has only the amount of time to process a receive block, until the next block is fully received.

Therefore, multi-carrier block transmission is an adequate practice. In particular, Orthogonal Frequency Division Multiplexing (OFDM) is a well researched multi-carrier technique that fulfills the need to cope with severe multipath propagation as well as simple processing. Its origins [Wei09] stem from the 1960s, where the foundations of OFDM in the modern form were laid [Sal67, Cha66]. In [CG68, WE71] the theoretical foundations were brought into a more practical and usable form. For practical implementations and lab realizations, however, it took until the early eighties [PR80, Hir81], when hardware became capable of handling the technique and OFDM started its track record.

Currently in 2013, in almost every modern digital communication standard, OFDM plays an important role. It is used in the audio and video broadcasting standards DAB and DVB [Eur95, Eur97] that are realized in wireless as well as wire-bound modes. Internet service providers lay out their last mile to the customer usually via DSL [ANS95]. In-building networks can be realized by power line communication [GL08] or wireless via the IEEE 802.11 standards [IEE12]. Finally, OFDM is also used in the LTE cellular communications standard [DPS11], permeating most communication standards currently used.

On the downside of OFDM is the need of a guard interval that collects the channel transient caused by multipath propagation. The guard interval separates the OFDM symbols as a countermeasure for inter-symbol interference, and is, depending on the sequence transmitted during that interval, a rather wasteful consumer of transmit energy and in some way a waste of transmit time reducing the bandwidth efficiency. There have been investigations to reduce this dissipation by diminishing the length of the guard interval, but giving up a part of the orthogonality feature of OFDM [HTO+02].

Along with other ideas that make different use of the guard interval and are cited in the next section, Unique Word OFDM transmits a deterministic sequence instead of a cyclic prefix, and is able to avoid most drawbacks of a guard interval while introducing some very nice and interesting properties, that can be exploited at the receiver.

1.2. State of the Art

The classical OFDM flavor using cyclic prefixes (CP) to ensure cyclicity of the OFDM symbols, as it is described in Section 2.1, is dominant and well known; a reference on a text book [NP00] shall stand for a huge variety of publications on this topic. This cyclic prefix depends on the data and is hence random. There are many variations of OFDM using CPs that change the content of the guard interval. For example, by rotating the prefix part in the complex domain, the transmitter of a CP-OFDM system is able to mitigate bad channels [CDCY11, Coo12]. This is under the assumption of channel knowledge at the transmitter and transmission of side information.

Related to UW-OFDM but in detail also very different is KSP-OFDM (known symbol padding) [TYP+07, WSM08, WS10], where the guard interval is occupied by a deterministic sequence. This technique is used, for example, in [CNS06]. In contrast to UW-OFDM the guard interval is not part of the DFT (discrete Fourier transform) interval. By removing the known portion of the known sequence in the guard interval (taking the channel dispersion into account), and adding the remaining transient of the data to the beginning of the observed block interval, the cyclicity of the KSP-OFDM symbols is achieved.

For UW-OFDM, the UW needs to be removed at the receiver in the same way, under the knowledge of the multipath characteristics of the channel. However, in case of UW-OFDM, the generation of the UW within the DFT interval introduces correlations among the subcarriers which can be advantageously exploited by the receiver to improve the bit error performance. Whilst in both schemes, UW- and KSP-OFDM, the deterministic sequences can be used for synchronization and channel estimation purposes, these correlations are not present in KSP-OFDM.

Actually under the label of KSP-OFDM, a method was shown in [CM01] that resembles the UW-OFDM concept according to the direct approach, described in Section 2.3.2. Due to the huge excess energies for the redundant subcarriers this method did not perform well, as shown in this work as well. Another variant can be derived from KSP-OFDM by setting the deterministic guard interval sequence to zero. This is called ZP-OFDM (zero padded) [LPV10, SM07] and endures the same shortcomings over UW-OFDM as KSP-OFDM. Also equivalent with KSP-OFDM, only termed differently, is TDS-OFDM (time domain synchronous OFDM). In a recent approach [DWC11] the TDS principle is adapted in order to mitigate the orthogonality loss problem in an elegant way. However, no redundancy is introduced in frequency domain, which is one of the main UW-OFDM features.

Several other attempts to place a deterministic sequence in the guard interval can be found, for example [MdCD06, WLL11, JHJR02]. All of them differ from the proposed UW-OFDM scheme in the way that the guard interval is not part of the DFT interval, and thus no correlations and coding is introduced by these approaches.

Since UW-OFDM time domain symbols contain a block of fixed samples, which is the UW, the set of all corresponding vectors in the discrete frequency domain forms a coset to a Reed Solomon (RS) code. Usually RS codes are defined for a finite field along with a suited discrete Fourier transform. The set of code words is specified by the fact, that the (inverse) DFT of all code words contains a block of $d_{\min} - 1$ successive zeros, where $d_{\min}$ is the minimum Hamming distance of the RS code. Due to the introduction of a fixed unique word, a coset code to an RS code is generated with respect to this Fourier transform with the same minimum distance $d_{\min}$ (see [Bla03]). All these definitions apply for the field of complex numbers and a usual DFT of length N as well and thus to UW-OFDM. All these parallels to RS codes suggest algebraic decoding. In [HHH10a, HHK09] it is shown that this leads to an ill-conditioned system of equations that is very sensitive to noise. Hence, other methods, like those introduced in this work, need to be applied.

The term "unique word" is derived from the corresponding concept for SC/FDE (single-carrier / frequency domain equalization) transmission. In single-carrier transmission also a cyclic prefix or a unique word can be used for establishing block cyclicity and simpler processing. The introduction of CPs [SKJ94, Czy97, Kad97] as well as UWs [DGE01, CM01, WMS$^+$02a, WMS$^+$02b, HWH03, Wit04, RBH05] in SC/FDE has been studied very well. Even the parallels to RS codes could be identified for SC/FDE [MMM09]. The introduction of UWs in SC/FDE systems is straightforward, since the data symbols as well as the UW symbols are defined in time domain. In UW-OFDM the data symbols are defined in frequency domain, whereas the UW symbols are defined in time domain, which leads to some difficulties. In [HHH10d] the similarities and differences of the UW approach for OFDM and SC/FDE are studied.

1.3. Scope of this Work

Although, some believe that there is not much more to learn on single-antenna OFDM, this work is dedicated to exactly this topic, and to show that OFDM with the unique word prefix still offers a lot challenges and chances. Specifically, the introduction of redundancy during the UW-OFDM symbol generation process poses an interesting task, as this redundancy can be beneficially exploited at the receiver in order to get better estimates of the data vector. This is what this work is dedicated to.

In Chapter 2 the principles of OFDM are introduced descriptively and mathematically, using the example of the 'classical' OFDM utilizing cyclic prefixes at first, in order to come to the main topic of this work: *Unique Word OFDM*. This chapter leads to specific receiver processing instructions in order to obtain a simple linear system model, describing the UW-OFDM transmission from modulation to the input of the data estimator. This 'basic' UW-OFDM scheme is called systematically generated UW-OFDM and requires the dedication of subcarriers to the role of data or redundant subcarriers. A study of OFDM symbol energies and the impact on the bit error ratio of the introduced UW-OFDM generation approach, along with a very similar and intuitive symbol generation alternative is shown. The OFDM symbol energy also depends on the positions of the redundant subcarriers. A well performing set can be found using the heuristic, introduced in this chapter as well. Then, a non-systematic approach will be briefly introduced, that lifts this role dedication from the subcarriers in order get all-purpose UW-OFDM subcarriers. While a big attenuation of a data subcarrier in a particular realization of the multipath channel has a very bad impact on the decoding results in systematic UW-OFDM, this effect is highly mitigated in the non-systematic approach.

After the presentation how to formulate the UW-OFDM transmission procedure in terms of a linear system model, some linear receivers are presented in Chapter 3. Two estimators are derived using classical and Bayesian estimation theory, and compared against two rather intuitive equalizers. Furthermore, an estimator will be shown that is better suited for improper transmit constellations. An inherent issue of the Bayesian receivers is the bias introduction, which consists of an inter-symbol interference and a symbol scaling part and poses a severe bit error degradation when using higher order transmit constellations. Symbol scaling compensated versions of the two introduced Bayesian estimators will be derived, in addition. For some of the linear estimators, complexity optimized versions are shown, and a complexity analysis proves the effectivity of these optimized equalizers. In a bit error comparison, the estimators will be evaluated regarding their performance. The improved estimators, suited for the special cases of improper or higher order constellations highlight a better bit error performance in the use case they were built for.

Several nonlinear receivers are shown in Chapter 4, which are well known in the MIMO (multiple input multiple output) world, where multiple transmit and/or receive antennas are used. Although UW-OFDM is designed for the use of only one transmit and receive antenna, the interpretation as a MIMO system is valid. The noise interpolation method is based on the correlation inherent in the remaining noise after an initial linear estimation, and is derived from Wiener filter theory. The problem of sample selection is introduced: The noise samples that shall be estimated as well as the samples that these samples are estimated from need to be picked very carefully, in order to improve the initial estimates. This behavior is analyzed in the same section. As another MIMO receiver, decision feedback equalization is shown in the ZF and MMSE version. After the

presentation of (due to its computational complexity mostly unfeasible) maximum likelihood sequence estimation, the sphere decoder (SD) as a much more practical receiver is shown. The sphere decoder algorithm is introduced for the UW-OFDM context, and its complexity is investigated. A performance comparison in the end of the chapter reveals the best receivers.

For all linear and nonlinear receivers, possibilities are shown to determine soft information, to be used by a channel decoder, a Viterbi decoder for example. With this soft information, as a measure of reliability, a channel decoder is able to deliver much better decoding results as opposed to only getting hardly decided binary symbols as input.

A QR matrix decomposition plays an important role as a tool for some of the presented receivers. Therefore, some methods to achieve this are shown in Chapter 5, along with an analysis of their impact. Chapter 6 will provide a conclusion, and an outlook to interesting research possibilities and questions left open by this work.

The mathematical notation throughout this work is as follows: Vectors and matrices are denoted in bold face lower case $\mathbf{a}$ and upper case $\mathbf{A}$ letters. The element number k of a vector is addressed by a_k. If appropriate, the k-th column of a matrix $\mathbf{A}$ is denoted by $\mathbf{a}_k$, such that a matrix with N columns is given by $\mathbf{A} = [\mathbf{a}_0, \mathbf{a}_1, \mathbf{a}_2, \ldots, \mathbf{a}_{N-1}]$. In order to name the element in row k and column l of the same matrix the notation $[\mathbf{A}]_{k,l}$ will be used. To address row number k of a matrix the same notation will be used with an asterisk to denote the whole range of column indices, as in $[\mathbf{A}]_{k,*}$. The conjugate complex of a scalar or of all elements of a vector or matrix is given by $\mathbf{a}^*$. Applicable to scalars, whole vectors or matrices, the operations $\Re\{\mathbf{a}\}$ and $\Im\{\mathbf{a}\}$ return the real part or the imaginary part of the complex numbers in $\mathbf{a}$. The transpose operation on a vector or a matrix is denoted by $\mathbf{A}^\mathsf{T}$, the conjugate transpose $\left(\mathbf{A}^\mathsf{T}\right)^* = \left(\mathbf{A}^*\right)^\mathsf{T}$, or Hermitian of a matrix is $\mathbf{A}^\mathsf{H}$. The identity matrix is given by $\mathbf{I}$.

A binary symbol $b \in \{0, 1\}$ is flipped to the other possibility by $\overline{b}$. The probability of an event is displayed as $\Pr(\cdot)$. A probability density function (pdf) is shown as $p(\cdot)$, and if the context is not adequate to distinguish different pdfs, an index $p_a(\cdot)$ is added to show its affiliation to random variable a.

Starting in Chapter 3, furthermore, the following notational rules aid comprehension. A vector with a tilde $\tilde{\mathbf{a}}$ denotes an estimate of a vector $\mathbf{a}$ and a hat denotes a vector with its values sliced to a valid data symbol, for example $\hat{\mathbf{a}} = \lfloor \tilde{\mathbf{a}} \rceil$. With a slicing operation $\lfloor \cdot \rceil$, the resulting vector contains values of the given transmit symbol alphabet $\mathcal{A}$.

2. The Unique Word OFDM Signaling Scheme

In this section, the multi-carrier transmission technique OFDM is introduced by starting with a quite intuitive solution, resulting in cyclic prefix OFDM, which will be further elaborated in order to come to the main topic of this work: Unique Word OFDM. A few preparatory steps at the receiver will be formulated that help to formulate the UW-OFDM transmission scheme as a linear model. This simple model will be the basis for the further receiver considerations in this work.

The basic idea of *multi-carrier* transmission in contrast to *single-carrier* transmission is to put the data symbols one after another in frequency domain, instead of putting them one after another in the time domain transmission stream. Therefore, a block of N data symbols is put into a data vector $\mathbf{d} \in \mathbb{C}^{N\times 1}$, which shall be transmitted at once. By modulating the data symbols on a set of subcarriers that are orthogonal to each other, all symbols can be easily separated again after the transmission. A set of N orthogonal signals can be formed easily from complex exponentials

$$f_n[k] = \frac{1}{N} \mathrm{e}^{\mathrm{j}\frac{2\pi}{N}nk}, \qquad n,k = 0,\ldots,N-1. \tag{2.1}$$

By putting all N signals $f_n[k]$ in the columns of a matrix

$$\mathbf{F}_N^{-1} = \begin{bmatrix} f_0[0] & f_1[0] & \cdots & f_{N-1}[0] \\ f_0[1] & f_1[1] & & f_{N-1}[1] \\ \vdots & & & \vdots \\ f_0[N-1] & f_1[N-1] & \cdots & f_{N-1}[N-1] \end{bmatrix}, \tag{2.2}$$

the inverse DFT (IDFT) matrix is obtained, having the element in the k-th row and l-th column

$$\left[\mathbf{F}_N^{-1}\right]_{k,l} = \frac{1}{N} \mathrm{e}^{\mathrm{j}\frac{2\pi}{N}kl}. \tag{2.3}$$

The modulation of the data symbols onto different subcarriers is implied by the choice of the orthogonal signals in (2.1). Consequently, with these signals in vector form $\mathbf{f}_n = [f_n[0], f_n[1], \ldots, f_n[N-1]]^{\mathsf{T}}$ and a data vector $\mathbf{d}$ consisting of the data symbols to be transmitted, the time domain transmit signal is computed by

$$\begin{aligned} \mathbf{x} &= \sum_{n=0}^{N-1} \mathbf{f}_n d_n \\ &= \mathbf{F}_N^{-1}\mathbf{d}. \end{aligned} \tag{2.4}$$

By the definition of the IDFT matrix $\mathbf{F}_N^{-1}$, the matrix $\mathbf{F}_N$ constitutes the DFT matrix with the elements

$$[\mathbf{F}_N]_{k,l} = \mathrm{e}^{-\mathrm{j}\frac{2\pi}{N}kl}. \tag{2.5}$$

Using these definitions of the DFT and IDFT matrices as in (2.5) and (2.3), the N-point discrete Fourier transform of a sequence $x[k] \circ\!\!-\!\!\bullet\, X[n]$ with $n, k = 0, 1, \ldots, N-1$, and the inverse discrete Fourier transform, defined as

$$X[n] = \sum_{k=0}^{N-1} x[k]\mathrm{e}^{-\mathrm{j}\frac{2\pi}{N}kn}, \qquad x[k] = \frac{1}{N}\sum_{n=0}^{N-1} X[n]\mathrm{e}^{\mathrm{j}\frac{2\pi}{N}kn}, \tag{2.6}$$

can be elegantly expressed in vector notation (only in this case the upper case $\mathbf{X}$ is a vector and not a matrix)

$$\mathbf{X} = \mathbf{F}_N\mathbf{x}, \qquad \mathbf{x} = \mathbf{F}_N^{-1}\mathbf{X}.$$

Another common definition of the DFT matrices is

$$\left[\mathbf{F}'_N\right]_{kl} = \frac{1}{\sqrt{N}}\mathrm{e}^{-\mathrm{j}\frac{2\pi}{N}kl}$$
$$\text{and } \left[\mathbf{F}'^{-1}_N\right]_{kl} = \frac{1}{\sqrt{N}}\mathrm{e}^{\mathrm{j}\frac{2\pi}{N}kl},$$

which has the advantages that both matrices are *unitary* and thus, the inverse of the DFT matrix is simply its Hermitian:

$$\mathbf{F}'_N\mathbf{F}'^{\mathsf{H}}_N = \mathbf{I} \quad \Rightarrow \quad \mathbf{F}'^{-1}_N = \mathbf{F}'^{\mathsf{H}}_N.$$

The definition does not have further impact on the operation of OFDM or other mathematical coherence in this work, but in order to coincide with recent publications covering UW-OFDM, this work uses the definition as in (2.5), such that

$$\mathbf{F}_N^{-1} = \frac{1}{N}\mathbf{F}_N^{\mathsf{H}}. \tag{2.7}$$

With these tools the following explanation of the classical OFDM using cyclic prefixes is well supported.

After assembly of the OFDM symbol according to (2.4), it is transmitted over a dispersive propagation channel. This channel is modeled as a multipath channel, which in turn is realized as a tapped delay line with the channel impulse response (CIR) as its coefficients. The noise introduced at the receiver is usually modeled as additive white Gaussian noise (AWGN). Thus, the channel propagation of a signal $x[k]$ resulting in $y[k]$ can be written as

$$\begin{aligned} y[k] &= x[k] * h[k] + n[k] \\ &= \sum_{n=0}^{\infty} h[n]x[k-n] + n[k] \end{aligned} \tag{2.8}$$

and visualized as in Figure 2.1, with the CIR $h[k]$, the sequence of additive white Gaussian noise $n[k]$ and the linear convolution operation '$*$'.

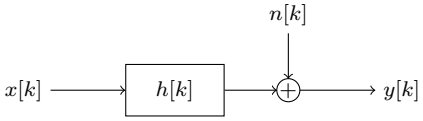


Figure 2.1.: Model of channel propagation.

2.1. Orthogonal Frequency Division Multiplexing with Cyclic Prefixing

To accommodate the intention to tackle the effect of multipath propagation and keep individual transmit blocks (from here OFDM symbols) free of inter-symbol interference (ISI), a *guard interval* is introduced between them. The duration of the guard interval of N_G samples should be defined at system design and adapted to the propagation scenario: The channel impulse response should not outlast the guard interval, as shown in Figure 2.2. A design like this ensures that the transient of the current OFDM symbol caused by the CIR and shown as the decaying curve in the guard interval of Figure 2.2 does never affect the next symbol, thus avoiding ISI.

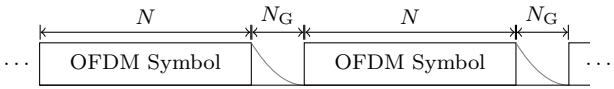


Figure 2.2.: Insertion of a guard interval between OFDM symbols.

Since the only purpose of the guard interval until now is to separate the OFDM symbols to avoid ISI, the transmitter can transmit anything during that time. Although transmitting nothing in the guard interval preserves transmit energy, it makes sense to insert a non-zero block, which is going to be reasoned in the following part.

Convolution Theorem A very important property of the (continuous) Fourier transform is that it transforms the *linear* convolution in time domain into a multiplication in frequency domain. However, switching to a discrete time signal and employing the DFT[1], the representative of the multiplication in frequency domain is the *cyclic* convolution in time domain [GRS01]

$$y[k] = x_1[k] \overset{N}{\circledast} x_2[k] \quad \circ\!\!-\!\!\bullet \quad Y[n] = X_1[n] \cdot X_2[n], \tag{2.9}$$

[1] In this section, a sequence denoted with an uppercase letter shall represent the DFT domain version of the time domain sequence denoted in lowercase, for instance $y[k] \circ\!\!-\!\!\bullet Y[n]$.

relating to sequences of length N. The time discrete cyclic convolution of length N is defined as

$$\begin{aligned} y[k] &= x_1[k] \overset{N}{\circledast} x_2[k] \\ &= \sum_{n=0}^{N-1} x_1[n] x_2[(k-n) \bmod N], \\ k &= 0, 1, \ldots, N-1. \end{aligned} \tag{2.10}$$

Cyclic Prefix In classical OFDM, the guard interval preceding each OFDM symbol is filled with a copy of the N_G last values of the symbol in time domain, as shown in Figure 2.3.

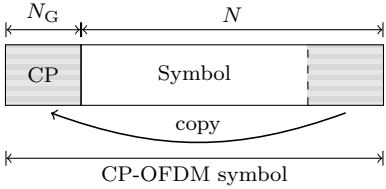


Figure 2.3.: Construction of an OFDM symbol using a cyclic prefix.

With the cyclic convolution theorem (2.9) the use of the cyclic prefix is understood clearly: By cutting off the guard interval at the receiver, all the influence of the preceding OFDM symbol is removed and thus, ISI is avoided. Additionally, when observing the remaining length N OFDM symbol, a *cyclic* convolution can be used to describe the dispersive channel propagation effect on the OFDM symbol, when being observed for the DFT duration N only (Figure 2.3)

$$y[k] = x[k] \overset{N}{\circledast} h[k] + n[k], \quad k = 0, 1, \ldots, N-1. \tag{2.11}$$

Here, the signals $x[k]$ and $y[k]$ are the transmitted and received signals, $h[k]$ is the CIR and $n[k]$ is complex additive white Gaussian noise with zero mean and a variance of σ_n^2.

The convolution theorem justifies to express the propagation also in frequency domain by

$$Y[n] = X[n]H[n] + N[n], \quad n = 0, 1, \ldots, N-1. \tag{2.12}$$

With the transmit symbol vector $\mathbf{d}$, its time domain representation $\mathbf{x}$ given by (2.4) and additive white Gaussian noise $\mathbf{n}'$ in time domain and $\mathbf{n}$ in frequency domain, this can be expressed in vector notation by

$$\begin{aligned} \mathbf{y} &= \widetilde{\mathbf{H}} \cdot \mathbf{F}_N \mathbf{x} + \mathbf{F}_N \mathbf{n}' \\ &= \widetilde{\mathbf{H}} \mathbf{d} + \mathbf{n}, \end{aligned} \tag{2.13}$$

introducing the diagonal matrix $\widetilde{\mathbf{H}}$ with the sampled channel frequency response $\mathbf{F}_N\mathbf{h}$ on its main diagonal[2]. While the time domain noise vector is a realization of complex white Gaussian noise[3] and its statistics are $\mathbf{n}' \sim \mathcal{N}_\mathrm{C}(\mathbf{0}; \sigma_n^2\mathbf{I})$, the statistics of its frequency domain representation $\mathbf{n} = \mathbf{F}_N\mathbf{n}'$ need to be derived. The assumption of $\mathbf{n}'$ to be white and having a diagonal covariance matrix is valid, as optimal band-limited analog signal processing can be assumed. After a linear transformation $\mathbf{y} = \mathbf{A}\mathbf{x}$ of a Gaussian distributed vector $\mathbf{x} \sim \mathcal{N}_\mathrm{C}(\boldsymbol{\mu}_x; \mathbf{C}_{xx})$ with a full rank matrix $\mathbf{A}$ the resulting mean vector $\boldsymbol{\mu}_y$ and covariance matrix $\mathbf{C}_{yy}$ can be determined as [Kay93]:

$$\begin{aligned} \boldsymbol{\mu}_y &= \mathrm{E}\{\mathbf{y}\} = \mathrm{E}\{\mathbf{A}\mathbf{x}\} = \mathbf{A}\mathrm{E}\{\mathbf{x}\} \\ &= \mathbf{A}\boldsymbol{\mu}_x, \\ \mathbf{C}_{yy} &= \mathrm{E}\left\{\mathbf{y}\mathbf{y}^\mathsf{H}\right\} = \mathrm{E}\left\{\mathbf{A}\mathbf{x}\mathbf{x}^\mathsf{H}\mathbf{A}^\mathsf{H}\right\} = \mathbf{A}\mathrm{E}\left\{\mathbf{x}\mathbf{x}^\mathsf{H}\right\}\mathbf{A}^\mathsf{H} \\ &= \mathbf{A}\mathbf{C}_{xx}\mathbf{A}^\mathsf{H}. \end{aligned} \tag{2.14}$$

Obviously the zero-mean property holds for $\mathbf{n}$, and as $\sigma_n^2\mathbf{F}_N\mathbf{I}\mathbf{F}_N^\mathsf{H} = \sigma_n^2 N\mathbf{I}$, the frequency domain noise vector $\mathbf{n}$ can be considered as the realization of a multivariate Gaussian random variable $\mathbf{n} \sim \mathcal{N}_\mathrm{C}(\mathbf{0}; N\sigma_n^2\mathbf{I})$.

For the receive vector as in (2.13), the most obvious receiver is the zero-forcing receiver that simply inverts the channel influence by

$$\tilde{\mathbf{d}} = \widetilde{\mathbf{H}}^{-1}\mathbf{y}, \tag{2.15}$$

which is also the best zero forcing receiver for the here described OFDM system using cyclic prefixes, or in short CP-OFDM.

Using a CP in the guard interval creates these nice mathematical circumstances, but besides this it is a waste of energy and bandwidth efficiency, as the CP is a dispensable copy of the random data part. The problem of the randomness and the dispensability of the guard interval content, while keeping the necessary mathematical properties and offering a lot more possibilities, is dealt with a method that is introduced in the next section.

2.2. The UW-OFDM Symbol Generation and System Model

Let $\mathbf{x}_\mathrm{u} \in \mathbb{C}^{N_u \times 1}$ be a predefined sequence of length N_u that is called unique word. This unique word shall form the tail of each OFDM time domain symbol of total length N and occupy the guard interval, as shown in Figure 2.4.

Hence, an OFDM time domain symbol vector, as the result of a length-N IDFT, consists of two parts and is of the form

$$\mathbf{x}' = \begin{bmatrix} \mathbf{x}_\mathrm{p} \\ \mathbf{x}_\mathrm{u} \end{bmatrix}, \tag{2.16}$$

[2] $\mathbf{h}$ is the channel impulse response with a duration not longer than the guard interval N_G, zero padded to the DFT length N

[3] The short notation of a normally distributed complex random vector $\mathbf{n}$ with mean vector $\boldsymbol{\mu}_n$ and covariance matrix $\mathbf{C}_{nn}$ is $\mathbf{n} \sim \mathcal{N}_\mathrm{C}(\boldsymbol{\mu}_n; \mathbf{C}_{nn})$.

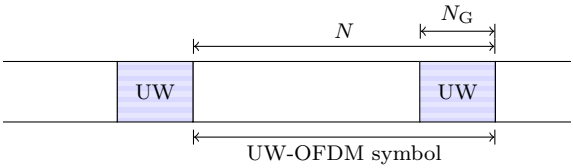


Figure 2.4.: Structure of an OFDM symbol using a unique word.

whereas $\mathbf{x}_{\mathrm{u}} \in \mathbb{C}^{N_u \times 1}$ is the unique word and only $\mathbf{x}_{\mathrm{p}} \in \mathbb{C}^{(N-N_u)\times 1}$ is random and affected by the data, the only part holding payload. In the concept suggested in [HHH10c] an OFDM symbol

$$\mathbf{x} = \begin{bmatrix} \mathbf{x}_{\mathrm{p}} \\ \mathbf{0} \end{bmatrix} \tag{2.17}$$

with a zero UW is generated in a first step. The final transmit symbol is determined by adding the desired UW in time domain in a second step

$$\mathbf{x}' = \mathbf{x} + \begin{bmatrix} \mathbf{0} \\ \mathbf{x}_{\mathrm{u}} \end{bmatrix}, \tag{2.18}$$

as it is visualized in Figure 2.5. The reason for this two-step approach instead of a direct generation of the UW is due to energy reasons and explained in Section 2.3.

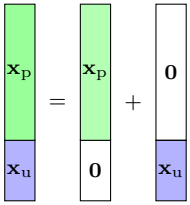


Figure 2.5.: Two-step UW-OFDM symbol generation.

As in conventional OFDM, the data symbols are from a symbol alphabet $\mathcal{A}$, usually QAM (quadrature amplitude modulation), PSK (phase shift keying) or ASK (amplitude shift keying) constellations, and put into the vector $\mathbf{d} \in \mathcal{A}^{N_d \times 1}$. This data vector shall be specified in frequency domain as part of a vector $\bar{\mathbf{x}}$, but here in addition the zero-word is specified in time domain as part of the vector $\mathbf{x} = \mathbf{F}_N^{-1}\bar{\mathbf{x}}$. As N_u values are preset in the time domain vector $\mathbf{x}$, at least the same number of data subcarriers has to be spent in frequency domain from $\bar{\mathbf{x}}$, in order to introduce some kind of redundancy. At first, the values on the redundant subcarriers are gathered in a vector $\mathbf{r}$ of length N_r (later in Section 2.4, another UW-OFDM symbol generation method is presented that lifts the distinction of data/redundant subcarriers again). Then an OFDM symbol is formed in frequency domain by

$$\bar{\mathbf{x}} = \mathbf{B}\mathbf{P}\begin{bmatrix} \mathbf{d} \\ \mathbf{r} \end{bmatrix}, \tag{2.19}$$

introducing a permutation matrix $\mathbf{P} \in \{0,1\}^{(N_d+N_r)\times(N_d+N_r)}$, which places the data and redundant values on their dedicated subcarrier positions. The position of these subcarriers in the transmission bandwidth is very important and will be explained in more detail in Section 2.3. Although the solution of the linear system of equations requires $N_r \geq N_u$, this work only deals with the case

$$N_r = N_u. \tag{2.20}$$

An analysis of the usage of more redundant subcarriers $N_r > N_u$ is given in [HHH10b]. Furthermore, a subcarrier selection matrix $\mathbf{B} \in \{0,1\}^{N\times(N_d+N_r)}$ is introduced to insert unused zero subcarriers, usually at the band edges and the DC subcarrier. This is done in many standards that employ OFDM in order to control the out-of-band radiation and obey spectral masks. The matrix $\mathbf{B}$ is constructed by starting with an identity matrix of size $(N_d + N_r)$ and adding zero rows at the subcarrier positions that are supposed to remain unused.

The time–frequency relation $\mathbf{x} = \mathbf{F}_N^{-1}\bar{\mathbf{x}}$ can now be written as

$$\begin{aligned} \mathbf{x} &= \overbrace{\mathbf{F}_N^{-1}\mathbf{B}\mathbf{P}}^{\mathbf{M}} \begin{bmatrix}\mathbf{d}\\ \mathbf{r}\end{bmatrix} \\ &\overset{\text{def.}}{=} \begin{bmatrix}\mathbf{M}_{11} & \mathbf{M}_{12}\\ \mathbf{M}_{21} & \mathbf{M}_{22}\end{bmatrix} \begin{bmatrix}\mathbf{d}\\ \mathbf{r}\end{bmatrix} = \begin{bmatrix}\mathbf{x}_\mathrm{p}\\ \mathbf{0}\end{bmatrix} \end{aligned} \tag{2.21}$$

which is visualized in Figure 2.6 by detailing in- and output of the IDFT operation.

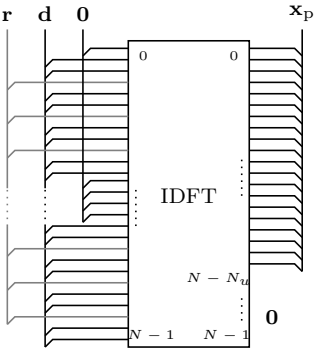


Figure 2.6.: Time and frequency domain partitioning at the transmitter IDFT.

With the appropriately sized sub-matrices $\mathbf{M}_{kl}$ of $\mathbf{M}$ in (2.21), it follows that

$$\begin{aligned} \mathbf{0} &= \mathbf{M}_{21}\mathbf{d} + \mathbf{M}_{22}\mathbf{r}, \\ \mathbf{r} &= -\mathbf{M}_{22}^{-1}\mathbf{M}_{21}\mathbf{d}. \end{aligned} \tag{2.22}$$

With the matrix

$$\mathbf{T} = -\mathbf{M}_{22}^{-1}\mathbf{M}_{21} \in \mathbb{C}^{N_r\times N_d}, \tag{2.23}$$

the vector of redundant subcarrier symbols can be determined by the linear mapping

$$\mathbf{r} = \mathbf{T}\mathbf{d}. \tag{2.24}$$

This presentation makes it clear why the values in $\mathbf{r}$ are called the values on the redundant subcarriers or redundant values, as they solely depend on the data, and hence introduce redundancy that can be exploited at the receiver. The construction of $\mathbf{T}$ and accordingly also the energy of the redundant subcarrier symbols highly depend on the choice of $\mathbf{P}$, see Section 2.3 for details on the energy of a UW-OFDM symbol and for optimization methods.

With (2.24) the assembled UW-OFDM symbol in frequency domain is

$$\bar{\mathbf{x}} = \mathbf{B}\mathbf{P}\begin{bmatrix}\mathbf{d}\\ \mathbf{r}\end{bmatrix} = \mathbf{B}\mathbf{P}\begin{bmatrix}\mathbf{I}\\ \mathbf{T}\end{bmatrix}\mathbf{d} = \mathbf{B}\mathbf{G}\mathbf{d}. \tag{2.25}$$

The matrix in (2.25)

$$\mathbf{G} = \mathbf{P}\begin{bmatrix}\mathbf{I}\\ \mathbf{T}\end{bmatrix}, \quad \mathbf{G} \in \mathbb{C}^{(N_d+N_r)\times N_d} \tag{2.26}$$

can be considered as a UW-OFDM generator matrix, which allows to rewrite the transmit symbol in time domain as

$$\mathbf{x}' = \mathbf{F}_N^{-1}\mathbf{B}\mathbf{G}\mathbf{d} + \begin{bmatrix}\mathbf{0}\\ \mathbf{x}_\mathrm{u}\end{bmatrix}, \tag{2.27}$$

Identifying the frequency domain influence of the UW part as $\bar{\mathbf{x}}_\mathrm{u} = \mathbf{F}_N\begin{bmatrix}\mathbf{0}\\ \mathbf{x}_\mathrm{u}\end{bmatrix}$, this becomes

$$\mathbf{x}' = \mathbf{F}_N^{-1}(\mathbf{B}\mathbf{G}\mathbf{d} + \bar{\mathbf{x}}_\mathrm{u}). \tag{2.28}$$

After transmission over a dispersive channel, a received OFDM time domain symbol can be modeled as

$$\mathbf{y}_r = \mathbf{H}_c\mathbf{x}' + \mathbf{n}' \tag{2.29}$$

$$= \mathbf{H}_c\mathbf{F}_N^{-1}(\mathbf{B}\mathbf{G}\mathbf{d} + \bar{\mathbf{x}}_\mathrm{u}) + \mathbf{n}', \tag{2.30}$$

where $\mathbf{n}' \sim \mathcal{N}_\mathrm{C}\left(\mathbf{0}; \sigma_n^2\mathbf{I}\right)$ represents a complex Gaussian noise vector of length N, which is white and zero-mean analogous to (2.14), and $\mathbf{H}_c \in \mathbb{C}^{N\times N}$ denotes a cyclic convolution matrix with the zero-padded CIR vector $\mathbf{h} \in \mathbb{C}^{N\times 1}$ in its first column. It is not directly evident to model the propagation using a *cyclic* convolution instead of the linear convolution in the same way as for CP-OFDM, but it becomes clear due to the repeating unique words. This is shown in detail in Appendix B. After applying the DFT at the receiver the zero subcarriers are excluded from further operation, which leads to the down-sized vector of length $(N_d + N_r)$

$$\mathbf{y}_d = \mathbf{B}^\mathsf{T}\mathbf{F}_N\mathbf{y}_r \tag{2.31}$$

$$= \mathbf{B}^\mathsf{T}\mathbf{F}_N\mathbf{H}_c\mathbf{F}_N^{-1}(\mathbf{B}\mathbf{G}\mathbf{d} + \bar{\mathbf{x}}_\mathrm{u}) + \mathbf{B}^\mathsf{T}\mathbf{F}_N\mathbf{n}'. \tag{2.32}$$

As any circulant matrix is diagonalized by the DFT matrix, the matrix $\mathbf{F}_N\mathbf{H}_c\mathbf{F}_N^{-1}$ is diagonal and contains the sampled channel frequency response on its main diagonal. Without altering $\mathbf{y}_d$, it is allowed to replace $\bar{\mathbf{x}}_\mathrm{u}$ in (2.32) by $\mathbf{B}\mathbf{B}^\mathsf{T}\bar{\mathbf{x}}_\mathrm{u}$. This operation first removes possibly non-zero values on the zero subcarrier positions from $\bar{\mathbf{x}}_\mathrm{u}$ ($\mathbf{B}^\mathsf{T}$), to insert the zeros again ($\mathbf{B}$). Although some frequency components of the unique word are lost in this procedure, the operations are allowed as $\mathbf{F}_N\mathbf{H}_c\mathbf{F}_N^{-1}$ is diagonal and due to the final removal of the zero subcarriers. With this step, the down-sized vector becomes

$$\mathbf{y}_d = \mathbf{B}^\mathsf{T}\mathbf{F}_N\mathbf{H}_c\mathbf{F}_N^{-1}\mathbf{B}(\mathbf{G}\mathbf{d} + \mathbf{B}^\mathsf{T}\bar{\mathbf{x}}_\mathrm{u}) + \mathbf{B}^\mathsf{T}\mathbf{F}_N\mathbf{n}'. \tag{2.33}$$

Furthermore, excluding the entries from the diagonal $\mathbf{F}_N\mathbf{H}_c\mathbf{F}_N^{-1}$ that correspond to the zero subcarriers yields

$$\begin{aligned} \widetilde{\mathbf{H}} &= \mathbf{B}^\mathsf{T}\mathbf{F}_N\mathbf{H}_c\mathbf{F}_N^{-1}\mathbf{B}, \\ \text{where } \widetilde{\mathbf{H}} &\in \mathbb{C}^{(N_d+N_r)\times(N_d+N_r)}, \end{aligned} \tag{2.34}$$

which yields the receive vector in frequency domain, written in the form of the *affine* model

$$\mathbf{y}_d = \widetilde{\mathbf{H}}\mathbf{G}\mathbf{d} + \widetilde{\mathbf{H}}\mathbf{B}^\mathsf{T}\bar{\mathbf{x}}_\mathrm{u} + \underbrace{\mathbf{B}^\mathsf{T}\mathbf{F}_N\mathbf{n}'}_{\mathbf{n}}. \tag{2.35}$$

Note that the channel characteristics and thus $\widetilde{\mathbf{H}}$ are unknown at the receiver at first, which are usually estimated. However, in this work perfect channel knowledge is assumed at any time. $\widetilde{\mathbf{H}}\mathbf{B}^\mathsf{T}\bar{\mathbf{x}}_\mathrm{u}$ represents the known portion contained in the received vector $\mathbf{y}_d$ originating from the UW. Therefore, as the next preparatory step for data estimation, the UW influence is subtracted

$$\mathbf{y} = \mathbf{y}_d - \widetilde{\mathbf{H}}\mathbf{B}^\mathsf{T}\bar{\mathbf{x}}_\mathrm{u} \tag{2.36}$$

to obtain the corrected OFDM symbol. Removing the constant offset due to the UW from (2.35) produces the *linear* model form of UW-OFDM transmission

$$\mathbf{y} = \underbrace{\widetilde{\mathbf{H}}\mathbf{G}}_{\mathbf{H}}\mathbf{d} + \mathbf{n}, \tag{2.37}$$

with the complex noise vector $\mathbf{n} \sim \mathcal{N}_\mathrm{C}\left(\mathbf{0}; N\sigma_n^2\mathbf{I}\right)$ of length $(N_d{+}N_r)$.

The model in (2.37) constitutes the starting point, from which the data symbols shall be estimated.

Link to Reed-Solomon Codes The matrix $\mathbf{G}$ in (2.25) can be interpreted as the code generator matrix for a systematic complex valued block code that generates the code words $\mathbf{c} = \mathbf{G}\mathbf{d}$. Furthermore, in time domain this code has defined N_u consecutive zeros. This shows many parallels to the definition of a Reed-Solomon (RS) code used for channel coding, which is traditionally defined on a Galois field, instead of the field of complex numbers and has consecutive zeros defined in frequency domain [Bla03]. Since UW-OFDM time domain symbols contain a block of fixed samples, the set of all corresponding vectors in discrete frequency domain forms a coset to a Reed Solomon code. Usually RS codes of length N are defined for a finite field $\mathbb{F}_Q^N$ using an element $w \in \mathbb{F}_Q$ of order N, $N \cdot l = Q-1$,

with $N, l, Q \in \mathbb{N}$ to define a discrete Fourier transform $\mathbb{F}_Q^N \to \mathbb{F}_Q^N$ in $\mathbb{F}_Q$. The set of code words is specified by the fact that the (inverse) DFT of all code words contains a block of $d_{\min} - 1$ successive zeros, where $d_{\min}$ is the minimum Hamming distance of the RS code. If this block of $d_{\min} - 1$ successive symbols differs from zero, but is also fixed for all code words, a coset code to an RS code is generated in the other domain with respect to this Fourier transform with the same minimum distance $d_{\min}$ (see [Bla03]). In contrast to the usual approach using RS codes over a finite field, for example $\mathbb{F}_{2^8}$, for an outer code in a concatenated code scheme, in UW-OFDM there is an inner RS code over the field of complex numbers. If any further channel coding scheme is applied, the OFDM guard interval is here additionally exploited for redundancy of an inner channel coding scheme in a natural way. All these parallels to RS codes suggest algebraic decoding. In [HHH10a, HHK09] it is shown that this leads to an ill-conditioned system of equations that is very sensitive to noise. Hence, other methods, like those introduced in this work, need to be applied.

2.3. UW-OFDM Symbol Energies

Up to now the UW-OFDM symbols were generated by a *two-step* approach, that added the unique word in a second step, after generating zeros at the UW position. An approach yielding much higher OFDM symbol energies is to generate the UW at the IDFT output in a *direct* way. The mean energy of a UW-OFDM symbol has direct impact on the performance on the whole communication chain. This energy does not only depend on the data symbols in $\mathbf{d}$, but is also affected by the redundant values in $\mathbf{r}$, which in turn depend on the data, the positions of the redundant subcarriers within the system bandwidth and most importantly the symbol generation method. Therefore, in this section, the mean OFDM symbol energies are derived analytically for the two-step approach and the briefly introduced direct approach.

2.3.1. Symbol Energy for Two-Step Approach

The mean energy of an UW-OFDM symbol when averaging over all possible data vectors, assembled according to (2.27), is given by [OH10]

$$\begin{aligned} E_x &= \mathrm{E}\left\{\mathbf{x}'^{\mathsf{H}}\mathbf{x}'\right\} = \mathrm{E}\left\{\mathbf{x}^{\mathsf{H}}\mathbf{x}\right\} + \mathbf{x}_{\mathrm{u}}^{\mathsf{H}}\mathbf{x}_{\mathrm{u}} \\ &= \frac{1}{N}\mathrm{E}\left\{\bar{\mathbf{x}}^{\mathsf{H}}\bar{\mathbf{x}}\right\} + \mathbf{x}_{\mathrm{u}}^{\mathsf{H}}\mathbf{x}_{\mathrm{u}} \\ &= \frac{1}{N}\mathrm{E}\left\{\begin{bmatrix}\mathbf{d}^{\mathsf{H}} & \mathbf{r}^{\mathsf{H}}\end{bmatrix}\begin{bmatrix}\mathbf{d}\\ \mathbf{r}\end{bmatrix}\right\} + \mathbf{x}_{\mathrm{u}}^{\mathsf{H}}\mathbf{x}_{\mathrm{u}} \\ &= \underbrace{\frac{1}{N}\mathrm{E}\left\{\mathbf{d}^{\mathsf{H}}\mathbf{d}\right\}}_{E_d} + \underbrace{\frac{1}{N}\mathrm{E}\left\{\mathbf{r}^{\mathsf{H}}\mathbf{r}\right\}}_{E_r} + \underbrace{\mathbf{x}_{\mathrm{u}}^{\mathsf{H}}\mathbf{x}_{\mathrm{u}}}_{E_u}. \end{aligned} \tag{2.38}$$

The data symbols are assumed to be uncorrelated and from an alphabet $\mathcal{A}$ with zero mean and variance σ_d^2, which provides

$$E_d = \frac{N_d \sigma_d^2}{N}. \tag{2.39}$$

The trace operation $\mathrm{tr}\,(\cdot)$[4] and (2.24) help to rewrite the energy of the redundant subcarriers as

$$\begin{aligned} E_r &= \frac{1}{N}\mathrm{E}\left\{\mathbf{r}^{\mathsf{H}}\mathbf{r}\right\} = \frac{1}{N}\mathrm{E}\left\{\mathrm{tr}\left(\mathbf{r}\mathbf{r}^{\mathsf{H}}\right)\right\} \\ &= \frac{1}{N}\mathrm{tr}\left(\mathrm{E}\left\{\mathbf{r}\mathbf{r}^{\mathsf{H}}\right\}\right) = \frac{1}{N}\mathrm{tr}\left(\mathrm{E}\left\{\mathbf{T}\mathbf{d}\mathbf{d}^{\mathsf{H}}\mathbf{T}^{\mathsf{H}}\right\}\right) \\ &= \frac{1}{N}\mathrm{tr}\left(\mathbf{T}\mathrm{E}\left\{\mathbf{d}\mathbf{d}^{\mathsf{H}}\right\}\mathbf{T}^{\mathsf{H}}\right) \\ &= \frac{\sigma_d^2}{N}\mathrm{tr}\left(\mathbf{T}\mathbf{T}^{\mathsf{H}}\right). \end{aligned} \tag{2.40}$$

Hence, the average amount of energy present on the redundant subcarriers depends solely on $\mathbf{T}$, which is only influenced by the number and positions of the redundant subcarriers and thus by the matrix $\mathbf{P}$.

To conclude, the overall mean energy of UW-OFDM symbols generated following the two-step approach becomes

$$E_x = \underbrace{\frac{N_d\sigma_d^2}{N}}_{E_d} + \underbrace{\frac{\sigma_d^2}{N}\mathrm{tr}\left(\mathbf{T}\mathbf{T}^{\mathsf{H}}\right)}_{E_r} + \underbrace{\mathbf{x}_{\mathrm{u}}^{\mathsf{H}}\mathbf{x}_{\mathrm{u}}}_{E_u}. \tag{2.41}$$

It is possible to identify the energies descriptively as related to the data E_d, the redundant subcarriers E_r and the unique word E_u.

2.3.2. Symbol Energy for Direct Approach

In contrast to the two-step approach (2.27), in a direct approach the UW can be generated at the output of the IDFT directly, skipping the addition (2.18). This requires different vectors and matrices, which are denoted with a superscript (d) (direct approach) in this section. The initial relation (2.21) has to be written as

$$\mathbf{x}^{(\mathrm{d})} = \mathbf{F}_N^{-1}\mathbf{B}\mathbf{P}\begin{bmatrix}\mathbf{d}\\ \mathbf{r}^{(\mathrm{d})}\end{bmatrix} = \begin{bmatrix}\mathbf{x}_{\mathrm{p}}^{(\mathrm{d})}\\ \mathbf{x}_{\mathrm{u}}\end{bmatrix}. \tag{2.42}$$

With the same definition of $\mathbf{M}$ as in (2.21), the derivation becomes $\mathbf{x}_{\mathrm{u}} = \mathbf{M}_{21}\mathbf{d} + \mathbf{M}_{22}\mathbf{r}^{(\mathrm{d})}$ and further

$$\begin{aligned} \mathbf{r}^{(\mathrm{d})} &= \mathbf{M}_{22}^{-1}\mathbf{x}_{\mathrm{u}} - \mathbf{M}_{22}^{-1}\mathbf{M}_{21}\mathbf{d} \\ &= \mathbf{M}_{22}^{-1}\mathbf{x}_{\mathrm{u}} + \mathbf{T}\mathbf{d}, \end{aligned} \tag{2.43}$$

which finally leads to

$$\begin{aligned} \bar{\mathbf{x}}^{(\mathrm{d})} &= \mathbf{B}\mathbf{P}\begin{bmatrix}\mathbf{I}\\ \mathbf{T}\end{bmatrix}\mathbf{d} + \mathbf{B}\mathbf{P}\begin{bmatrix}\mathbf{0}\\ \mathbf{M}_{22}^{-1}\end{bmatrix}\mathbf{x}_{\mathrm{u}} \\ &= \mathbf{B}\mathbf{G}\mathbf{d} + \mathbf{B}\mathbf{P}\begin{bmatrix}\mathbf{0}\\ \mathbf{M}_{22}^{-1}\end{bmatrix}\mathbf{x}_{\mathrm{u}}. \end{aligned} \tag{2.44}$$

[4] The trace of a matrix is the sum of the elements on its main diagonal.

Starting in frequency domain, the mean energy per OFDM symbol can be determined as

$$E_x^{(\mathrm{d})} = \frac{1}{N}\mathrm{E}\left\{\bar{\mathbf{x}}^{(\mathrm{d})\mathsf{H}}\bar{\mathbf{x}}^{(\mathrm{d})}\right\} = \frac{1}{N}\mathrm{E}\left\{\mathbf{d}^{\mathsf{H}}\mathbf{d}\right\} + \frac{1}{N}\mathrm{E}\left\{\mathbf{r}^{(\mathrm{d})\mathsf{H}}\mathbf{r}^{(\mathrm{d})}\right\}. \tag{2.45}$$

Applying (2.43) expands the last term further to

$$\begin{aligned}\mathrm{E}\left\{\mathbf{r}^{(\mathrm{d})\mathsf{H}}\mathbf{r}^{(\mathrm{d})}\right\} &= \mathrm{E}\left\{\mathrm{tr}\left(\mathbf{r}^{(\mathrm{d})}\mathbf{r}^{(\mathrm{d})\mathsf{H}}\right)\right\} = \mathrm{tr}\left(\mathrm{E}\left\{\mathbf{r}^{(\mathrm{d})}\mathbf{r}^{(\mathrm{d})\mathsf{H}}\right\}\right)\\ &= \mathrm{tr}\left(\mathrm{E}\left\{\left(\mathbf{T}\mathbf{d}+\mathbf{M}_{22}^{-1}\mathbf{x}_{\mathrm{u}}\right)\left(\mathbf{T}\mathbf{d}+\mathbf{M}_{22}^{-1}\mathbf{x}_{\mathrm{u}}\right)^{\mathsf{H}}\right\}\right)\\ &= \mathrm{tr}\left(\mathrm{E}\left\{\mathbf{T}\mathbf{d}\mathbf{d}^{\mathsf{H}}\mathbf{T}^{\mathsf{H}} + \mathbf{M}_{22}^{-1}\mathbf{x}_{\mathrm{u}}\mathbf{d}^{\mathsf{H}}\mathbf{T}^{\mathsf{H}} + \mathbf{T}\mathbf{d}\mathbf{x}_{\mathrm{u}}^{\mathsf{H}}\mathbf{M}_{22}^{-\mathsf{H}} + \mathbf{M}_{22}^{-1}\mathbf{x}_{\mathrm{u}}\mathbf{x}_{\mathrm{u}}^{\mathsf{H}}\mathbf{M}_{22}^{-\mathsf{H}}\right\}\right).\end{aligned}$$

As $\mathbf{x}_{\mathrm{u}}$ is deterministic and $\mathbf{d}$ has zero mean, the mixed terms vanish and the average redundant energy is[5]

$$\mathrm{E}\left\{\mathbf{r}^{(\mathrm{d})\mathsf{H}}\mathbf{r}^{(\mathrm{d})}\right\} = \sigma_d^2\mathrm{tr}\left(\mathbf{T}\mathbf{T}^{\mathsf{H}}\right) + \mathrm{tr}\left(\mathbf{M}_{22}^{-1}\mathbf{x}_{\mathrm{u}}\mathbf{x}_{\mathrm{u}}^{\mathsf{H}}\mathbf{M}_{22}^{-\mathsf{H}}\right). \tag{2.46}$$

Putting everything together produces

$$E_x^{(\mathrm{d})} = \underbrace{\frac{N_d\sigma_d^2}{N}}_{E_d} + \underbrace{\frac{\sigma_d^2}{N}\mathrm{tr}\left(\mathbf{T}\mathbf{T}^{\mathsf{H}}\right)}_{E_r} + \underbrace{\frac{1}{N}\mathbf{x}_{\mathrm{u}}^{\mathsf{H}}\mathbf{M}_{22}^{-\mathsf{H}}\mathbf{M}_{22}^{-1}\mathbf{x}_{\mathrm{u}}}_{E_u^{(\mathrm{d})}}, \tag{2.47}$$

which only differs from (2.41) in $E_u^{(\mathrm{d})}$. This small difference has a big impact on the final mean OFDM symbol energy and is quantified with exemplary sequences in the next section. Furthermore, it can be shown that the inequality

$$\frac{1}{N}\mathbf{x}_{\mathrm{u}}^{\mathsf{H}}\mathbf{M}_{22}^{-\mathsf{H}}\mathbf{M}_{22}^{-1}\mathbf{x}_{\mathrm{u}} \geq \mathbf{x}_{\mathrm{u}}^{\mathsf{H}}\mathbf{x}_{\mathrm{u}} \tag{2.48}$$

always holds, which is shown in detail in Appendix C.

The main differences between direct and two-step approach can be summarized in two points:

- The direct approach yields higher or equal mean energy on the redundant subcarriers, compared to the two-step approach.
- In direct approach, the UW affects only the redundant subcarriers spectrally, as seen in (2.44). In the two-step approach, the UW sequence is added afterwards and thus its frequency representation affects the whole spectrum. Thus, in direct approach a subtraction of the UW from the receive vector is not necessary, but the simple channel inversion estimator that is introduced in Section 3.1.3 would still work. However, more sophisticated data estimators are not usable in this case.

[5] As the operation order of the Hermitian transposition and the inversion does not matter $\left(\mathbf{M}_{22}^{\mathsf{H}}\right)^{-1} = \left(\mathbf{M}_{22}^{-1}\right)^{\mathsf{H}}$, it is written as $\mathbf{M}_{22}^{-\mathsf{H}}$.

2.3.3. Numerical Examples

With the inequality (2.48) it is obvious that the energy of an OFDM symbol generated by the two-step approach is always equal or less than the energy of a symbol generated using the direct approach. The fact that the UW only affects the redundant subcarriers in the final UW-OFDM symbol $\bar{\mathbf{x}}$ could be exploited as an advantage of the approach, if the energy penalty is rather low. For actual and reasonable unique words the excess energy in fact dominates the OFDM symbol energy budget and makes the direct approach unusable for communications.

In this section, some insight to the dimension of this issue for practical setups is given. Therefore, three common unique word sequences are considered, their energy consumption in the two-step and the direct approach are compared, and finally their impact on the bit error probability (BER – bit error ratio) is shown [OH10].

For the investigation the parameters of the UW-OFDM systems with a DFT length of 64, with 12 zero subcarriers and without zero subcarriers is applied, as introduced as exemplary UW-OFDM system A and B in Appendix A. The unique word is scaled in order to consume 25% of the overall average OFDM symbol energy generated by the two-step approach

$$\frac{\mathbf{x}_\mathrm{u}^\mathsf{H}\mathbf{x}_\mathrm{u}}{E_x} = 0.25, \tag{2.49}$$

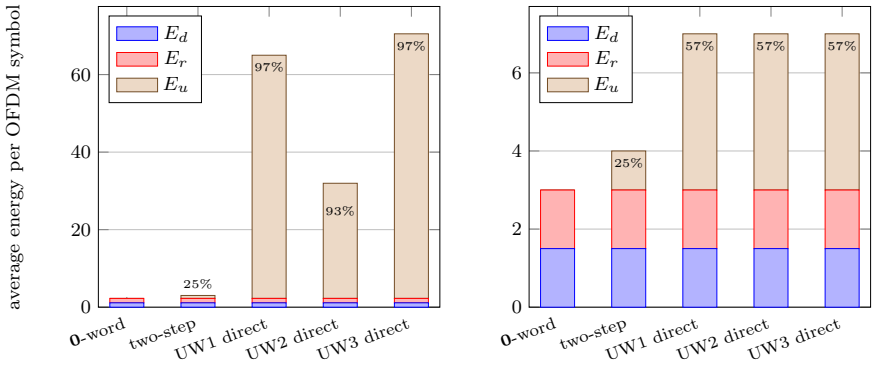
as it also occupies one fourth of the OFDM symbol duration in time domain.

Therefore, the OFDM symbol energies are calculated using the following sequences as unique words:

1. the generalized Barker sequence [GS65] of length 12 padded with zeros to the target length (UW1)
2. a CAZAC sequence (constant amplitude, zero autocorrelation) from [Pop92], as often used for channel estimation, frequency offset estimation and timing synchronization (UW2)
3. the Frank-Zadoff sequence of length 16, which is used as a unique word in [IEE04] and also has CAZAC properties (UW3)

For data symbols from the 4-QAM alphabet, the average energy demand per OFDM symbol when using these sequences as unique words is shown in Figure 2.7, split in the three components data energy E_d, redundant energy E_r and UW generation energy E_u. The latter is either just the energy of the unique word itself for the two-step approach E_u, or the whole term in (2.47), which was identified as the UW related energy $E_u^{(\mathrm{d})}$. Additionally, the UW generation energy in relation to the whole symbol energy E_u/E_x is given on top of each bar.

In both subplots the leftmost bar shows the symbol energy for the zero word as UW. Obviously, the parts E_d and E_r are the same as for all other UWs. The next bar shows the symbol energy for all UWs with the two-step approach, which does not differ between different UWs. The additional surplus due to the UW is 25% of the whole symbol energy budget, as postulated earlier. The three remaining bars show the average energies of the OFDM symbols generated by the direct approach for the chosen UWs. For the setup without zero subcarriers it is observed in Figure 2.7b that more than half of the symbol



(a) system A, with 12 zero subcarriers

(b) system B, without zero subcarriers

Figure 2.7.: Energies of UW-OFDM symbols generated by two-step and direct approach, with different UWs and for UW-OFDM systems according to Table A.1, partitioned in data, redundant and UW related part.

energy is related to the UW. The setup inserting 12 zero subcarriers, displayed in subfigure (a), increases the excess energy demand even more, up to a ratio, where more than 90% of the transmit energy is spent on the unique word's influence on the redundant subcarriers.

The excess energy is more or less wasted. It cannot be utilized for data symbol recovery at all, as the UW is subtracted in the beginning of the receiver processing. This can be perfectly observed, when looking at the bit error performance of UW-OFDM systems with an LMMSE (linear minimum mean square error) receiver, which will be introduced in Section 3.2, and these particular UWs in the AWGN channel. For the discussed five methods the bit error performance is displayed in Figure 2.8 revealing a huge advantage of the two-step approach compared to all direct approach results, that cannot be expected to be diminished by more sophisticated UW-OFDM receivers.

In particular, relating the direct approach with the highest OFDM symbol energy (UW3, Frank-Zadoff sequence) to the two-step approach symbol energy yields

$$\frac{E_x^{(\mathrm{UW3})}}{E_x^{(\text{two-step})}} = \frac{70.51}{3.02} = 23.35 \triangleq 13.7\,\mathrm{dB}. \tag{2.50}$$

These 13.7 dB of energy difference can be observed as gap between the BER curve for the two-step approach and the UW3, direct approach.

When using the zero word both approaches perform equally, as $E_u = 0$. This of course yields the best bit error results, but no UW is present that could be exploited beneficially for estimation, tracking or synchronization.

Due to these findings, the direct approach is dropped and the two-step approach is used throughout this work, instead.

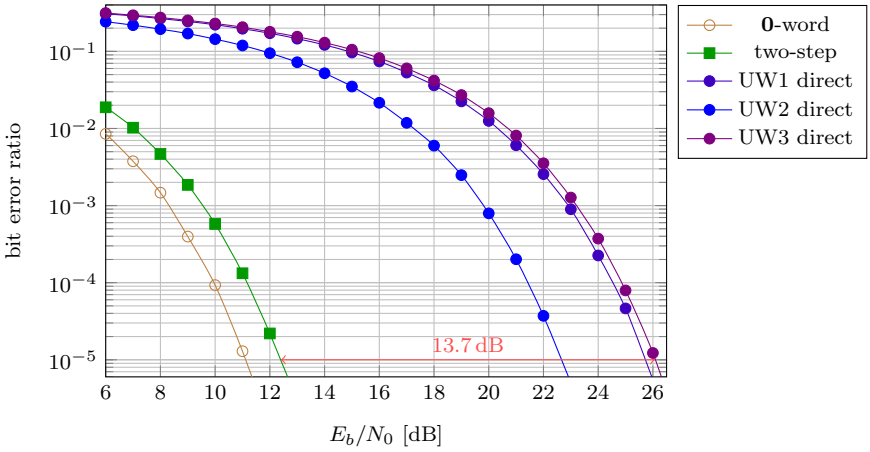


Figure 2.8.: Bit error performance of UW-OFDM system A according to Table A.1 with two-step and direct UW generation and different UWs in the AWGN channel.

2.3.4. Optimization of the Redundant Subcarrier Positions

In previous publications, for example [HHH10c, HOH11], and in Section 2.2 the necessity of finding good positions of the redundant subcarrier symbols and hence a suited permutation matrix $\mathbf{P}$ was stressed. These positions can be determined by an optimization that minimizes a given criterion, which needs to be defined. In [HHOH12] a cost function is used that incorporates the variances of the remaining error values after a supposed LMMSE data estimation at the receiver, which will be introduced in Section 3.2. It was found that minimizing the sum of these variances produces practically the same results, and hence $\mathbf{P}$, as the rather intuitive criterion to minimize the redundant energy

$$E_r = \frac{\sigma_d^2}{N} \mathrm{tr}\left(\mathbf{T}\mathbf{T}^{\mathsf{H}}\right), \tag{2.51}$$

which, for fixed system parameters σ_d^2 and N, permits the cost function

$$J_{E_r}(\mathbf{P}) = \mathrm{tr}\left(\mathbf{T}\mathbf{T}^{\mathsf{H}}\right). \tag{2.52}$$

The matrix $\mathbf{T}$ of course depends on $\mathbf{P}$ according to (2.21) and (2.23).

In [Hof] more optimization criteria are developed. Since it is also shown that the adjusted cost functions yield mostly the same $\mathbf{P}$, the redundant energy criterion considered in this work, is sufficient.

For either optimization goal and reasonable choices of N and N_r, an exhaustive search turns out to be unfeasible. In fact, the energy that has to be spent for the redundant subcarriers depending on their positions has been under research recently [Ste12, Ste13], but

a simple heuristic optimization method turned out to be most successful and convenient. This heuristic is presented in this section and is able to solve this integer valued optimization problem in reasonable computation time [MF04].

Instead of working with a permutation matrix, index sets and index vectors are used in the following. The index sets of the redundant and the data subcarriers are defined as $\mathcal{I}_r$ and $\mathcal{I}_d$ respectively, which have to fulfill

$$\begin{aligned} &\mathcal{I}_r \cup \mathcal{I}_d = \{0, 1, ..., N_r + N_d - 1\}, \\ &\mathcal{I}_r \cap \mathcal{I}_d = \emptyset. \end{aligned} \tag{2.53}$$

Furthermore, the corresponding index vectors $\mathbf{i}_r$ and $\mathbf{i}_d$ are used. It shall be noted that the permutation matrix $\mathbf{P}$ can unambiguously be derived from the sorted index vectors $\mathbf{i}_r$ and $\mathbf{i}_d$, and hence also the cost according to (2.52)

$$J_{E_r}(\mathbf{P}) = J\left(\mathbf{i}_r, \mathbf{i}_d\right). \tag{2.54}$$

In Algorithm 1 and 2 this heuristic is reproduced in pseudo-code. The function HEURISTIC_OPTIMIZATION starts with index vectors $\mathbf{i}_r$ and $\mathbf{i}_d$ that are randomly chosen but valid according to (2.53). Then in the function FIND_BETTER_INDEXSET a better index set is searched for by exchanging one element of $\mathbf{i}_r$ with one element of $\mathbf{i}_d$, such that the cost function decreases by a maximum amount. This function is called repeatedly until at least a local minimum is found. Fortunately, the supposedly global minimum can be found by executing Algorithm 1 only a few times (typically below 10 times for the parameter sets in this work, see Table A.1).

In summary, this algorithm delivers index sets in a local minimum within a few seconds or at most a few minutes (depending on N) when implemented in Matlab and executed on a standard PC system. For some setups, there exists more than one index set minimizing the cost function. The positions of the redundant subcarriers, given in Appendix A, were determined by these algorithms.

Algorithm 1 HEURISTIC_OPTIMIZATION

1: **choose** valid index vectors $\mathbf{i}_r$ and $\mathbf{i}_d$ randomly
2: $\mathbf{i}'_r \leftarrow \mathbf{i}_r, \mathbf{i}'_d \leftarrow \mathbf{i}_d$
3: $J_{\text{new}} \leftarrow \infty$
4: **repeat**
5: $\quad \mathbf{i}_r \leftarrow \mathbf{i}'_r, \mathbf{i}_d \leftarrow \mathbf{i}'_d$
6: $\quad J_{\text{old}} \leftarrow J_{\text{new}}$
7: $\quad (\mathbf{i}'_r, \mathbf{i}'_d) \leftarrow$ FIND_BETTER_INDEXSET$(\mathbf{i}_r, \mathbf{i}_d)$
8: $\quad J_{\text{new}} \leftarrow J\left(\mathbf{i}'_r, \mathbf{i}'_d\right)$
9: **until** $J_{\text{new}} \geq J_{\text{old}}$
10: $J_{\text{best}} \leftarrow J_{\text{old}}$
11: **sort** $\mathbf{i}_r$ and $\mathbf{i}_d$
12: **determine** $\mathbf{P}$ from $\mathbf{i}_r$ and $\mathbf{i}_d$

Algorithm 2 FIND_BETTER_INDEXSET

```
function [i_{r,out}, i_{d,out}] = FIND_BETTER_INDEXSET(i_r, i_d)
    J_best ← J(i_r, i_d)
    i_{r,out} ← i_r
    i_{d,out} ← i_d
    for k = 0, 1, ..., N_r − 1 do
        for l = 0, 1, ..., N_d − 1 do
            i_{r,tmp} ← i_r
            i_{d,tmp} ← i_d
            swap i_{r,tmp}[k] and i_{d,tmp}[l]
            J_new ← J(i_{r,tmp}, i_{d,tmp})
            if J_new < J_best then
                i_{r,out} ← i_{r,tmp}
                i_{d,out} ← i_{d,tmp}
                J_best ← J_new
            end if
        end for
    end for
end function
```

2.4. Non-Systematic UW-OFDM

In the last section the basic UW-OFDM symbol structure was shown, where each subcarrier is dedicated to carry either data or redundant values. In [HHH12] a *non-systematic* generation of UW-OFDM symbols is suggested that lifts off the dedication of the subcarriers to a data or redundant role and improves the system performance. More precisely, an altered generator matrix $\mathbf{G}$ is employed, which disperses the redundancy over all subcarriers in use. As the data is not immediately visible in the UW-OFDM symbol anymore, this approach is called *non-systematic* generation, analogously to the same term in channel coding. The other way round, *systematically* generated UW-OFDM symbols possess dedicated data subcarriers that directly show the information symbols.

Although the $\mathbf{G}$ for non-systematic UW-OFDM, as presented in this section, is different from the $\mathbf{G}$ for systematic UW-OFDM, it is denoted with the same symbol, as most processing in this work is also suited to both UW-OFDM generator matrices. Whenever a distinction between a systematic and non-systematic $\mathbf{G}$ is necessary, it is mentioned explicitly, or it becomes clear from the context.

2.4.1. Generating UW-OFDM Symbols Non-Systematically

In [HHH12] a different generator matrix was assumed to have the form

$$\mathbf{G} = \mathbf{AP}\begin{bmatrix}\mathbf{I}\\ \mathbf{T}\end{bmatrix}, \tag{2.55}$$

including, opposed to (2.26), an altered version of $\mathbf{T}$ and an additional real valued non-singular matrix $\mathbf{A} \in \mathbb{R}^{(N_d+N_r)\times(N_d+N_r)}$, to be found by the minimization of an appropriate system based cost function. Analogous to (2.21), this generator matrix still has to

ensure the zero word at the position of the guard interval

$$\mathbf{F}_N^{-1}\mathbf{BGd} = \begin{bmatrix}\mathbf{x}_\mathrm{p}\\ \mathbf{0}\end{bmatrix}. \tag{2.56}$$

With the dispersion of data and redundant values by $\mathbf{A}$, the frequency domain UW-OFDM symbol in (2.25) becomes

$$\bar{\mathbf{x}} = \mathbf{Bc} = \mathbf{BGd}, \tag{2.57}$$

with a vector $\mathbf{c} \in \mathbb{C}^{(N_d+N_r)\times 1}, \mathbf{c} = \mathbf{Gd}$ gathering all values that depend on the data, taking the place of the permuted vector with data and redundant symbols.

The cost function defined in [HHH12] for finding optimum generator matrices is based on the covariance matrix of the estimation error $\mathbf{C}_{ee}$ after an LMMSE data estimation, as defined in (3.30). The LMMSE topic will be introduced later in Section 3.2. However, the solution to this optimization problem is ambiguous. Particular solutions can be found, for example by applying the steepest descent algorithm. As in [HHH12, Eq. (44)] the initialization can be chosen as

$$\mathbf{A}^{(0)} = \mathbf{I}, \tag{2.58}$$

which implies $\mathbf{T}^{(0)}$ and $\mathbf{G}^{(0)}$ to be chosen as $\mathbf{T}$ and $\mathbf{G}$ for systematic UW-OFDM. The iterative optimization process consequently starts with the code generator matrix $\mathbf{G}$ of the systematic UW-OFDM concept, which can be considered a good initial guess. Note that again $\mathbf{T}$ is unambiguously given by the zero word constraint after the IDFT.

While this approach yields a mainly local dispersion of the energy of each data symbol, a second approach shown in [HHH12] that uses a random initialization of $\mathbf{A}^{(0)}$ shows a global energy dispersion over the whole system bandwidth. For this work only the first case with local energy dispersion and initialization as in (2.58) is considered.

Interestingly, it was found that a matrix $\mathbf{G}$, generated as introduced above, fulfills the property

$$\mathbf{G}^\mathsf{H}\mathbf{G} = \alpha\mathbf{I}, \quad \alpha \in \mathbb{R}, \tag{2.59}$$

which will be helpful when determining the non-systematic UW-OFDM symbol energy in Section 2.4.2.

It was found in [HHH12] that any matrix $\mathbf{G}$ fulfilling the properties (2.56) and (2.59) is a valid generator matrix for non-systematic UW-OFDM.

2.4.2. Energy of Non-Systematically Generated UW-OFDM Symbols

For non-systematically generated UW-OFDM the distinction between data and redundant energy is of course not possible anymore, but the property (2.59) simplifies the determination of the mean OFDM symbol energy a lot:

$$\begin{aligned} E_x &= \mathrm{E}\left\{\mathbf{x}'^\mathsf{H}\mathbf{x}'\right\} = \frac{1}{N}\mathrm{E}\left\{\bar{\mathbf{x}}^\mathsf{H}\bar{\mathbf{x}}\right\} + \mathbf{x}_\mathrm{u}^\mathsf{H}\mathbf{x}_\mathrm{u} \\ &= \frac{1}{N}\mathrm{E}\left\{\mathbf{d}^\mathsf{H}\mathbf{G}^\mathsf{H}\mathbf{Gd}\right\} + \mathbf{x}_\mathrm{u}^\mathsf{H}\mathbf{x}_\mathrm{u} \\ &= \frac{\alpha}{N}\mathrm{E}\left\{\mathbf{d}^\mathsf{H}\mathbf{d}\right\} + \mathbf{x}_\mathrm{u}^\mathsf{H}\mathbf{x}_\mathrm{u} \\ &= \frac{\alpha N_d \sigma_d^2}{N} + E_u. \end{aligned} \tag{2.60}$$

It makes sense to normalize the non-systematic generator matrix $\mathbf{G}$, such that $\alpha = 1$, in order to have OFDM symbol energies independent of the actually used generator matrix.

2.5. Simulation Setup

Linear and nonlinear receiver concepts for Unique Word OFDM are the main contribution of this work and covered in Chapter 3 and 4. The performance of the presented receiver concepts will be evaluated by numerical simulation in terms of the bit error ratio they achieve at a given signal-to-noise-ratio (SNR).

Signal-to-Noise Ratio As measure for the SNR, the average energy per bit of information E_b and the noise energy are put into relation. The noise energy can be derived from the noise power by multiplication with the sampling time T_s. Usually, in a digital signal processing setup, T_s is normalized to 1. Thus, the noise energy double-acts as noise power. The noise power again can be derived from the noise power spectral density N_0. Assuming optimum transmit and receive filters, alias-free sampling with sampling time T_s and representation in the ECB (equivalent complex baseband) domain, the noise power is given by $N_0/T_\mathrm{s} = N_0$. This reduces the regarded SNR measure to the E_b/N_0-ratio, which is defined by the average transmit energy per bit of information E_b and the one-sided noise power spectral density N_0 (corresponds to a two-sided noise power spectral density in ECB domain). Furthermore, as additive noise is considered to be free of mean, N_0 can be considered as noise variance σ_n^2, which is considerably done in this work.

Block Diagram The transmission chain of the UW-OFDM system is modeled as a block diagram as shown in Figure 2.9 with the following steps: The binary transmit data, which is assumed to be uniformly distributed and iid.[6], is either transmitted directly, or channel coded and interleaved prior to further processing. Both, the channel coding and interleaving operation, which are detailed later in this section, are optional and hence shown in dashed boxes. The sequence of binary symbols is then mapped to transmit symbols. While, in general these transmit symbols can be from any alphabet, in this work only 4-QAM will be used, having constellation points at $\{1+\mathrm{j}, 1-\mathrm{j}, -1+\mathrm{j}, -1-\mathrm{j}\}$, with the exception of 2-, 4- and 8-ASK (constellation points at $\{-1, 1\}$, $\{-3, -1, 1, 3\}$ and $\{-7, -5, \ldots, 7\}$), when discussing the widely linear receiver in Section 3.2.3, and 16-QAM (with the values $\{-3, -1, 1, 3\}$ in real and imaginary part) for the symbol scaling compensation in Section 3.2.4.

With channel coding, a conjunction of coding and mapping should be realized for optimality. However, a separate realization with an interleaver of a suitable depth together with bit-interleaved coded modulation (BICM) is close to optimal, if not optimal in some scenarios. According to [CTB98], it is asymptotically optimal and practically sufficient to employ Gray labeling in the mapping operation. According to these findings, in this work only Gray labeled mapping as well as a block interleaver, as detailed later in this section, is used.

[6] independent and identically distributed

An OFDM symbol is assembled in frequency domain according to (2.25) and then transformed into time domain, where it is equipped with the UW (2.18). Next, the channel propagation is modeled, which consists of a multipath propagation and additive white Gaussian noise. In the receiver, first the DFT is performed to get the frequency domain representation, from which the UW influence is subtracted, as described in (2.36). The next block performs the data estimation using any of the receivers that will be introduced in Chapter 3 and 4. After that, these estimates are mapped to a sequence of binary symbols that corresponds to the closest symbol of the transmit alphabet. If applicable, the deinterleaving and channel decoding follows then.

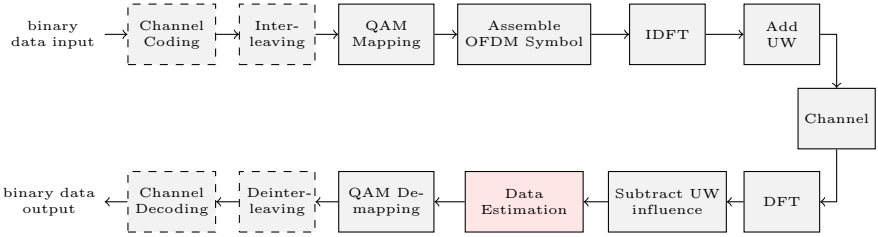


Figure 2.9.: Block diagram of the simulated Unique Word OFDM systems.

Channel Coding Transmission with and without channel coding is considered in this work. For the case when a channel code is used, the block diagram in Figure 2.9 shows a channel encoder at the transmitter side and a decoder at the receiver side in dashed boxes. The convolutional encoder with rate $r = 1/2$, constraint length 7 and the generator polynomials $(133, 171)$ is used, which is based on the definitions used during the standardization process of the IEEE 802.11a standard [IEE99]. A soft decision Viterbi algorithm is applied for decoding. If the Viterbi decoder is provided with reliability information for the receive symbols, the so-called soft information, it is able to further improve upon the hard-decision result. In this work, soft information is determined for all receiver types and soft-decision is done in all simulations using channel coding, rather than hard-decision.

Reliability Information For most receivers shown in this work, the statistics of the remaining error (that is the difference between estimated and actual data symbol) are used to determine the soft information. In particular, the main diagonal of the error covariance matrix is of interest, which holds the variances of the error. In the binary representation of the receive symbols, after QAM demapping, the ratio of the probabilities of each symbol to be 0 and 1 is an adequate reliability measure: A data symbol estimate $\tilde{d} = d + e$ is assumed to consist of the actual transmitted symbol and a complex error, which is zero-mean and Gaussian distributed with a variance given by σ_e^2. 4-QAM with neighboring QAM symbols separated by $d_{\min}$ is considered, where the real and imaginary part are independently detectable, such that both parts affect one detected bit b_0 and b_1 only. Then, the so-called log-likelihood ratio L_0 of the bit representing the real part is (equiprobable symbols assumed with complex Gaussian noise of zero mean and vari-

ance σ_e^2)

$$L_0 \triangleq \ln \frac{\Pr\left(b_0 = 1 \middle| \Re\left\{\tilde{d}\right\}\right)}{\Pr\left(b_0 = 0 \middle| \Re\left\{\tilde{d}\right\}\right)}. \tag{2.61}$$

By applying Bayes' theorem (a more detailed derivation and discussion is given in Section 4.3.2), this becomes

$$\begin{aligned} L_0 &= \ln \frac{p\left(\Re\left\{\tilde{d}\right\} \middle| b_0 = 1\right)}{p\left(\Re\left\{\tilde{d}\right\} \middle| b_0 = 0\right)} \\ &= \ln \frac{\exp\left\{-\frac{\left(\Re\{\tilde{d}\} - d_{\min}/2\right)^2}{\sigma_e^2}\right\}}{\exp\left\{-\frac{\left(\Re\{\tilde{d}\} + d_{\min}/2\right)^2}{\sigma_e^2}\right\}} \\ &= \frac{\left(\Re\left\{\tilde{d}\right\} + d_{\min}/2\right)^2}{\sigma_e^2} - \frac{\left(\Re\left\{\tilde{d}\right\} - d_{\min}/2\right)^2}{\sigma_e^2} \\ &= \frac{2d_{\min}}{\sigma_e^2} \Re\left\{\tilde{d}\right\} \end{aligned} \tag{2.62}$$

which yields $\frac{4}{\sigma_e^2} \Re\left\{\tilde{d}\right\}$ for the definition of the QAM constellations in this work. The same applies to the bit for the imaginary part b_1. This is how the reliability measure is determined from the data estimate and the noise variance for all estimates. It is exact for 4-QAM constellation, and acceptably accurate for higher order QAM constellations. Note that the same log-likelihood ratio applies for 2-ASK (and approximately for higher order ASK) with complex Gaussian noise, as it is used in this work.

Interleaving In order to avoid bundle errors, the coded bits are interleaved using a block interleaver over the complete bit stream. The block interleaver separates adjacent code bits by as many positions as indicated by the interleaving factor, which has to be chosen in order to spread adjacent bits further apart, than the constraint length of the code. This operation is reversed by a deinterleaver before channel decoding at the receiver.

For the choice of the interleaving factor, the time-invariance of the channel and the burst-wise transmission pose the following problem: The interleaving factor shall be large in order to spread symbols, which are affected by possibly correlated channel influence[7], as far as possible apart. On the other hand, due to the time-invariance and the transmission of many OFDM symbols over one realization of the channel, it should be avoided that neighboring bits are transmitted on the same subcarrier (or ones with correlated channel attenuation) in one of the *next* OFDM symbols. In general it can be stated:

[7] The channel attenuation factors given by $\widetilde{\mathbf{H}}$ are necessarily correlated, as only $N_G + 1 < N$ taps of the CIR determine the channel frequency response of length N.

- the further apart the bit positions inside the OFDM symbol, the lower the correlation of the channel attenuation
- the further two symbols are separated in the non-interleaved stream, the less negative impact has a possible correlation in the decoding process
- it should be avoided that both of these distances are low

This trade-off is taken care of by the choice of interleaving factors (number of positions, neighboring code bits are spread apart), as given for each system setup in Appendix A. In Figure 2.10, an interleaving factor of 15 is visualized, as it is used in the exemplary UW-OFDM systems B. The system shown here uses 4-QAM and therefore packs 96 coded binary symbols in one OFDM symbol. The blue marks indicate initially adjacent binary symbols that are separated by 15 other symbols after interleaving. Observing the relation to symbol 2, which is in position 16 of OFDM symbol 1 and marked with a black asterisk, a couple of symbols (circled in red) can be identified in positions that are highly correlated to the channel attenuation in the same position. Symbol number 8, 15 and 21 are the first symbols that lie in closest vicinity of symbol 2, at the positions 10, 19 and 13, compared to 16. This is an acceptable trade-off that was empirically found to show good BER results, even when used with different transmit constellations. For all other utilized UW-OFDM system setups, the same considerations were pursued in order to find a suitable interleaving factor, also under consideration of other transmit constellations used in this work.

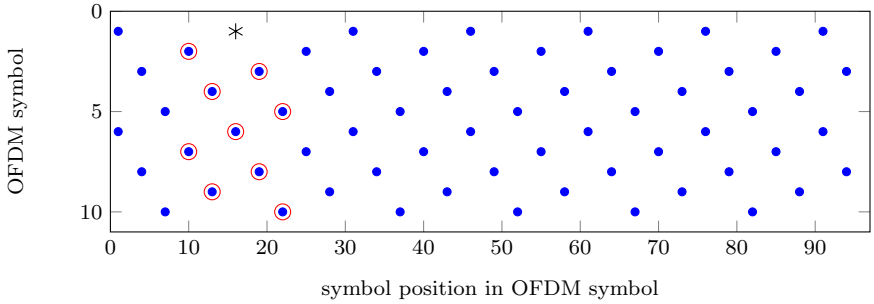


Figure 2.10.: Visualization of the separation of initially adjacent binary symbols with interleaving factor 15 in exemplary UW-OFDM system B and 4-QAM, according to Appendix A.

Simulation For the simulations, the multipath channel propagation is modeled as the convolution of the transmit signal with a time-invariant discrete channel impulse response vector $\mathbf{h}$ not exceeding the guard interval, which is with a maximum length of $N_\mathrm{G}+1$ taps. In any case additive white Gaussian noise is added after this. Its variance is set according to the desired E_b/N_0-ratio, determined at the transmitter output. For the simulations of the multipath environment, channel impulse responses (CIRs) are generated according to the model presented in [Fak97], which has also been used during the IEEE 802.11a standardization process. A fixed set of 10 000 CIRs is used as realization of the channel model, having a channel delay spread $\tau_\mathrm{rms} = 100\,\mathrm{ns}$ for the exemplary UW-OFDM systems A and

B and $\tau_{\mathrm{rms}} = 50\,\mathrm{ns}$ for systems SD and ML (see Appendix A for the system parameters). The CIR is assumed to be constant during the transmission of a burst of several hundred OFDM symbols. The BER values are averaged by dividing the total number of bit errors by the total number of transmitted bits over all channel realizations, which are used for transmission equally often. When the transmission over an AWGN channel is simulated, the CIR is simply $\mathbf{h} = [1]$.

For all operations arbitrarily accurate arithmetic is assumed, which means that quantization effects and numerical effects due to arithmetic implementation are neglected. Perfect channel knowledge, which is the CIR and the E_b/N_0-ratio, is assumed in the simulations. Since the focus lies on data estimation procedures in this work rather than on synchronization or channel estimation approaches, the zero UW is chosen for the BER simulations. When discussing the simulation results, performance differences are given in dB, measured at a BER of 10^{-5}, if not stated otherwise.

3. Linear Receivers for Unique Word OFDM

With the derivations in Section 2.2, the linear model of the Unique Word OFDM transmission system (2.37) is at hand. This linear model allows for many receiver types in order to recover the data from the receive vector $\mathbf{y}$. It has already been stated that the inversion of the channel influence by multiplication with $\widetilde{\mathbf{H}}^{-1}$ represents the optimum receiver for CP-OFDM. The UW-OFDM receiver, however, can be much more sophisticated and complex. This puts an important feature of OFDM and in particular CP-OFDM, its simplicity at the receiver side, into perspective. But, as shown in a performance comparison, the additional effort is not in vain.

In this chapter, several linear receivers will be derived that are either intuitive or emerge from classical or Bayesian estimation theory [Kay93, HOH11]. Also methods of complexity reduction will be shown, which require a slightly different definition of the linear model in (2.37). Therefore, the linear model for the systematic approach for UW-OFDM symbol generation (2.37) is expanded back to

$$\mathbf{y} = \underbrace{\mathbf{B}^{\mathsf{T}}\mathbf{F}_N\mathbf{H}_c\mathbf{F}_N^{-1}\mathbf{B}}_{\widetilde{\mathbf{H}}}\underbrace{\mathbf{P}\begin{bmatrix}\mathbf{I}\\ \mathbf{T}\end{bmatrix}}_{\mathbf{G}}\mathbf{d} + \underbrace{\mathbf{B}^{\mathsf{T}}\mathbf{F}_N\mathbf{n}'}_{\mathbf{n}}, \tag{3.1}$$

and the permutation is reverted $\mathbf{y}_s = \mathbf{P}^{\mathsf{T}}\mathbf{y}$, in order to get a sorted receive vector

$$\mathbf{y}_s = \underbrace{\mathbf{P}^{\mathsf{T}}\mathbf{B}^{\mathsf{T}}\mathbf{F}_N\mathbf{H}_c\mathbf{F}_N^{-1}\mathbf{B}\mathbf{P}}_{\widetilde{\mathbf{H}}_s}\underbrace{\begin{bmatrix}\mathbf{I}\\ \mathbf{T}\end{bmatrix}}_{\mathbf{G}_s}\mathbf{d} + \underbrace{\mathbf{P}^{\mathsf{T}}\mathbf{B}^{\mathsf{T}}\mathbf{F}_N\mathbf{n}'}_{\mathbf{n}_s}. \tag{3.2}$$

This also yields sorted versions of the channel matrix $\widetilde{\mathbf{H}}_s$ (which is diagonal, just as $\widetilde{\mathbf{H}}$) and the generator matrix $\mathbf{G}_s$. This sorted version places the data subcarriers at the first N_d and the redundant subcarriers at the following N_r positions, instead of an order of the subcarrier positions as formulated by $\mathbf{P}$. This yields again a linear model

$$\mathbf{y}_s = \widetilde{\mathbf{H}}_s\mathbf{G}_s\mathbf{d} + \mathbf{n}_s. \tag{3.3}$$

The advantage of the sorted notation becomes apparent, when the complexity optimized versions of the equalizers are introduced and their complexities compared. For the data estimates though, it does not matter at all if (2.37) or (3.3) is used. As the definition of $\mathbf{G}$ for non-systematic UW-OFDM is different, this approach cannot benefit from the sorting, and thus the complexity optimization shown later in this chapter is not feasible. For non-systematic UW-OFDM systems, (2.37) applies and $\widetilde{\mathbf{H}}$, $\mathbf{G}$ and $\mathbf{y}$ need to be used instead of $\widetilde{\mathbf{H}}_s$, $\mathbf{G}_s$ and $\mathbf{y}_s$ for the remainder of this chapter.

For now, linear data estimators of the form

$$\tilde{\mathbf{d}} = \mathbf{E}\mathbf{y}_s \tag{3.4}$$

are considered, where $\mathbf{E} \in \mathbb{C}^{N_d \times (N_d+N_r)}$ is the estimator matrix. These estimators are derived with the help of classical estimation theory in Section 3.1 and Bayesian estimation theory in Section 3.2. In case of a complex valued transmit alphabet that does not show first and second order statistics being invariant with respect to rotation within the complex plane, a so-called *improper* constellation [NM93], an even better performing Bayesian estimator exists. This estimator and a definition to 'properness' will be introduced in Section 3.2.3. Additionally, the computational complexity will be analyzed in Section 3.3, for most linear receivers. A performance comparison follows in Section 3.4, before summarizing the chapter in Section 3.5.

3.1. Classical Data Estimators – Zero Forcing Solutions

Note, that in classical estimation, the data vector is assumed to be deterministic but unknown. In order for the estimator to be unbiased, it is required that

$$\mathrm{E}\left\{\tilde{\mathbf{d}}\right\} = \mathrm{E}\left\{\mathbf{E}\mathbf{y}_s\right\} = \mathbf{E}\widetilde{\mathbf{H}}_s\mathbf{G}_s\mathbf{d} = \mathbf{d}. \tag{3.5}$$

Consequently, the unbiased constraint takes on the form

$$\mathbf{E}\widetilde{\mathbf{H}}_s\mathbf{G}_s = \mathbf{I}, \tag{3.6}$$

which is equivalent to the zero forcing (ZF) criterion for linear equalizers. The solution to (3.6) is ambiguous, though. To show this, a singular value decomposition of $\widetilde{\mathbf{H}}_s\mathbf{G}_s \in \mathbb{C}^{(N_d+N_r)\times N_d}$ is considered, as

$$\widetilde{\mathbf{H}}_s\mathbf{G}_s = \mathbf{U}\begin{bmatrix}\mathbf{\Sigma}\\ \mathbf{0}\end{bmatrix}\mathbf{V}^{\mathsf{H}}, \tag{3.7}$$

with unitary matrices $\mathbf{U} \in \mathbb{C}^{(N_d+N_r)\times(N_d+N_r)}$ and $\mathbf{V} \in \mathbb{C}^{N_d\times N_d}$, and with the diagonal matrix $\mathbf{\Sigma} \in \mathbb{R}^{N_d\times N_d}$ having the singular values of $\widetilde{\mathbf{H}}_s\mathbf{G}_s$ on its main diagonal. With (3.7) the postulation for the absence of a bias (also: ZF criterion) (3.6) becomes

$$\mathbf{E}\mathbf{U}\begin{bmatrix}\mathbf{\Sigma}\\ \mathbf{0}\end{bmatrix}\mathbf{V}^{\mathsf{H}} = \mathbf{I}. \tag{3.8}$$

It is easy to see that (3.8) and therefore also (3.6) is fulfilled by every equalizer of the form

$$\mathbf{E} = \mathbf{V}\begin{bmatrix}\mathbf{\Sigma}^{-1} & \mathbf{A}\end{bmatrix}\mathbf{U}^{\mathsf{H}} \tag{3.9}$$

with arbitrary $\mathbf{A} \in \mathbb{C}^{N_d\times N_r}$. Due to this fact, the ZF solution is ambiguous, which distinguishes UW-OFDM from competing block oriented single input single output (SISO) approaches like CP-OFDM and CP-SC/FDE. For CP-OFDM the channel inversion receiver $\mathbf{E} = \widetilde{\mathbf{H}}^{-1}$ represents the unambiguous ZF solution, which also corresponds to the optimum data estimator, see [NP00]. For CP-SC/FDE the ZF solution is also unambiguous, as soon as the receiver filter (for example a matched filter) preceding the equalizer is specified. It is given by the inverse of the diagonal symbol spaced channel matrix which

contains the influence of the transmit pulse shaping filter, the multipath channel and the receiver filter [HKWR03]. As the solution to the unbiased constraint is not unambiguous, it makes sense to look for the optimum solution which is commonly known as the best linear unbiased estimator.

3.1.1. Best Linear Unbiased Estimator

By applying the Gauss-Markov theorem [Kay93] to (3.3) and with the noise covariance matrix $\mathbf{C}_{nn} = \mathrm{E}\left\{\mathbf{n}_s\mathbf{n}_s^{\mathsf{H}}\right\} = N\sigma_n^2\mathbf{I}$, the best linear unbiased estimator (BLUE) and consequently the optimum ZF equalizer follows to

$$\mathbf{E}_{\mathrm{BLUE}} = \left(\mathbf{G}_s^{\mathsf{H}}\widetilde{\mathbf{H}}_s^{\mathsf{H}}\widetilde{\mathbf{H}}_s\mathbf{G}_s\right)^{-1}\mathbf{G}_s^{\mathsf{H}}\widetilde{\mathbf{H}}_s^{\mathsf{H}}. \tag{3.10}$$

$\mathbf{E}_{\mathrm{BLUE}}$ as given in (3.10) represents the pseudo-inverse of $\widetilde{\mathbf{H}}_s\mathbf{G}_s$. Since the noise in (3.3) is assumed to be Gaussian, (3.10) is also the minimum variance unbiased (MVU) estimator. The covariance matrix of $\tilde{\mathbf{d}} = \mathbf{E}_{\mathrm{BLUE}}\mathbf{y}_s$, or equivalently the covariance matrix of the error $\mathbf{e} = \mathbf{d} - \tilde{\mathbf{d}}$ is given by

$$\mathbf{C}_{ee} = N\sigma_n^2\left(\mathbf{G}_s^{\mathsf{H}}\widetilde{\mathbf{H}}_s^{\mathsf{H}}\widetilde{\mathbf{H}}_s\mathbf{G}_s\right)^{-1}. \tag{3.11}$$

With the singular value decomposition as in (3.7) and after some rearrangements using standard matrix algebra, (3.10) can immediately be rewritten as

$$\mathbf{E}_{\mathrm{BLUE}} = \mathbf{V}\begin{bmatrix}\boldsymbol{\Sigma}^{-1} & \mathbf{0}\end{bmatrix}\mathbf{U}^{\mathsf{H}}. \tag{3.12}$$

By comparing this result with (3.9) it can be concluded that $\mathbf{E}_{\mathrm{BLUE}}$ corresponds to the solution in (3.9) for the particular case $\mathbf{A} = \mathbf{0}$. $\mathbf{E}_{\mathrm{BLUE}}$ is in general a full matrix, which distinguishes UW-OFDM from CP-OFDM and CP-SC/FDE, where the BLUE is given by a diagonal matrix.

3.1.2. Complexity Optimized Version of the BLUE

One drawback of the BLUE represented as in (3.10) is the fact that an $N_d \times N_d$ matrix has to be inverted to determine the equalizer. In this section, a significantly complexity reduced version of the BLUE is derived by exploiting the simple structures of $\mathbf{G}_s$ and $\widetilde{\mathbf{H}}_s$, respectively. This structure however, is due to the re-sorting of the data and redundant subcarriers. The complexity improvement does not apply to non-systematically generated UW-OFDM, as the non-systematic $\mathbf{G}$ does not show a comparable simple structure.

For the complexity optimized BLUE, $\widetilde{\mathbf{H}}_s$ is decomposed as

$$\widetilde{\mathbf{H}}_s = \begin{bmatrix}\widetilde{\mathbf{H}}_{s,d} & \mathbf{0} \\ \mathbf{0} & \widetilde{\mathbf{H}}_{s,r}\end{bmatrix}, \tag{3.13}$$

with the diagonal matrices $\widetilde{\mathbf{H}}_{s,d} \in \mathbb{C}^{N_d \times N_d}$ and $\widetilde{\mathbf{H}}_{s,r} \in \mathbb{C}^{N_r \times N_r}$. With

$$\mathbf{G}_s = \begin{bmatrix}\mathbf{I} \\ \mathbf{T}\end{bmatrix} \tag{3.14}$$

from (3.3) it follows that

$$\widetilde{\mathbf{H}}_s\mathbf{G}_s = \begin{bmatrix} \widetilde{\mathbf{H}}_{s,d} & \mathbf{0} \\ \mathbf{0} & \widetilde{\mathbf{H}}_{s,r} \end{bmatrix} \begin{bmatrix} \mathbf{I} \\ \mathbf{T} \end{bmatrix} = \begin{bmatrix} \widetilde{\mathbf{H}}_{s,d} \\ \widetilde{\mathbf{H}}_{s,r}\mathbf{T} \end{bmatrix}, \tag{3.15}$$

and the expression $(\mathbf{G}_s^{\mathsf{H}}\widetilde{\mathbf{H}}_s^{\mathsf{H}}\widetilde{\mathbf{H}}_s\mathbf{G}_s)^{-1}$ appearing in (3.10) and (3.11) can be written as

$$(\mathbf{G}_s^{\mathsf{H}}\widetilde{\mathbf{H}}_s^{\mathsf{H}}\widetilde{\mathbf{H}}_s\mathbf{G}_s)^{-1} = (\widetilde{\mathbf{H}}_{s,d}^{\mathsf{H}}\widetilde{\mathbf{H}}_{s,d} + \mathbf{T}^{\mathsf{H}}\widetilde{\mathbf{H}}_{s,r}^{\mathsf{H}}\widetilde{\mathbf{H}}_{s,r}\mathbf{T})^{-1}. \tag{3.16}$$

With the introduction of the real diagonal matrices

$$\mathbf{D}_d = \widetilde{\mathbf{H}}_{s,d}^{\mathsf{H}}\widetilde{\mathbf{H}}_{s,d}, \quad \mathbf{D}_d \in \mathbb{R}^{N_d \times N_d}, \tag{3.17}$$

$$\mathbf{D}_r = \widetilde{\mathbf{H}}_{s,r}^{\mathsf{H}}\widetilde{\mathbf{H}}_{s,r}, \quad \mathbf{D}_r \in \mathbb{R}^{N_r \times N_r}, \tag{3.18}$$

the matrix inversion lemma [Kay93] can be applied to the right hand side of (3.16) to obtain

$$(\mathbf{G}_s^{\mathsf{H}}\widetilde{\mathbf{H}}_s^{\mathsf{H}}\widetilde{\mathbf{H}}_s\mathbf{G}_s)^{-1} = \mathbf{D}_d^{-1} - \mathbf{D}_d^{-1}\mathbf{T}^{\mathsf{H}}(\mathbf{T}\mathbf{D}_d^{-1}\mathbf{T}^{\mathsf{H}} + \mathbf{D}_r^{-1})^{-1}\mathbf{T}\mathbf{D}_d^{-1}. \tag{3.19}$$

The inversions of the real diagonal matrices $\mathbf{D}_d$ and $\mathbf{D}_r$ are trivial, and the additional matrix $(\mathbf{T}\mathbf{D}_d^{-1}\mathbf{T}^{\mathsf{H}}+\mathbf{D}_r^{-1})$ to be inverted is Hermitian and only has the dimension $N_r \times N_r$. Furthermore, the expression $\mathbf{T}\mathbf{D}_d^{-1}$ (and its Hermitian transpose) occurs repeatedly in (3.19) which allows for further complexity reduction.

Summarizing the complexity reduction, the BLUE estimator becomes

$$\mathbf{E}_{\mathrm{BLUE}} = \left(\mathbf{D}_d^{-1} - \mathbf{D}_d^{-1}\mathbf{T}^{\mathsf{H}}(\mathbf{T}\mathbf{D}_d^{-1}\mathbf{T}^{\mathsf{H}} + \mathbf{D}_r^{-1})^{-1}\mathbf{T}\mathbf{D}_d^{-1}\right)\left[\widetilde{\mathbf{H}}_{s,d}^{\mathsf{H}} \quad \mathbf{T}^{\mathsf{H}}\widetilde{\mathbf{H}}_{s,r}^{\mathsf{H}}\right]. \tag{3.20}$$

In Section 3.3, the complexity of the different representations of the BLUE is studied. It should be pointed out that the derivation of the complexity reduced version of the BLUE has mainly been made possible by the re-sorting (multiplication with $\mathbf{P}^{\mathsf{T}}$) of the data and redundant subcarrier symbols in (3.3). As this separation is nonexistent for non-systematically generated UW-OFDM, the complexity optimized version of the BLUE is not applicable for this UW-OFDM scheme.

3.1.3. Sub-Optimum ZF Receiver Structures

Any linear zero forcing equalizer has to fulfill (3.6) in order to be a valid estimator according to (3.4). As already shown above, the ZF solution is ambiguous for the UW-OFDM transmission model described in (2.37) or (3.3). Another quite intuitive and straightforward ZF solution is given by

$$\mathbf{E}_{\mathrm{CI}} = \begin{bmatrix} \mathbf{I} & \mathbf{0} \end{bmatrix} \widetilde{\mathbf{H}}_s^{-1}. \tag{3.21}$$

This equalizer inverts the channel $\widetilde{\mathbf{H}}_s$ first, and the data symbols are extracted subsequently. Clearly this procedure fulfills (3.6). In the following, this equalizer will be referred to as the channel inversion (CI) receiver. Using the decomposition of $\widetilde{\mathbf{H}}_s$ as in (3.13), (3.21) can be simplified to

$$\mathbf{E}_{\mathrm{CI}} = \widetilde{\mathbf{H}}_{s,d}^{-1}\begin{bmatrix} \mathbf{I} & \mathbf{0} \end{bmatrix}. \tag{3.22}$$

The channel inversion receiver represents a low complex solution since $\widetilde{\mathbf{H}}_{s,d}$ has a diagonal structure, but it does not take advantage of the correlations introduced by $\mathbf{G}_s$ at the transmitter side. The covariance matrix of $\tilde{\mathbf{d}} = \mathbf{E}_{\mathrm{CI}}\mathbf{y}_s$, or equivalently the covariance matrix of the error $\mathbf{e} = \mathbf{d} - \tilde{\mathbf{d}}$ can easily shown to be

$$\mathbf{C}_{ee} = N\sigma_n^2 \left(\widetilde{\mathbf{H}}_{s,d}^{\mathsf{H}}\widetilde{\mathbf{H}}_{s,d}\right)^{-1}. \tag{3.23}$$

Next, another quite intuitive equalizer is addressed, which exploits the a-priori knowledge that the guard interval samples of an UW-OFDM symbol must be zero after the channel inversion in the noiseless case. In the presence of noise, simply the guard interval samples are forced to zero, which is achieved by the time domain windowing (TDW) equalizer of the form

$$\mathbf{E}_{\mathrm{TDW}} = \begin{bmatrix}\mathbf{I} & \mathbf{0}\end{bmatrix}\mathbf{P}^{\mathsf{T}}\mathbf{B}^{\mathsf{T}}\mathbf{F}_N\mathbf{\Theta}\mathbf{F}_N^{-1}\mathbf{B}\mathbf{P}\widetilde{\mathbf{H}}_s^{-1}, \tag{3.24}$$

where

$$\mathbf{\Theta} = \begin{bmatrix}\mathbf{I} & \mathbf{0}\\ \mathbf{0} & \mathbf{0}\end{bmatrix}. \tag{3.25}$$

The TDW equalizer starts with an inversion of the channel. The permutation is applied next and the zero subcarrier symbols are added again in order to be able to transform back to time domain with an IDFT of length N. Here, a windowing is performed by $\mathbf{\Theta}$, where the guard interval samples are forced to zero. Next, a transformation back to frequency domain is carried out, the zero subcarriers are excluded again, a re-sorting is done, and finally the data symbols are extracted. It can easily be shown that $\mathbf{E}_{\mathrm{TDW}}$ also fulfills (3.6). Note, that the TDW equalizer also represents a quite low complex solution since none of the individual operations requires a full matrix multiplication, in fact most of the steps apart from DFT and IDFT are trivial. The covariance matrix of $\tilde{\mathbf{d}} = \mathbf{E}_{\mathrm{TDW}}\mathbf{y}_s$, or equivalently the covariance matrix of the estimation error $\mathbf{e} = \mathbf{d} - \tilde{\mathbf{d}}$ is given by

$$\mathbf{C}_{ee} = N\sigma_n^2\mathbf{E}_{\mathrm{TDW}}\mathbf{E}_{\mathrm{TDW}}^{\mathsf{H}}. \tag{3.26}$$

The two receivers (3.21) and (3.24) need the actual data symbols visible in the frequency domain code word $\mathbf{c}$, and thus work fine if the system generates the UW-OFDM symbol systematically. For non-systematic UW-OFDM however, this is not the case, and an additional inversion of the potentially fully occupied $\mathbf{A}$ would be necessary. Hence, the receivers for non-systematically generated UW-OFDM with CI or TDW data estimators are not regarded in this work, as they are not feasible in this simplicity and perform inferiorly.

3.2. Linear Bayesian Data Estimators — LMMSE Solutions

Now, the commonly used linear minimum mean square error (LMMSE) data estimator is discussed, which is derived with the help of the Bayesian approach. In the Bayesian approach, the data vector is assumed to be the realization of a random vector instead of a deterministic and unknown vector as in classical estimation theory. In the following, the LMMSE batch solution is derived, then a complexity optimized version of the LMMSE batch solution is formulated. Furthermore, in [HOH11] a highly complexity optimized version of the sequential LMMSE estimator is introduced. This work will exclude its derivation; it is referred to [HOH11, Kay93].

3.2.1. LMMSE Batch Solution

The LMMSE batch estimator is a linear estimator that is used as in (3.4), just like the estimators introduced up to now. The term "batch" indicates the fact that the equalizer can be determined completely before the actual data estimation. The result of the estimator determination is a *batch* of instructions (one row of matrix $\mathbf{E}$ for every data value to be estimated) in order to estimate the whole data vector at once. It also distinguishes this version from the sequential LMMSE, which is omitted in this work.

By applying the Bayesian Gauss-Markov theorem [Kay93] to (3.3), where $\mathbf{d}$ is now assumed to be the realization of a random vector, and by using $\mathbf{C}_{dd} = \sigma_d^2\mathbf{I}$ and $\mathbf{C}_{nn} = N\sigma_n^2\mathbf{I}$ the *split* LMMSE equalizer follows to

$$\mathbf{E}_{\mathrm{LMMSE}} = \mathbf{W}\widetilde{\mathbf{H}}_s^{-1}, \tag{3.27}$$

where $\mathbf{W}$ represents a Wiener smoothing matrix given by

$$\mathbf{W} = \mathbf{G}_s^{\mathsf{H}}\left(\mathbf{G}_s\mathbf{G}_s^{\mathsf{H}} + \frac{N\sigma_n^2}{\sigma_d^2}(\widetilde{\mathbf{H}}_s^{\mathsf{H}}\widetilde{\mathbf{H}}_s)^{-1}\right)^{-1}. \tag{3.28}$$

The following interpretation of the LMMSE estimator's mode of operation is allowed for (3.27): The LMMSE equalizer acts as a split composition of a simple channel inversion stage (multiplication with $\widetilde{\mathbf{H}}_s^{-1}$) and a Wiener smoothing operation (multiplication with $\mathbf{W}$). The Wiener smoothing operation exploits the correlations between subcarrier symbols which have been introduced by (2.24) at the transmitter and acts as a noise reduction operation on the subcarriers.

For the split equalizer in (3.27), a rather large $(N_d + N_r) \times (N_d + N_r)$ matrix has to be inverted. By applying the matrix inversion lemma, it can easily be shown that the equalizer can equivalently be determined by

$$\mathbf{E}_{\mathrm{LMMSE}} = \left(\mathbf{G}_s^{\mathsf{H}}\widetilde{\mathbf{H}}_s^{\mathsf{H}}\widetilde{\mathbf{H}}_s\mathbf{G}_s + \frac{N\sigma_n^2}{\sigma_d^2}\mathbf{I}\right)^{-1}\mathbf{G}_s^{\mathsf{H}}\widetilde{\mathbf{H}}_s^{\mathsf{H}}. \tag{3.29}$$

Equation (3.29) shows strong similarities to the BLUE in (3.10). For $\sigma_n^2 = 0$ the expressions for the LMMSE equalizer and the BLUE coincide. Note that by using (3.29) instead of (3.27) for the LMMSE equalizer determination, the matrix to be inverted only has the dimension $N_d \times N_d$. The error $\mathbf{e} = \mathbf{d} - \tilde{\mathbf{d}}$ has zero mean, and its covariance matrix is given by

$$\mathbf{C}_{ee} = N\sigma_n^2\left(\mathbf{G}_s^{\mathsf{H}}\widetilde{\mathbf{H}}_s^{\mathsf{H}}\widetilde{\mathbf{H}}_s\mathbf{G}_s + \frac{N\sigma_n^2}{\sigma_d^2}\mathbf{I}\right)^{-1}. \tag{3.30}$$

3.2.2. Complexity Optimized LMMSE Batch Equalizer

For the LMMSE equalizer, a complexity reduced version can be derived similar as for the BLUE in Section 3.1.2. Using (3.13) and (3.15), the expression in (3.29) to be inverted can be written as

$$\left(\mathbf{G}_s^{\mathsf{H}}\widetilde{\mathbf{H}}_s^{\mathsf{H}}\widetilde{\mathbf{H}}_s\mathbf{G}_s + \frac{N\sigma_n^2}{\sigma_d^2}\mathbf{I}\right)^{-1} = \left(\widetilde{\mathbf{H}}_{s,d}^{\mathsf{H}}\widetilde{\mathbf{H}}_{s,d} + \mathbf{T}^{\mathsf{H}}\widetilde{\mathbf{H}}_{s,r}^{\mathsf{H}}\widetilde{\mathbf{H}}_{s,r}\mathbf{T} + \frac{N\sigma_n^2}{\sigma_d^2}\mathbf{I}\right)^{-1}. \tag{3.31}$$

Now, defining

$$\mathbf{D}_d = \widetilde{\mathbf{H}}_{s,d}^{\mathsf{H}} \widetilde{\mathbf{H}}_{s,d} + \frac{N\sigma_n^2}{\sigma_d^2}\mathbf{I}, \qquad \mathbf{D}_d \in \mathbb{C}^{N_d \times N_d}, \tag{3.32}$$

$$\mathbf{D}_r = \widetilde{\mathbf{H}}_{s,r}^{\mathsf{H}} \widetilde{\mathbf{H}}_{s,r}, \qquad \mathbf{D}_r \in \mathbb{C}^{N_r \times N_r}, \tag{3.33}$$

and following the same steps as in Section 3.1.2, the equation for the complexity optimized version of the LMMSE equalizer is obtained by

$$\mathbf{E}_{\mathrm{LMMSE}} = \left(\mathbf{D}_d^{-1} - \mathbf{D}_d^{-1}\mathbf{T}^{\mathsf{H}}(\mathbf{T}\mathbf{D}_d^{-1}\mathbf{T}^{\mathsf{H}} + \mathbf{D}_r^{-1})^{-1}\mathbf{T}\mathbf{D}_d^{-1}\right)\left[\widetilde{\mathbf{H}}_{s,d}^{\mathsf{H}} \quad \mathbf{T}^{\mathsf{H}}\widetilde{\mathbf{H}}_{s,r\cdot}^{\mathsf{H}}\right], \tag{3.34}$$

which is exactly the same as (3.20) with just a slightly different definition of $\mathbf{D}_d$. Again, this version only applies to systematically generated UW-OFDM, the non-systematic approach does not allow this optimization.

3.2.3. Widely Linear MMSE Estimation

The BLUE and LMMSE estimator are optimal only for a special case that was implicitly assumed: They assume a data symbol constellation $\mathcal{A}$ that has *proper* statistics [NM93]. Properness is a property of complex random variables or vectors, which requires that

$$\mathrm{E}\left\{(\mathbf{x} - \mathrm{E}\{\mathbf{x}\})(\mathbf{x} - \mathrm{E}\{\mathbf{x}\})^{\mathsf{T}}\right\} = 0. \tag{3.35}$$

In the scalar case, this is equivalent with the fact that the real and imaginary part of the stochastic variable have the same variance and are uncorrelated to each other. Properness of course applies to most practical QAM constellations (assuming iid. binary input symbols). However, not always a proper alphabet is used for communication. For example, small ASK constellations defining only real valued data symbols are used in some standards as a simple "fallback" constellation, if communication otherwise fails. ASK constellations do not utilize the imaginary part at all, thus ASK is an improper constellation.

When the data symbol alphabet is improper, this can be taken into account when deriving a linear Bayesian estimator. For ASK data symbol alphabets, the so-called *widely* linear MMSE estimator is able to achieve better performance results compared to the LMMSE estimator [PC95, GSL03]. The derivation of the WLMMSE data estimator is straightforward and shown in [ASS11]. In this work the theory will not be elaborated and only the final results are shown. The 'unsorted' linear model (2.37) is used in this section, as the sorting does not offer any possibilities for complexity optimization. For widely linear estimators, the equalizing equation (3.4) is altered to[1]

$$\tilde{\mathbf{d}} = \mathbf{E}_1\mathbf{y} + \mathbf{E}_2\mathbf{y}^*, \tag{3.36}$$

with the two equalizer matrices defined as

$$\begin{aligned} \mathbf{E}_1 &= \left(\mathbf{C}_{dy} - \mathbf{C}_{dy^*}\mathbf{C}_{yy}^{-*}\mathbf{C}_{yy^*}^{*}\right)\mathbf{P}_{yy}^{-1}, \\ \mathbf{E}_2 &= \left(\mathbf{C}_{dy^*} - \mathbf{C}_{dy}\mathbf{C}_{yy}^{-1}\mathbf{C}_{yy^*}\right)\mathbf{P}_{yy}^{-*}, \end{aligned} \tag{3.37}$$

[1] As the operation order of the complex conjugate and the inversion does not matter, that is $\left(\mathbf{C}_{yy}^{*}\right)^{-1} = \left(\mathbf{C}_{yy}^{-1}\right)^{*}$, it is written as $\mathbf{C}_{yy}^{-*}$.

and the Schur complement

$$\mathbf{P}_{yy} = \mathbf{C}_{yy} - \mathbf{C}_{yy^*}\mathbf{C}_{yy}^{-*}\mathbf{C}_{yy^*}^{*}. \tag{3.38}$$

These equations make use of a covariance matrix of two zero mean vectors $\mathbf{a}$ and $\mathbf{b}$, which is defined as

$$\begin{aligned} \mathbf{C}_{ab} &= \mathrm{E}\left\{\mathbf{a}\mathbf{b}^{\mathsf{H}}\right\}, \\ \mathbf{C}_{ab^*} &= \mathrm{E}\left\{\mathbf{a}\left(\mathbf{b}^{*}\right)^{\mathsf{H}}\right\} = \mathrm{E}\left\{\mathbf{a}\mathbf{b}^{\mathsf{T}}\right\}. \end{aligned} \tag{3.39}$$

The matrix $\mathbf{C}_{ab^*}$ is sometimes also referred to as *complementary* or *pseudo*-covariance matrix.

For the special case of proper transmit data vectors, $\mathbf{C}_{dy^*} = \mathbf{0}$ and $\mathbf{C}_{yy^*} = \mathbf{0}$, reducing the WLMMSE to the LMMSE equalizer, such that $\mathbf{E}_1 = \mathbf{E}_{\mathrm{LMMSE}}$ and $\mathbf{E}_2 = \mathbf{0}$, as shown in [Tra13]. For ASK constellations, $\mathbf{d}$ is a vector of real data symbols. Then $\mathbf{C}_{dy^*} = \mathbf{C}_{dy}^{*}$ and subsequently $\mathbf{E}_2 = \mathbf{E}_1^{*}$, which results in the WLMMSE data estimation of the form [Tra13]

$$\tilde{\mathbf{d}} = \mathbf{E}_1\mathbf{y} + \mathbf{E}_1^{*}\mathbf{y}^{*} = 2\Re\left\{\mathbf{E}_{\mathrm{WLMMSE}}\mathbf{y}\right\}, \tag{3.40}$$

$$\mathbf{E}_{\mathrm{WLMMSE}} = \mathbf{E}_1 = \left(\mathbf{C}_{dy} - \mathbf{C}_{dy}^{*}\mathbf{C}_{yy}^{-*}\mathbf{C}_{yy^*}^{*}\right)\mathbf{P}_{yy}^{-1}. \tag{3.41}$$

With the covariance matrices defined as in (3.39), the necessary parts of the WLMMSE estimator for UW-OFDM in combination with real data vectors can be identified as

$$\begin{aligned} \mathbf{C}_{dy} &= \sigma_d^2\mathbf{G}^{\mathsf{H}}\widetilde{\mathbf{H}}^{\mathsf{H}}, \\ \mathbf{C}_{yy} &= \sigma_d^2\widetilde{\mathbf{H}}\mathbf{G}\mathbf{G}^{\mathsf{H}}\widetilde{\mathbf{H}}^{\mathsf{H}} + N\sigma_n^2\mathbf{I}, \\ \mathbf{C}_{yy^*} &= \sigma_d^2\widetilde{\mathbf{H}}\mathbf{G}\mathbf{G}^{\mathsf{T}}\widetilde{\mathbf{H}}^{\mathsf{T}}, \end{aligned} \tag{3.42}$$

and the Schur complement $\mathbf{P}_{yy}$ is composed as in (3.38).

The error covariance matrix can be determined by [ASS11]

$$\begin{aligned} \mathbf{C}_{ee} = \mathbf{C}_{dd} &- \left(\mathbf{C}_{dy} - \mathbf{C}_{dy^*}\mathbf{C}_{yy}^{-*}\mathbf{C}_{yy^*}^{*}\right)\mathbf{P}_{yy}^{-1}\mathbf{C}_{dy}^{\mathsf{H}} \\ &- \left(\mathbf{C}_{dy^*} - \mathbf{C}_{dy}\mathbf{C}_{yy}^{-1}\mathbf{C}_{yy^*}\right)\mathbf{P}_{yy}^{-*}\mathbf{C}_{dy^*}^{\mathsf{H}}, \end{aligned} \tag{3.43}$$

which, in the context of UW-OFDM with a real valued transmit vector, becomes

$$\begin{aligned} \mathbf{C}_{ee} &= \sigma_d^2\mathbf{I} - \mathbf{E}_{\mathrm{WLMMSE}}\mathbf{C}_{dy}^{\mathsf{H}} - \left(\mathbf{E}_{\mathrm{WLMMSE}}\mathbf{C}_{dy}^{\mathsf{H}}\right)^{*} \\ &= \sigma_d^2\mathbf{I} - 2\Re\left\{\mathbf{E}_{\mathrm{WLMMSE}}\mathbf{C}_{dy}^{\mathsf{H}}\right\}, \end{aligned} \tag{3.44}$$

with $\mathbf{E}_{\mathrm{WLMMSE}}$ as in (3.41).

3.2.4. Linear MMSE Estimation with Symbol Scaling Compensation

In Section 3.1 several ZF estimators $\mathbf{E}$ were introduced, which are unbiased as a prerequisite, such that

$$\mathrm{E}\left\{\tilde{\mathbf{d}}\right\} = \mathrm{E}\left\{\mathbf{E}\mathbf{y}\right\} = \mathbf{d} \tag{3.45}$$

In classical estimation, the parameter to be estimated $\mathbf{d}$ is assumed to be deterministic. Then, it follows

$$\mathrm{E}\left\{\mathbf{E}\mathbf{y}\right\} = \mathrm{E}\left\{\mathbf{E}\widetilde{\mathbf{H}}\mathbf{G}\mathbf{d} + \mathbf{E}\mathbf{n}\right\} = \mathbf{E}\widetilde{\mathbf{H}}\mathbf{G}\mathbf{d}$$

and, according to (3.6), the ZF estimators need to fulfill $\mathbf{E}\widetilde{\mathbf{H}}\mathbf{G} = \mathbf{I}$.

The LMMSE as well as the WLMMSE data estimators as introduced in Section 3.2, are unbiased in the *Bayesian* sense. However, as they do not imply (3.6), they are biased when regarding them in the classical sense [Hay96, Fis02]: While using $\mathbf{E}_{\mathrm{LMMSE}}$ as data estimator, the estimates are given by

$$\tilde{\mathbf{d}} = \mathbf{E}_{\mathrm{LMMSE}}\widetilde{\mathbf{H}}\mathbf{G}\mathbf{d} + \mathbf{E}_{\mathrm{LMMSE}}\mathbf{n}. \tag{3.46}$$

Defining $\mathbf{M} = \mathbf{E}_{\mathrm{LMMSE}}\widetilde{\mathbf{H}}\mathbf{G}$ and treating the whole vector $\mathbf{d}$ in the classical sense as deterministic, the expectation of a single data estimate $\tilde{d}_k, k = 0, \ldots, N_d - 1$ can be found as

$$\begin{aligned}\mathrm{E}\left\{\tilde{d}_k\right\} &= \sum_{l=0}^{N_d-1} [\mathbf{M}]_{k,l}\,\mathrm{E}\left\{d_l\right\} + \sum_{l=0}^{N_d+N_r-1} [\mathbf{E}_{\mathrm{LMMSE}}]_{k,l}\,\mathrm{E}\left\{n_l\right\} \\ &= \underbrace{\sum_{l=0,l\neq k}^{N_d-1} [\mathbf{M}]_{k,l}\,d_l}_{\text{inter-symbol interference}} + \underbrace{[\mathbf{M}]_{k,k}}_{\text{symbol scaling}}\; d_k,\end{aligned} \tag{3.47}$$

where $\mathrm{E}\{n_l\}$ vanishes for noise with zero mean. A bias, according to classical estimation theory, can be identified as

$$\begin{aligned}b_k &= \mathrm{E}\left\{\tilde{d}_k\right\} - d_k = \sum_{l=0,l\neq k}^{N_d-1} [\mathbf{M}]_{k,l}\,d_l + [\mathbf{M}]_{k,k}\,d_k - d_k \\ &= \sum_{l=0,l\neq k}^{N_d-1} [\mathbf{M}]_{k,l}\,d_l + \Big([\mathbf{M}]_{k,k} - 1\Big)\,d_k.\end{aligned} \tag{3.48}$$

For the LMMSE data estimator, clearly two effects can be identified that make up the bias:

- an inter-symbol interference that depends on the actual values of the other transmit symbols excluding number k
- a symbol scaling effect with a factor that is apparent on the main diagonal of $\mathbf{E}_{\mathrm{LMMSE}}\widetilde{\mathbf{H}}\mathbf{G}$, whose magnitude depends on the transmit symbol number k

It is worth mentioning that the BLUE does neither show inter-symbol interference, nor the scaling effect, as it is constructed in agreement with (3.6). As the LMMSE estimator becomes the BLUE with $E_b/N_0 \to \infty$, all the stronger becomes the difference between $\mathbf{E}_{\mathrm{LMMSE}}\widetilde{\mathbf{H}}\mathbf{G}$ and $\mathbf{E}_{\mathrm{BLUE}}\widetilde{\mathbf{H}}\mathbf{G} = \mathbf{I}$, and thus ISI and scaling effect at low SNR. While the ISI has noise-like impact on the decision of the data symbol, the scaling effect has a systematic influence: For a 4-QAM or 2-ASK data constellation, the scaling effect has no impact at all on the bit error probability, as the decision is not affected by scaling. For higher order constellations, however, the impact on the BER may be severe. When 16-QAM, 4-ASK or 8-ASK is used, it is possible that the symbol scaling effect alone causes systematic false decisions.

In this case, the LMMSE estimator can be provided with symbol scaling compensation (scc). The estimator matrix of the linear MMSE estimator with symbol scaling compensation (LMMSE,ssc) is defined by

$$\mathbf{E}_{\mathrm{LMMSE,ssc}} = \mathrm{diag}\left\{1/\left[\mathbf{M}\right]_{0,0}, 1/\left[\mathbf{M}\right]_{1,1}, \ldots, 1/\left[\mathbf{M}\right]_{N_d-1,N_d-1}\right\}\mathbf{E}_{\mathrm{LMMSE}} \tag{3.49}$$

with $\mathbf{E}_{\mathrm{LMMSE}}$ as in (3.29) and

$$\mathbf{M} = \mathbf{E}_{\mathrm{LMMSE}}\widetilde{\mathbf{H}}\mathbf{G}.$$

Using this new LMMSE estimator with ssc according to (3.49) causes the expectation of the estimate of the k-th symbol and its bias to become

$$\begin{aligned}\mathrm{E}\left\{\tilde{d}_k\right\} &= \frac{1}{\left[\mathbf{M}\right]_{k,k}}\sum_{l=0}^{N_d-1}\left[\mathbf{M}\right]_{k,l} d_l \\ &= \underbrace{\frac{1}{\left[\mathbf{M}\right]_{k,k}}\sum_{l=0,l\neq k}^{N_d-1}\left[\mathbf{M}\right]_{k,l} d_l}_{\text{bias } b_k} + d_k.\end{aligned} \tag{3.50}$$

Especially at low E_b/N_0-ratios, where the LMMSE estimator shows a stronger scaling effect, a benefit of the LMMSE estimation with ssc can be expected. This makes the method very valuable for coded transmission. When using this receiver, the covariance matrix of the remaining error $\mathbf{e} = \mathbf{d} - \tilde{\mathbf{d}}$ is determined in the Bayesian sense by

$$\begin{aligned}&\mathbf{C}_{ee} = \mathrm{E}\left\{\left(\mathbf{d}-\tilde{\mathbf{d}}\right)\left(\mathbf{d}-\tilde{\mathbf{d}}\right)^{\mathsf{H}}\right\} \\ &= \mathrm{E}\left\{\left(\mathbf{d}-\mathbf{E}_{\mathrm{LMMSE,ssc}}\widetilde{\mathbf{H}}\mathbf{G}\mathbf{d}-\mathbf{E}_{\mathrm{LMMSE,ssc}}\mathbf{n}\right)\left(\mathbf{d}-\mathbf{E}_{\mathrm{LMMSE,ssc}}\widetilde{\mathbf{H}}\mathbf{G}\mathbf{d}-\mathbf{E}_{\mathrm{LMMSE,ssc}}\mathbf{n}\right)^{\mathsf{H}}\right\} \\ &= \sigma_d^2\left(\mathbf{I}+\mathbf{E}_{\mathrm{LMMSE,ssc}}\widetilde{\mathbf{H}}\mathbf{G}\mathbf{G}^{\mathsf{H}}\widetilde{\mathbf{H}}^{\mathsf{H}}\mathbf{E}_{\mathrm{LMMSE,ssc}}^{\mathsf{H}} - \mathbf{E}_{\mathrm{LMMSE,ssc}}\widetilde{\mathbf{H}}\mathbf{G} - \mathbf{G}^{\mathsf{H}}\widetilde{\mathbf{H}}^{\mathsf{H}}\mathbf{E}_{\mathrm{LMMSE,ssc}}^{\mathsf{H}}\right) + \\ &\qquad + N\sigma_n^2\mathbf{E}_{\mathrm{LMMSE,ssc}}\mathbf{E}_{\mathrm{LMMSE,ssc}}^{\mathsf{H}}.\end{aligned} \tag{3.51}$$

Although for the simulations in this work mainly 4-QAM is considered, in order to show the impact of the LMMSE scaling, a 16-QAM constellation is employed. The scaling

effect can be observed in Figure 3.1, which shows a scatter plot of data estimates after regular LMMSE estimation and again after the symbol scaling compensation [RH13] in exemplary UW-OFDM system B and exemplary multipath channel A. The estimator is designed for an E_b/N_0-ratio of 6 dB, but additive noise is omitted for this presentation. Therefore, the plot only shows the scaling effect and inter-symbol interference. The cloud of receive symbols from the LMMSE with ssc around each constellation point is only due to ISI and not from noise, while the regular LMMSE prominently displays the scaling, additionally. The valid 16-QAM transmit symbols at the positions $\{-3, -1, 1, 3\}$ on the real axis and the same positions on the imaginary axis are plotted as a reference.

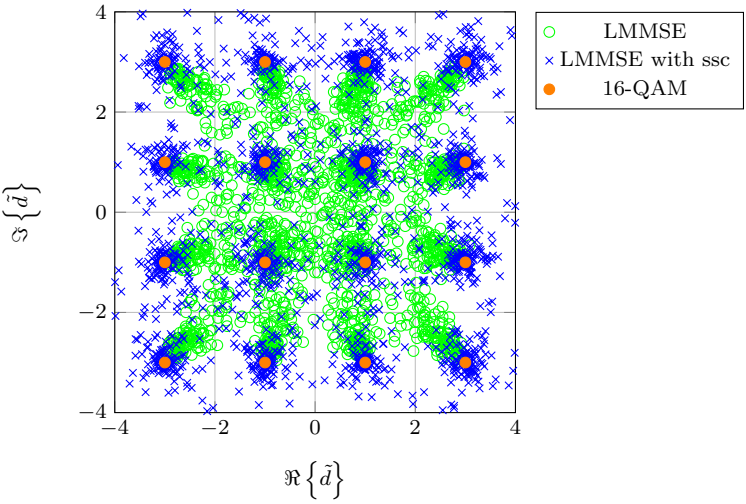


Figure 3.1.: Scatter plot of estimated data symbols after regular LMMSE estimation and LMMSE estimation with symbol scaling compensation in exemplary UW-OFDM system B and exemplary multipath channel A with the estimator designed for an E_b/N_0-ratio of 6 dB.

In case of the WLMMSE and possibly higher order n-ASK constellations with $n \geq 4$, the same considerations apply, too. Here, the data estimation process is

$$\begin{aligned}\tilde{\mathbf{d}} &= 2\Re\left\{\mathbf{E}_{\mathrm{WLMMSE}}\mathbf{y}\right\} \\ &= \mathbf{E}_{\mathrm{WLMMSE}}\left(\widetilde{\mathbf{H}}\mathbf{G}\mathbf{d} + \mathbf{n}\right) + \mathbf{E}^{*}_{\mathrm{WLMMSE}}\left(\widetilde{\mathbf{H}}\mathbf{G}\mathbf{d} + \mathbf{n}\right)^{*}.\end{aligned} \tag{3.52}$$

Analogous to (3.47) and now with $\mathbf{M} = \mathbf{E}_{\mathrm{WLMMSE}}\widetilde{\mathbf{H}}\mathbf{G}$, the expectation of the estimated

symbol number k is

$$\begin{aligned}\mathrm{E}\left\{\tilde{d}_k\right\} &= \sum_{l=0}^{N_d-1} [\mathbf{M}]_{k,l}\, d_l + \sum_{l=0}^{N_d-1} [\mathbf{M}^*]_{k,l}\, d_l^* \\ &= \sum_{l=0,l\neq k}^{N_d-1} [\mathbf{M}]_{k,l}\, d_l + [\mathbf{M}]_{k,k}\, d_k + \sum_{l=0,l\neq k}^{N_d-1} [\mathbf{M}^*]_{k,l}\, d_l^* + [\mathbf{M}^*]_{k,k}\, d_k^* \\ &= \sum_{l=0,l\neq k}^{N_d-1} 2\Re\left\{[\mathbf{M}]_{k,l}\, d_l\right\} + 2\Re\left\{[\mathbf{M}]_{k,k}\, d_k\right\}.\end{aligned} \tag{3.53}$$

As the transmit symbol alphabet is real when the WLMMSE estimator is used, this can be put into

$$\mathrm{E}\left\{\tilde{d}_k\right\} = \sum_{l=0,l\neq k}^{N_d-1} 2\Re\left\{[\mathbf{M}]_{k,l}\right\} d_l + 2\Re\left\{[\mathbf{M}]_{k,k}\right\} d_k, \tag{3.54}$$

Thus a WLMMSE estimator with symbol scaling compensation can be suggested in the form

$$\mathbf{E}_{\mathrm{WLMMSE,ssc}} = \mathrm{diag}\left\{1/\Re\left\{2\,[\mathbf{M}]_{0,0}\right\}, \ldots, 1/\Re\left\{2\,[\mathbf{M}]_{N_d-1,N_d-1}\right\}\right\} \mathbf{E}_{\mathrm{WLMMSE}} \tag{3.55}$$

now with

$$\mathbf{M} = \mathbf{E}_{\mathrm{WLMMSE}}\widetilde{\mathbf{H}}\mathbf{G}.$$

After some calculations, also the error covariance matrix can be evaluated as

$$\begin{aligned}\mathbf{C}_{ee} = \sigma_d^2\mathbf{I} &+ 2\Re\left\{E_{\mathrm{WLMMSE,ssc}}\mathbf{C}_{yy}E_{\mathrm{WLMMSE,ssc}}^{\mathsf{H}}\right\} + \\ &+ 2\Re\left\{E_{\mathrm{WLMMSE,ssc}}\mathbf{C}_{yy^*}E_{\mathrm{WLMMSE,ssc}}^{\mathsf{T}}\right\} - \\ &- 2\Re\left\{E_{\mathrm{WLMMSE,ssc}}\mathbf{C}_{dy}^{\mathsf{H}}\right\} - 2\Re\left\{\mathbf{C}_{dy}E_{\mathrm{WLMMSE,ssc}}^{\mathsf{H}}\right\}\end{aligned} \tag{3.56}$$

with $\mathbf{C}_{dy}$, $\mathbf{C}_{yy}$ and $\mathbf{C}_{yy^*}$ as in (3.42) and $\mathbf{E}_{\mathrm{WLMMSE,ssc}}$ as in (3.55).

Note that the symbol scaling compensated estimators, introduced in this section, are definitely inferior to the regular LMMSE and WLMMSE regarding the Bayesian MSE criterion. In terms of the bit error rate however, they can be expected to perform better, which is proven in Section 3.4.3.

3.3. Complexity Analysis of the Linear Estimators

In this section, the computational complexity of the derived equalizers will be analyzed on the one hand, and of the corresponding data estimation procedures on the other hand. These investigations will clearly show the benefits of the complexity reduced versions. In practice, the equalizers need to be determined each time the channel estimate is updated.

The potentially very complex WLMMSE receiver from Section 3.2.3 is excluded from this analysis. Also, the symbol scaling compensated versions of the LMMSE and WLMMSE are not regarded in this section, but a slight complexity increase can be expected in order to determine the compensating weights, while utilizing any previously calculations. It needs to be noted that non-systematically generated UW-OFDM presents a different situation, as it is not distinguished between data and redundant subcarriers anymore and $\mathbf{G}$ is fully occupied, as well as the sorted version $\mathbf{G}_s$.

3.3.1. Prerequisites

It should be pointed out that it is difficult or even impossible to declare an equitable measure of complexity, since the complexity of an implementation strongly depends on the choice of the hardware and software architecture and of many implementation details. Some operations can even be implemented in many different ways, which might have advantages on certain architectures as well. To simplify things, the number of complex multiplication equivalents (CME) for each individual equalizer and for the corresponding data estimation procedure is counted. Additions are ignored. Complex division are counted as 1 CME. Since the number of required divisions is negligible, this simplification does not effect the final complexity considerably. Real multiplications and real divisions are counted as $\frac{1}{4}$ CME.

For many of the derived equalizer implementations, matrix products of the form $\mathbf{A}^{-1}\mathbf{B}$ with a positive definite Hermitian matrix $\mathbf{A} \in \mathbb{C}^{m\times m}$ and with $\mathbf{B} \in \mathbb{C}^{m\times n_c}$ need to be considered. Calculating $\mathbf{X} = \mathbf{A}^{-1}\mathbf{B}$ is equivalent to solving the set of linear equations

$$\mathbf{A}\mathbf{X} = \mathbf{B} \tag{3.57}$$

for $\mathbf{X}$. For the complexity calculations, it is assumed that (3.57) is solved with the help of a Cholesky decomposition of $\mathbf{A}$ given by $\mathbf{A} = \mathbf{L}\mathbf{L}^{\mathsf{H}}$, where $\mathbf{L}$ is a lower triangular matrix having real and positive values on its main diagonal. $\mathbf{A}\mathbf{X} = \mathbf{B}$ can be rewritten as $\mathbf{L}(\mathbf{L}^{\mathsf{H}}\mathbf{X}) = \mathbf{B}$. To obtain $\mathbf{X}$ one can solve $\mathbf{L}\mathbf{Y} = \mathbf{B}$ for $\mathbf{Y}$ with the help of a forward substitution, and subsequently solve $\mathbf{L}^{\mathsf{H}}\mathbf{X} = \mathbf{Y}$ for $\mathbf{X}$ by backward substitution. The Cholesky decomposition requires $\frac{1}{6}m^3$ complex multiplications/divisions and m square roots [GL96, Sch]. By neglecting square roots, this ends up with $\frac{1}{6}m^3$ CME. A single forward or backward substitution requires $\frac{1}{2}m^2 + \frac{1}{2}m$ CME. In order to solve (3.57) with the aid of a Cholesky decomposition, finally a total count of $\frac{1}{6}m^3 + m^2 n_c + m n_c$ CME can be assumed.

Whenever possible, any simplification that a special matrix structure (for example a diagonal, a real or a Hermitian matrix) could offer is taken into account. Exemplarily, if the result of a matrix product is Hermitian as in $\mathbf{X} = \mathbf{A}^{\mathsf{H}}\mathbf{A}$, then only the main diagonal and the lower triangular part needs to be computed.

3.3.2. Complexity of the Investigated Data Estimation Procedures

Before performing the data estimation with the help of one of the investigated equalizers, an OFDM symbol has to be transformed to frequency domain with an FFT (fast Fourier transform) of length N, which requires $\frac{1}{2}N\log_2(N)$ CME. Furthermore, as one of the preparatory steps the influence of the UW has to be subtracted as described in (2.36).

Since additions/subtractions are not counted in these complexity considerations this step does not increase the CME count for the data estimation procedure.

In the following, the complexity of the equalizers investigated above is considered. The investigations begin with the simplest equalizer $\mathbf{E}_{\mathrm{CI}}$ as given in (3.22). To determine $\mathbf{E}_{\mathrm{CI}}$ only N_d CME (namely complex divisions to invert $\widetilde{\mathbf{H}}_{s,d}$) are required. The data estimation procedure for an OFDM symbol in frequency domain need N_d CME (namely complex multiplications). So the number of CMEs needed for equalizer determination and the actual equalization for the CI estimator is given by

$$\begin{aligned} n_{\mathrm{CI,det}} &= N_d, \\ n_{\mathrm{CI,eq}} &= N_d. \end{aligned} \tag{3.58}$$

To estimate the data part of an OFDM symbol with the help of $\mathbf{E}_{\mathrm{TDW}}$, one could first determine its matrix representation as in (3.24) and then estimate the data vector by performing the full matrix-vector product $\tilde{\mathbf{d}} = \mathbf{E}_{\mathrm{TDW}}\mathbf{y}_s$ which requires $N_d(N_d + N_r)$ operations. However, most of the individual operations needed to perform the data estimation are trivial. The procedure starts with the multiplication $\widetilde{\mathbf{H}}_s^{-1}\mathbf{y}_s$ ($N_d + N_r$ CME), next the permutation is applied and the zero subcarrier symbols are added (zero CME) in order to be able to transform back to time domain with an IFFT of length N ($\frac{N}{2}\log_2(N)$ CME). In time domain a windowing takes place, where the guard interval samples are forced to zero (zero CME). Next a transformation back to frequency domain is performed ($\frac{N}{2}\log_2(N)$ CME), the zero subcarriers are excluded again, a re-sorting is done, and finally the data symbols are extracted (zero CME). So in total, the data estimation procedure per OFDM symbol requires $N\log_2(N) + N_d + N_r$ CME. For the equalizer determination only $\widetilde{\mathbf{H}}_s$ needs to be inverted which requires $N_d + N_r$ CME (namely complex divisions).

$$\begin{aligned} n_{\mathrm{TDW,det}} &= N_d + N_r, \\ n_{\mathrm{TDW,eq}} &= N\log_2(N) + N_d + N_r. \end{aligned} \tag{3.59}$$

The complexity of the different BLUE and LMMSE estimator batch representations is investigated. For all implementations the data vector estimation for one OFDM symbol requires a full matrix vector product $\tilde{\mathbf{d}} = \mathbf{E}\mathbf{y}_s$ with

$$n_{\mathrm{LMMSE,eq}} = N_d(N_d + N_r)\ \mathrm{CME}. \tag{3.60}$$

The CME count for the equalizer determination differs significantly for the different implementations. At first the standard representation of the BLUE as in (3.10) and of the LMMSE estimator as in (3.29) is handled. These two expressions merely differ in the regularization term which only adds a single arithmetic operation. This single operation is neglected, and (3.10) and (3.29) are treated as equally complex. Using (3.15) it is easy to see that the matrix multiplication $\mathbf{X}_1 = \widetilde{\mathbf{H}}_s\mathbf{G}_s$ only leads to N_dN_r CME. For the product $\mathbf{X}_2 = \mathbf{X}_1^{\mathsf{H}}\mathbf{X}_1$ the findings from (3.16) can be used, which is

$$\mathbf{X}_2 = \widetilde{\mathbf{H}}_{s,d}^{\mathsf{H}}\widetilde{\mathbf{H}}_{s,d} + \mathbf{T}^{\mathsf{H}}\widetilde{\mathbf{H}}_{s,r}^{\mathsf{H}}\widetilde{\mathbf{H}}_{s,r}\mathbf{T}.$$

By additionally exploiting the fact that $\mathbf{X}_2$ is Hermitian, one can easily find that the matrix product $\mathbf{X}_2 = \mathbf{X}_1^{\mathsf{H}}\mathbf{X}_1$ requires $\frac{1}{2}N_d^2N_r + N_dN_r + N_d + N_r$ CME. Finally the operation $(\mathbf{X}_2)^{-1}\mathbf{X}_1^{\mathsf{H}}$ takes up

$$\tfrac{7}{6}N_d^3 + N_d^2N_r + N_d^2 + N_dN_r$$

CME by using the Cholesky decomposition together with the forward and backward substitutions as mentioned above. The overall CME count for the BLUE in (3.10) and the LMMSE estimator in (3.29) therefore adds up to

$$n_{\text{LMMSE,det}} = \tfrac{7}{6}N_d^3 + \tfrac{3}{2}N_d^2N_r + 3N_dN_r + N_d^2 + N_d + N_r \text{ CME}. \tag{3.61}$$

Using similar considerations it can be shown that the LMMSE equalizer, which is split regarding its functionality into channel inversion part and Wiener smoothing filter as expressed in (3.27), requires

$$\begin{aligned} n_{\text{LMMSE}_{\text{split}},\text{det}} = {} & \tfrac{7}{6}N_d^3 + \tfrac{5}{2}N_d^2N_r + 2N_dN_r^2 + \tfrac{1}{6}N_r^3 \\ & + N_d^2 + \tfrac{3}{2}N_dN_r + \tfrac{5}{2}N_d + \tfrac{5}{2}N_r \text{ CME}. \end{aligned} \tag{3.62}$$

For the complexity optimized batch representations of the BLUE and the LMMSE estimator one has to determine the expressions in (3.20) and (3.34), respectively. The simple inverses $\mathbf{D}_d^{-1}$ and $\mathbf{D}_r^{-1}$ need $\frac{5}{4}N_d + \frac{5}{4}N_r$ CME, to determine $\mathbf{D}_d^{-1}\mathbf{T}^{\mathsf{H}}$ another $\frac{1}{2}N_dN_r$, and for $\mathbf{T}[\mathbf{D}_d^{-1}\mathbf{T}^{\mathsf{H}}]$ additional $\frac{1}{2}N_dN_r^2 + \frac{1}{2}N_dN_r$ CME are required. The operation $(\cdot)^{-1}\mathbf{T}\mathbf{D}_d^{-1}$ demands $\frac{1}{6}N_r^3 + N_dN_r^2 + N_dN_r$ CME, and the multiplication with $\mathbf{D}_d^{-1}\mathbf{T}^{\mathsf{H}}$ adds $N_d^2N_r$ CME. The determination of $\mathbf{G}_s^{\mathsf{H}}\widetilde{\mathbf{H}}_s^{\mathsf{H}}$ and the final multiplication add $N_dN_r + N_d^2N_r + N_d^2$ CME (see (3.15)), which totals to

$$n_{\text{LMMSE}_{\text{opt}},\text{det}} = \tfrac{1}{6}N_r^3 + 2N_d^2N_r + \tfrac{3}{2}N_dN_r^2 + N_d^2 + 3N_dN_r + \tfrac{5}{4}N_d + \tfrac{5}{4}N_r \text{ CME}. \tag{3.63}$$

Considering the fully occupied matrix $\mathbf{G}$ for the non-systematic UW-OFDM approach, the two LMMSE equalizers are determined with the following complexities:

$$\begin{aligned} n_{\text{LMMSE}_{\text{split}}^{\text{non-syst}},\text{det}} &= \tfrac{8}{3}N_d^3 + \tfrac{3}{2}N_d^2 + \tfrac{11}{2}N_d^2N_r + \tfrac{3}{2}N_dN_r + 3N_dN_r^2 + \tfrac{1}{6}N_r^3 \text{ CME}, \\ n_{\text{LMMSE}^{\text{non-syst}},\text{det}} &= \tfrac{5}{3}N_d^3 + \tfrac{5}{2}N_d^2 + \tfrac{3}{2}N_d^2N_r + \tfrac{5}{2}N_dN_r \text{ CME}. \end{aligned} \tag{3.64}$$

The complexity optimized version of the BLUE/LMMSE is not applicable for non-systematically generated UW-OFDM, as in this case $\mathbf{G}_s^{\mathsf{H}}\widetilde{\mathbf{H}}_s^{\mathsf{H}}\widetilde{\mathbf{H}}_s\mathbf{G}_s$ is not separable into a sum as in (3.16). Therefore, no comparable complexity improvement can be expected. The equalization complexity resembles that of a full matrix multiplication and is identical to the complexity of the UW-OFDM system generating symbols systematically, as in (3.60).

In order to summarize these results, all CME counts are put in Table 3.1 for reference. The estimators shown there are

- CI (3.22), systematic
- TDW (3.24), systematic
- split LMMSE (3.27), systematic and non-systematic
- standard BLUE (3.10) and LMMSE (3.29), systematic and non-systematic
- optimized BLUE (3.20) and LMMSE (3.34), systematic.

3.3.3. Numerical Example

In the simulation part in Section 3.4, results for a particular parameter setup will be shown. For the complexity considerations only N_d, N_r and N are important. In Table 3.2 the complexities of the different equalizer representations and data estimation procedures for exemplary UW-OFDM systems A, B and SD (for system parameters see Appendix A) are compared. The candidate estimators under comparison are the same as summarized at the end of the last section.

For each equalizer, the number of CMEs needed for equalizer determination, as well as for the data estimation process is given. In the last two columns the number of CMEs in relation to the systematic standard LMMSE estimator is given. Note that the contribution of the FFT ($\frac{N}{2}\log_2(N)$ CME) per OFDM symbol is counted for the data estimation, which is required in all cases, along with the additional effort contributed by the particular equalization procedure.

It can be observed that the simple equalizers $\mathbf{E}_{\mathrm{CI}}$ and $\mathbf{E}_{\mathrm{TDW}}$ show a significantly lower complexity for the equalizer determination as well as for the data estimation per OFDM symbol. For the BLUE and LMMSE estimator, it can be stated that the complexity optimized batch solutions reduce the equalizer determination complexity by at least 25% compared to the straightforward implementations in (3.10) and (3.29) for the examined UW-OFDM systems. The split version of the LMMSE according to (3.27) is at least one third more complex than the standard version and can be therefore excluded from considerations.

The standard version of the BLUE/LMMSE estimator for non-systematically generated UW-OFDM requires approximately 30% additional CMEs, compared to the systematic version, as many simplifications cannot be used with this approach. The non-systematic split estimator requiring three times as many CMEs, compared to the standard systematic version, disqualifies itself again from consideration.

estimator type CME counts for estimator determination	CME counts for equalizing operation
CI, systematic N_d	N_d
TDW, systematic $N_d + N_r$	$N \log_2(N) + N_d + N_r$
split LMMSE, systematic $\frac{7}{6}N_d^3 + \frac{5}{2}N_d^2 N_r + 2N_d N_r^2 + \frac{1}{6}N_r^3 + N_d^2 + \frac{3}{2}N_d N_r + \frac{5}{2}N_d + \frac{5}{2}N_r$	$N_d^2 + N_d N_r$
standard BLUE/LMMSE, systematic $\frac{7}{6}N_d^3 + \frac{3}{2}N_d^2 N_r + 3N_d N_r + N_d^2 + N_d + N_r$	$N_d^2 + N_d N_r$
optimized BLUE/LMMSE, systematic $\frac{1}{6}N_r^3 + 2N_d^2 N_r + \frac{3}{2}N_d N_r^2 + N_d^2 + 3N_d N_r + \frac{5}{4}N_d + \frac{5}{4}N_r$	$N_d^2 + N_d N_r$
split LMMSE, non-systematic $\frac{8}{3}N_d^3 + \frac{3}{2}N_d^2 + \frac{11}{2}N_d^2 N_r + \frac{3}{2}N_d N_r + 3N_d N_r^2 + \frac{1}{6}N_r^3$	$N_d^2 + N_d N_r$
standard BLUE/LMMSE, non-systematic $\frac{5}{3}N_d^3 + \frac{5}{2}N_d^2 + \frac{3}{2}N_d^2 N_r + \frac{5}{2}N_d N_r$	$N_d^2 + N_d N_r$

Table 3.1.: CME counts for determination and equalizing operation of the introduced linear data estimators.

Equalization method	CME for		relative CME for	
system A	eq. determination	data estimation	eq. determination	data estimation
CI, systematic	36	228	0.04%	11%
TDW, systematic	52	628	0.06%	30%
split LMMSE, systematic	127677	2064	144%	100%
std. BLUE/LMMSE, systematic	88612	2064	100%	100%
opt. BLUE/LMMSE, systematic	59068	2064	67%	100%
split BLUE/LMMSE, non-syst.	269603	2064	304%	100%
std. BLUE/LMMSE, non-syst.	113544	2064	128%	100%
system B				
CI, systematic	48	240	0.03%	7%
TDW, systematic	64	640	0.03%	20%
split LMMSE, systematic	250059	3264	132%	100%
std. BLUE/LMMSE, systematic	188992	3264	100%	100%
opt. BLUE/LMMSE, systematic	97531	3264	52%	100%
split BLUE/LMMSE, non-syst.	539819	3264	286%	100%
std. BLUE/LMMSE, non-syst.	247296	3264	131%	100%
system SD				
CI, systematic	16	71	0.19%	16%
TDW, systematic	24	189	0.28%	43%
split LMMSE, systematic	12540	439	147%	100%
std. BLUE/LMMSE, systematic	8515	439	100%	100%
opt. BLUE/LMMSE, systematic	6387	439	75%	100%
split BLUE/LMMSE, non-syst.	25920	439	304%	100%
std. BLUE/LMMSE, non-syst.	10859	439	128%	100%

Table 3.2.: CME counts of the introduced equalizers and data estimation procedures for exemplary UW-OFDM systems A, B and SD, according to Table A.1.

3.4. Performance Comparison

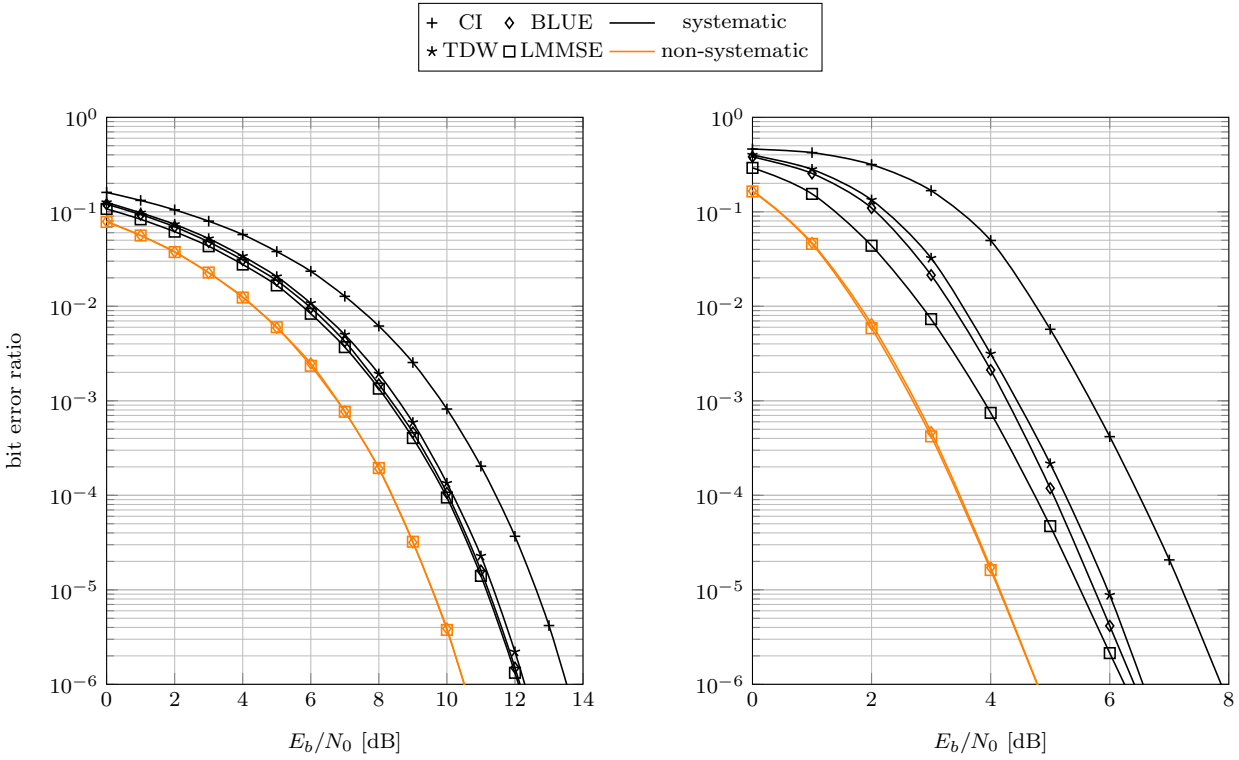
3.4.1. Simulation Results of the Linear Receivers

In this section, the introduced receiver concepts are evaluated in terms of their BER performance. All derived variants of an estimator, that is the conventional form and the complexity optimized version of the BLUE, as well as the split, batch and complexity optimized version of the LMMSE, produce the same results and hence perform equivalently. Different performance of distinct versions of an estimator would only be expected if fixed point implementations were regarded, which is not the focus of the investigations in this work. Results for the exemplary UW-OFDM setups A and B (see Table A.1) will be shown, and the results of both setups will be discussed together, as far as possible. The results for systematic as well as non-systematic UW-OFDM are shown. For non-systematic UW-OFDM the CI and TDW estimators are omitted due to the reasons discussed earlier.

Clearly, OFDM is designed for data transmission in frequency selective environments. Nevertheless, the comparison starts with simulation results in the AWGN channel, since these results provide first interesting insights. In Figure 3.2 and 3.3, the BER performance of the different data estimators is compared for the exemplary UW-OFDM systems A and B (see Appendix A) under AWGN conditions.

The discussion starts with the uncoded case, shown in the sub-figures (a): As expected the CI estimator shows the worst performance, since it completely ignores the information present on the redundant subcarriers. Surprisingly, the very simple and intuitive TDW data estimator performs almost as well as the BLUE and the LMMSE estimator in the AWGN environment. At a BER of 10^{-5} these three estimators, which all make use of the a-priori knowledge introduced by the zero UW, outperform the CI estimator by around 1.5 dB in both UW-OFDM setups A and B. The trend is similar for coded transmission. However, the fact that the LMMSE estimator and the BLUE show a different performance in an AWGN environment is completely in contrast to single-carrier systems [HOH11], for example SC/FDE. Nevertheless, the performance gain of the LMMSE estimator is quite small, and the BLUE approaches the LMMSE estimator performance for high E_b/N_0, as the term $N\sigma_n^2/\sigma_d^2$ in (3.29) converges to zero.

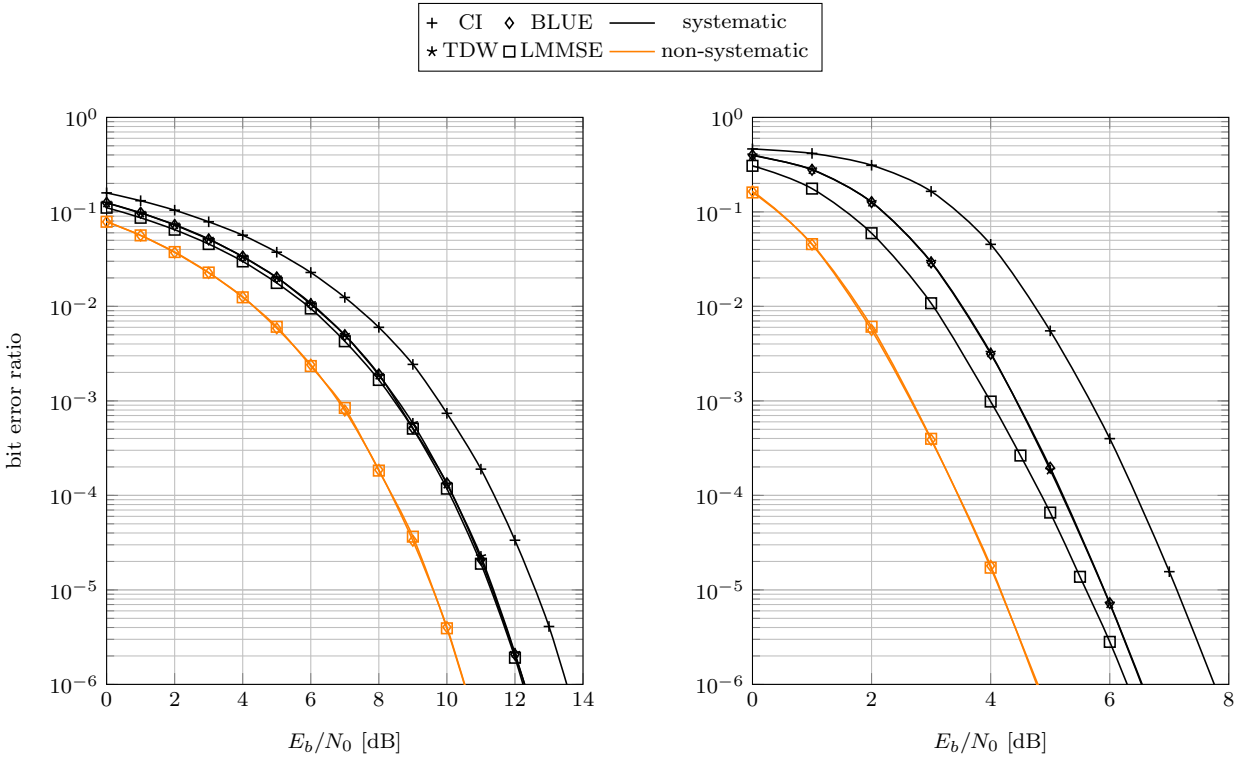
Non-systematic UW-OFDM in combination with the BLUE/LMMSE estimator significantly outperforms the systematic UW-OFDM counterparts by almost 2 dB. Both, the BLUE and the LMMSE estimator however, show the exact same performance in the non-systematic setup. Very similar observations can be made for coded transmission, with equivalent performance for BLUE and LMMSE estimator.



(a) UW-OFDM system A, uncoded, AWGN

(b) UW-OFDM system A, coded, AWGN

Figure 3.2.: BER performance of linear data estimators for system A according to Table A.1 in the AWGN channel.



(a) UW-OFDM system B, uncoded, AWGN

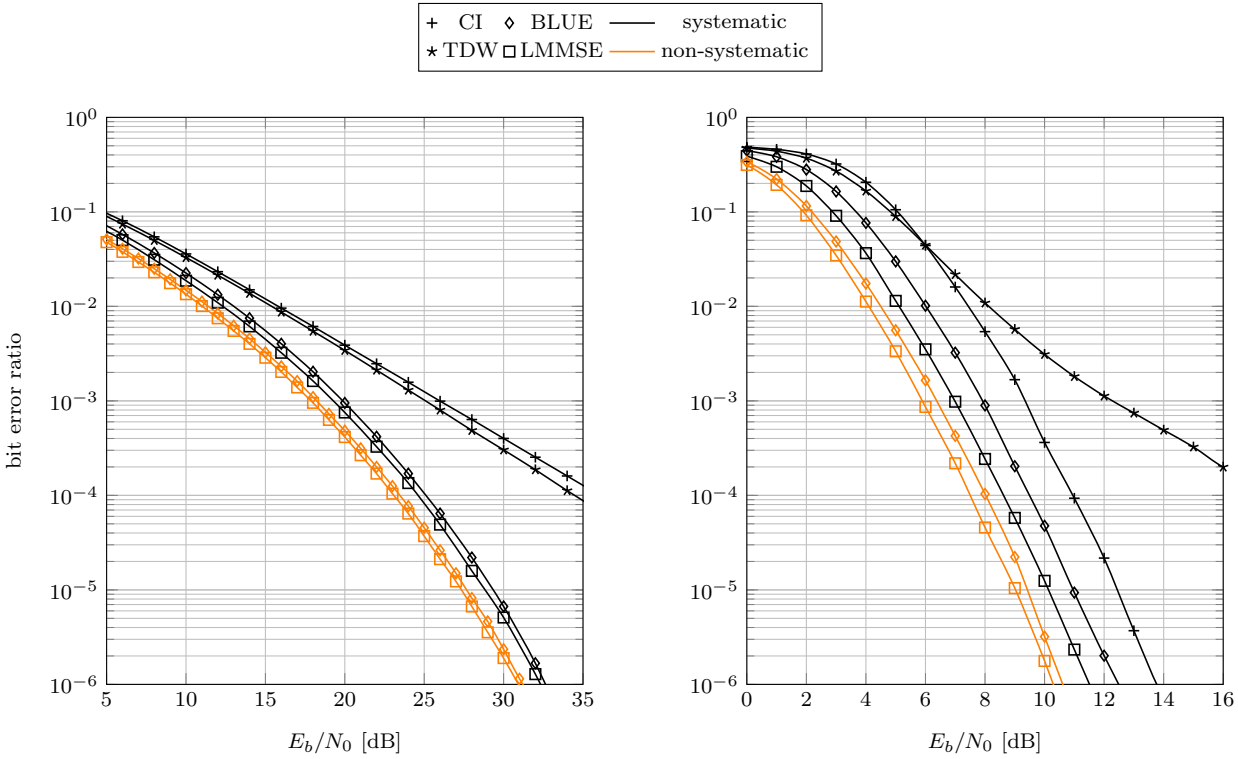
(b) UW-OFDM system B, coded, AWGN

Figure 3.3.: BER performance of linear data estimators for system B according to Table A.1 in the AWGN channel.

When interpreting the results for the multipath environment in Figure 3.4 and 3.5 similar trends as in the AWGN case can be observed, but now the LMMSE estimator and the BLUE clearly outperform the other estimators. For uncoded transmission, the TDW outperforms the CI estimator by more than 2 dB (again at a BER of 10^{-5}, which is cut off in the figures in order to increase the visibility for the other estimators). The BLUE and LMMSE estimator show a significantly faster decay of the BER than the CI and TDW, resulting in a more and more increasing gain of already more than 10 dB. This documents some coding gain for the BLUE and the LMMSE estimator, which originates from the exploitation of the RS code by means of the redundant subcarriers that were introduced by the UW-OFDM symbol generation process. At a BER of 10^{-5}, the LMMSE estimator still shows a gain of about 0.4 dB over the BLUE in both systems A and B, which narrows with increasing E_b/N_0 as explained for the AWGN case.

For coded transmission, the CI, BLUE and LMMSE estimators perform with almost constant gain of about 1.6 dB (BLUE over CI) and 1 dB (LMMSE over BLUE) for setup A and 1.4 and 0.8 dB for setup B, respectively. Somewhat unexpected and in contrast to the uncoded results and those in the AWGN channel, the TDW shows a very bad BER performance. Judging from the uncoded results, where the TDW outperforms the CI estimator, this should also be visible in the coded case. A deeper analysis reveals that the gain of the TDW estimator in the uncoded case, especially at low E_b/N_0-ratios that are relevant for coded transmission, depends a lot on the channel realization. For certain channel conditions, the TDW estimator even has to endure a performance loss, compared to the CI estimator. As the BER degradation is only small for most channels, the TDW supersedes the CI estimator in the uncoded case *on average*. However, a small degradation of the data estimates at low E_b/N_0-ratios is able to cause a quite big degradation of the BER, if measured after the channel decoder. Some 'severe' channels cause a small degradation of the uncoded BER, but impair the coded BER severely. Those channels dominate the coded BER performance and cause the unexpected loss of the coded TDW estimator against CI. A more detailed explanation and analysis of why the TDW performs worse than the CI estimator for some channels can be found in [HOH11].

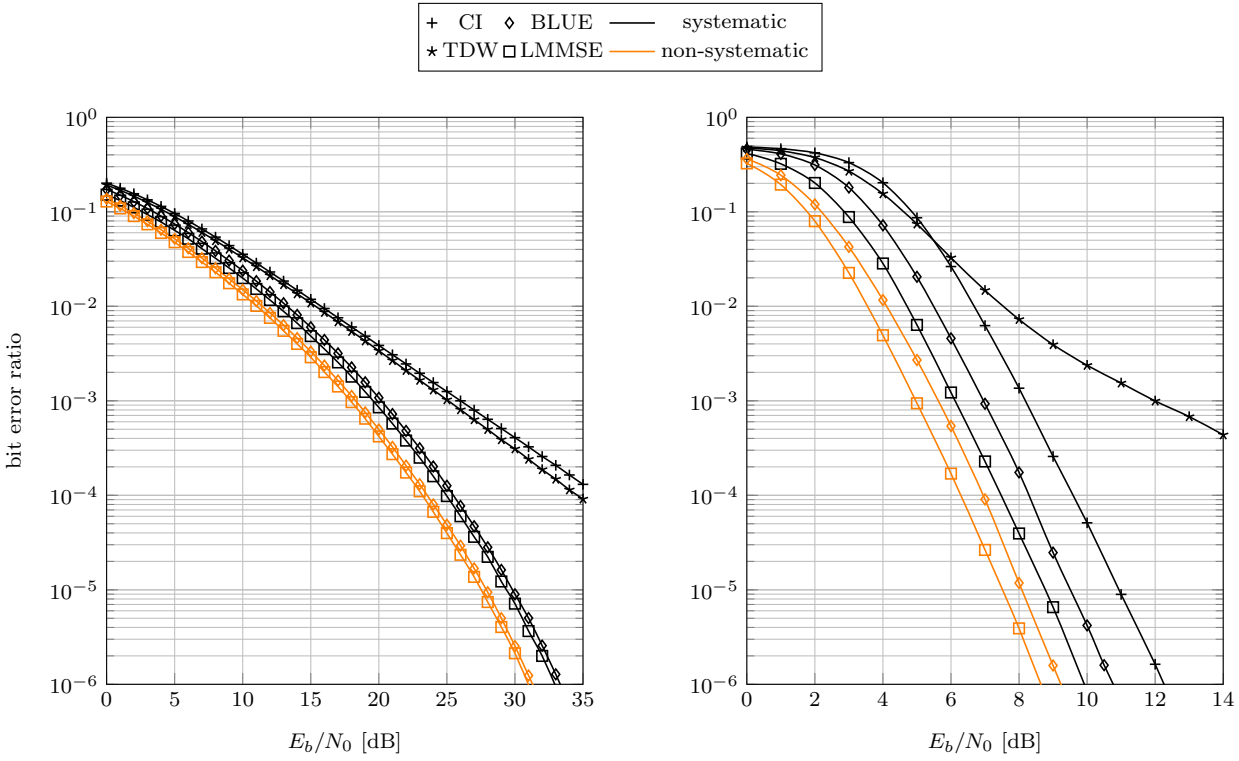
The systems that generate the UW-OFDM symbols non-systematically show the best overall performance. Besides spreading each data symbol over the whole OFDM symbol duration, as with the systematic approach, the non-systematic systems also spread each data symbol over the whole system bandwidth, making it robust against severe fading additionally. The non-systematic UW-OFDM systems outperform their systematic counterparts by 1.6 dB for the BLUE and 1.5 dB for the LMMSE with uncoded transmission and by 1.0 and 1.5 dB, respectively, with coded transmission in system A. For system B these advantages stay in the same range at about 1.8 and 1.9 dB for uncoded transmission and to 1.4 and 1.2 dB, respectively, for coded transmission. Other than in the AWGN channel, BLUE and LMMSE are now clearly distinguishable from each other.



(a) UW-OFDM system A, uncoded, multipath

(b) UW-OFDM system A, coded, multipath

Figure 3.4.: BER performance of linear data estimators for system A according to Table A.1 in multipath environment.



(a) UW-OFDM system B, uncoded, multipath

(b) UW-OFDM system B, coded, multipath

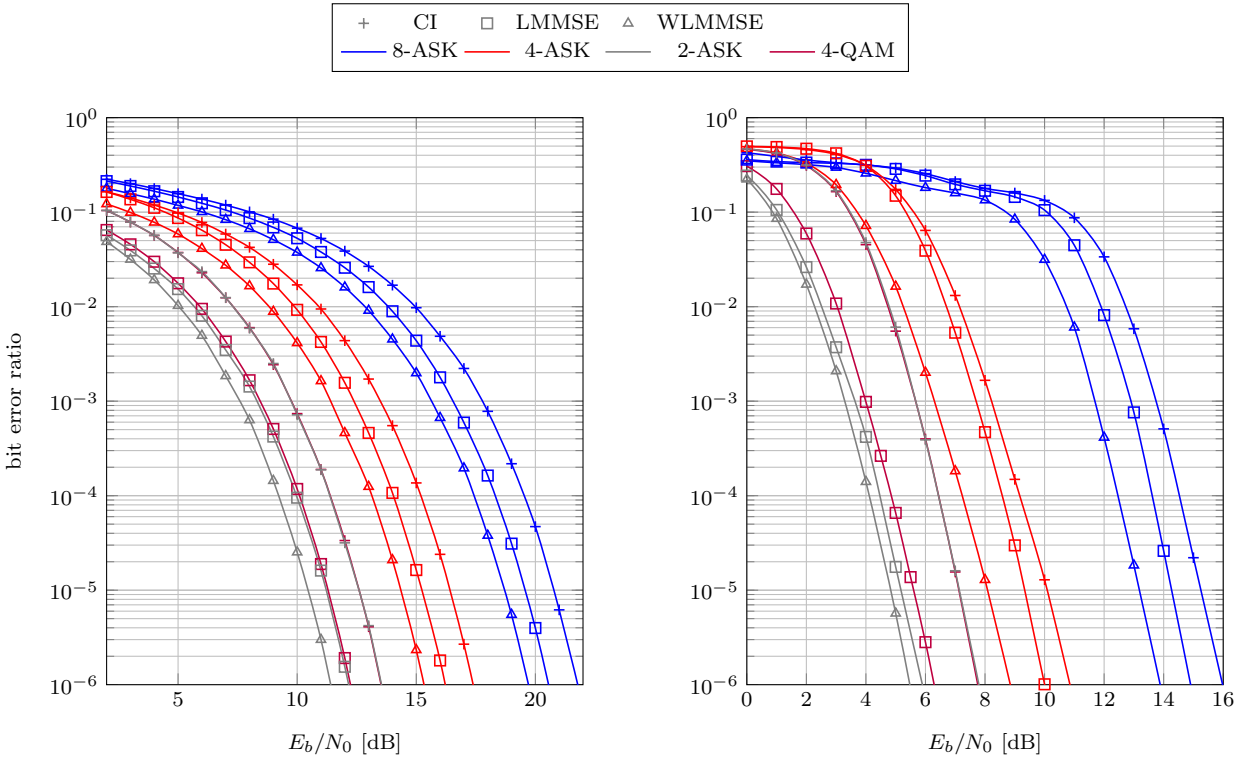
Figure 3.5.: BER performance of linear data estimators for system B according to Table A.1 in multipath environment.

3.4.2. Simulation Results of the Widely Linear Receivers

As, for the WLMMSE estimator, an improvement of the BER results is only expected for improper constellations, but 4-QAM is mainly utilized in this work, the simulation results are discussed separated from the other linear data estimators. In Figure 3.6 and 3.7 the bit error performance of the WLMMSE is compared to the CI and LMMSE estimators for ASK constellations. Despite of losing the whole imaginary dimension with ASK compared to QAM, what makes it mostly useless, 2-ASK is still worth to be considered from the implementation perspective: When using adaptive modulation, the modulator for a very bad subcarrier with very low SNR could switch from 4-QAM to 2-ASK. Then, besides gaining 3 dB in transmission quality, the very same filters can be utilized, while only the capability to transmit one bit on that particular subcarrier is lost.

On the left side of the Figures 3.6 and 3.7, the uncoded results are shown for the AWGN channel and multipath environment. For any ASK constellation, the WLMMSE supersedes the results of the LMMSE estimator by up to 0.9 and 4 dB in the AWGN channel and multipath environment, respectively. For the coded results on the right side, gains vary between 0.2 and over 1.7 dB, but always in favor of the WLMMSE estimator.

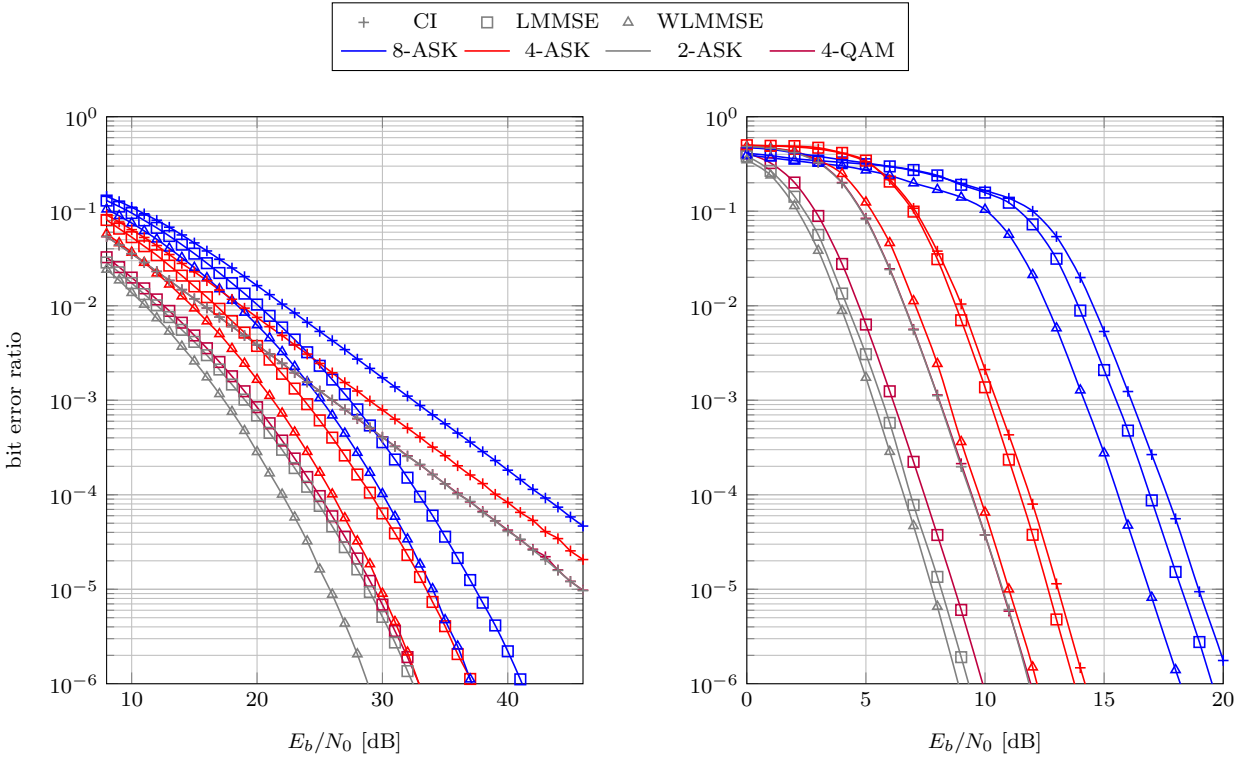
As a reference, the CI and LMMSE results for 4-QAM are included. In a BER-over-E_b/N_0 representation, ASK and 4-QAM feature similar performance. While for CP-OFDM these two systems perform exactly the same, they perform only equivalently for systematic UW-OFDM, if the CI estimator is used. For the LMMSE estimator, the 2-ASK constellations shows a small gain over 4-QAM of about 0.5 dB uncoded and 0.6 dB coded in the multipath environment. As the redundant subcarriers are still complex valued for 2-ASK, more redundancy is introduced on average *per bit* in the process of the UW-OFDM symbol generation, compared to 4-QAM. As seen before, the LMMSE estimator is quite good at utilizing this redundancy in order to recover the data symbols, and hence, the 2-ASK estimates are a bit more accurate than those of the complex constellation. The CI estimator ignores all redundancy and performs equally for 2-ASK and 4-QAM. Together with the additional gain of the WLMMSE over the LMMSE estimator, this makes transmission with 2-ASK compared to 4-QAM a much safer transmission mode for systematic UW-OFDM.



(a) uncoded, AWGN channel

(b) coded, AWGN channel

Figure 3.6.: WLMMSE performance compared to CI and LMMSE estimator for system B in the AWGN channel, according to Appendix A.



(a) uncoded, multipath environment

(b) coded, multipath environment

Figure 3.7.: WLMMSE performance compared to CI and LMMSE estimator for system B in multipath environment, according to Appendix A.

3.4.3. Simulation Results of the Bayesian Receivers with Symbol Scaling Compensation

In this section, the performance of the Bayesian estimators with symbol scaling compensation are compared to their regular counterparts, with relevant constellations. As mentioned in Section 3.2.4, the scaling compensation only has an effect for higher order transmit constellations, which are 4-ASK or higher for ASK constellations and 16-QAM or higher for QAM constellations. In this work, in general 4-QAM is used for the simulations, with the exception of some ASK constellations to show the potential of the widely linear MMSE estimators. In order to show the potential of the scaling compensated Bayesian receivers, the results for 16-QAM, 4- and 8-ASK are presented, in this section. The results of the BLUE are included as a reference.

For uncoded transmission, only a very small gain of the estimators with symbol scaling compensation can be observed. The gain even vanishes with increasing E_b/N_0-ratio and is only noticeable at low SNR, as predicted in Section 3.2.4. Figures as well as a detailed discussion for uncoded transmission are skipped here, as the maximum gain in multipath environment at a BER of 10^{-5} achieved by the scaling compensation for WLMMSE does not exceed 0.1 dB for any constellation. Symbol scaling compensation for the LMMSE is rewarded with up to 0.15 dB. In the AWGN channel the gains are even less.

In coded transmission, however, the gain by introducing symbol scaling compensation is quite remarkable. First of all it is noteworthy that the BLUE outperforms the regular LMMSE for the considered higher order constellations. This is in accordance to the theory, as the BLUE suffers neither from the symbol scaling effect, nor from ISI. When introducing symbol scaling compensation, some nice gains can be achieved: As seen for transmission in the AWGN channel in Figure 3.8, the scaling compensated LMMSE with 16-QAM improves the regular LMMSE by up to 0.5 dB at low SNR. At a BER of 10^{-5}, the symbol scaling compensated LMMSE supersedes the regular one still by 0.4 dB, whereas the BLUE only gains 0.1 dB over the LMMSE without ssc. With 4-ASK, the scaling compensation improves the results by 0.74 dB at a BER of 10^{-5}. The regular WLMMSE estimator is another 0.5 dB better than the scaling compensated LMMSE, and superseded by the scaling compensated WLMMSE by another 0.22 dB. The BLUE only gains 0.48 dB against the regular LMMSE. Very similar results are observed using 8-ASK. While the BLUE supersedes the regular LMMSE by 0.24 dB, the symbol scaling compensation achieves 0.3 dB gain for the LMMSE over the regular version and only 0.06 dB for the WLMMSE, which is again by far better than both LMMSE estimators.

In the multipath environment, shown in Figure 3.9, the achievable gain by symbol scaling compensation numbers to 0.4 dB for the LMMSE with 16-QAM, compared to the BLUE superseding the LMMSE by only 0.1 dB. With 4-ASK, the gain increases to impressive 1.4 dB, when regarding the LMMSE, and 0.7 dB for the BLUE. Still ahead by far is the scaling compensated WLMMSE estimator, which outperforms its non-compensated counterpart by 0.45 dB. With 8-ASK, the scaling compensation for the LMMSE performs almost 0.6 dB better than the regular version, while the BLUE improves it by only 0.4 dB. Above that, the better performing regular WLMMSE is only outperformed by the WLMMSE with scaling compensation by 0.1 dB.

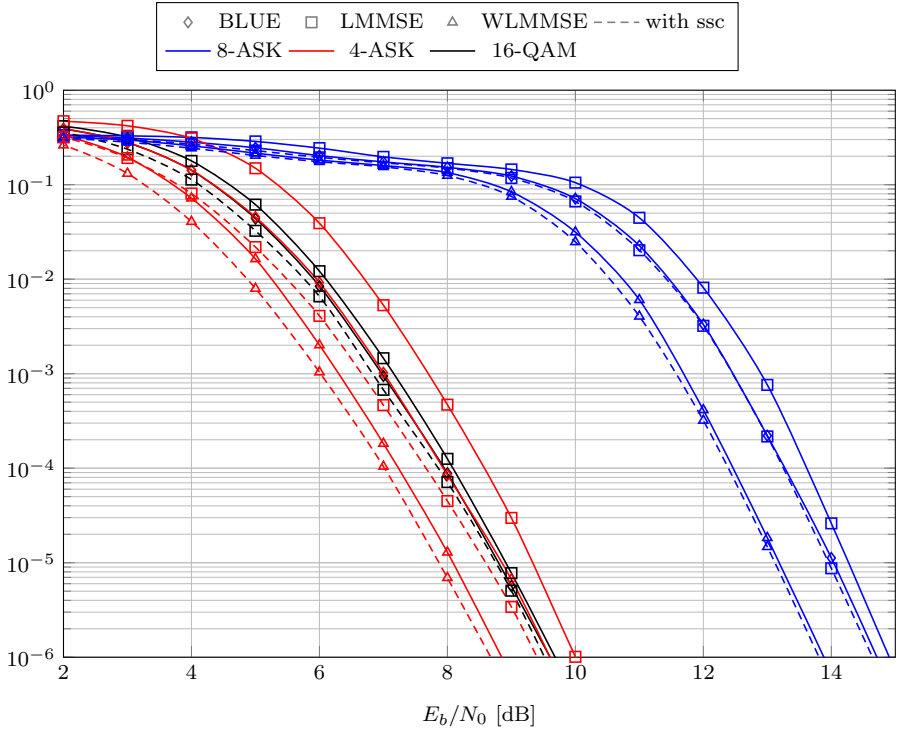


Figure 3.8.: Performance of the Bayesian estimators LMMSE and WLMMSE with and without symbol scaling compensation, compared to the BLUE for coded system B in the AWGN channel, according to Appendix A.

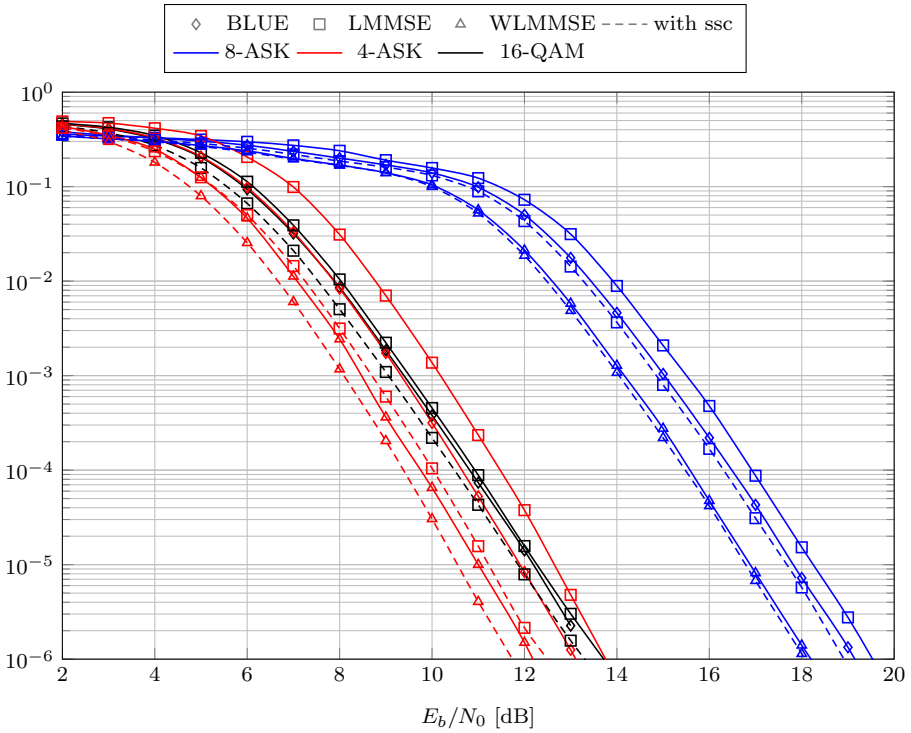


Figure 3.9.: Performance of the Bayesian estimators LMMSE and WLMMSE with and without symbol scaling compensation, compared to the BLUE for coded system B in multipath environment, according to Appendix A.

3.5. Summary of Linear Receivers

In this chapter, several linear data estimators specifically designed for UW-OFDM were investigated. Data estimators following the principles of classical estimation theory were introduced, which lead to ZF equalizers. Two simple and intuitive ZF equalizers and the optimum ZF equalizer constituting the BLUE have been discussed. Following the Bayesian estimation principle, the LMMSE estimator has been presented. Highly complexity reduced versions of these estimators were derived and their complexity investigated in terms of equivalent complex multiplication counts. The complexity optimized BLUE and LMMSE estimator versions feature considerably reduced CME counts compared to their straightforward counterparts, but still they show a significantly higher complexity compared to the simple ZF solutions. Furthermore, the WLMMSE estimator was derived, that poses a better performing alternative to the LMMSE for improper data symbol constellations, such as ASK.

Simulations for the AWGN channel give an initial idea of the performance of the estimators. Not unexpectedly, the more complex estimators BLUE and LMMSE perform best and the TDW estimator supersedes the very simplistic CI estimator. The LMMSE yields the best BER performance. The simulations in the multipath environment revealed that the more complex estimators BLUE and LMMSE are able to exploit the correlations on the redundant subcarriers effectively and yield a coding gain, outperforming CI and TDW by far. The TDW estimator presents itself as a reasonable alternative to the CI estimator, but only for the uncoded case. In coded transmission however, and on average over many multipath channels, the TDW estimator is not able to compete with all other estimators. This result is unexpected.

When using an ASK symbol alphabet, the WLMMSE receiver is able to further improve upon the already very good results of the LMMSE estimator. While, for CP-OFDM as well as for UW-OFDM in combination with the CI estimator, 2-ASK and 4-QAM perform exactly the same in terms of a BER-over-E_b/N_0 representation, 2-ASK outperforms 4-QAM for systematic UW-OFDM using an LMMSE estimator. Together with the gain due to WLMMSE estimation, 2-ASK proves to be a quite safe communication mode in the framework of systematic UW-OFDM.

In general, Bayesian estimators show a scaling effect, which is due to a bias, when considering the estimators' operation in the classical sense. If higher order constellations are used, which is n-QAM with $n \geq 16$ or n-ASK with $n \geq 4$, the symbol scaling effect of the LMMSE and WLMMSE data estimators has a severe and observable impact on the bit error performance. This is not the case for the BLUE, which is derived using classical estimation theory. Whenever higher order constellations are used it makes sense to switch to the BLUE, or even better, to employ a symbol scaling compensation for the LMMSE or WLMMSE estimators, in order to improve the bit error probability considerably. For higher order ASK constellations however, switching from the LMMSE to the WLMMSE yields a much higher gain than introducing the symbol scaling compensation.

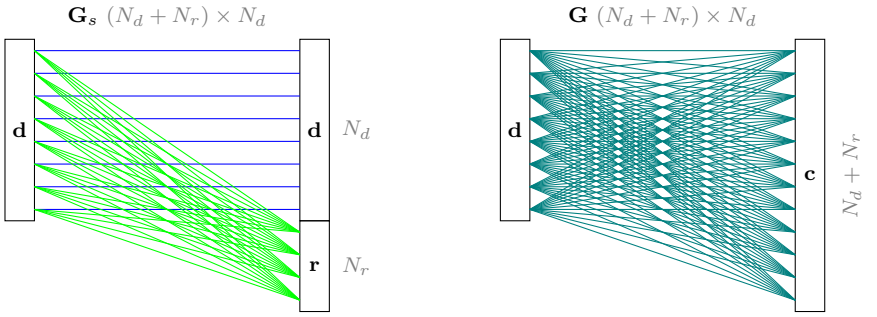
4. Nonlinear Receivers for Unique Word OFDM

In this chapter, several nonlinear receiver structures are discussed, which all base on the linear UW-OFDM system model in (2.37). To simplify notation, the context to UW-OFDM is dropped partially and the UW-OFDM symbol generator matrix $\mathbf{G}$ and channel frequency response $\widetilde{\mathbf{H}}$ (zero subcarriers omitted) are treated together as one matrix $\mathbf{H} = \widetilde{\mathbf{H}}\mathbf{G}$ with the dimensions $(N_d + N_r) \times N_d$. The same model then becomes

$$\mathbf{y} = \mathbf{H}\mathbf{d} + \mathbf{n}. \tag{4.1}$$

With this linear system model and the properties of the involved vectors and matrices, it is possible to read this chapter independently from the general UW-OFDM system description. The receive vector $\mathbf{y} \in \mathbb{C}^{(N_d+N_r)\times 1}$ is calculated with a channel matrix $\mathbf{H} = \widetilde{\mathbf{H}}\mathbf{G} \in \mathbb{C}^{(N_d+N_r)\times N_d}$ and the data vector $\mathbf{d} \in \mathcal{A}^{N_d\times 1}$ with elements from the transmit alphabet $\mathcal{A}$ – most commonly QAM. The vector $\mathbf{n} \sim \mathcal{N}_\mathrm{C}\left(0; N\sigma_n^2\mathbf{I}\right)$ of length $N_d + N_r$ is uncorrelated complex white Gaussian noise with zero mean and each value with variance $N\sigma_n^2$, where N is the size of the DFT used for the UW-OFDM processing, and σ_n^2 the actual noise variance in time domain.

This channel matrix $\mathbf{H}$ can now be interpreted as the channel matrix of a complex MIMO channel with N_d transmit and $N_d + N_r$ receive antennas, as the dimensions suggest. However, the channel matrix $\mathbf{H}$ is not fully occupied like in a general MIMO model for systematic UW-OFDM. The systematic UW-OFDM model implies a structure as outlined in Figure 4.1a with the sorted $\mathbf{G}_s$ as in (3.3). Sub-figure (b) displays the same using the altered $\mathbf{G}$ for non-systematic UW-OFDM and $\mathbf{c}$ as in (2.57), where all values carry information and redundancy.



(a) systematically generated UW-OFDM

(b) non-systematically generated UW-OFDM

Figure 4.1.: Interpretation of the UW-OFDM generator matrix as a MIMO system.

4.1. Noise Interpolation

For a linear data estimation, as a result from the last Chapter, a linear estimator matrix $\mathbf{E}$ and the statistics of the remaining estimation error in form of its covariance matrix $\mathbf{C}_{ee}$ are at hand. An inspection of $\mathbf{C}_{ee}$ reveals that its off-diagonals are generally non-zero, which means that, after the linear estimation, the estimation error values $\mathbf{e} = \mathbf{d} - \tilde{\mathbf{d}}$ are correlated, which is also identified as *colored* error values. Their correlation indicates that, if an error value is known, knowledge about other error values is gained as well. This knowledge is to be exploited by a receiver called "noise filtering" in general. In this work the noise filtering concept will be narrowed down to *noise interpolation* (NI).

Exploiting error statistics in receiver systems has been done for a long time. Regardless, if it is called noise prediction, noise interpolation, noise whitening or precoding, in SC/FDE and MIMO systems, it refers usually to similar techniques that are based on the Wiener filter theory. Before applying it to the purposes of noise filtering for UW-OFDM, the theory is recapitulated briefly.

4.1.1. Optimal Wiener Filter

In this section, the basics of general Wiener filtering are briefly recapitulated, in order to offer the tools to derive the noise filters. Wiener filtering is covered in countless text books, exemplarily [Wie49, Orf88, Kay93, Zak05, Hay96, Kam04]. While this technique is referred to as Wiener or noise *filtering* as a general term, this work addresses the special cases widely known as *interpolation* and *smoothing*.

In traditional Wiener filtering problems like prediction, interpolation or smoothing, signal values are observed in a time based manner one after another. In this work, block processing is employed, which means that a block of N observed values $\mathbf{x} = [x_0, x_1, \ldots, x_{N-1}]^\mathsf{T}$ is considered for filtering. These observations $\mathbf{x} = \mathbf{s} + \mathbf{u}$ consist of additive noise $\mathbf{u}$ and the desired signal $\mathbf{s}$. However, neither the desired signal values in $\mathbf{s}$, nor the noise values in $\mathbf{u}$ can be considered white[1], but their second-order and thus auto-correlation statistics are known.

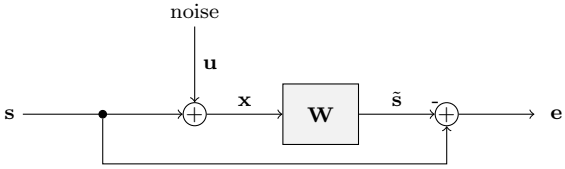


Figure 4.2.: Wiener filtering system model.

The task of Wiener filtering is to find a good estimate $\tilde{s}_k$ for a single value s_k by linear combinations of the observed vector $\mathbf{x}$ with the coefficients $\mathbf{w}^{(k)} = \left[w_0^{(k)}, w_1^{(k)}, \ldots, w_{N-1}^{(k)}\right]^\mathsf{T}$.

[1] If the values in a vector $\mathbf{a}$ are white, each of its values a_k is uncorrelated to any other value $a_l, l \neq k$.

These weights need to be determined in an optimum way and are used to easily compute the estimate number k by

$$\begin{aligned}\tilde{s}_k &= \sum_{n=0}^{N-1} w_n^{(k)} x_n \\ &= \mathbf{w}^{(k)\mathsf{T}} \mathbf{x}.\end{aligned} \tag{4.2}$$

Good coefficients in the Wiener sense are found, if the power of the estimation error $\mathrm{E}\left\{|e_k|^2\right\} = \mathrm{E}\left\{e_k e_k^*\right\}$ with

$$e_k = s_k - \tilde{s}_k = s_k - \mathbf{w}^{(k)\mathsf{T}} \mathbf{x} \tag{4.3}$$

is minimized. This can be determined by setting the Wirtinger derivative (that is its derivative with respect to all $w_l^{(k)*}$) [Fis02] to zero

$$\frac{\partial}{\partial w_l^{(k)*}} \mathrm{E}\left\{e_k e_k^*\right\} = \mathrm{E}\left\{ e_k \frac{\partial}{\partial w_l^{(k)*}} \left(s_k - \sum_{n=0}^{N-1} w_n^{(k)} x_n \right)^* \right\} \tag{4.4}$$

$$= -\mathrm{E}\left\{e_k x_l^*\right\} \stackrel{!}{=} 0, \quad \forall l = 0, 1, \ldots, N-1, \tag{4.5}$$

which is known as the *orthogonality principle*. From this, it can be noted that the observations and the estimation error are forced to be orthogonal and uncorrelated, which furthermore yields exactly N of the so-called *normal equations* by expanding e_k

$$\begin{aligned}\mathrm{E}\left\{ \left(s_k - \mathbf{w}^{(k)\mathsf{T}} \mathbf{x} \right) x_l^* \right\} &= 0, \\ \mathrm{E}\left\{s_k x_l^*\right\} &= \mathbf{w}^{(k)\mathsf{T}} \mathrm{E}\left\{\mathbf{x} x_l^*\right\}, \\ \forall l &= 0, 1, \ldots, N-1.\end{aligned} \tag{4.6}$$

This can be formulated for all N observations at once in vector notation as

$$\mathrm{E}\left\{ s_k \mathbf{x}^{\mathsf{H}} \right\} = \mathbf{w}^{(k)\mathsf{T}} \mathrm{E}\left\{ \mathbf{x}\mathbf{x}^{\mathsf{H}} \right\}, \tag{4.7}$$

which yields

$$\mathbf{w}^{(k)\mathsf{T}} = \mathrm{E}\left\{ s_k \mathbf{x}^{\mathsf{H}} \right\} \mathrm{E}\left\{ \mathbf{x}\mathbf{x}^{\mathsf{H}} \right\}^{-1}. \tag{4.8}$$

Now, the cross-correlation vector between the desired signal sample and the observation vector and the autocorrelation[2] matrix of the observations can be identified as

$$\begin{aligned}\mathbf{r}_{sx}^{(k)\mathsf{T}} &= \mathrm{E}\left\{ s_k \mathbf{x}^{\mathsf{H}} \right\}, \qquad \mathbf{r}_{sx}^{(k)\mathsf{T}} \in \mathbb{C}^{1 \times N} \\ \mathbf{R}_{xx} &= \mathrm{E}\left\{ \mathbf{x}\mathbf{x}^{\mathsf{H}} \right\}, \qquad \mathbf{R}_{xx} \in \mathbb{C}^{N \times N},\end{aligned} \tag{4.9}$$

[2] Correlation and covariance statistics are equivalent, when the random variables are considered zero mean, which is the case here.

such that

$$\mathbf{w}^{(k)\mathsf{T}} = \mathbf{r}_{sx}^{(k)\mathsf{T}} \mathbf{R}_{xx}^{-1}. \tag{4.10}$$

These statistical properties need to be known in order to determine the optimal Wiener coefficients for the estimation of signal value s_k. Now, after all N Wiener filters are computed, all of them can be put into an estimator matrix

$$\mathbf{W} = \left[\mathbf{w}^{(0)}, \mathbf{w}^{(1)}, \ldots, \mathbf{w}^{(N-1)}\right]^{\mathsf{T}} \tag{4.11}$$

of size $(N \times N)$, which is used to produce the whole vector of estimates in one equation by

$$\tilde{\mathbf{s}} = \mathbf{W}\mathbf{x}. \tag{4.12}$$

The vector $\mathbf{r}_{sx}^{(k)\mathsf{T}}$ can be also obtained by extracting the k-th row of $\mathbf{R}_{sx} = \mathrm{E}\left\{\mathbf{s}\mathbf{x}^{\mathsf{H}}\right\}$, which further simplifies the determination of the whole filter matrix at once to $\mathbf{W} = \mathbf{R}_{sx}\mathbf{R}_{xx}^{-1}$.

With the Wiener filter matrix $\mathbf{W}$, the power of each value of the estimation error vector $\mathbf{e} = \mathbf{s} - \tilde{\mathbf{s}}$ is minimized, as apparent on the main diagonal of the error covariance matrix $\mathbf{C}_{ee} = \mathrm{E}\left\{\mathbf{e}\mathbf{e}^{\mathsf{H}}\right\}$. Being aware of the fact that $\mathbf{R}_{xs} = \mathbf{R}_{sx}^{\mathsf{H}}$, it can be determined as

$$\begin{aligned}
\mathbf{C}_{ee} &= \mathrm{E}\left\{\mathbf{e}\mathbf{e}^{\mathsf{H}}\right\} = \mathrm{E}\left\{\mathbf{e}\mathbf{s}^{\mathsf{H}}\right\} - \overbrace{\mathrm{E}\left\{\mathbf{e}\mathbf{x}^{\mathsf{H}}\right\}}^{\overset{(4.5)}{=}0}\mathbf{W}^{\mathsf{H}} \\
&= \mathrm{E}\left\{\mathbf{s}\mathbf{s}^{\mathsf{H}}\right\} - \mathbf{W}\mathrm{E}\left\{\mathbf{x}\mathbf{s}^{\mathsf{H}}\right\} \\
&= \mathbf{R}_{ss} - \mathbf{W}\mathbf{R}_{sx}^{\mathsf{H}}.
\end{aligned} \tag{4.13}$$

It needs to be noted that this is the approach to estimate all values $s_k, k = 0, \ldots, N-1$, by using the set of all N observations. However, it might be reasonable not to utilize all N observations for the estimation of the desired sample s_k, but only a selected set of samples, whose indices are gathered in $\mathcal{S}_k$. Then, for the derivation of the Wiener filters, it is necessary to diminish the participating vectors and matrices in size, such that only the filter weights with indices in $\mathcal{S}_k$ are being calculated. The Wiener filter is derived exactly the same way as above and becomes analogous to (4.10)

$$\left[\mathbf{w}^{(k)\mathsf{T}}\right]_{\mathcal{S}_k} = \left[\mathbf{R}_{sx}\right]_{k,\mathcal{S}_k} \left(\left[\mathbf{R}_{xx}\right]_{\mathcal{S}_k,\mathcal{S}_k}\right)^{-1}, \tag{4.14}$$

where the filter vector $\mathbf{w}^{(k)\mathsf{T}}$ is filled only at the positions indicated by $\mathcal{S}_k$. All other positions are set to zero, such that the batch processing of the whole vector as in (4.12) is still possible. $\mathbf{w}^{(k)\mathsf{T}}$ is determined from $[\mathbf{R}_{sx}]_{k,\mathcal{S}_k}$, the k-th row of $\mathbf{R}_{sx}$ with the columns indicated by $\mathcal{S}_k$, and $[\mathbf{R}_{xx}]_{\mathcal{S}_k,\mathcal{S}_k}$, the submatrix of $\mathbf{R}_{xx}$ with only the rows in $\mathcal{S}_k$ and the columns in $\mathcal{S}_k$. One might also choose not to do Wiener filtering for every sample s_k, but only those with indices given in the set $\mathcal{T}$. In that case, all previous observations apply; only the estimator matrix $\mathbf{W}$ has zero rows at the indices not included in $\mathcal{T}$. In any case, the error covariance matrix can be still determined as shown in (4.13).

Depending on the specific application these equations may be further simplified. As stated in the beginning of this section the vectors $\mathbf{s}$ and $\mathbf{u}$ are mutually *uncorrelated*, implying $\mathrm{E}\left\{s_k u_l^*\right\} = 0, \forall k, l$. Also being aware that $\mathbf{x} = \mathbf{s} + \mathbf{u}$ the vector indicating the cross-correlation between all desired signal values $\mathbf{s}$ and the observations $\mathbf{x}$ can be further developed to

$$\begin{aligned}\mathbf{R}_{sx} &= \mathrm{E}\left\{\mathbf{s}\mathbf{x}^{\mathsf{H}}\right\} = \mathrm{E}\left\{\mathbf{s}\mathbf{s}^{\mathsf{H}}\right\} + \mathrm{E}\left\{\mathbf{s}\mathbf{u}^{\mathsf{H}}\right\} \\ &= \mathrm{E}\left\{\mathbf{s}\mathbf{s}^{\mathsf{H}}\right\} = \mathbf{R}_{ss}.\end{aligned} \tag{4.15}$$

Furthermore, the autocorrelation matrix of the observations becomes

$$\begin{aligned}\mathbf{R}_{xx} &= \mathrm{E}\left\{\mathbf{x}\mathbf{x}^{\mathsf{H}}\right\} \\ &= \mathrm{E}\left\{(\mathbf{s}+\mathbf{u})(\mathbf{s}+\mathbf{u})^{\mathsf{H}}\right\} \\ &= \mathbf{R}_{ss} + \mathbf{R}_{uu}.\end{aligned} \tag{4.16}$$

Finally, including these simplifications and considering the fact that $\mathbf{R}_{ss}^{\mathsf{H}} = \mathbf{R}_{ss}$, this section can be concluded with the summarizing results

$$\left[\mathbf{w}^{(k)\mathsf{T}}\right]_{\mathcal{S}_k} = [\mathbf{R}_{ss}]_{k,\mathcal{S}_k}\left([\mathbf{R}_{ss}+\mathbf{R}_{uu}]_{\mathcal{S}_k,\mathcal{S}_k}\right)^{-1}, k \in \mathcal{T}$$
$$\mathbf{W} = \left[\mathbf{w}^{(0)}, \mathbf{w}^{(1)}, \ldots, \mathbf{w}^{(N-1)}\right]^{\mathsf{T}}, \tag{4.17}$$

$$\tilde{\mathbf{s}} = \mathbf{W}\mathbf{x}, \tag{4.18}$$

$$\mathbf{C}_{ee} = \mathbf{R}_{ss} - \mathbf{W}\mathbf{R}_{ss} = (\mathbf{I} - \mathbf{W})\mathbf{R}_{ss}. \tag{4.19}$$

4.1.2. Wiener Noise Filtering

In this section, the results of Wiener filter theory from the last section are applied to develop methods for noise filtering. As, due to the block processing in OFDM, there are no causality problems as in time based filtering here, the filter can operate on any sample of the received OFDM symbol and estimate any noise value.

Prerequisite to noise filtering is that noise values show correlations among each other and their correlation statistics are known. This is the case after any linear data estimation, for example all linear estimators shown in Chapter 3, with the exception of non-systematically generated UW-OFDM in the AWGN channel, where the remaining noise vector is white. As the best performing linear receiver, an LMMSE estimation is chosen prior to noise filtering, according to (3.29)

$$\tilde{\mathbf{d}}^{\mathrm{lin}} = \mathbf{E}_{\mathrm{LMMSE}}\mathbf{y}. \tag{4.20}$$

The statistics of the actual error of the linear estimates

$$\mathbf{e} = \tilde{\mathbf{d}}^{\mathrm{lin}} - \mathbf{d}, \quad \mathbf{e} \in \mathbb{C}^{N_d \times 1} \tag{4.21}$$

are given by the error covariance matrix $\mathbf{C}_{ee}$ (3.30), while $\mathbf{e}$ itself is of course unknown as the correct transmit vector cannot be known at the receiver. Despite the uncertainty about $\mathbf{d}$ and hence $\mathbf{e}$, here it is pretended to be able to recover the data symbols perfectly by slicing without making a mistake. With this assumption, the estimate of the noise vector

$$\tilde{\mathbf{e}} = \tilde{\mathbf{d}}^{\mathrm{lin}} - \left\lfloor \tilde{\mathbf{d}}^{\mathrm{lin}} \right\rceil \tag{4.22}$$

is treated as the actual correct error vector, and it is assumed that the error statistics in $\mathbf{C}_{ee}$, as determined during the linear estimation, apply.

The slicing operation $\lfloor \cdot \rceil$ in (4.22) of course, which is the main reason of the nonlinearity of the noise filtering method, does not always detect the correct data symbols. In the presence of noise there is the chance that slicing of the linear estimate yields a data symbol other than the transmitted one, and a faulty error vector $\tilde{\mathbf{e}}$ may result. Hence, the observed error vector can be modeled as

$$\underbrace{\tilde{\mathbf{e}}}_{\text{observed error } \triangleq \mathbf{x}} = \underbrace{\mathbf{e}}_{\text{actual error } \triangleq \mathbf{s}} + \mathbf{v}. \tag{4.23}$$

If the slicing in (4.22) and thus the observed error values $\tilde{\mathbf{e}}$ are correct, the remaining noise $\mathbf{v}$ vanishes. In this case, the decoding does not produce any bit errors and an improvement cannot be expected from noise interpolation. Only if a false decision occurs, $\mathbf{v}$ is non-zero, which is visualized for a binary alphabet in Figure 4.3, and the decoding result has bit errors. This is the situation where the noise interpolation method is supposed to improve the linear estimate, such that the linear data estimate is pushed beyond the decision boundary into the domain of the correct data symbol.

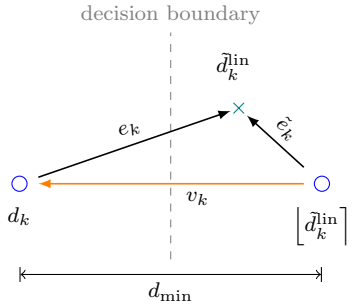


Figure 4.3.: Error values in (4.23) in case of a wrong decision.

At this point, the knowledge about the correlation of the error vector $\mathbf{e}$, quantified by the off-diagonal elements of $\mathbf{C}_{ee}$, can be exploited. In order to find a better noise estimate $\tilde{e}_k^{\mathrm{NI}}$ a Wiener filter can be applied, taking any observed error sample $\tilde{e}_l$ with $l = 0, 1, \ldots, N_d - 1$ into account. This noise estimate shall be subtracted from the initial linear estimate in order to get a hopefully improved value

$$\tilde{d}_k^{\mathrm{NI}} = \tilde{d}_k^{\mathrm{lin}} - \tilde{e}_k^{\mathrm{NI}}. \tag{4.24}$$

In this context, the vectors known from Wiener filter theory in Figure 4.2 can be identified as follows: The desired signal $\mathbf{s}$ is in this case the correct noise vector $\mathbf{e}$, the noisy measurements $\mathbf{x}$ correspond to $\tilde{\mathbf{e}}$ and the improved estimate $\tilde{\mathbf{s}}$ corresponds to $\tilde{\mathbf{e}}^{\mathrm{NI}}$.

Furthermore, some statistics for the remaining noise will be needed, denoted as $\mathbf{u}$ in Figure 4.2 and as $\mathbf{v}$ in (4.23). The autocorrelation matrix $\mathbf{C}_{vv} = \mathrm{E}\left\{\mathbf{v}\mathbf{v}^{\mathsf{H}}\right\}$ can be derived quite intuitively. First, it is assumed that there is no correlation between a possible wrong slicing of any linear estimate $\tilde{d}_l^{\mathrm{lin}}$ and the decision of the k-th value, if $l \neq k$. This allows to consider the samples of the remaining error $\mathbf{v}$ to be uncorrelated as well and all off-diagonal entries of its autocorrelation matrix $\mathbf{R}_{vv}$ can be set to zero. Second, the main diagonal indicates the mean power of the values of $\mathbf{v}$, which can be approximated. The mean power of noise value v_k is given by

$$\mathrm{E}\left\{|v_k|^2\right\} = \sigma_{v_k}^2 = (1-\chi_k)|v_{\mathrm{correct}}|^2 + \chi_k |v_{\mathrm{wrong}}|^2, \tag{4.25}$$

with the probability, that slicing of the linear estimate yields an incorrect decision χ_k (also called symbol error ratio, SER), and the contributing error values v_{correct} and v_{wrong} in case of correct or wrong decision. In case of a correct decision, the remaining error is zero, of course, and does not contribute to its power, thus $v_{\mathrm{correct}} = 0$. For a wrong decision, however, the remaining error (v_k in Figure 4.3) is simply the distance between the two symbols, with the value $v_{\mathrm{wrong}} = d_{\mathrm{min}} = |a_0 - a_1|, \quad a_n \in \mathcal{A}$, when using a binary transmit symbol alphabet. In this case, the power of v_k is then $\sigma_{v_k}^2 = \chi_k d_{\mathrm{min}}^2$. For higher order transmit alphabets it is assumed that only errors occur which are in minimum distance of the transmit symbols

$$v_{\mathrm{wrong}} = d_{\mathrm{min}} = \min_{\substack{a_k, a_l \in \mathcal{A} \\ a_k \neq a_l}} |a_k - a_l| .$$

In Appendix D it is reasoned that it is legitimate to neglect errors of higher distance, as the probability is noneffectively small. Therefore, for higher order constellations $\sigma_{v_k}^2 \approx \chi_k d_{\mathrm{min}}^2$ is only an approximation. The SER χ_k is the probability that slicing will yield an incorrect decision and can be approximated analytically, which is also explicated in Appendix D. Therewith, the main diagonal of $\mathbf{R}_{vv}$ can be built from the values

$$\begin{aligned} \sigma_{v_k}^2 &\approx \chi_k d_{\mathrm{min}}^2 \\ &\approx N_{\mathrm{min}} \mathrm{Q}\left(\sqrt{\frac{d_{\mathrm{min}}^2}{2\sigma_{e_k}^2}}\right) d_{\mathrm{min}}^2, \\ &\text{for } k = 0, 1, \ldots, N_d - 1, \end{aligned} \tag{4.26}$$

for any QAM constellation, where d_{min} and N_{min} depend on the used constellation size and scaling, and with $\sigma_{e_k}^2 = [\mathbf{C}_{ee}]_{k,k}$ being the variance of the complex noise on receive symbol $\tilde{d}_k^{\mathrm{lin}}$, which is apparent on the k-th entry of the main diagonal of $\mathbf{C}_{ee}$. With this, the matrix

$$\mathbf{R}_{vv} = \mathrm{diag}\left\{\sigma_{v_0}^2, \sigma_{v_1}^2, \ldots, \sigma_{v_{N_d-1}}^2\right\} \tag{4.27}$$

can be constructed. It needs to be noted that the mean and covariance statistics provide all information that is to know about a random vector, if it is Gaussian distributed. The vector $\mathbf{v}$ in this work, however, is far from Gaussian. The elements of $\mathbf{v}$ have a high probability to be zero and only a small chance to take one of a few possible error values. Hence, even if the determination of the covariance matrix $\mathbf{R}_{vv}$ is exact, Wiener filtering

is not exploiting the whole information about $\mathbf{v}$. This poses quite an inaccuracy for noise filtering.

After identifying the entities from Wiener filter theory in the noise filtering context and an analytical expression for the remaining error statistics, it can be summarized:

$$\begin{aligned}
\mathbf{s} &\rightarrow \mathbf{e} \\
\mathbf{x} &\rightarrow \tilde{\mathbf{e}} \\
\tilde{\mathbf{s}} &\rightarrow \tilde{\mathbf{e}}^{\mathrm{NI}} \\
\mathbf{u} &\rightarrow \mathbf{v} \\
\mathbf{W} &\rightarrow \mathbf{W}_{\mathrm{NI}} \\
\mathbf{R}_{ss} &\rightarrow \mathbf{C}_{ee} \\
\mathbf{R}_{uu} &\rightarrow \mathbf{R}_{vv}.
\end{aligned}$$

Translating the orthogonality principle (4.5) into these terms yields

$$\mathrm{E}\left\{\left(e_k - \tilde{e}_k^{\mathrm{NI}}\right)\tilde{e}_l^*\right\} = 0, \quad \forall k \in \mathcal{T}, l \in \mathcal{S}_k.$$

The difference of the estimated error after noise interpolation and the actual error $e_k - \tilde{e}_k^{\mathrm{NI}}$, which should be preferably vanishing, is supposed to be orthogonal to any sample of the observed error vector $\tilde{e}_l$ that is at hand after the linear estimation and the slicing.

With these tools one should be able to optimally estimate error sample $\tilde{e}_k^{\mathrm{NI}}$. However, the choice of the samples from which $\tilde{e}_k^{\mathrm{NI}}$ is estimated is crucial; their selection will be dealt with later. For now, the set $\mathcal{S}_k$ is introduced and contains the indices of all error samples to be used for noise filtering of error sample $\tilde{e}_k^{\mathrm{NI}}$.

Elaborating the filter equations in order to determine the optimum Wiener filter for sample number k according to (4.17) further yields the filter coefficients

$$\left[\mathbf{w}^{(k)\mathsf{T}}\right]_{\mathcal{S}_k} = \mathbf{r}_k^{\mathsf{T}} \left(\mathbf{R}_k\right)^{-1}, \tag{4.28}$$

where $\mathbf{r}_k^{\mathsf{T}} = [\mathbf{C}_{ee}]_{k,\mathcal{S}_k}$ is the k-th row of $\mathbf{C}_{ee}$ with only the columns listed in $\mathcal{S}_k$. $\mathbf{R}_k = [\mathbf{C}_{ee} + \mathbf{R}_{vv}]_{\mathcal{S}_k,\mathcal{S}_k}$ is extracted from the matrix sum $\mathbf{C}_{ee} + \mathbf{R}_{vv}$ by selecting all the rows listed in $\mathcal{S}_k$ and all the columns listed in $\mathcal{S}_k$.

After estimation, the linearly equalized version of the receive vector is updated

$$\tilde{\mathbf{d}}^{\mathrm{NI}} = \tilde{\mathbf{d}}^{\mathrm{lin}} - \tilde{\mathbf{e}}^{\mathrm{NI}}, \tag{4.29}$$

in order to get an improved vector of data estimates.

4.1.3. Selection of Noise Samples

The starting point of the noise filtering procedure is always an initial estimation, where the remaining error is correlated, and the amount of the correlation is known. With a linear data estimator, good estimates $\tilde{\mathbf{d}}^{\mathrm{lin}}$ and the correlation statistics of the remaining error are obtained already. When applying noise filtering methods, those initial estimates shall be improved.

As mentioned earlier, the choice of samples that shall be improved (set $\mathcal{T}$), as well as the choice of samples that are used for their improvement (sets $\mathcal{S}_k$), is crucial to the performance of this method. A false decision when slicing $\tilde{d}_l^{\mathrm{lin}}$ in (4.22) must be absolutely avoided, thus the error value $\tilde{e}_l, l \in \mathcal{S}_k$ has to be as certain as possible. On the other hand the estimates shall be improved as good as possible before doing the final detection, in order to decrease the final BER. A very *good* initial estimate might not need any improvement and can be left as it is, in order to reduce the computational complexity. Then again a *bad* sample might even result in degradation of an estimate. What makes up a good or bad sample remains to be defined and will be investigated in this section.

In the end, one needs to be aware of the fact that the introduced Wiener filters are optimum in the MSE (means squared error) sense. This is not directly linked to the bit error probability, which is to be improved in the first place. A better MSE does not ensure a better BER as well. This makes most considerations in this section unfeasible to analyze analytically, and a more intuitive and rule-of-thumb-like approach becomes applicable.

Some criteria shall be defined in order to decide, whether or not to pick a noise sample to be estimated or to estimate from. First of all, the receiver can decide on sample-based criteria, by looking at their actual values. This approach, however, demands very high computational effort, as every single sample is examined. One sample-based method will be explained later in this section that yields very good BER results, but shall be excluded from further investigations in order to focus on stochastic rules that usually apply to a wider range of samples and possibly for more OFDM symbols at once. Second, noise samples can be picked for exceeding a defined threshold θ of any criterion, or because they are among the n best samples according to that criterion. This question will also be dealt with later.

Additionally, the optimal choice of values also depends on the current E_b/N_0 ratio. A couple of criteria to apply for sample selection follow in Section 4.1.3.2.

4.1.3.1. Interpolation vs. Smoothing

A special issue is the question, whether or not to use a noise sample $\tilde{e}_k$ to estimate $\tilde{e}_k^{\mathrm{NI}}$, to use *intrinsic* information. In Wiener filter theory the case where sample number k is not used to improve itself, and only *extrinsic* information is used, is called *interpolation*. The main diagonals of $\mathbf{C}_{ee}$ and $\mathbf{R}_{vv}$ in (4.28) are excluded in this case. The other case, where sample number k may be used for estimating $\tilde{e}_k^{\mathrm{NI}}$, is called *smoothing* [Kay93]. Smoothing is not considered in this work which can be reasoned with the results shown Figure 4.4, where the bit error performance for three noise filtering receivers in exemplary UW-OFDM system B with 4-QAM in a multipath environment, as introduced in Appendix A, are shown. For simplicity of this example, the receivers included in this simulation employ

the sample selection policy to estimate all N_d noise values. The noise filtering receiver denoted as "smoothing" uses all samples in $\tilde{\mathbf{e}}$ to estimate the full vector $\tilde{\mathbf{e}}^{\mathrm{NI}}$, while for the receiver called "interpolation", symbol number k is excluded from the list of symbols to be used for estimating $\tilde{e}_k^{\mathrm{NI}}$. For both, the approximation of $\mathbf{R}_{vv}$ according to (4.27) is used. The two receivers denoted as "simplified" smoothing and interpolation assume $\mathbf{R}_{vv} = 0$ and safe the effort to compute the matrix.

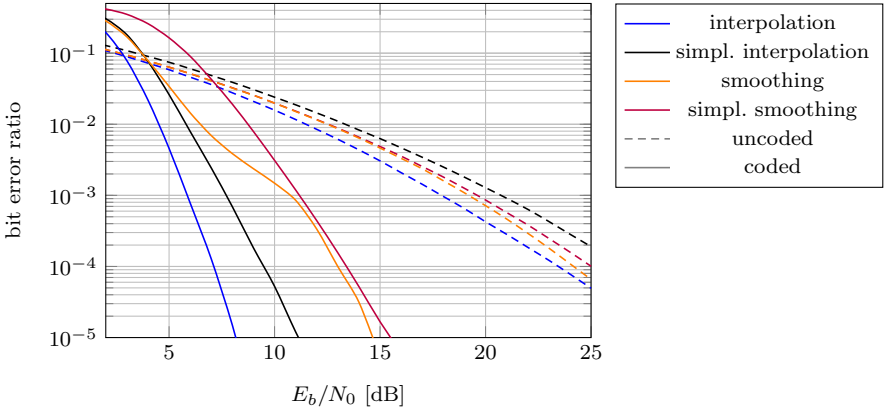


Figure 4.4.: BER performance for noise filtering receivers doing interpolation and smoothing, in uncoded and coded UW-OFDM system B in multipath environment.

The uncoded results show that the smoothers utilizing the approximated $\mathbf{R}_{vv}$ yield slightly better results than the smoother assuming $\mathbf{R}_{vv} = \mathbf{0}$. At a BER of 10^{-4}, this advantage is about 0.8 dB, and the simplified interpolator is outperformed by both smoothers by up to 2 dB. The presence of a gain tells that the smoothing approach is not wrong at all. The interpolator using the approximated $\mathbf{R}_{vv}$ yields an additional 0.7 dB gain over the smoother. Coded transmission relies on a correctly updated error covariance matrix $\mathbf{C}_{ee}$ after noise filtering, according to (4.19), which again depends on very accurate error statistics beforehand. Figure 4.4 suggests that the approximated statistics are not accurate enough for smoothing, as both coded smoothers fall far behind the interpolators, operating more than 3.5 dB worse, at a BER of 10^{-5}. From the definition in (4.26), the determination of $\mathbf{R}_{vv}$ was only an approximation of the actual probabilities that are not measurable. Surprisingly, if the sample to be estimated is excluded from the used noise samples, the approximated $\mathbf{R}_{vv}$ yields a much better result in coded interpolation. The interpolator with approximated $\mathbf{R}_{vv}$ supersedes the simplified interpolator by impressive 3 dB.

Although the interpolation method clearly achieves the best results for the given sample selection mode, *simplified noise interpolation* will be employed for noise filtering in this work, where no $\mathbf{R}_{vv}$ needs to be used at all. This decision removes the approximated $\mathbf{R}_{vv}$, and thus a severe uncertainty from the equation, and causes the noise statistics to be less sensible to errors and instabilities, due to multiple approximations. This issue is especially significant for iterative noise interpolation, introduced in Section 4.1.5, where the error statistics are used to update themselves multiple times and thus allow an approximation

error to propagate and amplify. Also, the investigations for this work showed that with the appropriate sample selection policies, the simplified interpolator can catch up very easily with the gain of the more accurate interpolation method. Only with the inferior sample selection rule, used to produce Figure 4.4, the interpolator with approximated $\mathbf{R}_{vv}$ is able to yield this impressive again. The assumption of $\mathbf{R}_{vv} = \mathbf{0}$ simplifies the filter determination, reduces computational complexity, enhances the mathematical stability and, in the end, has only to endure a very small performance loss.

4.1.3.2. Criteria for Noise Sample Selection

Up to now, it was discussed how to determine improved noise estimates, when the candidates are already picked. Besides excluding value $\tilde{e}_k$ to estimate $\tilde{e}_k^{\mathrm{NI}}$, these can be selected rather arbitrarily. In the following, some selection criteria are introduced.

Noise Variance Here, the variances of the actual error $\mathbf{e}$ (4.21) represent the qualifying measure. These variances are known in advance and present on the main diagonal of the error covariance matrix $\mathbf{C}_{ee}$ after linear estimation. The idea is that samples with higher variance are candidates to be estimated, while those with lower variance are safer and can be used for the estimation of others.

A-Priori Probability of Wrong Decision Here, the probability that the slicing of a linear estimate yields a different symbol than the actually transmitted one

$$\left\lfloor \tilde{d}_k^{\mathrm{lin}} \right\rceil \neq d_k$$

is taken as the selection criterion. In consideration of the computational complexity, the a-priori probabilities *before* observing the linear data estimates should be determined using the statistical information provided by the $\mathbf{C}_{ee}$ matrix. The probability that the slicing operation produces a wrong result is the analytical symbol error probability χ derived in Appendix D. Hence, it is named "SER criterion". The SER for symbol number k is according to (D.12)

$$\chi_k \approx N_{\min} \mathrm{Q}\left(\sqrt{\frac{d_{\min}^2}{2\sigma_{e_k}^2}} \right),$$

with $d_{\min}$ representing the distance to the nearest neighbor symbol and depending on the used QAM constellation and transmit power, $N_{\min}$ being constellation dependent as well and given in Table D.1, and $\sigma_{e_k}^2$ as the error variance, apparent on $[\mathbf{C}_{ee}]_{k,k}$.

Because the Q-function is strictly convex (see Figure D.4), a sorted list of χ_k will be in the exact same order as a list of the noise variances. Picking the n best or worst values from this list would yield the exact same results. Hence, the two criteria will only differ in performance when a threshold approach is chosen. Furthermore, the SER is a probability and thus definitely a number in the interval $[0; 1]$, which might be useful when selecting thresholds.

Correlation Lastly, it could make sense to pick noise samples, when they show high correlations to other samples, or are highly correlated to a sample selected for estimation. This information is also known in advance and present in $\mathbf{C}_{ee}$ excluding the main diagonal.

In Figure 4.5, the magnitude of the correlation coefficients in the $\mathbf{C}_{ee}$ matrix excluding the main diagonal of a typical UW-OFDM setup after LMMSE estimation is shown[3].

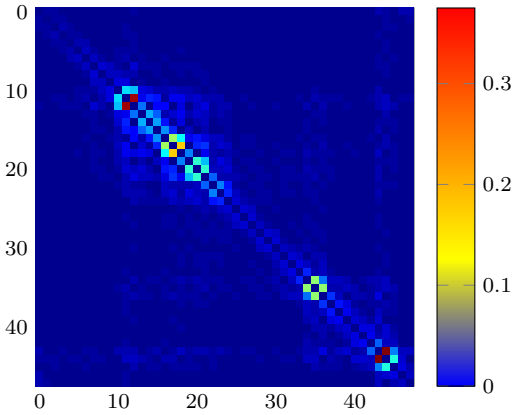


Figure 4.5.: $|\mathbf{C}_{ee}|$ after LMMSE estimation without main diagonal.

In simulations it turned out that this method does not give good answers on the question, how many samples or which thresholds to pick in order to achieve a good BER. Hence, it is not further investigated in this work.

Sample-based: A-Posteriori Probability of Wrong Decision In contrast to the earlier shown a-priori probabilities, also the a-posteriori probabilities of a wrong decision can be determined. This probability is determined under the knowledge of the linear data estimates and needs to be calculated per sample.

The a-posteriori probability that the symbol a_n from the alphabet $\mathcal{A}$ was transmitted as d_k, if the value $\tilde{d}_k^{\text{lin}}$ was received, can be expressed as the conditional probability $\Pr\left(d_k = a_n \middle| \tilde{d}_k^{\text{lin}}\right)$. Applying Bayes' law yields

$$\Pr\left(d_k = a_n \middle| \tilde{d}_k^{\text{lin}}\right) = \frac{p\left(\tilde{d}_k^{\text{lin}} \middle| d_k = a_n\right) \Pr\left(d_k = a_n\right)}{\sum_{l=0}^{|\mathcal{A}|-1} p\left(\tilde{d}_k^{\text{lin}} \middle| d_k = a_l\right) \Pr\left(d_k = a_l\right)}. \tag{4.30}$$

[3] system B in exemplary multipath channel A

In the present case of equiprobable transmit symbols $\Pr(d_k = a_n) = 1/|\mathcal{A}|$ this becomes furthermore

$$\Pr\left(d_k = a_n \middle| \tilde{d}_k^{\text{lin}}\right) = \frac{p\left(\tilde{d}_k^{\text{lin}} \middle| d_k = a_n\right)}{\sum_{l=0}^{|\mathcal{A}|-1} p\left(\tilde{d}_k^{\text{lin}} \middle| d_k = a_l\right)}. \tag{4.31}$$

Now, the probability of a wrong symbol decision can be determined as

$$p_{\text{wrong}} = 1 - \Pr\left(d_k = \left\lfloor \tilde{d}_k^{\text{lin}} \right\rceil \middle| \tilde{d}_k^{\text{lin}}\right) = 1 - \frac{p\left(\tilde{d}_k^{\text{lin}} \middle| d_k = \left\lfloor \tilde{d}_k^{\text{lin}} \right\rceil\right)}{\sum_{l=0}^{|\mathcal{A}|-1} p\left(\tilde{d}_k^{\text{lin}} \middle| d_k = a_l\right)}. \tag{4.32}$$

For the estimation of other samples, very safe samples with low p_{wrong} should be picked, while samples with a high p_{wrong} might be candidates for estimation from others.

By assuming that the true error $\tilde{d}_k^{\text{lin}} - d_k = e_k$ is normally distributed[4] with the statistics $e_k \sim \mathcal{N}_{\text{C}}\left(0; \sigma_{e_k}^2\right)$ with $\sigma_{e_k}^2 = [\mathbf{C}_{ee}]_{k,k}$, such that

$$p\left(\tilde{d}_k^{\text{lin}} \middle| d_k = a_n\right) = \frac{1}{\pi\sigma_{e_k}^2} \mathrm{e}^{-\frac{\left|\tilde{d}_k^{\text{lin}} - a_n\right|^2}{\sigma_{e_k}^2}}, \tag{4.33}$$

these probabilities can now be evaluated as follows:

$$p_{\text{wrong}} = 1 - \frac{\exp\left\{-\frac{\left|\tilde{d}_k^{\text{lin}} - \left\lfloor \tilde{d}_k^{\text{lin}} \right\rceil\right|^2}{\sigma_{e_k}^2}\right\}}{\sum_{l=0}^{|\mathcal{A}|-1} \exp\left\{-\frac{\left|\tilde{d}_k^{\text{lin}} - a_l\right|^2}{\sigma_{e_k}^2}\right\}} \tag{4.34}$$

The application of the Gaussian pdf with mean a_n in (4.33), however, is only correct for initial linear unbiased estimation like the BLUE, and also highly appropriate for the LMMSE with symbol scaling compensation. However, for small transmit alphabets, the assumption is also sufficiently accurate for the LMMSE estimator. Every received value $\tilde{d}_k^{\text{lin}}$ of every OFDM symbol has to be examined, for this method, what makes it computationally very complex. As it yields good results, this one is used as the only sample-based method as a coarse orientation for how good the methods based on stochastic measures perform in terms of the bit error probability. This will be shown in Section 4.1.6.

[4] The (proper) scalar complex Gaussian pdf (probability density function) with mean μ and variance σ^2 is given by $p(x) = \frac{1}{\pi\sigma^2}\mathrm{e}^{-\frac{|x-\mu|^2}{\sigma^2}}$, see (4.76).

4.1.4. Batch Filtering

In the last section, criteria were shown for how to make a selection of samples that are to be estimated and the samples that are used for estimation of those. In this section an approach is shown that is used for construction of a set of noise filters to be applied in one shot on the whole receive vector, hence its name "batch approach". This filter set could also be reused several times, if the channel statistics do not change quickly. The batch processing as well as the re-usability are measures which potentially reduce computational complexity.

The process is described in Algorithm 3, in which the set $\mathcal{T}$ contains the indices of all values that are to be estimated, while the values to be used for the estimation of the k-th value are gathered in the set $\mathcal{S}_k, k \in \mathcal{T}$.

Algorithm 3 Batch Noise Interpolation

1: **function** $[\tilde{\mathbf{d}}^{\mathrm{NI}}, \mathbf{C}_{ee}^{\mathrm{NI}}] = \mathrm{NI\_BATCH}(\tilde{\mathbf{d}}^{\mathrm{lin}}, \mathbf{C}_{ee}^{\mathrm{lin}})$
2: $\quad \mathbf{W}_{\mathrm{NI}} \leftarrow \mathbf{0}$
3: $\quad \tilde{\mathbf{e}} \leftarrow \tilde{\mathbf{d}}^{\mathrm{lin}} - \left\lfloor \tilde{\mathbf{d}}^{\mathrm{lin}} \right\rceil$
4: $\quad$ **for all** $k \in \mathcal{T}$ **do**
5: $\quad\quad$ **determine** $\mathcal{S}_k$ for the estimation of $\tilde{e}_k$
6: $\quad\quad [\mathbf{W}_{\mathrm{NI}}]_{k,\mathcal{S}_k} \leftarrow [\mathbf{C}_{ee}]_{k,\mathcal{S}_k} \left([\mathbf{C}_{ee} + \mathbf{R}_{vv}]_{\mathcal{S}_k,\mathcal{S}_k} \right)^{-1}$
7: $\quad$ **end for**
8: $\quad \tilde{\mathbf{d}}^{\mathrm{NI}} \leftarrow \tilde{\mathbf{d}}^{\mathrm{lin}} - \mathbf{W}_{\mathrm{NI}} \tilde{\mathbf{e}}$
9: $\quad \mathbf{C}_{ee}^{\mathrm{NI}} \leftarrow \mathbf{C}_{ee}^{\mathrm{lin}} - \mathbf{W}_{\mathrm{NI}} \mathbf{C}_{ee}^{\mathrm{lin}}$
10: **end function**

The filter determination in line 6 uses the vector and matrix defined in Section 4.1.2: $[\mathbf{C}_{ee}]_{k,\mathcal{S}_k}$ is the k-th row of $\mathbf{C}_{ee}$ with only the columns listed in $\mathcal{S}_k$. $[\mathbf{C}_{ee} + \mathbf{R}_{vv}]_{\mathcal{S}_k,\mathcal{S}_k}$ is extracted from the matrix sum $\mathbf{C}_{ee} + \mathbf{R}_{vv}$ by selecting all the rows listed in $\mathcal{S}_k$ and all the columns listed in $\mathcal{S}_k$. It needs to be emphasized that the error covariance matrix as determined in line 9 is only exact, when no slicing error occurs. In practical situations, this is just an approximation that is sufficiently accurate.

With this simple algorithm, one is ready to investigate the vast number of possible combinations of criteria for the selection of values to be estimated (set $\mathcal{T}$) and to estimate from (sets $\mathcal{S}_k$), introduced in the previous section. However, all these possible combinations and their BER dependency on the E_b/N_0 ratio, make it very difficult to give a comprehensive research of all possibilities. Furthermore, no general rule became apparent that allowed to break down all criteria to one optimum formula. Therefore, a few approaches that proved to perform well and the way to come to these solutions are given exemplarily. A final best candidate will be picked in the end that will be compared to the other receivers.

All simulations will be done in a multipath environment to obtain significant results. Also, a 4-QAM transmit symbol alphabet is used. It turned out that trying to improve *all* available values (that is $\mathcal{T} = \{0, 1, \ldots, N_d - 1\}$) is a good approach. Then in order to select $\mathcal{S}_k$, the two most intuitive selection approaches are shown, along with the best BER results.

4.1.4.1. Noise Variance Threshold for Sample Selection for Batch NI

In this section the criterion for the selection of the noise samples that are used for estimating others is the variance of the noise on each data symbol after linear estimation. The decision for selecting the set $\mathcal{S}_k$ is made upon a noise variance threshold that needs to be defined:

$$\mathcal{S}_k = \left\{ l \middle| \sigma_{e_l}^2 < \theta(\rho), l = \{0, \ldots, N_d - 1\} \backslash k \right\} \tag{4.35}$$

When trying to find a suitable noise variance threshold θ, the dependence on the SNR ρ needs to be taken into account. When determining the BER for different θ and E_b/N_0-ratios, the results can be visualized as in Figure 4.6, which shows the results of an exemplary UW-OFDM system B with 4-QAM and averaged over many multipath channels (see Appendix A). The two horizontal axes show the E_b/N_0 ratio in dB and the noise variance threshold θ. On the vertical z-axis the deviation of the BER relative to the best achieved BER at the given E_b/N_0-ratio, is displayed, in dependence on θ and in a logarithmic scale. Therefore, the best achieved ratio must be 1 for a given SNR and is visible as the deep blue valley.

In sub-figure (b) the contour plot of the same BER plane is displayed, making it much easier to identify the exact location of the minimum BER valley that is to be achieved.

As prerequisite for this work the E_b/N_0 ratio is assumed to be known at the receiver, which then can be respected by selecting an appropriate threshold. In order to meet the minimum BER as close as possible, the threshold can be selected by a function of $\rho_{\mathrm{dB}} = 10 \log_{10}(E_b/N_0)$, as shown by the red line. In Figure 4.7 and 4.8 the same plots are shown for the exemplary UW-OFDM systems SD and ML, which are described in Appendix A.

In each figure the red line was determined by a function that was found empirically by fitting a straight line into the logarithmic scaling of the E_b/N_0-axis and linear scaling of the threshold. In order to accommodate for different UW-OFDM system setups the function also depends on N_d:

$$\theta(\rho_{\mathrm{dB}}) = -0.015\rho_{\mathrm{dB}} + \log_{10}(N_d)/7 + 0.48 \tag{4.36}$$

4.1.4.2. SER Threshold for Sample Selection for Batch NI

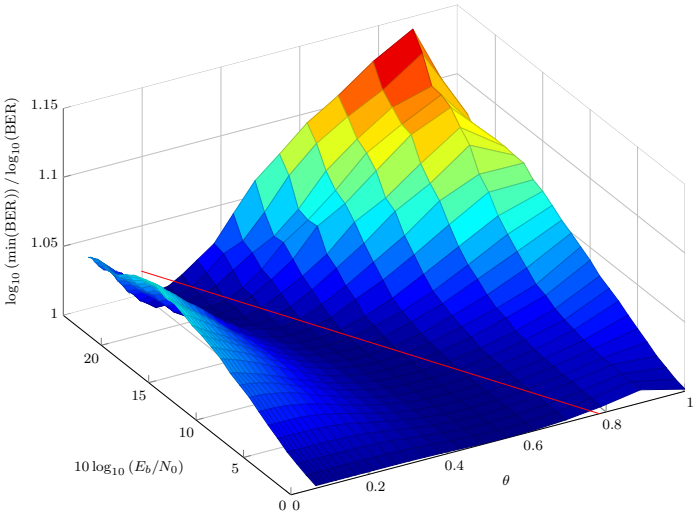
As second criterion to be investigated for batch noise interpolation, the SER threshold is chosen:

$$\mathcal{S}_k = \{ l | \chi_l < \theta(\rho), l = \{0, \ldots, N_d - 1\} \backslash k \} \tag{4.37}$$

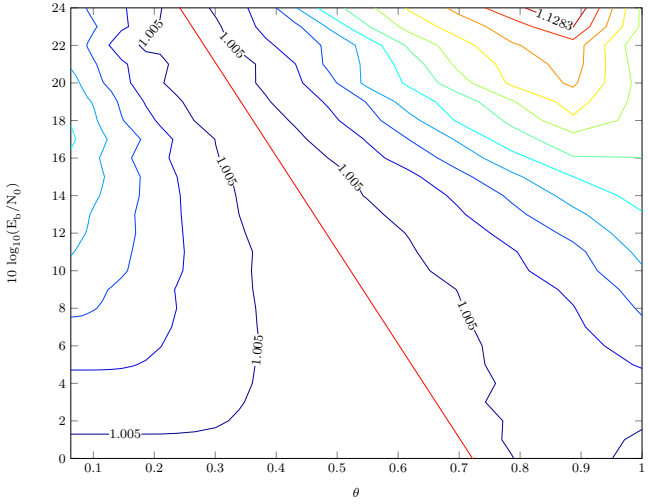
Following the same approach as for the noise variance threshold, an SER threshold function in $\rho_{\mathrm{lin}} = E_b/N_0$ was found. Again, the function is a straight line, only in a logarithmic representation of θ with linear display of E_b/N_0. By adapting slope and offset to the system size N_d the function was found empirically as

$$\theta(\rho_{\mathrm{lin}}) = 10^{\left((-N_d^{-1.65} - 0.002) \cdot \rho_{\mathrm{lin}} - N_d/80 - 0.7\right)}. \tag{4.38}$$

Again, the BER planes with the clearly visible minimum BER valley that shall be reached is shown for UW-OFDM system setups B, SD and ML in Figure 4.9, 4.10 and 4.11, with the red line being the evaluation of (4.38).

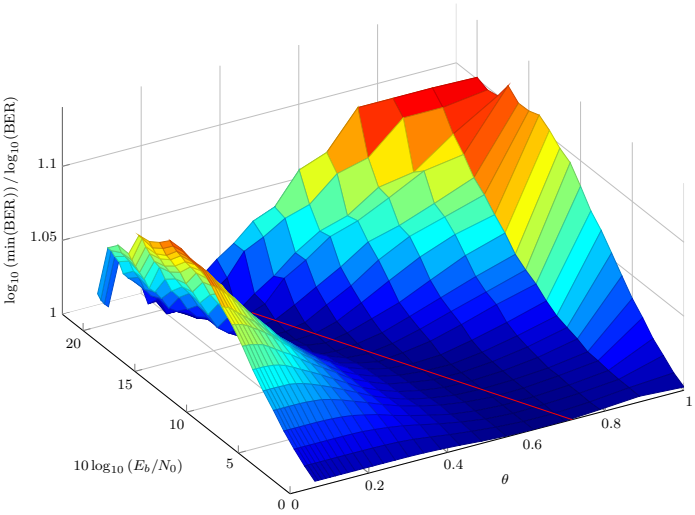


(a) BER deviation

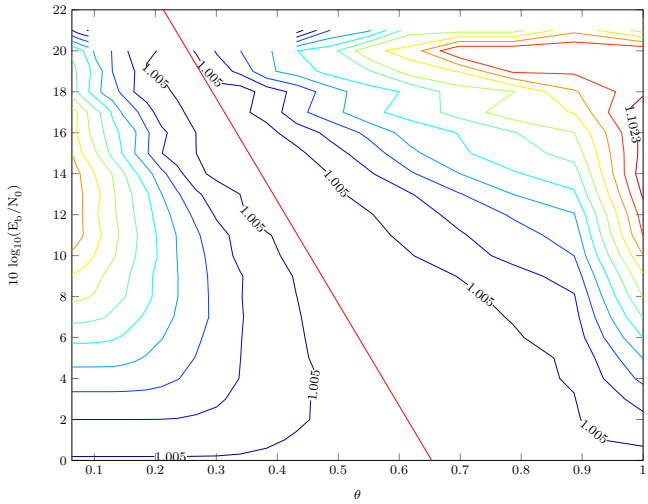


(b) Contour plot of 4.6a

Figure 4.6.: Impact of E_b/N_0-ratio and noise variance threshold θ on the BER for batch noise interpolation and system B, according to Table A.1.

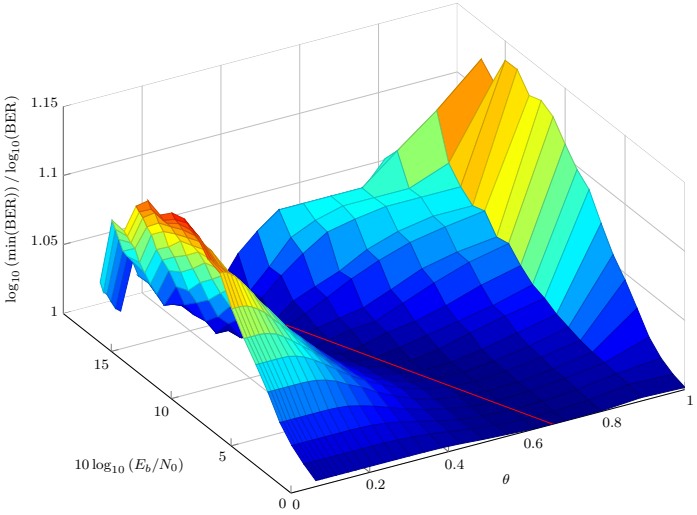


(a) BER deviation

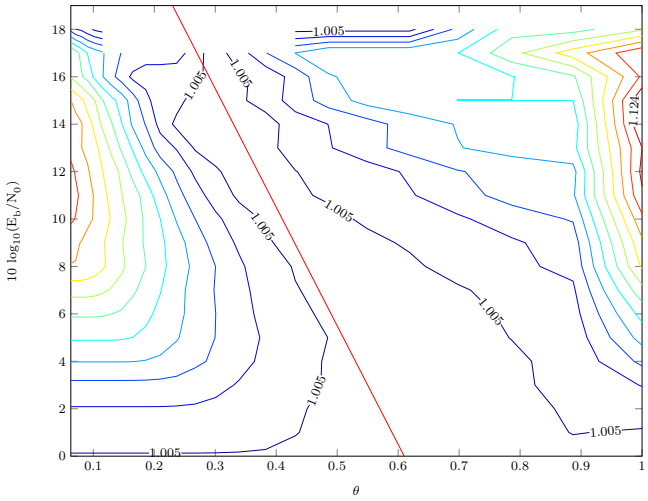


(b) Contour plot of 4.7a

Figure 4.7.: Impact of E_b/N_0-ratio and noise variance threshold θ on the BER for batch noise interpolation and system SD, according to Table A.1.

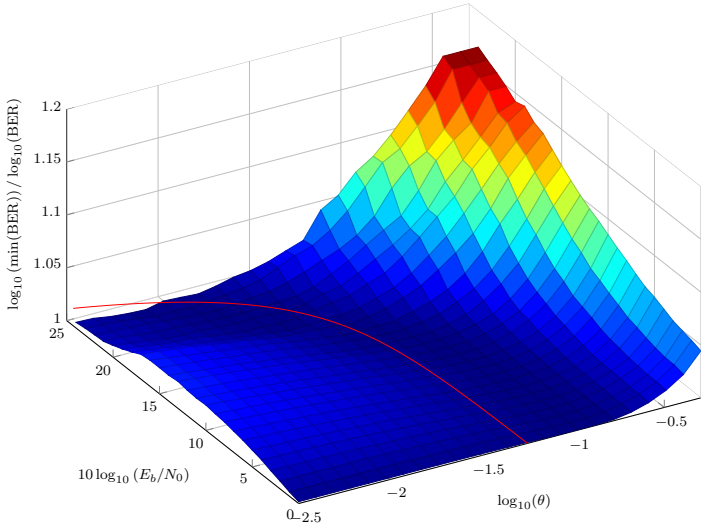


(a) BER deviation

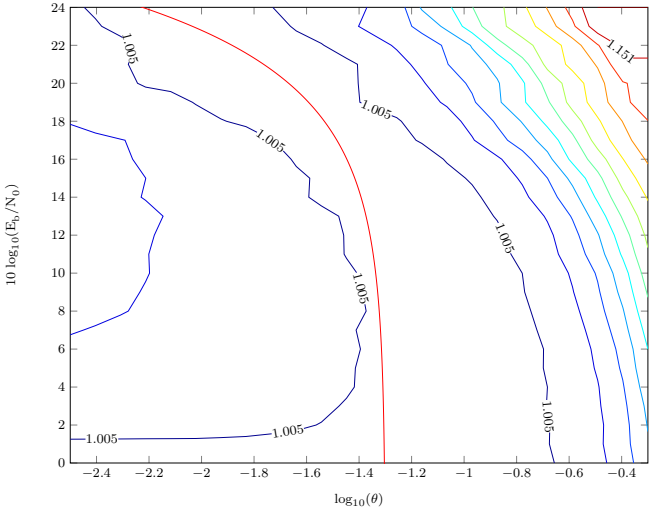


(b) Contour plot of 4.8a

Figure 4.8.: Impact of E_b/N_0-ratio and noise variance threshold θ on the BER for batch noise interpolation and system ML, according to Table A.1.

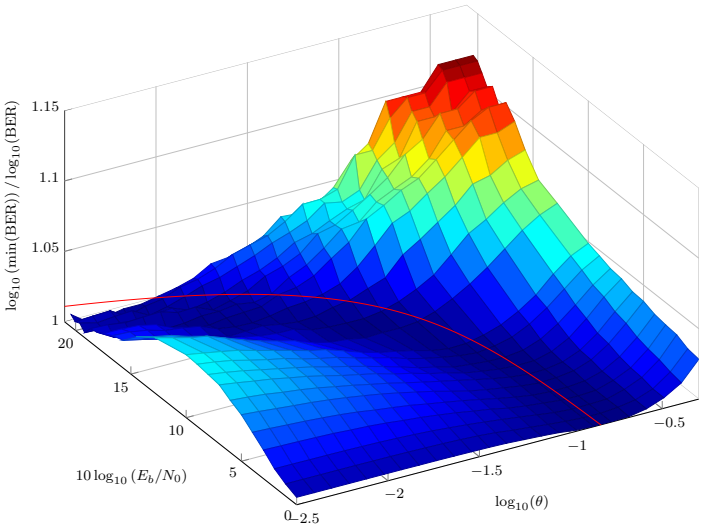


(a) BER deviation

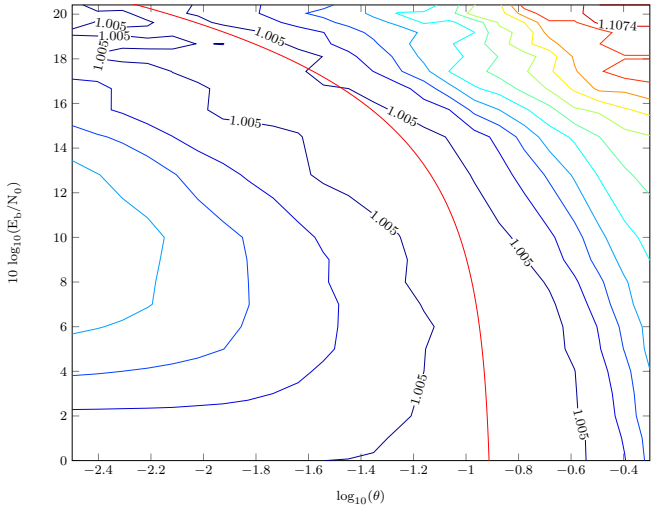


(b) Contour plot of 4.9a

Figure 4.9.: Impact of E_b/N_0-ratio and SER threshold θ on the BER for batch noise interpolation and system B, according to Table A.1.

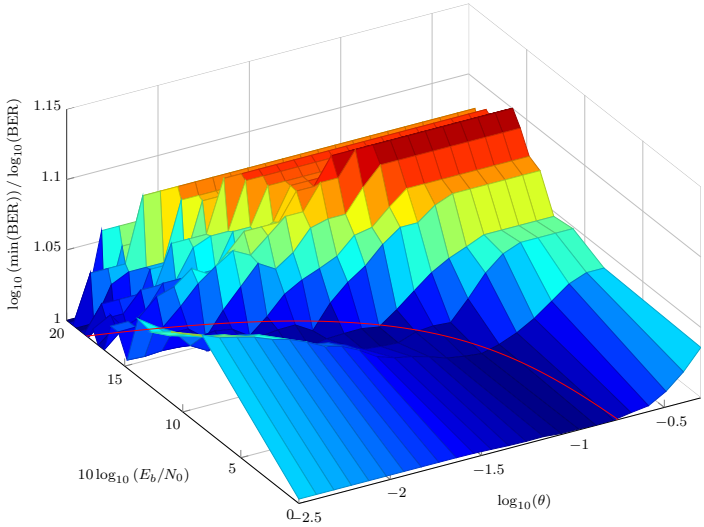


(a) BER deviation

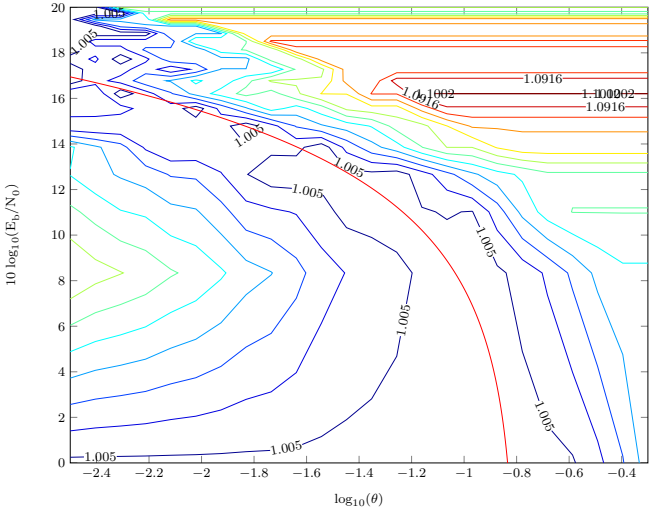


(b) Contour plot of 4.10a

Figure 4.10.: Impact of E_b/N_0-ratio and SER threshold θ on the BER for batch noise interpolation and system SD, according to Table A.1.



(a) BER deviation



(b) Contour plot of 4.11a

Figure 4.11.: Impact of E_b/N_0-ratio and SER threshold θ on the BER for batch noise interpolation and system ML, according to Table A.1.

4.1.5. Iterative Filtering

After describing batch filtering in the last section, which resulted in a simple matrix multiplication in the end, now an approach is introduced that is called *iterative filtering*. Instead of determining the whole Wiener filter at once, only one noise value is estimated here within one iteration. After updating the error statistics, the same procedure can be applied again. A brief outline is given in Algorithm 4, where the function NI_ITERATIVE is called with the results from the linear estimation $\tilde{\mathbf{d}}^{\text{lin}}$ and $\mathbf{C}_{ee}^{\text{lin}}$.

Algorithm 4 Iterative Noise Interpolation

1: **function** $[\tilde{\mathbf{d}}, \mathbf{C}_{ee}] = \text{NI\_ITERATIVE}(\tilde{\mathbf{d}}, \mathbf{C}_{ee})$
2: **repeat**
3: $\mathbf{w}^{\mathsf{T}} \leftarrow \mathbf{0}$
4: $\tilde{\mathbf{e}} \leftarrow \tilde{\mathbf{d}} - \left\lfloor \tilde{\mathbf{d}} \right\rceil$
5: $k \leftarrow \underset{l}{\text{argmax}} \, [\mathbf{C}_{ee}]_{l,l}$ ▷ Pick noisiest sample
6: **determine** $\mathcal{S}_k$ for the estimation of $\tilde{e}_k$
7: $\left[\mathbf{w}^{\mathsf{T}}\right]_{\mathcal{S}_k} \leftarrow [\mathbf{C}_{ee}]_{k,\mathcal{S}_k} \left([\mathbf{C}_{ee} + \mathbf{R}_{vv}]_{\mathcal{S}_k,\mathcal{S}_k} \right)^{-1}$
8: $\tilde{d}_k \leftarrow \tilde{d}_k - \mathbf{w}^{\mathsf{T}} \tilde{\mathbf{e}}$
9: $[\mathbf{C}_{ee}]_{k,*} \leftarrow [\mathbf{C}_{ee}]_{k,*} - \mathbf{w}^{\mathsf{T}} \mathbf{C}_{ee}$
10: $[\mathbf{C}_{ee}]_{*,k} \leftarrow [\mathbf{C}_{ee}]_{*,k} - \mathbf{C}_{ee} \mathbf{w}^{*}$
11: **until** iterated n_I times
12: **end function**

In line 5 the k-th sample with highest variance and thus highest prior error probability is selected for improvement. Then a set $\mathcal{S}_k$ needs to be found, from which $\tilde{e}_k$ shall be estimated. Here, the same criteria as formulated in Section 4.1.3.2 can be applied again. The filter determination in line 7 uses the same notation as for the batch noise interpolation. The statistics of the error are updated according to (4.19). However, as only one noise sample is estimated, the k-th row and column of $\mathbf{C}_{ee}$ need to be updated separately.

This procedure can be repeated as many times as an improvement is expected. How often this is can be easily answered for a simple sample selection method: A receiver employing a rather 'dumb' noise interpolation strategy by simply interpolating the current sample to be estimated from *all* available noise values except the current in each iteration k, such that

$$\mathcal{S}_k = \{0, \ldots, N_d - 1\} \backslash k.$$

Using this type of receiver, Figure 4.12 shows for different E_b/N_0-ratios the average number of bit errors *added*, compared to the number of errors in the previous iteration step. Thus, at iteration step n the number of bit errors that were introduced from step $n-1$ to step n is displayed. This number is normalized to the number of errors in step 0, which is after the linear estimation. Thus, the z-value of the plot at the position of iteration step number n and $\rho_{\text{dB}} = 10 \log_{10} (E_b/N_0)$ is determined by

$$z(\rho_{\text{dB}}, n) = \frac{\text{avg}\left(\epsilon_n^{(\rho_{\text{dB}})} - \epsilon_{n-1}^{(\rho_{\text{dB}})}\right)}{\text{avg}\left(\epsilon_0^{(\rho_{\text{dB}})}\right)},$$

where $\epsilon_n^{(\rho_{\mathrm{dB}})}$ is the number of bit errors after iteration step n at an logarithmic E_b/N_0-ratio of ρ_{dB}. All these counts are considered per OFDM symbol and averaged over 1 000 channel snapshots. For the simulation, an uncoded exemplary UW-OFDM system B (according to Table A.1) using 4-QAM was used.

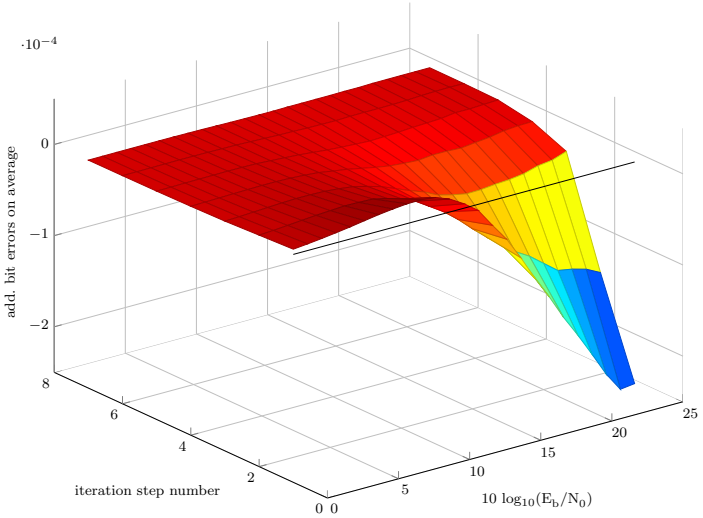


Figure 4.12.: Average bit error difference after an iteration for 'dumb' iterative NI in exemplary UW-OFDM system B.

For an effective receiver, this measure should always be negative, as positive values indicate added bit errors and thus pose a degradation. Above an SNR of about 13 dB the used noise interpolator proves to be effective. More interestingly, below that SNR the used noise interpolation actually *increases* the number of bit errors. This shows the importance of a careful selection of samples to estimate from. In Figure 4.12, a black line indicates the zero level at iteration step 1, to emphasize the error generation in this E_b/N_0-range. Anyway, after 7 iterations at the latest, another iteration does not yield any change in the number of bit errors on average. This number suffices in a system with $N_d = 48$ data values to exploit the whole potential of the noise interpolator.

Anticipating the study of threshold selection for iterative noise interpolation in the following, the effect of a carefully chosen sample selection criterion can be shown. Figure 4.13 shows the same graphic as before for the receiver with a noise variance threshold that will be introduced in Section 4.1.5.1 in exemplary system B. Here it can be seen that the noise interpolation method is effective over the whole E_b/N_0 range, most effective at 14 dB. At this SNR 6 iterations allow the NI to develop its whole potential, while at very high and low SNR no more than 2 iterations might be needed.

The results for exemplary system SD in Figure 4.14 prove the finding, that for the systems in this work in general not more than 8 iterations are needed to achieve the optimum

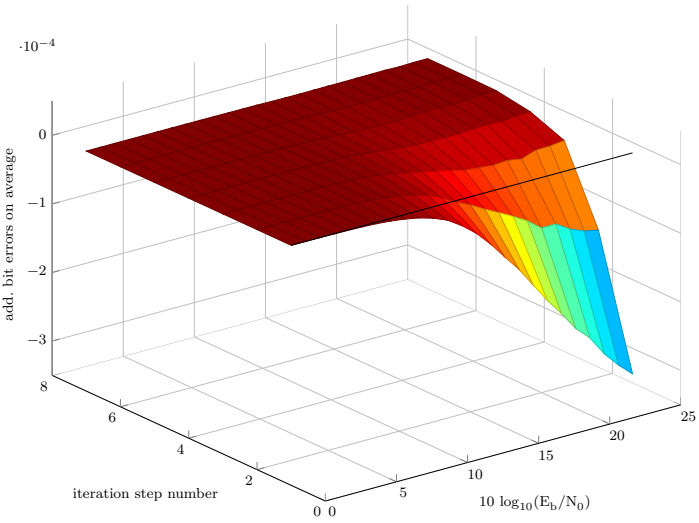


Figure 4.13.: Average bit error difference after an iteration for the noise variance thresholding NI in exemplary UW-OFDM system B.

NI performance. The parameters of all exemplary UW-OFDM systems are listed in Appendix A.

Investigating the sample selection topic again reveals that the values in $\mathcal{S}_k$ for estimation of the k-th value can be selected in the same way as it was done for the batch processing in Section 4.1.4. In the following, the same two selection approaches are investigated.

4.1.5.1. Noise Variance Threshold for Sample Selection for Iterative NI

When choosing a noise variance threshold for picking the samples to be used for estimation, such that

$$\mathcal{S}_k = \left\{ l \middle| \sigma_{e_l}^2 < \theta(\rho), l = \{0, \ldots, N_d - 1\} \backslash k \right\}, \tag{4.39}$$

a nice situation is found for the iterative noise filtering: A constant noise variance threshold of

$$\theta = 0.55 \tag{4.40}$$

yields very good results for all investigated UW-OFDM system setups. In Figure 4.15, the BER achieved by a receiver designed for system B with different noise variance thresholds θ over the SNR, in relation to the minimum achieved BER per SNR is shown, in order to prove this. The minimum BER valley is met pretty well by the red line representing $\theta = 0.55 = 10^{-0.26}$. Very similar results can be seen for UW-OFDM systems SD and ML in Figure 4.16 and 4.17. Due to the simple threshold, this is an interesting low complexity

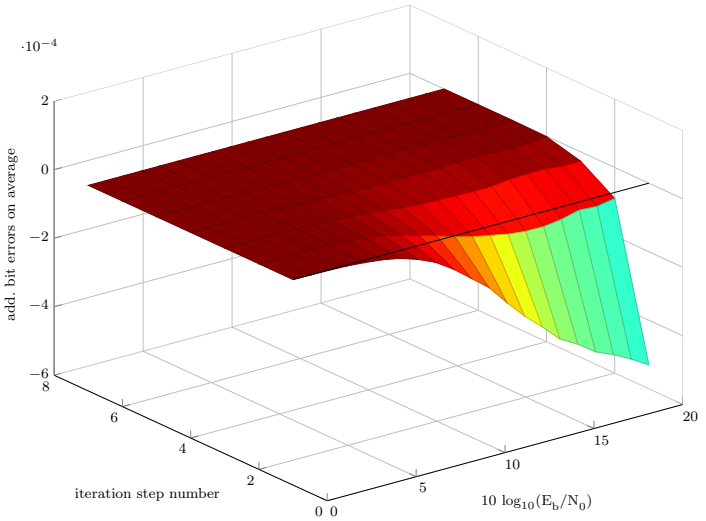
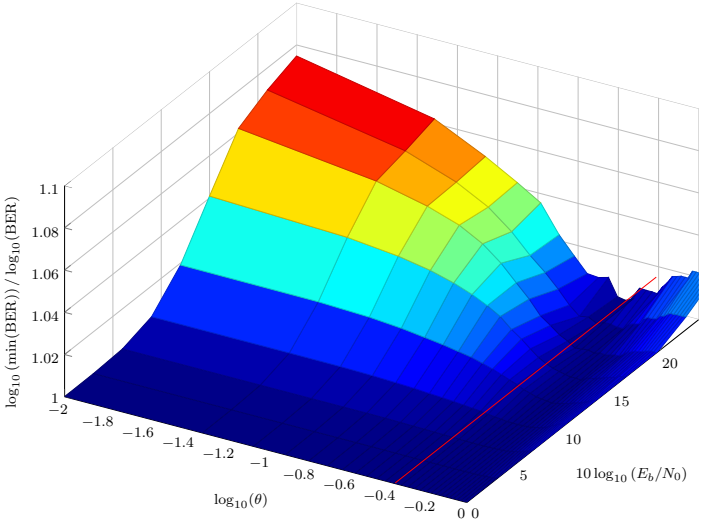
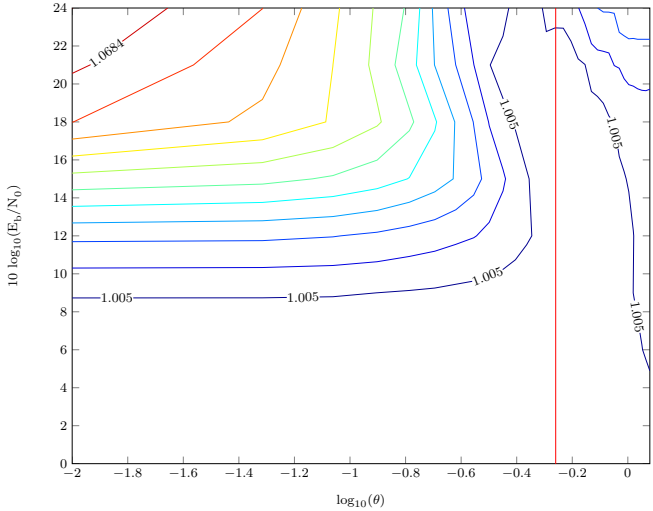


Figure 4.14.: Average bit error difference after an iteration for the noise variance thresholding NI in exemplary UW-OFDM system SD.

iterative noise interpolation method, where the receiver parameters need no tuning at all in order to achieve satisfying results.

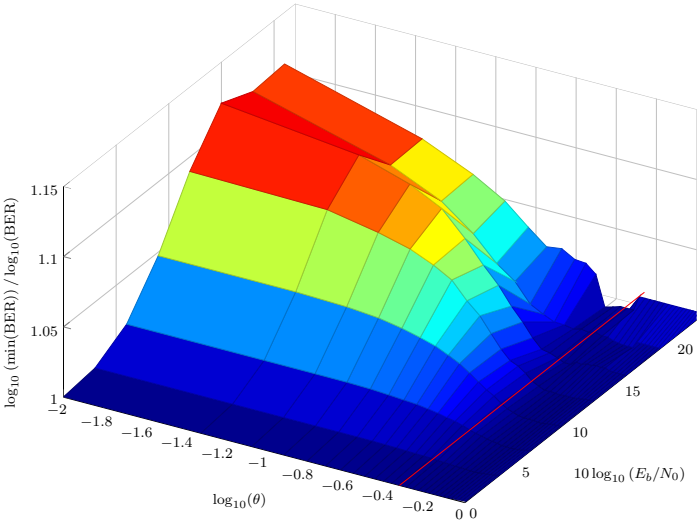


(a) BER deviation

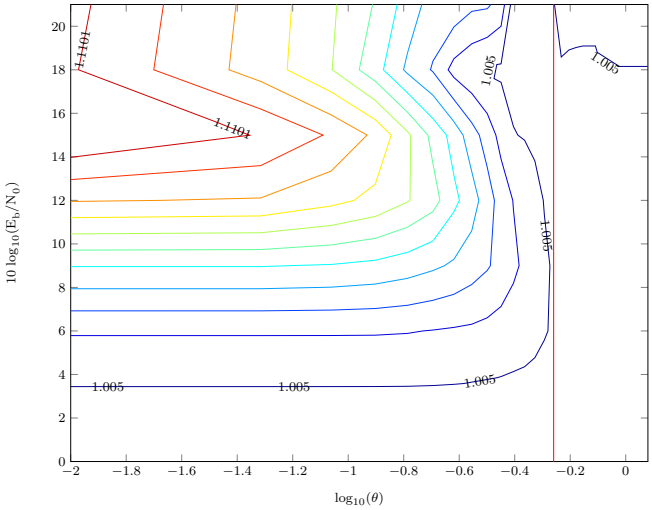


(b) Contour plot of 4.15a

Figure 4.15.: Impact of E_b/N_0-ratio and noise variance threshold θ on the BER for iterative noise interpolation and system B, according to Table A.1.

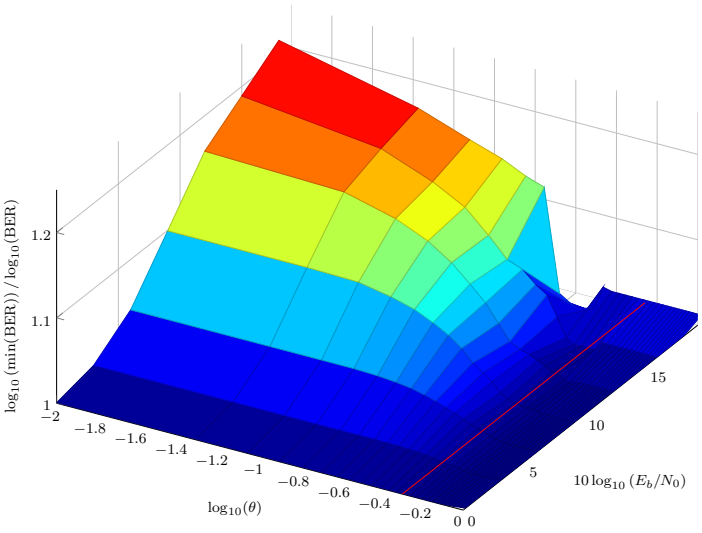


(a) BER deviation

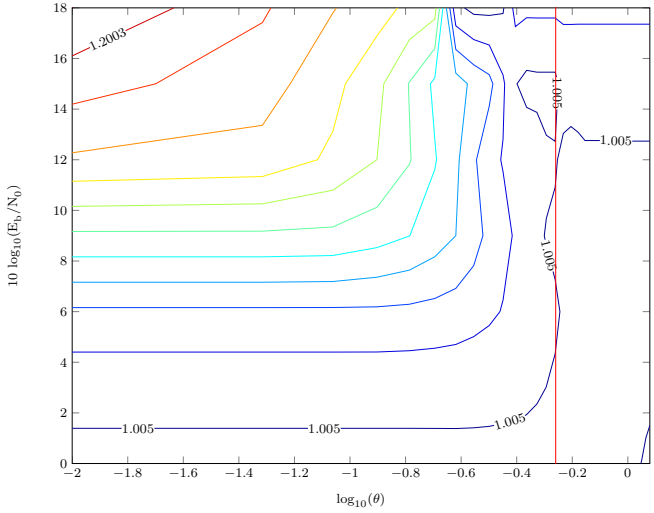


(b) Contour plot of 4.16a

Figure 4.16.: Impact of E_b/N_0-ratio and noise variance threshold θ on the BER for iterative noise interpolation and system SD, according to Table A.1.



(a) BER deviation



(b) Contour plot of 4.17a

Figure 4.17.: Impact of E_b/N_0-ratio and noise variance threshold θ on the BER for iterative noise interpolation and system ML, according to Table A.1.

4.1.5.2. SER Threshold for Sample Selection for Iterative NI

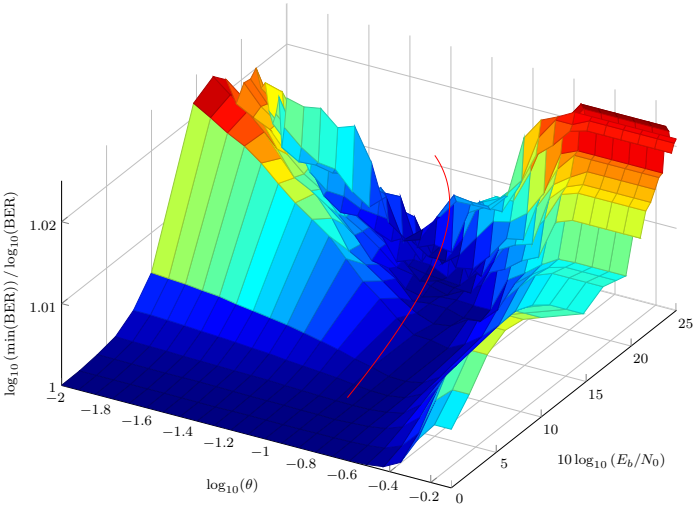
Using an SER threshold for iterative noise interpolation, such that the set of selected value indices to be used for estimation is

$$\mathcal{S}_k = \{l | \chi_l < \theta(\rho), l = \{0, \ldots, N_d - 1\} \backslash k\}\,, \tag{4.41}$$

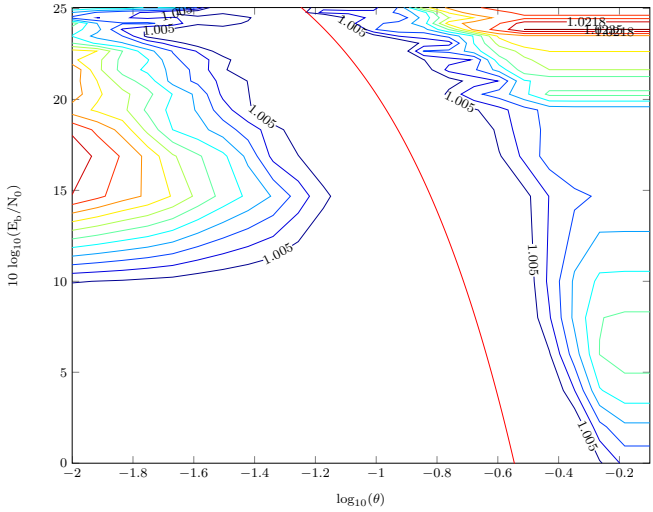
yields BER results as shown in Figure 4.18, 4.19 and 4.20 for UW-OFDM system setups B, SD and ML, respectively. The red line approximating the minimum BER achieving threshold is determined by the empirically found SER threshold function

$$\theta(\rho_{\mathrm{dB}}) = \left(-0.0123 + \frac{N_d - 16}{10000}\right)\rho_{\mathrm{dB}} + 0.405 - \frac{(N_d - 29)^2}{3000}. \tag{4.42}$$

Again, it still needs to be shown, if the computational effort is justified by the achieved BER gains, which is done in Section 4.1.6.

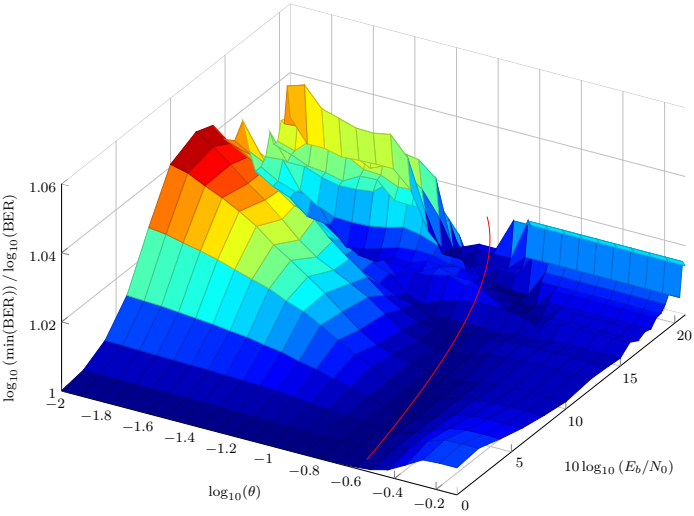


(a) BER deviation

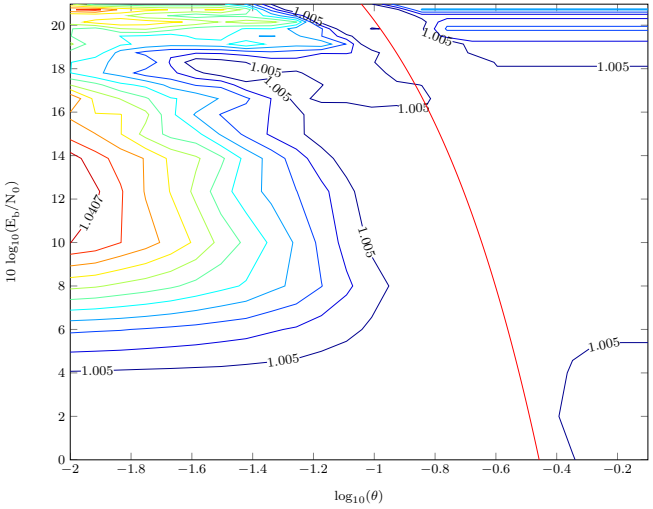


(b) Contour plot of 4.18a

Figure 4.18.: Impact of E_b/N_0-ratio and SER threshold θ on the BER for iterative noise interpolation and system B, according to Table A.1.

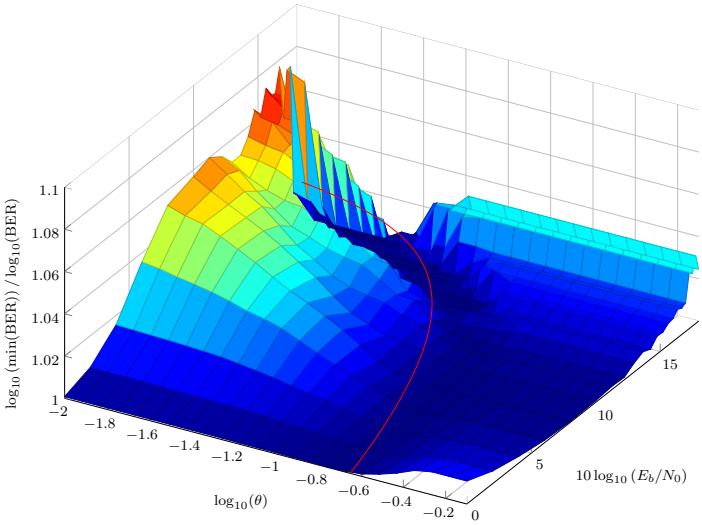


(a) BER deviation

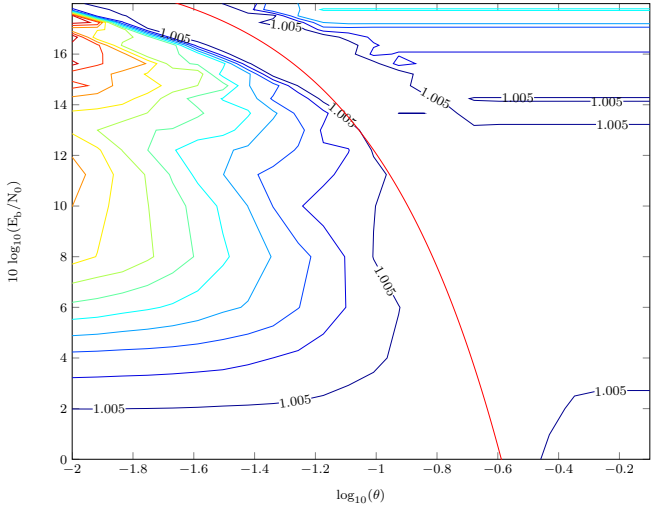


(b) Contour plot of 4.19a

Figure 4.19.: Impact of E_b/N_0-ratio and SER threshold θ on the BER for iterative noise interpolation and system SD, according to Table A.1.



(a) BER deviation



(b) Contour plot of 4.20a

Figure 4.20.: Impact of E_b/N_0-ratio and SER threshold θ on the BER for iterative noise interpolation and system ML, according to Table A.1.

4.1.6. Performance Evaluation

The noise filtering problem has two main parameters: The samples which are to be estimated and the samples, from which those shall be estimated. In Section 4.1.3.2, criteria were introduced to select these samples. These criteria could be applied in order to pick the n best values or the samples obeying a threshold θ. This left a vast set of possibilities open, hard to be covered comprehensively.

First, the k-th sample to be estimated was excluded from the set of candidates $\mathcal{S}_k$ to be used for estimation, which narrowed down the noise filtering to a *noise interpolation* procedure. When introducing the actual algorithms for noise interpolation in Section 4.1.4 and 4.1.5 the alternatives could be finally narrowed down by choosing *all* samples to be estimated for the batch approach, and picking at each iteration that sample with the highest noise variance for the iterative noise interpolation approach. Furthermore, noise and SER thresholding were picked as the criteria to select $\mathcal{S}_k$ and arrive with only a few different NI methods, that are possible to be surveyed.

Unfortunately, also a dependency of the thresholds on the SNR was discovered. As a consequence the thresholds could be approximated as a function of the E_b/N_0-ratio. Finally, all these possibilities were broken down to two candidates each, batch and iterative noise interpolation.

In this section, simulation results are shown for systematically generated UW-OFDM, as introduced in Section 2.2, in order to pick one final noise interpolation method that arises from this section to compete with the other UW-OFDM receivers. These findings are also correct for non-systematically generated UW-OFDM, as introduced in Section 2.4, as the final comparison of the nonlinear methods in Section 4.5 will show. As a reference, the best linear receiver known so far, the LMMSE estimator, as well as the noise interpolator which decides on a per-symbol basis is included in the comparison. For the latter, it turned out that a probability threshold of $\theta = 0.04$, selecting the samples

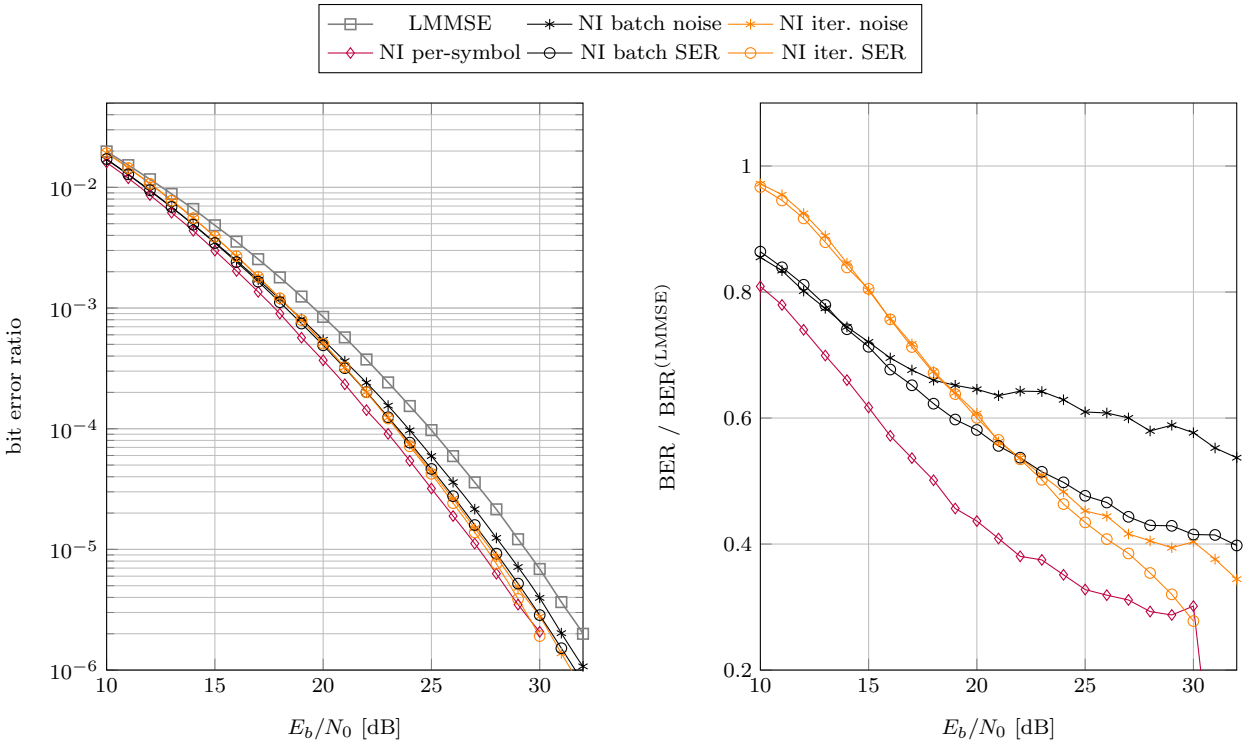
$$\mathcal{S}_k = \{k | p_{\text{wrong}}(k) < \theta\} \tag{4.43}$$

to be used for the estimation, yields very good results. All these receivers are applied to exemplary system B, as described in Appendix A.

Thus, six methods are compared in Figure 4.21a and 4.22a in terms of bit error probability:

- LMMSE estimator, according to (3.29) (LMMSE)
- sample-based NI according to (4.43), with a probability threshold for a wrong decision of $\theta = 0.04$ (NI per-symbol)
- batch NI using a noise variance threshold according to Section 4.1.4.1 (NI batch noise)
- batch NI using an SER threshold according to Section 4.1.4.2 (NI batch SER)
- iterative NI using a noise variance threshold according to Section 4.1.5.1 (NI iter. noise)
- iterative NI using an SER threshold according to Section 4.1.5.2 (NI iter. SER)

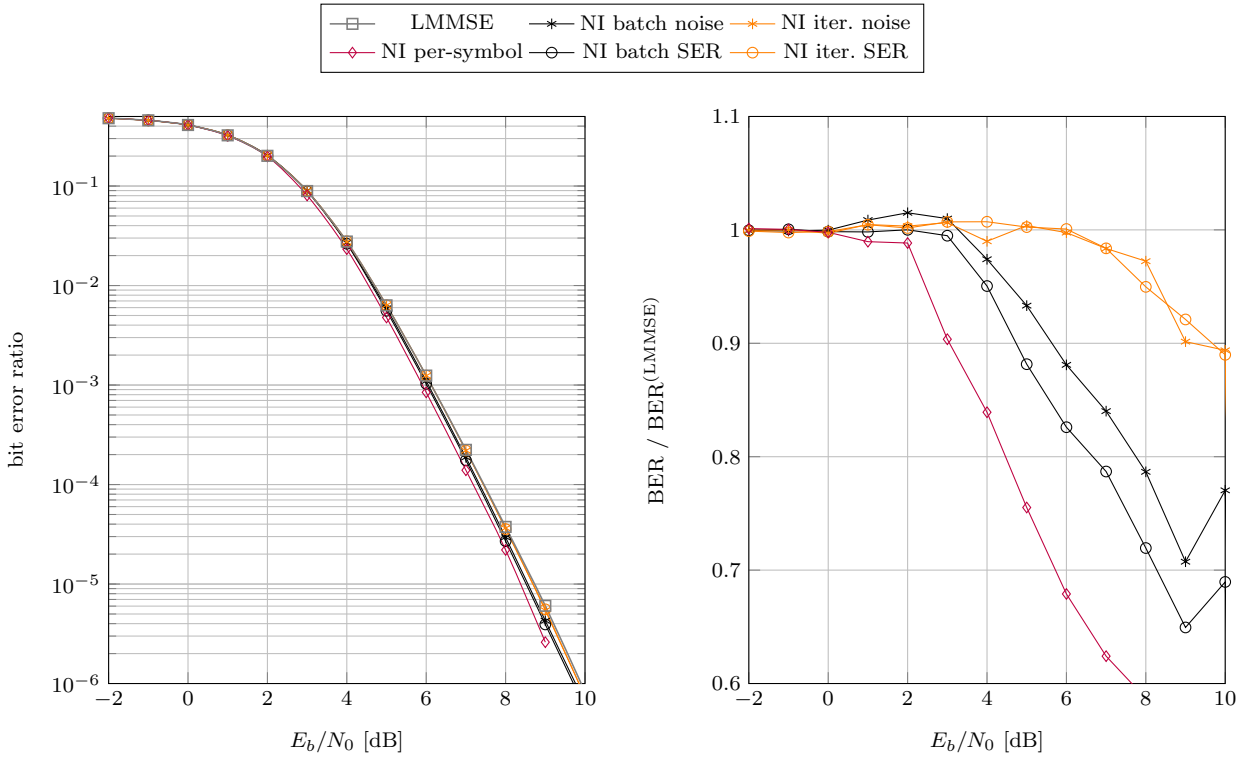
None of the four NI candidates can be easily identified as the best performing one. For this reason, Figure 4.21b and 4.22b show the BER relative to the BER achieved by the LMMSE receiver, in order to identify the better performing variants.



(a) uncoded results

(b) uncoded results, relative to LMMSE

Figure 4.21.: BER results for uncoded noise interpolation receivers with system B according to Table A.1 in multipath environment.



(a) coded results

(b) coded results, relative to LMMSE

Figure 4.22.: BER results for coded noise interpolation receivers with system B according to Table A.1 in multipath environment.

Due to the statistical imprecision, the latter presentation seems very rough and inaccurate. Although it is almost impossible to draw a quantitative conclusion from it, it is easier to make a qualitative observation than in the first plot. From this follows that NI methods perform well on higher SNR in uncoded transmission, but do not necessarily show the same advantage in coded transmission. However, the batch NI with SER thresholding is a good compromise that performs very well in uncoded and at least as good as the LMMSE estimator in coded transmission. Therefore, this candidate is picked as the NI representative that will be used for comparison with the nonlinear receivers to be introduced below.

The highly complex NI with per-symbol selections performs best, but not far from the other introduced NI methods. Considering the huge computational effort and the small gain, this method is excluded from further examinations, as discussed before.

4.2. Decision Feedback Equalization

The decision feedback equalization (DFE) detection scheme is a nonlinear method that detects symbols iteratively. In each step, it performs an operation where one data symbol is decided, which is also the part that incorporates the nonlinearity. The decided symbol is used in a feedback to help equalize the remaining values of the OFDM symbol, hence the name. Another name for the same method is successive interference cancellation (SIC), where the portion of each data symbol on the receive vector is considered as *interference*, which is *canceled* out *successively*. The BLAST architecture also coincides with DFE, while V-BLAST (Vertical Bell Labs Layered Space-Time) [Fos96, WFGV98] is the most prominent version. This method is described in Algorithm 5, as outlined in [WFGV98, Jan04, Win04] and many more, and as applied for UW-OFDM in [Ret10].

The data symbol estimate $\tilde{d}_k$ is obtained from the receive vector $\mathbf{y}$ by a linear estimation by

$$\tilde{d}_k = \mathbf{w}_k^{\mathsf{H}} \mathbf{y}, \tag{4.44}$$

where $\mathbf{w}_k$ is the linear estimator for symbol number k, which is also called nulling vector [Fos96], in this context. Then the symbol is decided (also called slicing)

$$\hat{d}_k = \left\lfloor \tilde{d}_k \right\rceil, \tag{4.45}$$

which represents the non-linear operation. As, according to the linear system model (4.1)

$$\mathbf{y} = \mathbf{H}\mathbf{d} + \mathbf{n} = \mathbf{h}_0 d_0 + \mathbf{h}_1 d_1 + \cdots + \mathbf{h}_{N_d-1} d_{N_d-1} + \mathbf{n},$$

the anticipated impact of the decided symbol on the receive vector is subtracted in the following step:

$$\mathbf{y}' = \mathbf{y} - \hat{d}_k \mathbf{h}_k \tag{4.46}$$

The nulling vector acts as an estimator for the k-th symbol and can be obtained in many ways, most commonly according to the ZF or MMSE criterion. Both ways to retrieve the nulling vector $\mathbf{w}_k$ will be elaborated in the Unique Word OFDM context in the next sections. After determination of the nulling vector for symbol k, decision of the data estimate and removal of its influence, the linear model is reduced in size, as if the transmission of data symbol number k never happened. Then, these steps are repeated all over again, until all data symbols of $\hat{\mathbf{d}}$ are determined.

It is implied that each decision is assumed to be correct; when wrong, a faulty part is subtracted from the receive vector, causing actually more interference for the detection of all following symbols. Just as for noise filtering, this is called error propagation and needs to be avoided. The probability of an error can be greatly reduced by changing the order, in which data symbols are decided. The best strategy is to work on those data symbols first which can be decided with lowest error probability. It is shown in Section 4.1.3.2 that the lowest *symbol error ratio* is achieved by the symbol with the lowest noise variance.

4.2.1. DFE with Zero Forcing Criterion

In ZF DFE the nulling vectors are determined by following the ZF principle that is already known from linear receivers: The BLUE. In order to get the nulling vector for symbol number k, only the k-th row of the BLUE, executed in (3.10), needs to be extracted:

$$\mathbf{D}_{\mathrm{ZF}} = \mathbf{H}^{+} = \mathbf{E}_{\mathrm{BLUE}} \tag{4.47}$$

$$\mathbf{w}_k^{\mathsf{H}} = [\mathbf{D}_{\mathrm{ZF}}]_{k,*} \tag{4.48}$$

In order to determine the symbol with the lowest symbol error probability, a look at the data estimates *after* a supposed ZF equalization should be taken. According to the linear model these are

$$\tilde{\mathbf{d}} = \mathbf{H}^{+}\mathbf{y} = \mathbf{d} + \mathbf{H}^{+}\mathbf{n}, \tag{4.49}$$

with independent noise samples and equal noise variances in $\mathbf{n} \sim \mathcal{N}_{\mathrm{C}}\left(0; N\sigma_n^2\mathbf{I}\right)$. For the k-th data estimate this means

$$\tilde{d}_k = d_k + \mathbf{w}_k^{\mathsf{H}}\mathbf{n}, \tag{4.50}$$

and the noise power after optimum ZF estimation becomes

$$\begin{aligned} \mathrm{E}\left\{\left(\tilde{d}_k - d_k\right)\left(\tilde{d}_k - d_k\right)^{*}\right\} &= \mathrm{E}\left\{\mathbf{w}_k^{\mathsf{H}}\mathbf{n}\mathbf{n}^{\mathsf{H}}\mathbf{w}_k\right\} \\ &= N\sigma_n^2 \left\|\mathbf{w}_k\right\|_2^2 . \end{aligned} \tag{4.51}$$

This implies that the noise on each data symbol is amplified according to the row norms of $\mathbf{H}^{+}$ during the equalization process, and the data symbol to decide most reliably is the one corresponding to the row of $\mathbf{H}^{+}$ with smallest norm. An equivalent way to formulate this is used in line 8 of Algorithm 5. The most reliable symbol is the one corresponding to the smallest element of the main diagonal of the error covariance matrix that is to be expected from a linear estimation, in this case

$$\mathbf{C}_{ee}^{\mathrm{BLUE}} = N\sigma_n^2 \left(\mathbf{H}^{\mathsf{H}}\mathbf{H}\right)^{-1},$$

according to (3.11). Beyond that, it is easy to show that

$$\left[\mathbf{C}_{ee}^{\mathrm{BLUE}}\right]_{k,k} = N\sigma_n^2 \left\|\mathbf{w}_k\right\|_2^2 .$$

The next step is important and distinguishes a linear detection from DFE: The decision in (4.45) is made, and it is assumed to be correct and no errors occur in this step. Not only the symbol's impact is removed from the receive vector according to (4.46) in line 13, but also from future considerations. This is done by deleting the corresponding column of the channel matrix $\mathbf{H}$ and reducing its size, as shown in line 14. In Algorithm 5, in contrast to other versions of the algorithm, the equalized value $\tilde{\mathbf{d}}$ is returned rather than the detected data symbol $\left\lfloor\tilde{\mathbf{d}}\right\rceil$ in order to provide for soft information, as detailed in Section 4.2.3.

Algorithm 5 Decision Feedback Equalization

1: **function** $[\tilde{\mathbf{d}}, \mathbf{C}_{ee}]$ =DFE$(\mathbf{y}, \mathbf{H}, \sigma_n^2)$
2: $\quad\mathbf{C}_{ee} \leftarrow \mathbf{0}$
3: $\quad\mathbf{M} \leftarrow \mathbf{H}$
4: $\quad\mathbf{i} \leftarrow \{0, 1, \ldots, N_d - 1\}$
5: $\quad$**for** $l = 0, \ldots, N_d - 1$ **do**
6: $\quad\quad\mathbf{D} \leftarrow \mathbf{M}^+$ ▷ pseudo-inverse
7: $\quad\quad\mathbf{C}_{ee}^{\text{lin}} \leftarrow N\sigma_n^2 \left(\mathbf{M}^{\mathsf{H}}\mathbf{M}\right)^{-1}$ ▷ error cov. matrix after linear estimation
8: $\quad\quad r \leftarrow \underset{m}{\text{argmin}} \left[\mathbf{C}_{ee}^{\text{lin}}\right]_{m,m}$ ▷ find minimum row norm
9: $\quad\quad k \leftarrow i_r$; **remove** element r from $\mathbf{i}$ ▷ determine affected symbol number
10: $\quad\quad\mathbf{w}_k^{\mathsf{H}} \leftarrow [\mathbf{D}]_{r,*}$ ▷ nulling vector
11: $\quad\quad[\mathbf{C}_{ee}]_{k,k} \leftarrow \left[\mathbf{C}_{ee}^{\text{lin}}\right]_{r,r}$ ▷ store soft information
12: $\quad\quad\tilde{d}_k \leftarrow \mathbf{w}_k^{\mathsf{H}}\mathbf{y}$ ▷ estimate symbol k
13: $\quad\quad\mathbf{y} \leftarrow \mathbf{y} - [\mathbf{H}]_{*,r} \left\lfloor \tilde{d}_k \right\rceil$ ▷ cancel interference
14: $\quad\quad$**remove** column r from $\mathbf{M}$
15: $\quad$**end for**
16: **end function**

4.2.2. DFE with Minimum Mean Square Error Criterion

As mentioned earlier, not only the ZF criterion can be applied in the determination of the nulling vectors. In the ZF version, the nulling vectors were determined from the BLUE

$$\begin{aligned}\mathbf{D}_{\text{ZF}} &= \mathbf{H}^+ \\ &= \left(\mathbf{H}^{\mathsf{H}}\mathbf{H}\right)^{-1}\mathbf{H}^{\mathsf{H}}.\end{aligned} \tag{4.52}$$

While the ZF nulling vectors extracted from the BLUE are the result when applying classical estimation theory, also Bayesian theory and thus the LMMSE estimator (3.29) can be used to determine the nulling vectors $\mathbf{w}_k$. Then

$$\mathbf{E}_{\text{LMMSE}} = \left(\mathbf{H}^{\mathsf{H}}\mathbf{H} + \frac{N\sigma_n^2}{\sigma_d^2}\mathbf{I}\right)^{-1}\mathbf{H}^{\mathsf{H}} \tag{4.53}$$

substitutes the instruction to determine $\mathbf{D}$ in line 6, which yields the MMSE version of the DFE and supposedly a better performance. With the definition of an *extended* channel matrix

$$\breve{\mathbf{H}} = \begin{bmatrix}\mathbf{H} \\ \sqrt{\frac{N\sigma_n^2}{\sigma_d^2}}\mathbf{I}\end{bmatrix}, \tag{4.54}$$

$$\breve{\mathbf{H}}^{\mathsf{H}}\breve{\mathbf{H}} = \begin{bmatrix}\mathbf{H}^{\mathsf{H}} & \sqrt{\frac{N\sigma_n^2}{\sigma_d^2}}\mathbf{I}\end{bmatrix}\begin{bmatrix}\mathbf{H} \\ \sqrt{\frac{N\sigma_n^2}{\sigma_d^2}}\mathbf{I}\end{bmatrix} \tag{4.55}$$

$$= \mathbf{H}^{\mathsf{H}}\mathbf{H} + \frac{N\sigma_n^2}{\sigma_d^2}\mathbf{I}, \tag{4.56}$$

an equivalent MMSE DFE receiver can also be determined by

$$\begin{aligned}\mathbf{D}_{\mathrm{MMSE}} = \breve{\mathbf{H}}^{+} &= \left(\breve{\mathbf{H}}^{\mathsf{H}}\breve{\mathbf{H}}\right)^{-1}\breve{\mathbf{H}}^{\mathsf{H}} \\ &= \left(\mathbf{H}^{\mathsf{H}}\mathbf{H} + \frac{N\sigma_n^2}{\sigma_d^2}\mathbf{I}\right)^{-1}\left[\mathbf{H}^{\mathsf{H}} \quad \sqrt{\frac{N\sigma_n^2}{\sigma_d^2}}\mathbf{I}\right].\end{aligned} \tag{4.57}$$

This matrix yields longer nulling vectors, with length $2N_d + N_r$ instead of $N_d + N_r$. Therefore, also the receive vector has to be extended [WBKK03] by an appropriate amount of zeros to

$$\breve{\mathbf{y}} = \begin{bmatrix}\mathbf{y} \\ \mathbf{0}\end{bmatrix}, \tag{4.58}$$

in order to be used in line 12 and 13. Thus, executing Algorithm 5 with $\breve{\mathbf{H}}$ and $\breve{\mathbf{y}}$ yields the MMSE version of the DFE [Has00], which is also the way it is implemented for the simulations in this work.

4.2.3. Soft Information from DFE

An optimum DFE for coded UW-OFDM transmission would require to join channel decoding and the DFE operation by incorporating the decoding into the feedback loop. Due to the restrictions of the UW-OFDM system considered in this work according to Figure 2.9, the DFE needs to be realized as in Figure 4.23, where no link between the channel decoding and the DFE is present.

Due to the nonlinear nature of the DFE processing, it is not possible to provide a full post-detection error covariance matrix for the error $\mathbf{e} = \mathbf{d} - \left\lfloor\tilde{\mathbf{d}}\right\rceil$, as it was derived for the linear receivers. There have been efforts to determine soft information for the DFE procedure [CCC00, WNH^{+}06, CWK06]. However, these methods are computationally very complex and especially for the N_d dimensions spanned by the UW-OFDM signaling scheme practically not realizable.

As soft information for the Viterbi decoder, the variances of the remaining error after each linear equalization step and therefore the corresponding main diagonal element of the error covariance matrix $\mathbf{C}_{ee}$ suffice for satisfying results. These error variances can be easily calculated [WFGV98]. By expanding (4.44) to

$$\begin{aligned}\tilde{d}_k &= \mathbf{w}_k^{\mathsf{H}}\mathbf{y} \\ &= \mathbf{w}_k^{\mathsf{H}}\mathbf{H}\mathbf{d} + \mathbf{w}_k^{\mathsf{H}}\mathbf{n} \\ &= d_k + \mathbf{w}_k^{\mathsf{H}}\mathbf{n},\end{aligned}$$

the amplification of the noise $\mathbf{n}$ for ZF DFE induced by the k-th nulling vector $\mathbf{w}_k$ can be quantified by its $l2$-norm, as shown in (4.51). Hence, the post-detection error variance on data symbol k, which is also on position k of the diagonal of the error covariance matrix, is determined by

$$\left[\mathbf{C}_{ee}\right]_{k,k} = N\sigma_n^2 \left\|\mathbf{w}_k\right\|_2^2, \tag{4.59}$$

or by reusing the error variances that are used to determine the symbol processing order, as it is done in line 11 in Algorithm 5.

As the extended versions in (4.54) and (4.58) along with the pseudo-inverse are used for the MMSE realization, (4.59) applies also for MMSE, not only in the ZF case. Therefore, the noise amplification as the nulling vector's row norm yields the same error statistics, as the evaluation of the main diagonal of the appropriate error covariance matrices in each iteration would do. However, if an extended receive vector according to (4.58) is used, also an extended noise vector

$$\breve{\mathbf{n}} = \begin{bmatrix} \mathbf{n} \\ \mathbf{0} \end{bmatrix}$$

needs to be introduced, for these considerations to be consistent.

It needs to be noted that these error variances are determined under the assumption of correct decisions in each DFE iteration. They do not cover the possibility of error propagation and can be considered as an approximation only.

4.2.4. DFE – Other Interpretations

Separated DFE In the version of Algorithm 5, the nulling vectors $\mathbf{w}_k$ are computed during the procedure in an iterative manner. However, if the channel conditions and thus $\mathbf{H}$ do not change, neither the order, nor the nulling vectors themselves change. This suggests a separation of a pre-computation of the nulling vectors and the data decision parts as presented in Algorithm 6. The permutation matrix $\mathbf{\Pi}$ that is built in line 10 documents the order, in which the data symbols need to be detected. Since the index k assigns the data symbols in linear order, the data vector needs to be re-sorted after the detection loop (line 22). By putting all nulling vectors into a matrix $\mathbf{W}_{\mathrm{DFE}} = \left[\mathbf{w}_0, \mathbf{w}_1, \ldots, \mathbf{w}_{N_d-1}\right]^{\mathsf{H}}$, a matrix

$$\mathbf{K} = \mathbf{W}_{\mathrm{DFE}}\mathbf{H}\mathbf{\Pi} \tag{4.60}$$

can be assembled. One can show that this $\mathbf{K}$ has a unit-diagonal and an upper triangular structure. Both matrices are shaped such that the algorithm can be expressed in the simple block diagram for DFE [Fis02], which is shown in Figure 4.23.

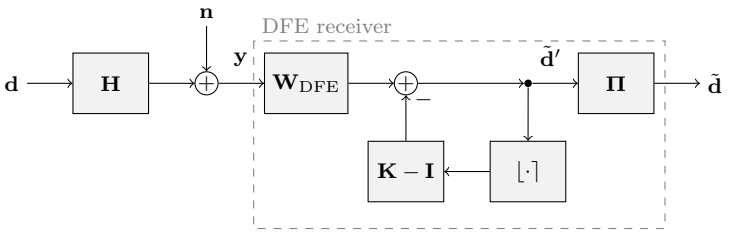


Figure 4.23.: Decision feedback equalization receiver.

Furthermore, it can be observed that all rows and thus all nulling vectors are mutually orthogonal, which is a direct consequence of the creation process of $\mathbf{w}_k$. Setting $\mathbf{M} = \mathbf{H}$, this can be shown as follows:

Algorithm 6 Separated DFE

$\mathbf{W}_{\mathrm{DFE}} \leftarrow \mathbf{0}$; $\mathbf{\Pi} \leftarrow \mathbf{0}$; $\mathbf{C}_{ee} \leftarrow \mathbf{0}$
$\mathbf{M} \leftarrow \mathbf{H}$
$\mathbf{i} \leftarrow \{0, 1, \ldots, N_d - 1\}$
for $l = N_d - 1, \ldots, 0$ **do**
    $\mathbf{D} \leftarrow \mathbf{M}^{+}$
    $\mathbf{C}_{ee}^{\mathrm{lin}} \leftarrow N\sigma_n^2 \left(\mathbf{M}^{\mathsf{H}}\mathbf{M}\right)^{-1}$
    $r \leftarrow \underset{m}{\operatorname{argmin}} \left[\mathbf{C}_{ee}^{\mathrm{lin}}\right]_{m,m}$
    $k \leftarrow i_r$; **remove** element r from $\mathbf{i}$
    $\mathbf{w}_l^{\mathsf{H}} \leftarrow [\mathbf{D}]_{r,*}$
    $[\mathbf{\Pi}]_{k,l} \leftarrow 1$ ▷ update permutation matrix
    $[\mathbf{C}_{ee}]_{k,k} \leftarrow \left[\mathbf{C}_{ee}^{\mathrm{lin}}\right]_{r,r}$
    remove column r from $\mathbf{M}$
end for

$\tilde{\mathbf{d}}' \leftarrow \mathbf{W}_{\mathrm{DFE}}\mathbf{y}$ ▷ $\mathbf{W}_{\mathrm{DFE}} = [\mathbf{w}_0, \mathbf{w}_1, \ldots, \mathbf{w}_{N_d-1}]^{\mathsf{H}}$
$\mathbf{K} \leftarrow \mathbf{W}_{\mathrm{DFE}}\mathbf{H}\mathbf{\Pi}$
$\mathbf{K}' \leftarrow \mathbf{K} - \mathbf{I}$
for $k = N_d - 1, \ldots, 0$ **do**
    $\tilde{\mathbf{d}}' \leftarrow \tilde{\mathbf{d}}' - \mathbf{k}'_k \left\lfloor \tilde{d}'_k \right\rceil$ ▷ cancel interference
end for

$\tilde{\mathbf{d}} \leftarrow \mathbf{\Pi}\tilde{\mathbf{d}}'$ ▷ re-sort data vector

1. Per definition, $\mathbf{DM} = \mathbf{I}$ for both $\mathbf{D}$ according to (4.47) and (4.57), which results to

$$\mathbf{w}_k^{\mathsf{H}} \cdot \mathbf{m}_l = \begin{cases} 1 & \text{for } l = k, \\ 0 & \text{else}, \end{cases} \tag{4.61}$$

   for the nulling vector $\mathbf{w}_k$ determined in the current iteration. This means $\mathbf{w}_k$ is orthonormal to any column vector in $\mathbf{M}$, but the k-th.

2. The column space of $\mathbf{M}$ corresponds to the Hermitian transpose of the row space of $\mathbf{M}^+$. This is true, as the row space of $\mathbf{M}^+ = (\mathbf{M}^{\mathsf{H}}\mathbf{M})^{-1} \cdot \mathbf{M}^{\mathsf{H}}$ is basically determined by $\mathbf{M}^{\mathsf{H}}$, as the multiplication with an invertible matrix $(\mathbf{M}^{\mathsf{H}}\mathbf{M})^{-1}$ does not change the subspace[5] spanned by the rows of $\mathbf{M}^{\mathsf{H}}$.

3. Because of 2., the removal of the k-th column of $\mathbf{M}$ ensures orthogonality of $\mathbf{w}_k$ with the next nulling vector.

The complete induction concludes the proof that the current nulling vector is also orthogonal to all nulling vectors determined in future iterations, and finally that the rows of $\mathbf{W}_{\text{DFE}}$ form an orthonormal basis.

Similarly, the triangular structure of $\mathbf{K} = \mathbf{W}_{\text{DFE}}\mathbf{H}\mathbf{\Pi}$ can be proven. In the first iteration, the nulling vector $\mathbf{w}_{N_d-1}^{\mathsf{H}}$ is defined in order to be orthogonal to all column vectors in $\mathbf{M}$, except the selected with minimum noise amplification. Hence, only a single one is in the (N_d-1)-th row of $\mathbf{K}$. In the next iterations, the nulling vector is defined only to be orthogonal to all *remaining* columns of $\mathbf{M} = \mathbf{H}$. It is not orthogonal to any previously selected columns, as they were removed from $\mathbf{M}$ after being processed. Hence, the product of the new nulling vector and $\mathbf{H}$ has non-zero values for previously removed column vectors, a one for the currently processes column and zeros for all other columns still present in $\mathbf{M}$. The permutation by $\mathbf{\Pi}$ is needed to meet the algorithms' way of ordering and produces finally the upper triangular structure of $\mathbf{K}$.

It needs to be noted that this does not apply for MMSE DFE using (4.53), rather than (4.57) to determine $\mathbf{D}_{\text{MMSE}}$. In case $\mathbf{D} = \mathbf{E}_{\text{LMMSE}}$, (4.61) does not hold and thus neither orthonormality of the resulting $\mathbf{W}_{\text{DFE}}$, nor the triangular structure of $\mathbf{K}$ are true. Algorithm 6 and the later introduced Algorithm 7 depend on these structures.

DFE by QR Decomposition These particular properties of $\mathbf{W}_{\text{DFE}}$ and $\mathbf{K}$ suggest the two definitions

$$\mathbf{Q} = \left(\mathbf{\Gamma}^{-1}\mathbf{W}_{\text{DFE}}\right)^{\mathsf{H}}, \tag{4.62}$$

$$\mathbf{R} = \mathbf{\Gamma}^{-1}\mathbf{K}, \tag{4.63}$$

with the diagonal matrix

$$[\mathbf{\Gamma}]_{k,k} \triangleq \|\mathbf{w}_k\|_2 = \left\| [\mathbf{W}_{\text{DFE}}]_{k,*} \right\|_2 = \sqrt{\sum_{l=0}^{N_d-1} \left| [\mathbf{W}_{\text{DFE}}]_{k,l} \right|^2}, \tag{4.64}$$

[5] The matrix multiplication is just a linear combination of the base vectors, hence no dimensions can be added. The invertibility ensures full rank, hence no dimensions are lost.

indicating the $l2$-norm of each row vector of $\mathbf{W}_{\mathrm{DFE}}$, its Euclidean length. Then, the error covariance matrix is simply $\mathbf{C}_{ee} = \mathbf{\Gamma}^2$. This allows for a different viewpoint: It introduces the matrices $\mathbf{Q}$ with orthonormal column vectors and an appropriate upper triangular $\mathbf{R}$, as they could have been produced by any QR decomposition. This topic will be dealt with in detail in Chapter 5. The DFE procedure based on a QR decomposed channel matrix is equivalent to the processing in Figure 4.23 (see [WBR+01, WRB+02]). For now, another adaption of the algorithm in Algorithm 7 is presented, which defines the functions VBLAST_QR and QR_DETECTION.

Algorithm 7 Separated DFE and QR detection

$[\mathbf{Q}, \mathbf{R}, \mathbf{\Pi}, \mathbf{C}_{ee}] \leftarrow \mathrm{VBLAST\_QR}(\mathbf{H})$
$\mathbf{d} \leftarrow \mathbf{\Pi} \cdot \mathrm{QR\_DETECTION}(\mathbf{y}, \mathbf{Q}, \mathbf{R})$

function $[\mathbf{Q}, \mathbf{R}, \mathbf{\Pi}] = \mathrm{VBLAST\_QR}(\mathbf{H})$
  $\mathbf{W} \leftarrow \mathbf{0}$; $\mathbf{\Pi} \leftarrow \mathbf{0}$; $\mathbf{C}_{ee} \leftarrow \mathbf{0}$
  $\mathbf{M} \leftarrow \mathbf{H}$
  $\mathbf{i} \leftarrow \{0, 1, \ldots, N_d - 1\}$
  for $l = N_d - 1, \ldots, 0$ **do**
    $\mathbf{D} \leftarrow \mathbf{M}^{+}$
    $\mathbf{C}_{ee}^{\mathrm{lin}} \leftarrow N\sigma_n^2 \left(\mathbf{M}^{\mathsf{H}}\mathbf{M}\right)^{-1}$
    $r \leftarrow \underset{m}{\operatorname{argmin}} \left[\mathbf{C}_{ee}^{\mathrm{lin}}\right]_{m,m}$
    $k \leftarrow i_r$; **remove** element r from $\mathbf{i}$
    $\mathbf{w}_l^{\mathsf{H}} \leftarrow [\mathbf{D}]_{r,*} / \sqrt{\left[\mathbf{C}_{ee}^{\mathrm{lin}}\right]_{r,r}}$ ▷ normalized nulling vector
    $[\mathbf{\Pi}]_{k,l} \leftarrow 1$
    $[\mathbf{C}_{ee}]_{k,k} \leftarrow \left[\mathbf{C}_{ee}^{\mathrm{lin}}\right]_{r,r}$
    remove column r from $\mathbf{M}$
  end for
  $\mathbf{Q} \leftarrow \mathbf{W}^{\mathsf{H}}$ ▷ $\mathbf{W} = [\mathbf{w}_0, \mathbf{w}_1, \ldots, \mathbf{w}_{N_d-1}]^{\mathsf{H}}$
  $\mathbf{R} \leftarrow \mathbf{W}\mathbf{H}\mathbf{\Pi}$
end function

function $\tilde{\mathbf{d}}' = \mathrm{QR\_DETECTION}(\mathbf{y}, \mathbf{Q}, \mathbf{R})$
  $\tilde{\mathbf{d}}' \leftarrow \mathbf{Q}^{\mathsf{H}}\mathbf{y}$
  for $k = N_d, \ldots, 1$ **do**
    $\tilde{d}'_k \leftarrow \tilde{d}'_k / [\mathbf{R}]_{k,k}$
    $\tilde{d}'_l \leftarrow \tilde{d}'_l - [\mathbf{R}]_{l,k} \left\lfloor \tilde{d}'_k / [\mathbf{R}]_{k,k} \right\rceil, \quad l = 0, \ldots, k-1$
  end for
end function

The matrix $\mathbf{W}$ is a variant of $\mathbf{W}_{\mathrm{DFE}}$, where the column vectors are normalized. It is used to produce the same matrices $\mathbf{Q}, \mathbf{R}$ and $\mathbf{\Pi}$, as they would be generated by the sorted QR decomposition [WBKK03], shown in Section 5.2. The changes that distinguish Algorithm 7 from the initial DFE in Algorithm 5 shall be pointed out in the following:

- the determination of the nulling matrix $\mathbf{W}$ and the permutation matrix $\mathbf{\Pi}$ to describe the order of processing is separated from the actual decision feedback loop and needs

to be calculated only once per channel update

- the iterator k descends from N_d-1 in order to receive a matrix $\mathbf{R}$ with upper instead of lower triangular structure
- the nulling vectors $\mathbf{w}_k$ are normalized, thus $\mathbf{W}$ consists of orthonormal column vectors
- in order not to affect the current data estimate $\tilde{d}'_k$, the decision feedback has to be omitted for symbol k explicitly in line 26 (iterator l excluding k), as it cannot be removed as simple as in Algorithm 6, where $\mathbf{K}-\mathbf{I}$ sufficed
- the detection function QR_DETECTION works with any matrix $\mathbf{Q}$ that fulfills $\mathbf{Q}^{\mathsf{H}}\mathbf{Q} = \mathbf{I}$ and an $\mathbf{R}$ with upper triangular structure, both satisfying $\mathbf{QR} = \mathbf{H\Pi}$

From this context, a QR decomposition (see Chapter 5) of the channel matrix

$$\mathbf{H} = \mathbf{QR\Pi}^{\mathsf{T}} \tag{4.65}$$

is introduced, with the matrix $\mathbf{Q} \in \mathbb{C}^{(N_d+N_r)\times N_d}$, consisting of orthonormal column vectors, so that with $\mathbf{Q}^{\mathsf{H}}\mathbf{Q} = \mathbf{I}$ partial unitarity[6] is fulfilled, and a matrix $\mathbf{R} \in \mathbb{C}^{N_d\times N_d}$ of upper triangular structure.

Rewriting the channel model (2.37)

$$\mathbf{y} = \mathbf{Hd} + \mathbf{n} = \mathbf{QR} \cdot \overbrace{\mathbf{\Pi}^{\mathsf{T}}\mathbf{d}}^{\mathbf{d}'} + \mathbf{n} \tag{4.66}$$

$$\mathbf{y}' = \mathbf{Q}^{\mathsf{H}}\mathbf{y} = \mathbf{Rd}' + \underbrace{\mathbf{Q}^{\mathsf{H}}\mathbf{n}}_{\mathbf{n}'}, \tag{4.67}$$

helps to understand the DFE and in particular the function QR_DETECTION better. The multiplication of $\mathbf{Q}^{\mathsf{H}}$, which is executed in line 23, can be interpreted as a rotation into the N_d-dimensional subspace, spanned by $\mathbf{H}$. It neither has impact on the length of receive or noise vectors, nor does it introduce correlation, hence the noise statistics $\mathbf{Q}^{\mathsf{H}}\mathbf{n} = \mathbf{n}' \sim \mathcal{N}_{\mathrm{C}}\left(0; N\sigma_n^2\mathbf{I}\right)$ are maintained. Furthermore, it reorders the dimensions in the way necessary for the correct ordering of the QR detection algorithm, as it was determined in VBLAST_QR. Moreover, due to the triangular structure of $\mathbf{R}$, it can be said that the channel is transformed into a *spatially causal* one [Win04], considering the MIMO interpretation of the UW-OFDM system model introduced earlier. Of course, for UW-OFDM the spatial interpretation is incorrect as the diversity is achieved by some kind of coding, but the conclusions remain same.

Elaborating the k-th receive symbol yields

$$y'_k = [\mathbf{R}]_{kk}\, d'_k + \sum_{l=k+1}^{N_d} [\mathbf{R}]_{kl}\, d'_l + n'_k. \tag{4.68}$$

[6] Actual unitarity is fulfilled only if both $\mathbf{Q}^{\mathsf{H}}\mathbf{Q} = \mathbf{I}$ and $\mathbf{QQ}^{\mathsf{H}} = \mathbf{I}$ hold; the latter is not the case here.

In this representation the data symbol with the number $N_d - 1$ can be immediately estimated

$$\begin{aligned}\tilde{d}'_{N_d-1} &= y'_{N_d-1} / [\mathbf{R}]_{N_d-1,N_d-1} \\ &= d'_{N_d-1} + n'_{N_d-1} / [\mathbf{R}]_{N_d-1,N_d-1}\,.\end{aligned} \tag{4.69}$$

Subtracting the decision's influence on the remaining receive values yields

$$y''_k = y'_k - [\mathbf{R}]_{k,N_d-1} \left\lfloor \tilde{d}'_{N_d-1} \right\rceil, \quad \text{for } k = 1, \ldots, N_d - 1. \tag{4.70}$$

This step is called the *decision feedback*, hence the term "decision feedback equalization".

After repeating these steps accordingly for the following N_d-1 symbols, the fully processed receive symbol vector $\tilde{\mathbf{d}}'$ is present. As a last step the vector of detected symbols needs to be re-sorted

$$\tilde{\mathbf{d}} = \mathbf{\Pi}\tilde{\mathbf{d}}', \tag{4.71}$$

in order to restore the correct order of the detected symbols, which was destroyed due to the algorithmic processing.

Note, that if the quantizer in the feedback loop is bypassed, the cascade of $\mathbf{Q}^{\mathsf{H}}$ and feedback loop $\mathbf{R}^{-1}$, or equivalently $\mathbf{W}_{\mathrm{DFE}}$ and $\mathbf{K}$, is equal to $\mathbf{H}^{+}$: Omitting the slicing operation, the output in Figure 4.23 can be expressed as

$$\begin{aligned}\tilde{\mathbf{d}}' &= \mathbf{W}_{\mathrm{DFE}}\mathbf{y} - (\mathbf{K} - \mathbf{I})\tilde{\mathbf{d}}' \\ (\mathbf{I} + \mathbf{K} - \mathbf{I})\tilde{\mathbf{d}}' &= \mathbf{W}_{\mathrm{DFE}}\mathbf{y} \\ \mathbf{W}_{\mathrm{DFE}}\mathbf{H}\mathbf{\Pi}\tilde{\mathbf{d}}' &= \mathbf{W}_{\mathrm{DFE}}\mathbf{y} \\ \mathbf{H}\tilde{\mathbf{d}} &= \mathbf{y} \\ \tilde{\mathbf{d}} &= \mathbf{H}^{+}\mathbf{y}\end{aligned}$$

Thus, only the nonlinear quantizing operation distinguishes DFE from a linear equalization using the Moore-Penrose pseudo-inverse, which corresponds to the BLUE, as in Section 3.1.1. This quantization along with the feedback implies an equalization that does not amplify the noise.

4.2.5. On the Equivalence of DFE and NI

For non-block-based processing, it is well known that DFE and NI can be converted into each other [HM84, Fis02] quite easily, and both variants perform equivalently. For the block-based operation with UW-OFDM, the situation is a bit different.

In this section, this equivalence is tried to be recreated for UW-OFDM, with a slightly altered version of the iterative noise interpolator in Algorithm 4. Only uncoded transmission is considered in this section, and the determination of soft information neglected. Furthermore, the NI shall be performed with an initial linear data estimation using the BLUE, according to (3.10), and the ZF DFE according to (4.47) is employed, in order to have common statistical measures to start with. Algorithm 8 differs from the more general form in the following points:

- the loop is iterated N_d times, in order to do one noise interpolation for each data symbol (line 3)
- the noise interpolators are gathered in the rows of matrix $\mathbf{W}_{\mathrm{NI}}$ in line 7, their execution is done in line 16
- a permutation matrix $\mathbf{\Pi}$ in line 8 documents the order in which to do the noise interpolation

Algorithm 8 Separated Iterative Noise Interpolation

1: $\mathbf{C}_{ee} \leftarrow \mathbf{C}_{ee}^{\mathrm{lin}}$
2: $\mathbf{\Pi} \leftarrow \mathbf{0}$
3: **for** $l = 0, 1, \ldots, N_d - 1$ **do**
4: $\quad k \leftarrow \underset{m}{\operatorname{argmax}} \left[\mathbf{C}_{ee}\right]_{m,m}$ ▷ Pick noisiest sample
5: $\quad \mathcal{S}_k \leftarrow \{0, 1, \ldots, N_d - 1\} \backslash k$
6: $\quad \mathbf{w}^{\mathsf{T}} = \mathbf{r}_k^{\mathsf{T}} \left(\mathbf{R}_k\right)^{-1}$
7: $\quad \left[\mathbf{W}_{\mathrm{NI}}\right]_{l,*} \leftarrow \mathbf{w}^{\mathsf{T}}$
8: $\quad \left[\mathbf{\Pi}\right]_{k,l} \leftarrow 1$
9: $\quad \left[\mathbf{C}_{ee}\right]_{k,*} \leftarrow \left[\mathbf{C}_{ee}\right]_{k,*} - \mathbf{w}^{\mathsf{T}} \mathbf{C}_{ee}^{\mathsf{H}}$
10: $\quad \left[\mathbf{C}_{ee}\right]_{*,k} \leftarrow \left[\mathbf{C}_{ee}\right]_{*,k} - \mathbf{C}_{ee} \mathbf{w}^{*}$
11: **end for**
12:
13: $\tilde{\mathbf{d}}' \leftarrow \mathbf{\Pi}^{\mathsf{T}} \tilde{\mathbf{d}}^{\mathrm{lin}}$
14: $\mathbf{K}_{\mathrm{NI}} \leftarrow \mathbf{W}_{\mathrm{NI}} \mathbf{\Pi}$
15: **for** $n = 0, 1, \ldots, N_d - 1$ **do**
16: $\quad \tilde{d}'_n \leftarrow \tilde{d}'_n - \left[\mathbf{K}_{\mathrm{NI}}\right]_{n,*} \left(\tilde{\mathbf{d}}' - \left\lfloor \tilde{\mathbf{d}}' \right\rceil\right)$
17: **end for**
18:
19: $\tilde{\mathbf{d}} \leftarrow \mathbf{\Pi} \tilde{\mathbf{d}}'$

Starting point is again an estimate of a linear data estimation $\tilde{\mathbf{d}}^{\mathrm{lin}}$ and the error statistics $\mathbf{C}_{ee}^{\mathrm{lin}}$. This resembles the same separation of filter determination that does not depend on the received data and filter operation as introduced for the DFE in Algorithm 6. The samples to be used to for estimation are now chosen as $\mathcal{S}_k = \{0, 1, \ldots, N_d - 1\} \backslash k$. Along with the notation used earlier, in line 6, $\mathbf{r}_k^{\mathsf{T}} = \left[\mathbf{C}_{ee}\right]_{k,\mathcal{S}_k}$ is the k-th row of the current $\mathbf{C}_{ee}$ with only the columns listed in $\mathcal{S}_k$. $\mathbf{R}_k = \left[\mathbf{C}_{ee}\right]_{\mathcal{S}_k,\mathcal{S}_k}$ is extracted from the matrix $\mathbf{C}_{ee}$ by selecting all the rows and all the columns listed in $\mathcal{S}_k$.

After bringing the linear estimates into the correct detection order in line 13, and adapting the columns of $\mathbf{W}_{\mathrm{NI}}$ accordingly (line 14), the noise interpolation can be performed one after another, iterating $n = 0, 1, \ldots, N_d - 1$. Another permutation in the end in line 19 re-sorts the data symbols to the correct order. Now, this can be expressed as the block diagram shown in Figure 4.24a, where the linear data estimation is represented by the block $\mathbf{E}$.

With a little modification [Fis02], the block diagram can be brought into an equivalent one with two filter loops, as shown in Figure 4.24b. Furthermore, the matrix $\mathbf{W}_{\mathrm{NI}}\mathbf{\Pi}$ can be identified as an upper triangular matrix with a zero main diagonal.

In a last step, the first loop can be replaced with a matrix multiplication with

$$\mathbf{K} = \mathbf{I} - \mathbf{W}_{\mathrm{NI}}\mathbf{\Pi},$$

and the filter in the second loop gets a minus, such that

$$\mathbf{K}' = -\mathbf{W}_{\mathrm{NI}}\mathbf{\Pi} = \mathbf{K} - \mathbf{I},$$

as shown in Figure 4.24c. As in the second loop, the definition of $\mathbf{K}$ agrees with Figure 4.23, it also suggests to apply the definition in (4.60), $\mathbf{K} = \mathbf{W}_{\mathrm{DFE}}\mathbf{H}\mathbf{\Pi}$. Now, this is the exact same as the DFE block diagram in Figure 4.23, as evaluating $\mathbf{y'}$ just before the addition in Figure 4.24c yields

$$\begin{aligned}\mathbf{y}' &= \mathbf{K}\mathbf{\Pi}^{\mathsf{T}}\mathbf{E}\mathbf{y} \\ &= \mathbf{W}_{\mathrm{DFE}}\mathbf{H}\mathbf{\Pi}\mathbf{\Pi}^{\mathsf{T}}\mathbf{E}\mathbf{y}.\end{aligned}$$

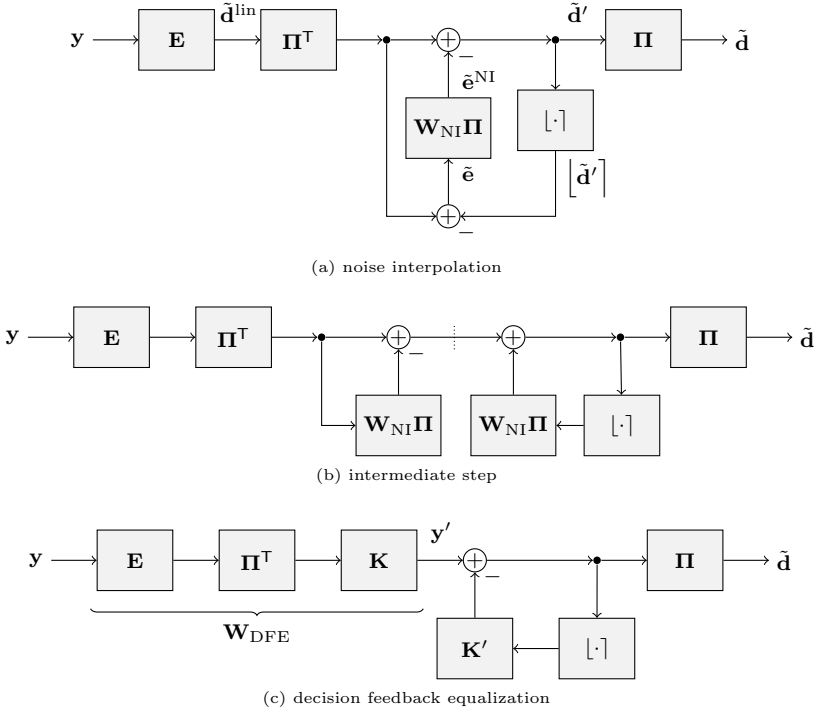
Since $\mathbf{\Pi}$ is orthogonal $\mathbf{\Pi}\mathbf{\Pi}^{\mathsf{T}} = \mathbf{I}$. Also remembering that $\mathbf{E}$ is the BLUE, and thus it resembles the Moore-Penrose pseudo-inverse of the channel, and $\mathbf{H}\mathbf{E} = \mathbf{I}$, this becomes

$$\mathbf{y}' = \mathbf{W}_{\mathrm{DFE}}\mathbf{y}.$$

This proves the equivalence to Figure 4.23 and Algorithm 6, with

$$\begin{aligned}\mathbf{W}_{\mathrm{NI}} &= \mathbf{I} - \mathbf{W}_{\mathrm{DFE}}\mathbf{H}\mathbf{\Pi}, \\ \mathbf{W}_{\mathrm{DFE}} &= (\mathbf{I} - \mathbf{W}_{\mathrm{NI}})\,\mathbf{\Pi}^{\mathsf{T}}\mathbf{E}.\end{aligned}$$

All those representations (a) to (c) in Figure 4.24 are equivalent in terms of a block diagram [Fis02]. Even though performing the filter determination parts in Algorithm 6 and 8 yield almost equal permutation matrices $\mathbf{\Pi}$ (only minor differences not affecting the final outcome could be observed in simulations), DFE and NI do not perform equally. It has not been mentioned yet that with the evolution to Figure (b) a switch in the order of operation has to take place, exactly in between the two loops: While the first forward loop needs to be executed in the order $n = 0, 1, \ldots, N_d - 1$, the second feedback loop needs to operate in reverse order $n = N_d{-}1, N_d{-}2, \ldots, 0$. Then, DFE performance is achieved, even with filter matrices generated with the separated iterative NI Algorithm 8. In this single step lies the difference between the NI and the DFE in this setup. Also, these considerations are valid up to some very limited extent only, as many restrictions had to be made for Algorithm 8. Any deviation from this algorithm prevents the creation of an upper triangular $\mathbf{K}$, which is a prerequisite for this DFE interpretation.



(a) noise interpolation

(b) intermediate step

(c) decision feedback equalization

Figure 4.24.: Conversion of noise interpolation to decision feedback equalization receiver.

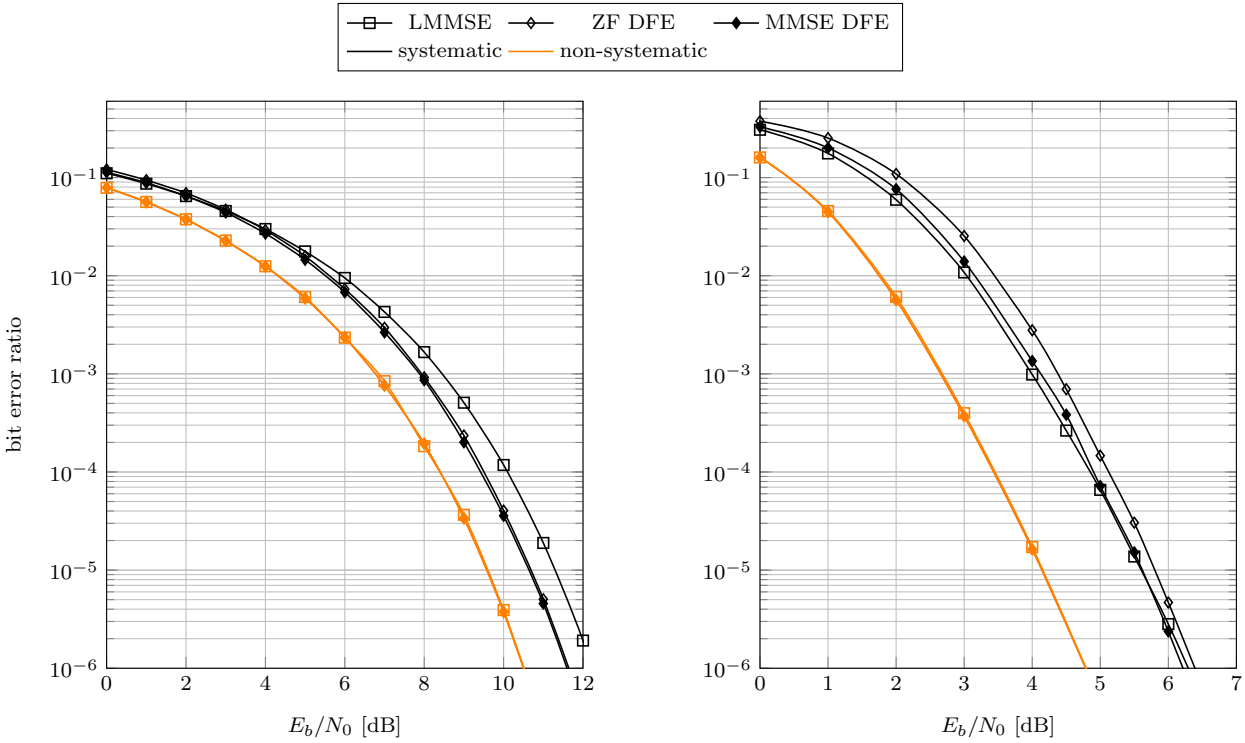
4.2.6. Performance Evaluation

For the evaluation of the bit error performance of the DFE receivers, the ZF and the MMSE version of the DFE are compared against the LMMSE data estimator. As usual, the results for uncoded and coded transmission are shown in the AWGN channel and multipath environment. Systematically as well as non-systematically generated UW-OFDM transmission is shown.

For uncoded transmission of systematically generated UW-OFDM symbols in the AWGN channel, the DFE receivers achieve a gain of up to 0.6 dB over the LMMSE estimator at high E_b/N_0-ratios. Then, the regularization term $N\sigma_n^2/\sigma_d^2$ in (4.53) becomes very small and the MMSE DFE and the ZF DFE coincide, as visible in the figures. At low SNR the MMSE DFE is slightly ahead of both, the LMMSE and the ZF DFE. For coded transmission, the DFE receivers unfortunately fall behind the LMMSE performance. Only the MMSE DFE can still supersede the LMMSE estimator in the simulated range: Above an E_b/N_0-ratio of about 5.3 dB and a BER of $1.5 \cdot 10^{-5}$ the MMSE DFE becomes the better data detector. For the non-systematic generation approach, the DFE performances concur with the LMMSE results completely and perform exactly the same, with a gain of 1.1 dB for uncoded and 1.4 dB for coded transmission, compared to the best performing receiver for systematically generated UW-OFDM, which is the MMSE DFE. This means that the DFE receivers are not able to beat the LMMSE estimator for non-systematic UW-OFDM in AWGN.

In multipath environment, a bigger advantage for the DFE receivers can be recognized. For uncoded transmission, the DFE variants have a big gain over the LMMSE, increasing with the SNR, of up to 3 dB for the systematically generated system and even up to 5.5 dB for non-systematically generated UW-OFDM. Furthermore, the MMSE version supersedes the ZF version relatively constant over the whole SNR range by 0.2 dB for non-systematic UW-OFDM. Surprisingly, for the systematic systems the MMSE DFE shows an advance of up to 0.5 dB, which is *increasing* with the SNR.

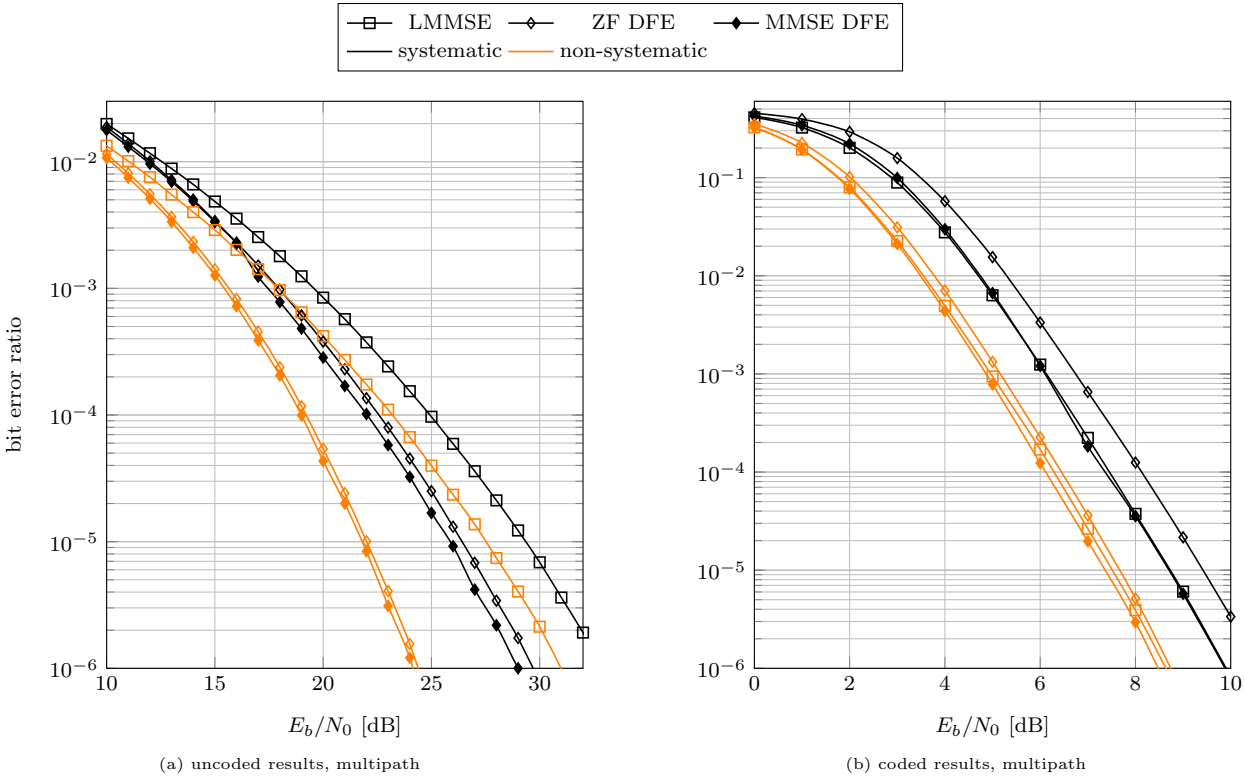
For coded transmission in multipath environment using the systematic approach, the MMSE DFE performs very close to the LMMSE. The ZF DFE, however, is not able to compete with the other two receivers and performs about 0.7 dB worse than the LMMSE estimator. For the non-systematically generating UW-OFDM system the ZF DFE is also not able to outperform the LMMSE, with a loss of 0.15 dB. The MMSE DFE supersedes the LMMSE estimator by 0.16 dB. Due to its overall good performance in coded and uncoded transmission, the MMSE DFE receiver will be used in later comparisons.



(a) uncoded results, AWGN

(b) coded results, AWGN

Figure 4.25.: BER performance of the DFE estimators for system B in AWGN according to Appendix A.



(a) uncoded results, multipath

(b) coded results, multipath

Figure 4.26.: BER performance of the DFE estimators for system B in multipath environment according to Appendix A.

4.3. Maximum Likelihood Sequence Estimation

The best detection performance can be achieved by a maximum a-posteriori (MAP) detection [Pro00]. The MAP detector seeks for the data vector that is most likely to be sent, once the receive vector $\mathbf{y}$ is known. In other words, it maximizes the a-posteriori probability of the transmit vector after receiving $\mathbf{y}$, which is the conditional probability of the data vector $\mathbf{d}$ under the knowledge of $\mathbf{y}$

$$\hat{\mathbf{d}} = \underset{\mathbf{d}\in\mathcal{A}^{N_d}}{\operatorname{argmax}} \Pr\left(\mathbf{d}|\mathbf{y}\right), \tag{4.72}$$

where $\mathcal{A}^{N_d}$ is the set of all possible data vectors. Using Bayes' theorem this a-posteriori probability can be rewritten as

$$\begin{aligned}\Pr\left(\mathbf{d}|\mathbf{y}\right) &= \frac{p_{y|d}\left(\mathbf{y}|\mathbf{d}\right)\Pr\left(\mathbf{d}\right)}{p_y\left(\mathbf{y}\right)} \\ &= \frac{p_{y|d}\left(\mathbf{y}|\mathbf{d}\right)\Pr\left(\mathbf{d}\right)}{\sum_{\mathbf{d'}\in\mathcal{A}^{N_d}} p_{y|d}\left(\mathbf{y}|\mathbf{d'}\right)\Pr\left(\mathbf{d'}\right)}.\end{aligned} \tag{4.73}$$

Equiprobable and iid. binary source symbols and thus independent and uniformly distributed transmit symbols are assumed. Hence, the probability of any data vector is $\Pr\left(\mathbf{d}\right) = |\mathcal{A}|^{-N_d}$, and the term can be reduced to

$$\Pr\left(\mathbf{d}|\mathbf{y}\right) = \frac{p_{y|d}\left(\mathbf{y}|\mathbf{d}\right)}{\sum_{\mathbf{d'}\in\mathcal{A}^{N_d}} p_{y|d}\left(\mathbf{y}|\mathbf{d'}\right)}. \tag{4.74}$$

The summation term in the denominator is constant for a given receive vector $\mathbf{y}$ and can be excluded in the maximization, transforming the MAP problem into a maximum-likelihood sequence estimation (MLSE)

$$\hat{\mathbf{d}} = \underset{\mathbf{d}\in\mathcal{A}^{N_d}}{\operatorname{argmax}}\, p_{y|d}\left(\mathbf{y}|\mathbf{d}\right). \tag{4.75}$$

4.3.1. MLSE Processing

The channel model (4.1) in mind, the covariance matrix of the additive noise is known as $\mathbf{C}_{nn} = N\sigma_n^2\mathbf{I}$. With the multivariate complex Gaussian distribution [Kam04] the probability in (4.75) can be written as

$$p_{y|d}\left(\mathbf{y}|\mathbf{d}\right) = \frac{1}{\pi^{N_d}\det\mathbf{C}_{nn}}\exp\left\{-\left(\mathbf{Hd}-\mathbf{y}\right)^{\mathsf{H}}\mathbf{C}_{nn}^{-1}\left(\mathbf{Hd}-\mathbf{y}\right)\right\} \tag{4.76}$$

$$= \frac{1}{(\pi N\sigma_n^2)^{N_d}}\exp\left\{-\frac{\|\mathbf{Hd}-\mathbf{y}\|_2^2}{N\sigma_n^2}\right\}. \tag{4.77}$$

The constant factor in front of the exponential function can be disregarded for the maximization. It is also allowed to search for the maximum of the logarithm of the expression, as the logarithm is a strictly monotonic function, which yields

$$\hat{\mathbf{d}} = \underset{\mathbf{d}\in\mathcal{A}^{N_d}}{\operatorname{argmax}} -\frac{\|\mathbf{Hd}-\mathbf{y}\|_2^2}{N\sigma_n^2}. \tag{4.78}$$

The denominator is constant and can be disregarded again, and the minus turns the maximization into a minimization. The remaining MLSE problem is to search for the minimizer of the distance between any valid transmit vector after channel propagation and the received vector:

$$\hat{\mathbf{d}} = \underset{\mathbf{d}\in\mathcal{A}^{N_d}}{\operatorname{argmin}} ||\mathbf{Hd}-\mathbf{y}||_2^2, \tag{4.79}$$

with the channel matrix $\mathbf{H}$ including the UW-OFDM generator matrix and the multipath channel, according to (4.1). In other words, this is the search for the closest point for all $\mathbf{Hd}$ out of the N_d-dimensional QAM grid $\mathcal{A}^{N_d}$ to the receive vector $\mathbf{y}$.

Both equations (4.72) and (4.75) are equivalent in case of equiprobable data symbols. Both expressions suggest to check *every* possible data vector of length N_d with symbols from the alphabet $\mathcal{A}$, also in the evolved form (4.79). For practical UW-OFDM setups this is an impossible task, since there are $|\mathcal{A}|^{N_d}$ valid data vectors, which is about 10^{29} possible data vectors[7] for the exemplary system B (see Appendix A) and a quaternary alphabet, respectively. In Section 4.4 an efficient algorithm is presented that yields the same results in a practical amount of time. For this reason, MLSE simulation results are omitted here.

4.3.2. Reliability Information for MLSE Results

For the MLSE, reliability information can be directly determined without the detour over the error statistics. Every data vector $\mathbf{d} \in \mathcal{A}^{(N_d\times 1)}$ can also be represented by its binary equivalent $\mathbf{b} \in \{0,1\}^{(N_d n_b\times 1)}$, when n_b binary symbols form each data symbol, such that $n_b = \mathrm{ld}|\mathcal{A}|$. As already suggested in Section 2.5, the probability for $b_l = 0$ is used as reliability information under the knowledge of the receive vector $\mathbf{y}$, as well as the probability for $b_l = 1$. These probabilities are usually very close to zero or one, what suggests the use of a presentation that has a better resolution near these borders. Therefore, a log-likelihood ratio (LLR) [Hub02] is the common choice that accomplishes these needs and shall be defined in this work as

$$L\left(\Pr\left(x=1|\cdots\right)\right) \triangleq \ln\frac{\Pr\left(x=1|\cdots\right)}{\Pr\left(x=0|\cdots\right)}. \tag{4.80}$$

For the UW-OFDM purpose here, the LLR L_l is specific for each symbol of $\hat{\mathbf{b}}$, which is the binary representation of the detected MLSE data vector $\hat{\mathbf{d}}$, and introduced as

$$L_l = L\left(\Pr\left(b_l=1|\mathbf{y}\right)\right) = \ln\frac{\Pr\left(b_l=1|\mathbf{y}\right)}{\Pr\left(b_l=0|\mathbf{y}\right)}. \tag{4.81}$$

[7] Assuming generous 10 billion data vectors to be checked per second, it would still take more than 250 billion years to do a brute-force MLSE on a single OFDM symbol in system B with 4-QAM.

Due to its definition, an LLR value close to zero indicates a probability of around 1/2, which poses a high uncertainty regarding its decision for a one or a zero. Higher magnitudes indicate higher certainty. The sign of the LLRs on the other hand, indicate the tendency. This means negative values indicate a higher probability for a zero symbol, while positive values stand for a one.

With Bayes' theorem and the knowledge of equiprobable input symbols the derivation as in (4.73) is followed and the probabilities can be changed to probability densities

$$L_l = \ln \frac{p_{y|b=1}\left(\mathbf{y}|b_l = 1\right)}{p_{y|b=0}\left(\mathbf{y}|b_l = 0\right)} = \ln \frac{\sum\limits_{\mathbf{b}\in\mathcal{B}_l^{(1)}} p_{y|b}\left(\mathbf{y}|\mathbf{b}\right)}{\sum\limits_{\mathbf{b}\in\mathcal{B}_l^{(0)}} p_{y|b}\left(\mathbf{y}|\mathbf{b}\right)}, \tag{4.82}$$

where $\mathcal{B}_l^{(\beta)}$ is the set of all binary symbol vectors $\{0,1\}^{(N_d n_b \times 1)}$, with the symbol at position l fixed to β. Using the equivalent representation of the binary symbol vector as a data symbol vector $\mathbf{b} \to \mathbf{d}(\mathbf{b})$ and the findings in (4.77), these probability densities are given as

$$p_{y|b}\left(\mathbf{y}|\mathbf{b}\right) = \frac{1}{(\pi N \sigma_n^2)^{N_d}} \exp\left\{-\frac{\|\mathbf{H}\mathbf{d}(\mathbf{b}) - \mathbf{y}\|_2^2}{N\sigma_n^2}\right\}. \tag{4.83}$$

and the LLR becomes

$$L_l = \ln \frac{\sum\limits_{\mathbf{b}\in\mathcal{B}_l^{(1)}} \exp\left\{-\frac{\|\mathbf{H}\mathbf{d}(\mathbf{b})-\mathbf{y}\|_2^2}{N\sigma_n^2}\right\}}{\sum\limits_{\mathbf{b}\in\mathcal{B}_l^{(0)}} \exp\left\{-\frac{\|\mathbf{H}\mathbf{d}(\mathbf{b})-\mathbf{y}\|_2^2}{N\sigma_n^2}\right\}} \tag{4.84}$$

$$= \ln \sum_{\mathbf{b}\in\mathcal{B}_l^{(1)}} \exp\left\{-\frac{\|\mathbf{H}\mathbf{d}(\mathbf{b}) - \mathbf{y}\|_2^2}{N\sigma_n^2}\right\} - \ln \sum_{\mathbf{b}\in\mathcal{B}_l^{(0)}} \exp\left\{-\frac{\|\mathbf{H}\mathbf{d}(\mathbf{b}) - \mathbf{y}\|_2^2}{N\sigma_n^2}\right\}. \tag{4.85}$$

Again, the unpractical situation is at hand that in order to compute the LLR values for one OFDM symbol, the probabilities of all possible data vectors need to be checked. An implementation for determining the exact LLRs is shown in Algorithm 9. The actual computation is only feasible for very small UW-OFDM systems, though.

This algorithm is able to retrieve the maximum-likelihood sequence estimate $\hat{\mathbf{b}}$ in its binary symbol representation from $\mathbf{y}$, and additionally provides the associated log-likelihood ratios $\mathbf{L} = [L_0, L_1, \ldots, L_{N_d-1}]^\mathsf{T}$ for every symbol. The value p, as calculated in line 6, is not a real probability, but represents one term of any sum in (4.85). Then, in the lines 8 and 9 this p contributes to the sum of LLR number l, depending on the value of the binary representation b_l of the current data vector $\mathbf{d}$. The final LLR vector is determined in line 11. Then, the binary representation of the most likely sequence is finally determined in line 12. The operations $\ln \cdot$ and $\mathrm{sign}(\cdot)$ work element-wise on the argument vectors.

Algorithm 9 Maximum-Likelihood Sequence Estimator with Soft-Output

function $[\hat{\mathbf{b}}, \mathbf{L}] = \mathrm{MLSO}(\mathbf{y}, \mathbf{H}, \sigma_n^2, \mathcal{A})$
  $\mathbf{p}^{(0)} \leftarrow \mathbf{0}$
  $\mathbf{p}^{(1)} \leftarrow \mathbf{0}$
  repeat
    $\mathbf{d} \leftarrow$ next valid data vector
    $p \leftarrow \exp\left\{-\frac{\|\mathbf{Hd}-\mathbf{y}\|_2^2}{N\sigma_n^2}\right\}$
    $\mathbf{b} \leftarrow$ binary representation of $\mathbf{d}$
    $p_l^{(0)} \leftarrow p_l^{(0)} + p, \quad \forall l$ with $b_l = 0$
    $p_l^{(1)} \leftarrow p_l^{(1)} + p, \quad \forall l$ with $b_l = 1$
  until all valid $\mathbf{d}$ tried
  $\mathbf{L} \leftarrow \ln \mathbf{p}_1 - \ln \mathbf{p}_0$
  $\hat{\mathbf{b}} \leftarrow \frac{\mathrm{sign}(\mathbf{L})+1}{2}$
end function

However, the same restrictions as in determining the binary symbol vector with the MLSE apply here: In practical UW-OFDM systems the problem is not solvable due the vast number of valid OFDM symbols. Therefore, it makes sense to consider the brute-force MLSE method for small UW-OFDM systems only. For this reason the exemplary system ML (see Appendix A) was introduced, which reduces the computation effort with a 4-ary alphabet to $4^8 = 65\,536$ possible data vectors. BER results of the MLSE will be shown as reference, together with the results of the sphere decoder, and in particular to quantify the sphere decoder's loss due to the max-log approximation in Section 4.4.2.

4.4. Sphere Decoding

In Section 4.3, the optimum way to decode UW-OFDM symbols is shown. However, for a brute-force MLSE according to (4.79) *every* possible data vector $\mathbf{d}$ needs to be considered in the minimization

$$\hat{\mathbf{d}} = \underset{\mathbf{d}\in\mathcal{A}^{N_d}}{\operatorname{argmin}} ||\mathbf{H}\mathbf{d} - \mathbf{y}||_2^2,$$

which is an impossible task for practical UW-OFDM systems due to its enormous complexity. For this kind of problem the sphere decoding (SD) algorithm is an attractive method, as it is able to solve (4.79) in a tractable amount of time.

As the data vector can only adopt discrete values and the UW-OFDM generator matrix can be considered as a lattice basis, the product $\mathbf{H}\mathbf{d}$ can be interpreted as a mathematical lattice. In contrast to most lattice descriptions it is finite, as the coefficients in $\mathbf{d}$ can only take values from a finite transmit symbol alphabet, and it is complex in value, as coefficients and basis are defined on $\mathbb{C}$.

The initial concept as search for short vectors in lattices was shown in [Poh81, FP85]. An improved interpretation was given in [SE94]. Amongst others, the evolution of the algorithm for data detection was shown in [Mow94], until [VB99] impressed the term "sphere decoding", which became very popular from then. In many publications the concept was adapted for different applications [DCB00, BBW+05], its possibilities further improved [AEVZ02, DGC03] and its potentially exponential complexity eased [HV05]. While sphere decoding became very popular for MIMO signal processing (for example [BBW+05, AA08]), it is dealt with the sphere decoder in the UW-OFDM context, here.

4.4.1. Sphere Decoding for Unique Word OFDM

In this section, an adapted version of the algorithm in [AEVZ02] for Unique Word OFDM is presented [OH11b]. To allow for sphere decoding, a QR decomposition of the channel matrix

$$\mathbf{H} = \underbrace{\begin{bmatrix}\mathbf{Q} & \mathbf{Q}_0\end{bmatrix}}_{\mathbf{Q}'} \begin{bmatrix}\mathbf{R} \\ \mathbf{0}\end{bmatrix} \tag{4.86}$$

enables the required simplifications, where $\mathbf{Q}' \in \mathbb{C}^{(N_d+N_r)\times(N_d+N_r)}$ is a unitary matrix and $\mathbf{R} \in \mathbb{C}^{N_d\times N_d}$ is upper triangular. The parts of $\mathbf{Q}'$ can be interpreted as follows: The submatrix $\mathbf{Q}$ corresponds to the co-image of $\mathbf{H}$ – similar to the $\mathbf{Q}$ for DFE in (4.65) – while $\mathbf{Q}_0$ represents the left null space of $\mathbf{H}$, which extends $\mathbf{Q}$ to be unitary. The order of the QR decomposition has great impact on the execution of the SD. As the SD always yields the MLSE solution (exception is soft-output SD), here the execution time is affected, rather than the bit error performance as for the DFE. The sorted QR decomposition (see Section 5.2) is selected for the SD receivers. Thus, for the bit error analysis, the QR decomposition has no importance, unless the number of node visits or sphere radii are restricted. By restricting sphere radii, execution time can be further influenced, which is not covered in this work.

Inserting the decomposed channel matrix into (4.79), the term to be minimized becomes [OH11b]

$$\begin{aligned}\|\mathbf{H}\mathbf{d}-\mathbf{y}\|_2^2 &= \left\|\mathbf{Q}'\begin{bmatrix}\mathbf{R}\\\mathbf{0}\end{bmatrix}\mathbf{d}-\mathbf{y}\right\|_2^2 \\ &= \left\|\mathbf{Q}'^{\mathsf{H}}\mathbf{Q}'\begin{bmatrix}\mathbf{R}\\\mathbf{0}\end{bmatrix}\mathbf{d}-\mathbf{Q}'^{\mathsf{H}}\mathbf{y}\right\|_2^2 \qquad (4.87)\\ &= \left\|\begin{bmatrix}\mathbf{R}\\\mathbf{0}\end{bmatrix}\mathbf{d}-\begin{bmatrix}\mathbf{Q}^{\mathsf{H}}\\\mathbf{Q}_0^{\mathsf{H}}\end{bmatrix}\mathbf{y}\right\|_2^2 \\ &= \left\|\mathbf{R}\mathbf{d}-\mathbf{Q}^{\mathsf{H}}\mathbf{y}\right\|_2^2+\left\|\mathbf{Q}_0^{\mathsf{H}}\mathbf{y}\right\|_2^2. \qquad (4.88)\end{aligned}$$

In the step to (4.87), the multiplication with the unitary $\mathbf{Q}'^{\mathsf{H}}$ can be geometrically interpreted as a rotation in the N_d-dimensional vector space and does not affect the norm at all. The matrix $\mathbf{R}$ on the other hand stretches the transmit vector, altering its norm. As the second term in (4.88) is independent of $\mathbf{d}$, it can be dropped and the minimization problem (4.79) transforms into

$$\hat{\mathbf{d}} = \underset{\mathbf{d}\in\mathcal{A}^{N_d}}{\operatorname{argmin}} \left\|\mathbf{R}\mathbf{d}-\mathbf{y}'\right\|_2^2, \qquad (4.89)$$

using

$$\begin{aligned}\mathbf{y}' &= \mathbf{Q}^{\mathsf{H}}\mathbf{y}\\ &= \mathbf{R}\mathbf{d}+\mathbf{Q}^{\mathsf{H}}\mathbf{n}.\end{aligned} \qquad (4.90)$$

Note that the unitary transformation $\mathbf{Q}^{\mathsf{H}}\mathbf{n}$ does not change the noise statistics of $\mathbf{n}$ at all. Due to the triangular structure of $\mathbf{R}$, (4.89) can be solved in an iterative fashion using the sphere decoder algorithm [VB99, AEVZ02]. This procedure is shown in Algorithm 10 in an adapted form to deal with a finite and complex valued transmit alphabet, and hence suited for UW-OFDM transmission.

In the following, some aspects of Algorithm 10 are explained: For the sphere decoding procedure, an inversion of $\mathbf{R}$ will be necessary, which is a simple and straightforward procedure due to its triangular structure. The matrix

$$\mathbf{F} = \mathbf{R}^{-1} \qquad (4.91)$$

will be of good use from here and has an upper triangular structure as well. Likewise it can be shown that the diagonal elements of $\mathbf{F}$ correspond to the reciprocal diagonal elements of $\mathbf{R}$ such that

$$[\mathbf{F}]_{k,k} = 1/[\mathbf{R}]_{k,k}. \qquad (4.92)$$

In order to get a better understanding of the operation method of the SD, three operating domains shall be identified that is switched in between:

- receive domain, with the unaltered receive vector: $\mathbf{y} = \mathbf{Q}\mathbf{R}\mathbf{d}+\mathbf{n}$
- rotated receive domain or stretched domain, where the receive vector is de-rotated by $\mathbf{Q}^{\mathsf{H}}$, but is left with the stretching influence by $\mathbf{R}$: $\mathbf{y}' = \mathbf{R}\mathbf{d}+\mathbf{Q}^{\mathsf{H}}\mathbf{n}$

- equalized domain, where additionally the stretching by $\mathbf{R}$ is removed, and thus all channel influence is canceled; here the transmit symbols are directly visible, up to the AWGN: $\tilde{\mathbf{y}} = \mathbf{d} + \mathbf{F}\mathbf{Q}^{\mathsf{H}}\mathbf{n}$

In which domain the calculatory steps are performed is selected in order to optimize computational performance.

At first in line 4, the de-rotated receive vector is equalized regarding the matrix $\mathbf{R}$, in order to switch to the equalized domain model

$$\tilde{\mathbf{y}} = \mathbf{F}\mathbf{y}' \overset{(4.90)}{=} \mathbf{F}\mathbf{Q}^{\mathsf{H}}\mathbf{y} = \underbrace{\mathbf{F}\mathbf{R}}_{\overset{(4.91)}{=}\mathbf{I}}\,\mathbf{d} + \mathbf{F}\mathbf{Q}^{\mathsf{H}}\mathbf{n}, \tag{4.93}$$

which changes the minimization problem (4.89) to

$$\hat{\mathbf{d}} = \underset{\mathbf{d}\in\mathcal{A}^{N_d}}{\operatorname{argmin}} \|\mathbf{d} - \tilde{\mathbf{y}}\|_2^2 \,. \tag{4.94}$$

Now, the operation of the algorithm is followed until the first decoding hypothesis is found. The detection process starts at the last symbol of the data vector $\mathbf{d}$ with number $k = N_d - 1$. The symbol number k is also called the *level*, indicating that $N_d - k$ data symbols are preliminarily decided and fixed after the processing for the given level k. A decrease of the level corresponds to deciding for another data symbol, while an increase is accompanied by giving up the decision of a symbol and reverting to an earlier state in order to consider other possibilities. Therefore, a version of $\tilde{\mathbf{y}}$ specific to each level needs to be stored, hence the superscript in Algorithm 10.

In the beginning, the initList function is called for the root level $k = N_d - 1$. This function creates a list $\mathbf{A}_k$, which is specific for each level k, with all elements of the transmit symbol alphabet $\mathcal{A}$ together with their additive metric δ. The additive metric δ is the weight a particular symbol would add to the overall Euclidean distance of the decoding hypothesis to the receive vector. As it is the easiest way to determine this distance in the equalized domain, but the additive metric applies in the stretched domain[8], the de-stretching by $\mathbf{F}$ needs to be reversed:

$$\begin{aligned} \delta &= [\mathbf{R}]_{k,k}\, d_k - y'_k \\ &= [\mathbf{R}]_{k,k}\, d_k - [\mathbf{F}]^{-1}_{k,k}\, \tilde{y}_k \\ &= \frac{d_k - \tilde{y}_k}{[\mathbf{F}]_{k,k}} \end{aligned} \tag{4.95}$$

This is performed in line 31 with a division by the diagonal element of $\mathbf{F}$, with $\mathbf{a}$ being the vector of elements from $\mathcal{A}$. The list $\mathbf{A}_k$ is sorted by this additive metric in ascending order, which also gives the testing order.

Due to the upper triangular structure of $\mathbf{R}$ and $\mathbf{F} = \mathbf{R}^{-1}$ all other symbols of the data vector $\mathbf{d}$ are not regarded yet and do not influence the cumulative metric Δ_k, at this point. This is why it is possible to consider symbol d_{N_d-1} alone and determine an additive metric δ like that. The vectors and matrices are visualized in Figure 4.27, where the dark

[8] The metric is actually only valid in the receive domain, but metrics do not change when switching between receive and stretched domain, due to the unitarity of $\mathbf{Q}$.

Algorithm 10 Sphere Decoder

function $\hat{\mathbf{d}} = \text{SD}(\mathbf{y}', \mathbf{F}, \mathcal{A})$
  $\Delta_{N_d} \leftarrow 0;\ \lambda^{\text{ML}} \leftarrow \infty$
  $k \leftarrow N_d - 1$
  $\tilde{\mathbf{y}}^{(k)} \leftarrow \mathbf{F}\mathbf{y}'$
  $\mathbf{A}_k \leftarrow$ initList$(k, \tilde{y}_k^{(k)}, [\mathbf{F}]_{k,k}, \mathcal{A})$
  while $k < N_d$ **do** ▷ termination criterion
    $[d_k, \delta, \mathbf{A}_k] \leftarrow$ getNextSymbol$(\mathbf{A}_k)$
    if isempty(d_k) **then** ▷ all symbols from $\mathcal{A}$ tried
      $k \leftarrow k + 1$
    else
      $\Delta_k \leftarrow \Delta_{k+1} + |\delta|^2$ ▷ update cumulative metric
      if $\Delta_k < \lambda^{\text{ML}}$ **then**
        if $k > 0$ **then** ▷ not at a leaf yet
          $\tilde{y}_l^{(k-1)} \leftarrow \tilde{y}_l^{(k)} - \delta\,[\mathbf{F}]_{l,k}, \quad l = 0, \ldots, k-1$
          $k \leftarrow k - 1$
          $\mathbf{A}_k \leftarrow$ initList$(k, \tilde{y}_k^{(k)}, [\mathbf{F}]_{k,k}, \mathcal{A})$
        else ▷ new ML hypothesis found
          $\hat{\mathbf{d}} \leftarrow \mathbf{d}$
          $\lambda^{\text{ML}} \leftarrow \Delta_k$
          $k \leftarrow k + 1$
        end if
      else
        $k \leftarrow k + 1$
      end if
    end if
  end while
end function

function $\mathbf{A}_k = \text{initList}(k, \tilde{y}, f, \mathcal{A})$
  $\mathbf{a} \leftarrow$ all elements of $\mathcal{A}$
  $\boldsymbol{\delta} \leftarrow (\mathbf{a} - \tilde{y})/f$
  $\mathbf{A}_k \leftarrow \begin{bmatrix}\mathbf{a} & \boldsymbol{\delta}\end{bmatrix}$
  sort rows of $\mathbf{A}_k$ according to the second column in ascending order
end function

function $[d, \delta, \mathbf{A}_k] = \text{getNextSymbol}(\mathbf{A}_k)$
  $d \leftarrow [\mathbf{A}_k]_{1,1}$
  $\delta \leftarrow [\mathbf{A}_k]_{1,2}$
  remove first row of $\mathbf{A}_k$
end function

orange outlined parts participate in the calculation of the additive metric δ at level $k = N_d - 1$. On a side note, deciding for the symbol with minimum additive metric, which is done here by sorting with respect to δ, is the equivalent to the slicing operation for the DFE Algorithm 7 in line 26.

Then, the iteration loop performs on each level k the following steps: First it retrieves the next symbol d_k and its additive metric δ from list $\mathbf{A}_k$ via GETNEXTSYMBOL and removes the entry (line 39). In line 11, the cumulative metric Δ_k for level k is calculated by adding the metric from one level above and the current additive metric. If this cumulative metric is smaller than the overall metric of the decoding hypothesis λ^{ML}, the current symbol is stored for the levels below and its influence from the equalized receive vector $\tilde{\mathbf{y}}^{(k-1)}$ is removed (line 14). As $\tilde{\mathbf{y}}$ is the equalized receive vector, the additive metric needs to be translated to the equalized domain as well. This operation is described in column $N_d - 1$ in the rows $0, \ldots, k-1$ of $\mathbf{F}$ or in the dark blue outline in Figure 4.27. This operation is the corresponding version of the interference cancellation step in line 13 of the DFE Algorithm 5 and is also present in the other DFE versions.

After k is decreased and the list $\mathbf{A}_k$ is created for the new level, the loop starts over, and the level is decreased with each iteration, until $k = 0$ is reached. Having worked level $k = 0$ means a complete valid data symbol vector is present, which is stored as decoding hypothesis $\hat{\mathbf{d}}$ along with the metric λ^{ML} this vector yields.

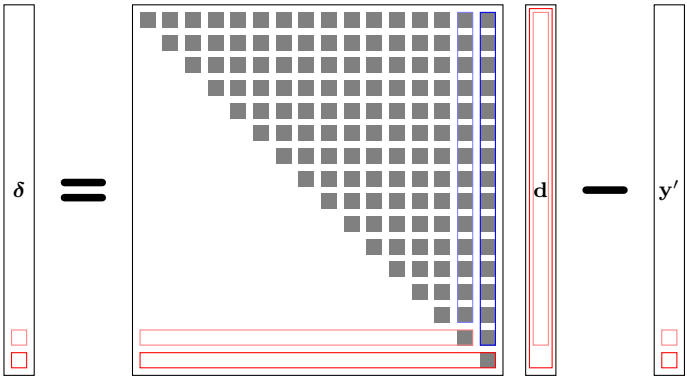


Figure 4.27.: Visualization of the dimensions of $\mathbf{Rd} - \mathbf{y}'$.

Assuming the same QR decomposition as for the DFE in (4.65) and comparing with the QR_DETECTION function in Algorithm 7, the very same operations were done, up to this point. And in fact, if the SD stopped here, the exact same decoding hypothesis $\hat{\mathbf{d}}$ is found. By deciding for the symbols adding the least amount to the cumulative metric Δ_k however, the possibility of a lower overall metric is ignored, which could have been found after walking a branch that is not the minimizer of the current additive metric. This is what the SD ensures by considering the other symbols at the levels $k > 0$ as well. After finding an ML hypothesis and moving up one level, the next-best symbol is picked from $\mathbf{A}_k$ together with its additive metric (line 7). If the list is empty (line 8), all possibilities on this level are exhausted and the level needs to be increased.

If the cumulative metric Δ_k exceeds a previously found best metric λ^{ML}, it does not make any sense to hold on to the current partial data vector under test, as the additive metric only increases the result. Hence, it is moved up one level (line 23). On the other hand, if the current cumulative metric is smaller than the currently best metric λ^{ML}, there is still the chance to find a better overall solution and two cases can be distinguished:

- If the current level is $k = 0$ a new best decoding hypothesis is reached (line 17) and stored. Next, the algorithm has to move up a level.
- If $k > 0$ (line 13), the level is decreased and the new symbol list $\mathbf{A}_k$ is prepared. Before that, the equalized receive vector for the new level is determined by removing the portion of the data symbol that was decided for before, from the receive vector.

As soon as k is increased beyond the root level $N_d - 1$, no more symbols worth checking are left and the sphere detection is finished. The decoding hypothesis $\hat{\mathbf{d}}$ now holds the MLSE solution.

The SD algorithm can also be sketched as a tree search of depth N_d. The root is at level $k = N_d - 1$, and the leaves representing a full decoding hypothesis are on level $k = 0$. An example for the processing of such a sphere decoding tree is shown in Figure 4.28. This very useful representation is enabled by the upper triangular structure of $\mathbf{R}$ and inherently visualizes the separable decision for individual data symbols.

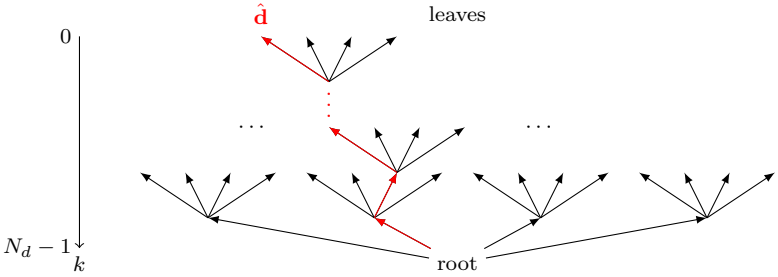


Figure 4.28.: Schematic SD tree for a 4-ary constellation and a depth of N_d.

4.4.2. Determination of Soft Information in Sphere Decoding

Analog to the MLSE as shown in Section 4.3, no remaining error statistics are known after sphere decoding. The soft information for the channel decoder needs to be determined in a similar way, using LLRs [OSHH12]. For the determination of the soft information for SD, (4.85) is rewritten in order to fit the derivations from before as

$$L_l = \ln \sum_{\mathbf{b} \in \mathcal{B}_l^{(1)}} \exp\left\{ -\frac{\|\mathbf{R}\mathbf{d}(\mathbf{b}) - \mathbf{y}'\|_2^2}{N\sigma_n^2} \right\} - \ln \sum_{\mathbf{b} \in \mathcal{B}_l^{(0)}} \exp\left\{ -\frac{\|\mathbf{R}\mathbf{d}(\mathbf{b}) - \mathbf{y}'\|_2^2}{N\sigma_n^2} \right\}, \quad (4.96)$$

with the same index sets of the sum as in (4.82).

An exemplary sum of two sum elements e^{a_1} and e^{a_2} in the log-domain can be expressed as [Hub02]

$$\ln\left(\mathrm{e}^{a_1}+\mathrm{e}^{a_2}\right)=\max\left\{a_1,a_2\right\}+f_c\left(|a_1-a_2|\right), \tag{4.97}$$

using a correctional function

$$f_c(x)=\ln\left(1+\mathrm{e}^{-x}\right) \tag{4.98}$$

that is plotted in Figure 4.29. The max-log approximation [HtB03, RVH95] states

$$\ln\left(\mathrm{e}^{a_1}+\mathrm{e}^{a_2}\right)\approx\max\left\{a_1,a_2\right\}. \tag{4.99}$$

This approximation is quite accurate, if the correctional function $f_c(x)$ is small, which is the case for a big difference $|a_1-a_2|$. Extending the max-log approximation to the whole sums in (4.96), the correctional function part is small if there is one particular big exponent and otherwise small sum values, which happens usually in safe decisions when only little noise is present. Thus, the max-log approximation is very accurate in a high-SNR regime and expected to become more and more inaccurate with decreasing E_b/N_0-ratio.

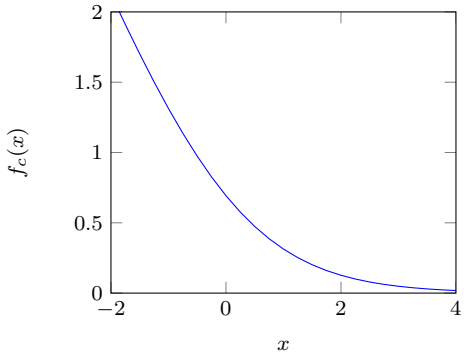


Figure 4.29.: Correctional function $f_c(x)$ as used in (4.97).

Therewith, the calculation is greatly simplified, as most of the addends of the sums in (4.96) can be omitted and only the largest one, coinciding with the minimum distance vector, is considered. Additionally, the noise variance term $N\sigma_n^2$ can be factored out for each LLR and even omitted for the channel decoding, as it is constant for the whole OFDM symbol. After turning the maximization to a minimization due to the minus in the exponent, the LLR determination problem (4.96) becomes [SBB08]

$$L_l=\min_{\mathbf{b}\in\mathcal{B}_l^{(1)}}\left\|\mathbf{R}\mathbf{d}(\mathbf{b})-\mathbf{y}'\right\|_2^2-\min_{\mathbf{b}\in\mathcal{B}_l^{(0)}}\left\|\mathbf{R}\mathbf{d}(\mathbf{b})-\mathbf{y}'\right\|_2^2. \tag{4.100}$$

For each symbol, one of the two minima is given by λ^{ML}. The other one is the best solution, where the current symbol b_l, whose reliability is investigated, is flipped, such that $b_l = \overline{\hat{b}_l}$. This results to the metric

$$\lambda_l^{\overline{\mathrm{ML}}} = \min_{\mathbf{b} \in \mathcal{B}_l^{\left(\overline{\hat{b}_l}\right)}} \left\| \mathbf{R}\mathbf{d}(\mathbf{b}) - \mathbf{y}' \right\|_2^2, \tag{4.101}$$

and finally the LLRs

$$L_l = \begin{cases} \lambda^{\mathrm{ML}} - \lambda_l^{\overline{\mathrm{ML}}} & \text{if } \hat{b}_l = 0, \\ \lambda_l^{\overline{\mathrm{ML}}} - \lambda^{\mathrm{ML}} & \text{if } \hat{b}_l = 1, \end{cases} \tag{4.102}$$

with the metric of the ML solution $\lambda^{\mathrm{ML}} = \left\| \mathbf{R}\hat{\mathbf{d}} - \mathbf{y}' \right\|_2^2$ and the metric of the counter-hypothesis

$$\lambda_l^{\overline{\mathrm{ML}}} = \min_{\mathbf{d} \in \mathcal{A}_l^{\left(\overline{\hat{b}_l}\right)}} \left\| \mathbf{R}\mathbf{d} - \mathbf{y}' \right\|_2^2. \tag{4.103}$$

The metric of the counter-hypothesis is specific to each binary symbol and determined by fixing the l-th symbol of the data vector $\mathbf{d}$ under test to the complement of the MLSE solution $\hat{\mathbf{b}}$. The flipped symbol number l is denoted as $\overline{\hat{b}_l}$ and $\mathcal{A}_l^{(\beta)}$ is the set of all valid data vectors $\mathbf{d}$ with the l-th symbol equal to β.

Now, simply these two values need to be found, which is the main problem of the soft-output sphere decoder. Besides finding the ML solution and metric, calculating the LLRs resorts to finding the $N_d n_b$ "next-best" minima of (4.103) for each symbol [SF10]. This interpretation can be sketched by a decoding tree of depth 3 for a 4-ary alphabet in Figure 4.30, where the ML solution is indicated by the red path, delivering the ML metric λ^{ML}. In order to find the metrics of the counter-hypotheses, the respective binary symbol of the ML solution needs to be flipped and fixed, what corresponds to pruning the SD tree by the appropriate branches and reducing the number of leaves by one half. For example, to determine the counter-hypothesis $\lambda_0^{\overline{\mathrm{ML}}}$ with respect to the first symbol b_0 (1 in the given example), the best solution with the first symbol fixed to $b_0 = 0$ is sought and indicated in green. Fixing the first symbol to 0 conforms with fixing the MSB and removing the right half of the SD tree from the search process. Fixing symbol b_1 to the opposite of the ML solution (fixing to 1), conforms with removing the first and the third branch from the root of the SD tree, as it represents the LSB.

One could solve these minimization problems subsequently by re-running the SD for each counter-hypothesis with correspondingly restricted search space, that is performing a so-called repeated tree search. This however requires to run the SD $N_d n_b + 1$ times per OFDM symbol, and hence, imposes a high complexity burden.

To alleviate this, the initial Algorithm 10 published in [AEVZ02, OH11b] is extended in order to provide soft information according to the *single tree search* principle shown in [JO05, SBB08], which ensures that one leaf of the SD tree is visited not more than once and branches are only followed if there is the chance to update either $\hat{\mathbf{d}}$ (and thus λ^{ML}) or one of the $N_d n_b$ counter-hypotheses' metrics $\lambda_l^{\overline{\mathrm{ML}}}$. This single tree search was implemented, while introducing the restriction to a finite, possibly complex valued transmit symbol alphabet $\mathcal{A}$, instead of the ability to operate on an infinite real lattice only.

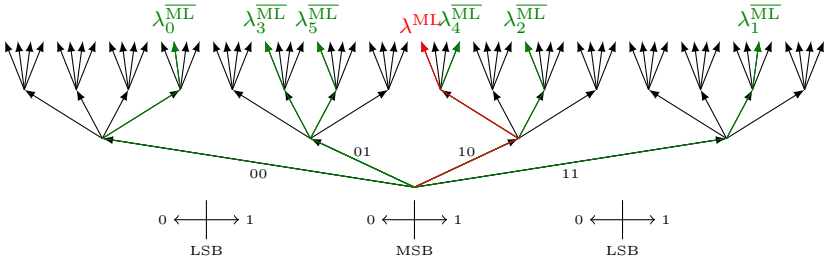


Figure 4.30.: Schematic SD tree for a 4-ary constellation and a depth of 3.

The implementation as published in [OSHH12] is outlined in Algorithm 11, following the pseudo-code notation in [AEVZ02, SF10]. Only the additions and changes compared to Algorithm 10 are highlighted, while accordant operations are displayed in faded gray.

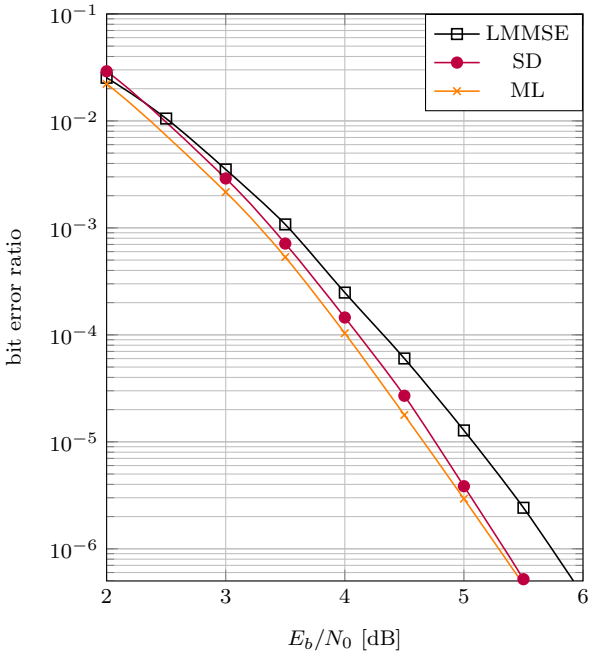
The functions INITLIST and GETNEXTSYMBOL remain the same as in the normal SD Algorithm 10.

LLR clipping during the SD search is an effective means to speed up the search process and hence reduce its computational complexity [SBB08]. This is achieved by limiting the SD search radius ρ (line 13) to include only counter-hypotheses within the LLR clipping level $L_{\max}$. LLR clipping is implemented in line 27. Of course, it affects the bit error performance negatively. However, LLR clipping enables an efficient trade-off between performance and complexity. In particular, the traditional hard-output SD algorithm can be modeled by setting $L_{\max} = 0$.

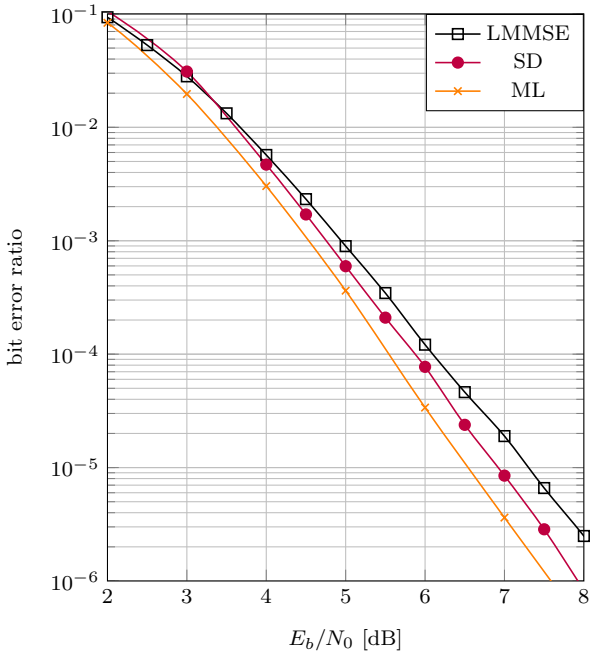
Of course, the use of the max-log approximation induces a loss in accuracy. The soft information found by the SOSD according to Algorithm 11 is less accurate than the one found by the soft-output maximum-likelihood decoder in Algorithm 9, which is exact and thus optimum. For a very small system this loss due to the approximation can be numbered. For the exemplary UW-OFDM system ML, according to Table A.1, with 4-QAM, bit error results are shown in Figure 4.31 for the case of an AWGN channel and in multipath environment. Non-systematically generated UW-OFDM is omitted in this plot, as it shows the very same max-log degradation as for systematic UW-OFDM. A comprehensive performance comparison is given in Section 4.4.3.

Algorithm 11 Soft-Output Sphere Decoder

function $[\hat{\mathbf{b}}, \mathbf{L}] =$ SOFT-OUTPUT SD$(\mathbf{y}', \mathbf{F}, \mathcal{A})$
  $\Delta_{N_d} \leftarrow 0;\ \lambda^{\mathrm{ML}} \leftarrow \infty;\ \boldsymbol{\lambda}^{\overline{\mathrm{ML}}} \leftarrow \boldsymbol{\infty}$
  $k \leftarrow N_d - 1$
  $\tilde{\mathbf{y}}^{(k)} \leftarrow \mathbf{F}\mathbf{y}'$
  $\mathbf{A}_k \leftarrow$ INITLIST$(k, \tilde{y}_k^{(k)}, [\mathbf{F}]_{k,k}, \mathcal{A})$
  while $k < N_d$ **do**
    $[d_k, \delta, \mathbf{A}_k] \leftarrow$ GETNEXTSYMBOL$(\mathbf{A}_k)$
    update binary symbol representation **b** of **d**
    if isempty(d_k) **then** ▷ all symbols from $\mathcal{A}$ tried
      $k \leftarrow k + 1$
    else
      $\Delta_k \leftarrow \Delta_{k+1} + |\delta|^2$
      $\rho \leftarrow \max \lambda_l^{\overline{\mathrm{ML}}}\Big|_{l=0,\ldots,(k+1)n_b-1\ \vee\ b_l \neq \hat{b}_l}$ ▷ tree pruning
      if $\Delta_k < \rho$ **then**
        if $k > 0$ **then**
          $\tilde{y}_l^{(k-1)} \leftarrow \tilde{y}_l^{(k)} - \delta\,[\mathbf{F}]_{l,k}, \quad l = 0, \ldots, k-1$
          $k \leftarrow k - 1$
          $\mathbf{A}_k \leftarrow$ INITLIST$(k, \tilde{y}_k^{(k)}, [\mathbf{F}]_{k,k}, \mathcal{A})$
        else
          if $\Delta_k < \lambda^{\mathrm{ML}}$ **then**
            $\lambda_l^{\overline{\mathrm{ML}}} \leftarrow \lambda^{\mathrm{ML}}, \quad \forall l$ with $b_l \neq \hat{b}_l$
            $\hat{\mathbf{b}} \leftarrow \mathbf{b}$
            $\lambda^{\mathrm{ML}} \leftarrow \Delta_k$
          else
            $\lambda_l^{\overline{\mathrm{ML}}} \leftarrow \min\left\{\lambda_l^{\overline{\mathrm{ML}}}, \Delta_k\right\}, \quad \forall l$ with $b_l \neq \hat{b}_l$
          end if
          $\lambda_l^{\overline{\mathrm{ML}}} \leftarrow \min\left\{\lambda_l^{\overline{\mathrm{ML}}}, \lambda^{\mathrm{ML}} + L_{\max}\right\}$ ▷ LLR clipping
        end if
      else
        $k \leftarrow k + 1$
      end if
    end if
  end while
  $L_l \leftarrow (\lambda^{\mathrm{ML}} - \lambda_l^{\overline{\mathrm{ML}}}) \cdot (2\hat{b}_l - 1), \quad \forall l = 0, \ldots, N_d n_b - 1$
end function



(a) BER results in the AWGN channel.



(b) BER results in multipath environment.

Figure 4.31.: Performance loss in coded transmission due to the max-log approximation for soft-output SD in system ML according to Table A.1 with 4-QAM.

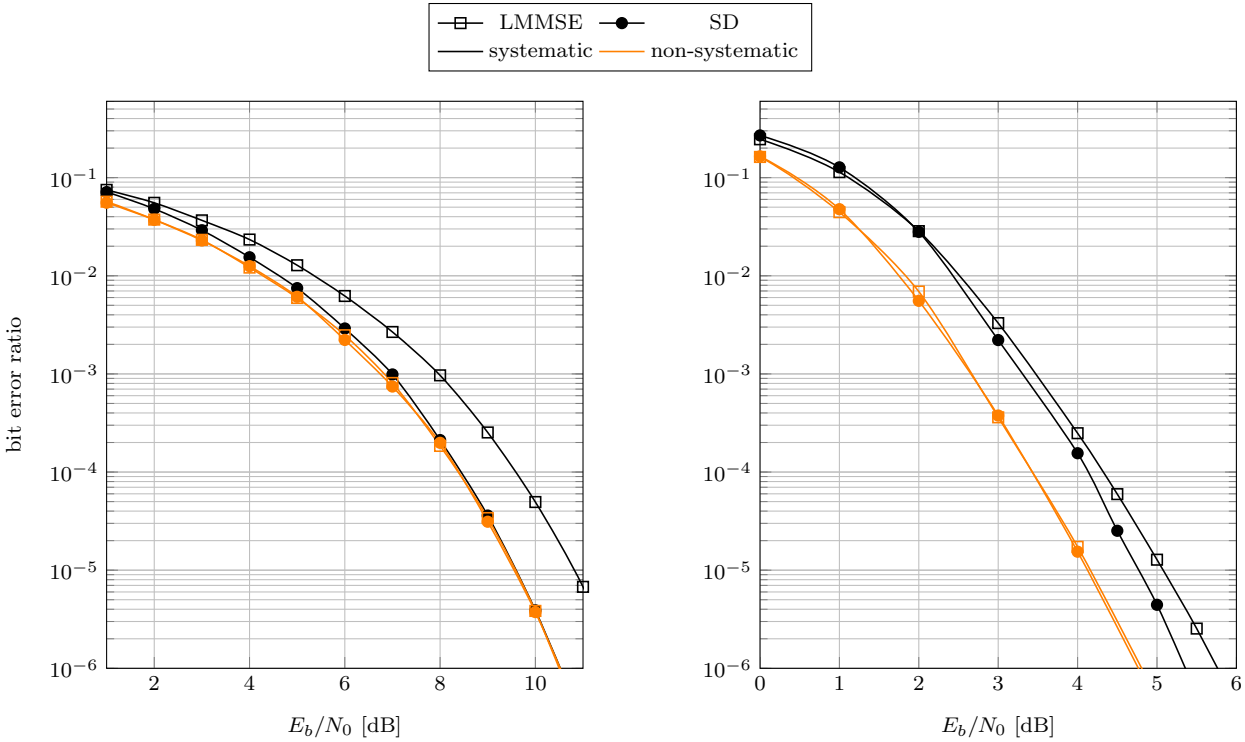
Comparing the curves for the soft-output ML to the soft-output SD for coded transmission with rate 1/2 at a BER of 10^{-5} reveals that the disadvantage due to the approximation used for the SD amounts to less than 0.1 dB in the AWGN channel. For reference, the LMMSE curve for the same system parameters are almost 0.4 dB behind the ML. The tendencies are the same, but the distances larger for the multipath environment: The SD and LMMSE are 0.45 dB respectively 0.75dB behind the soft-output ML detection, posing a considerable degradation. Nevertheless, this small gap between coded SD and ML detection can hardly be closed. Because of the high dimensionality and the vast amount of terms to consider in the calculation in (4.85) for practically sized UW-OFDM systems, the soft-output sphere decoder will constitute the practical implementation to achieve near-ML performance.

4.4.3. Performance Evaluation

In order to evaluate especially the soft-output SD results in a reasonable amount of time, the OFDM system had to be adapted, as the exemplary systems A and B took too long to simulate. Therefore the exemplary system SD (see Appendix A) was introduced, featuring only $N_d = 16$ data and $N_r = 8$ redundant subcarriers. The length of the channel impulse response is set to 9 in order to have the maximum CIR length without having to endure ISI. All channel realizations feature (on average) a root mean square delay spread of 50 ns.

Figure 4.32 depicts the bit error ratio of a transmission over the AWGN channel for uncoded as well as coded transmission. In spite of the different parameter choices compared to [HHOH12], the uncoded results show the very same tendencies. For systematically coded UW-OFDM the SD achieves a gain of about 1.2 dB uncoded and 0.4 dB in coded transmission. The non-systematically coded systems however clearly outperform all systematic UW-OFDM systems. Interestingly, in the AWGN channel both receiver types achieve the same results and the simple LMMSE estimator already represents the optimum data estimator for non-systematically generated UW-OFDM. Quite outstandingly, the uncoded systematic SD achieves the same results asymptotically at high E_b/N_0-ratios. For coded transmission, systematic generation of the UW-OFDM symbol induces a loss of approximately 0.5 dB over non-systematic generation for the SD. It needs to be noted that due to the max-log approximation in the determination of the soft information the soft-output SD is not optimal, but still performs better at higher E_b/N_0-ratios than the LMMSE, which uses exact statistics of the residual noise as soft information. Only at low E_b/N_0 for coded transmission, the approximation is not good enough, and the LMMSE slightly outperforms the SD.

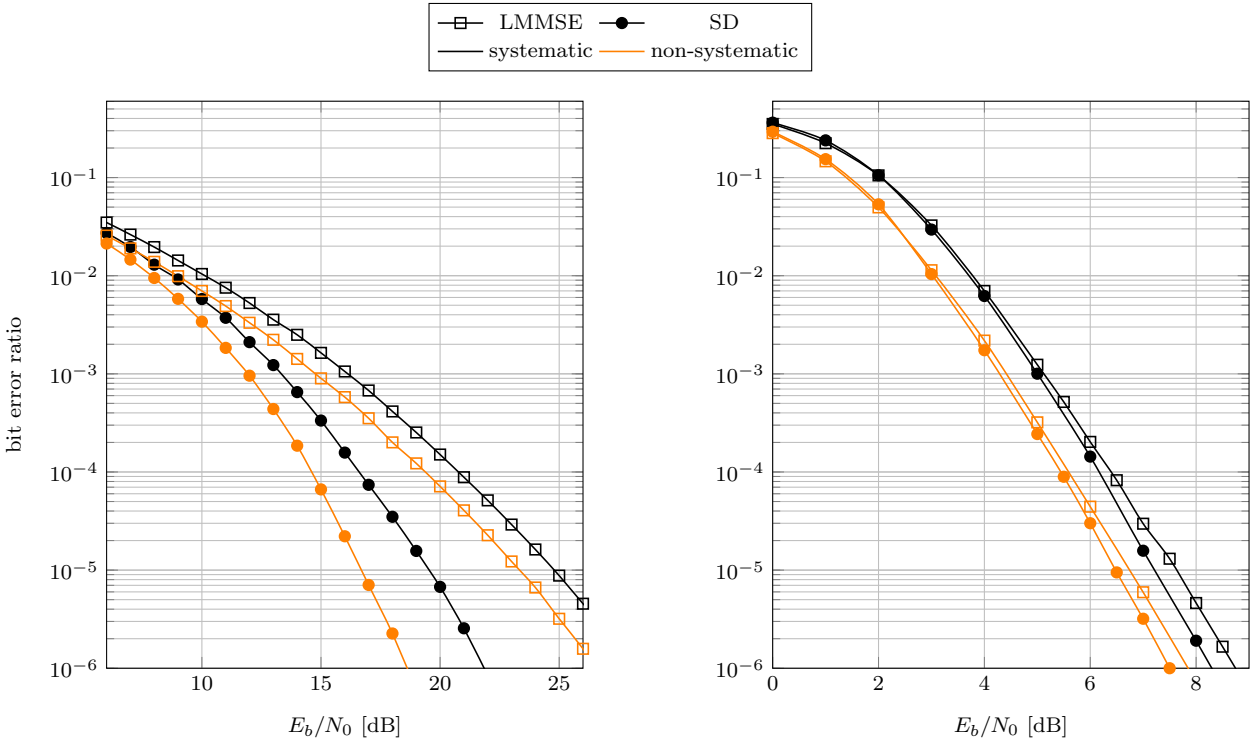
The more relevant results for OFDM transmission come from simulations over multipath channels, that are shown in Figure 4.33. The uncoded SD shows a significant advantage compared to the uncoded LMMSE with increasing tendency. At a BER of 10^{-5} the SD's gain has grown already to 5.4 / 6.8 dB using systematic / non-systematic UW-OFDM generation. This visualizes the enormous gap between the best linear estimator and the best result possible. Additionally, the non-systematic SD has a 2.8 dB advantage over the systematic SD. As expected, the gain for the coded transmission is much smaller but the soft-output SD still considerably outperforms the LMMSE by about 0.4 and 0.3 dB for systematically and non-systematically generated UW-OFDM respectively. Here, the non-systematic versions are by around 0.8 dB ahead of their systematic counterparts.



(a) uncoded results, AWGN

(b) coded results, AWGN

Figure 4.32.: BER performance of the SD for system SD according to Table A.1 in the AWGN channel.



(a) uncoded results, multipath

(b) coded results, multipath

Figure 4.33.: BER performance of the SD for system SD according to Table A.1 in multipath environment.

4.4.4. Complexity Evaluation

A comprehensive comparison of the presented detection schemes has to take the required computational complexity into account. Similar to [HOH11], complex multiplication equivalents (CMEs) are adopted as a measure for computational complexity. While the complexity of the regular SD only becomes an issue on low E_b/N_0-ratios, its limitation is covered in [OH11a] and is excluded from this work in order to focus on the soft-output sphere decoder.

In case of soft-output SD, the average number of CMEs per OFDM symbol depends on the actual channel conditions and especially on the signal-to-noise ratio. It can be stated that the complexities of systematic and non-systematic UW-OFDM symbol generation only differ slightly in the count of the CMEs, which is why only the systematic results are shown. The LMMSE data estimation itself is performed by a matrix multiplication, thus requiring $N_d^2 + N_d N_r$ CMEs per OFDM symbol (compare with Section 3.3).

Note that both detectors require a pre-processing step per channel estimation update, which is the computation of (3.29) for the LMMSE and the QR decomposition for the soft-output SD. In this work, however, only the CMEs required for data detection per bit of information (after channel decoding) are considered. A more detailed comparison of the pre-processing steps of both detectors reveals that both have comparable computational complexity. The number of CMEs for the matrix inversion for the LMMSE (see (3.61)) and the post-sorting QR decomposition required for the soft-output SD [WBKK03] is cubic in the number of subcarriers and will not be counted in this consideration.

In Algorithm 11, CMEs only occur upon execution of the function INITLIST in lines 5 and 18, and upon the 'decision feedback' in line 16. Figure 4.34 depicts the required CMEs per bit of information for the multipath setting and exemplary UW-OFDM system SD with 4-QAM transmit symbol constellation (see Appendix A). For the unrestricted soft-output SD without LLR clipping ($L_{\max} = \infty$), the plot shows the expected decreasing CME behavior, as early found hypotheses become more likely to be the most probable ones for increasing E_b/N_0 values. The complexity of the hard-output SD (Algorithm 10), labeled just SD, is included as a lower bound. Additionally, the CME count for the LMMSE data estimation is included. As stated earlier this is a simple matrix multiplication using $(N_d^2 + N_d N_r)/(N_d n_b R) = 24$ CMEs per bit of information for this setup[9], with the number of binary symbols per data symbol for 4-QAM being $n_b = 2$ and the used coding rate $R = 1/2$.

Using LLR clipping with clipping levels $L_{\max} = 3$ and 5, the complexity of the soft-output SD is reduced significantly. For $L_{\max} = 5$, the effect of LLR clipping on the bit error performance is almost negligible, as can be deduced from Figure 4.35. Only for $L_{\max} = 3$ the BER is impaired significantly.

Counting the CMEs as a complexity benchmark is usually a good measure for a comparison in highly integrated circuits with good optimization and parallelization. Anyhow, the huge structural overhead which appears in Algorithm 11 is not taken into account. In order to have a performance comparison on the other end of the implementation spectrum, the execution time of the detection methods on an office PC was measured. The PC,

[9] Opposed to Table 3.2, the effort of the DFT is not considered here.

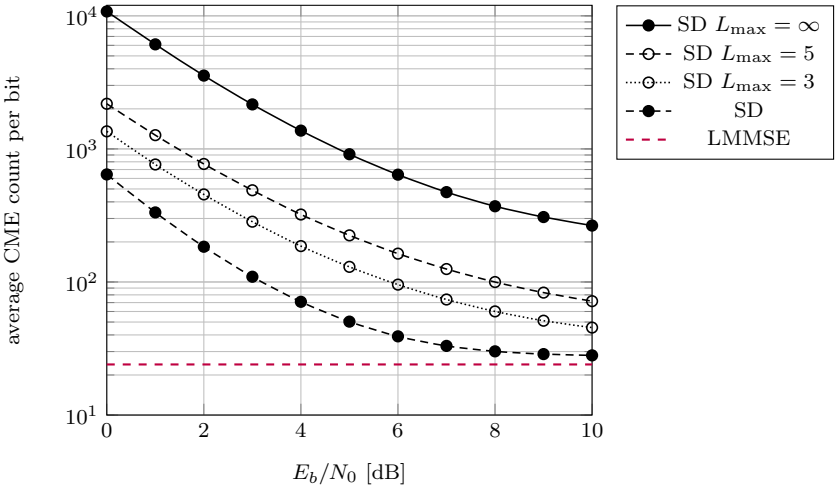


Figure 4.34.: Complexity (in CME per bit of information) for soft-output detection of UW-OFDM using the soft-output SD and the LMMSE estimator.

containing a Pentium D CPU with 2.8 GHz, was set up with Linux Mint 12 64-bit and a 2011 version of Mathworks Matlab as simulation software. The CPU was constrained in order to use one core only. The LMMSE data estimator is implemented in Matlab only, while the soft-output SD as in Algorithm 11 is implemented in C and connected to Matlab as MEX [Web12]. The preparatory steps for the soft-output SD (the QR decomposition) are computed by Matlab as well. This comparison seems unfair at first sight, as the Matlab execution is supposed to be much slower than the C implementation, but can be reasoned. It is worked on an array of many OFDM symbols for each channel at once. The LMMSE data estimation itself is one simple multiplication of the receive vector with the LMMSE matrix (3.29) for each OFDM symbol. By putting many receive vectors for one channel realization in the columns of a large receive matrix, this results in a single matrix-matrix multiplication for all OFDM symbols, whose internal Matlab implementation is very efficient using BLAS libraries and able to keep up with the soft-output SD in C.

The results of the run-time comparison are shown in Figure 4.36, which shows the time in seconds to detect one bit of information on average, in dependence on the E_b/N_0-ratio. The very same trends as before in Figure 4.34 can be observed, when the CMEs were counted only. This suggests that the execution time is approximately linear in the number of CMEs.

In comparison, it can be concluded that the LMMSE estimator represents a very competitive low-complexity variant for soft-output detection of UW-OFDM compared to the close-to-optimum sphere decoder with soft-output. Nevertheless, the SD represents the best possible receiver for uncoded transmission and performs close to the theoretical optimum MLSE, when soft information is desired, where it suffers from a very high complexity

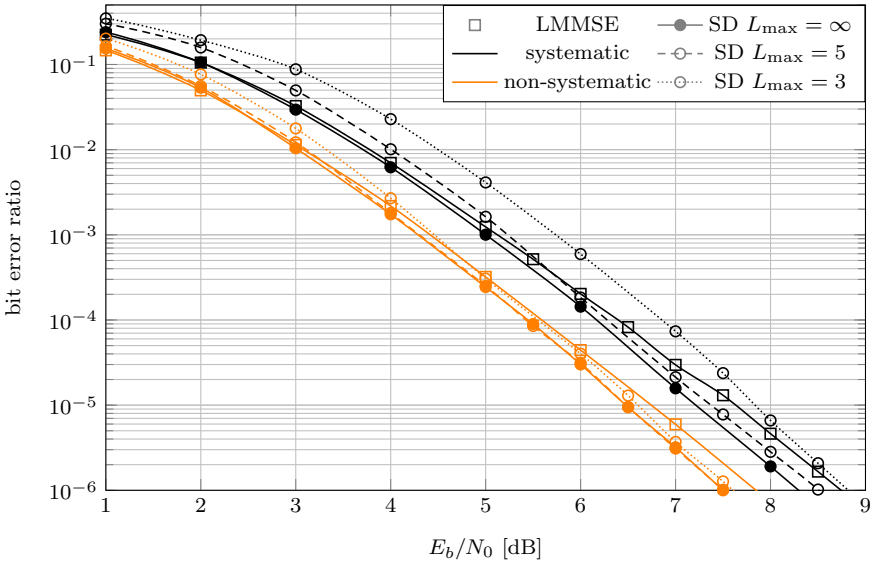


Figure 4.35.: BER results of coded UW-OFDM using the soft-output SD in multipath environment for different LLR clipping levels $L_{\max}$, in exemplary system SD according to Table A.1.

burden. However, with appropriate setting of the LLR clipping, a reasonable trade-off between complexity increase and performance gain over the LMMSE can be found. Still, it has to be noted that especially in case of larger number of subcarriers, as well as larger QAM alphabets compared to 4-QAM, the complexity of the SD may be prohibitively large for hardware implementation. Furthermore, this comparison does not account for computational processing steps like list sorting and memory access, the latter occurring in vast amounts for the SD algorithm.

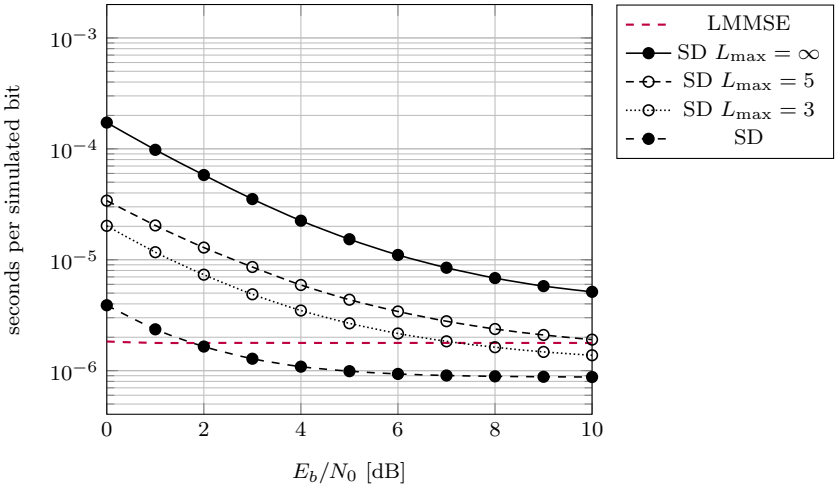


Figure 4.36.: Complexity (in average run-time per bit of information to simulate on the reference PC) of soft-output detection of UW-OFDM using the soft-output SD and the LMMSE estimator.

4.5. Performance Comparison

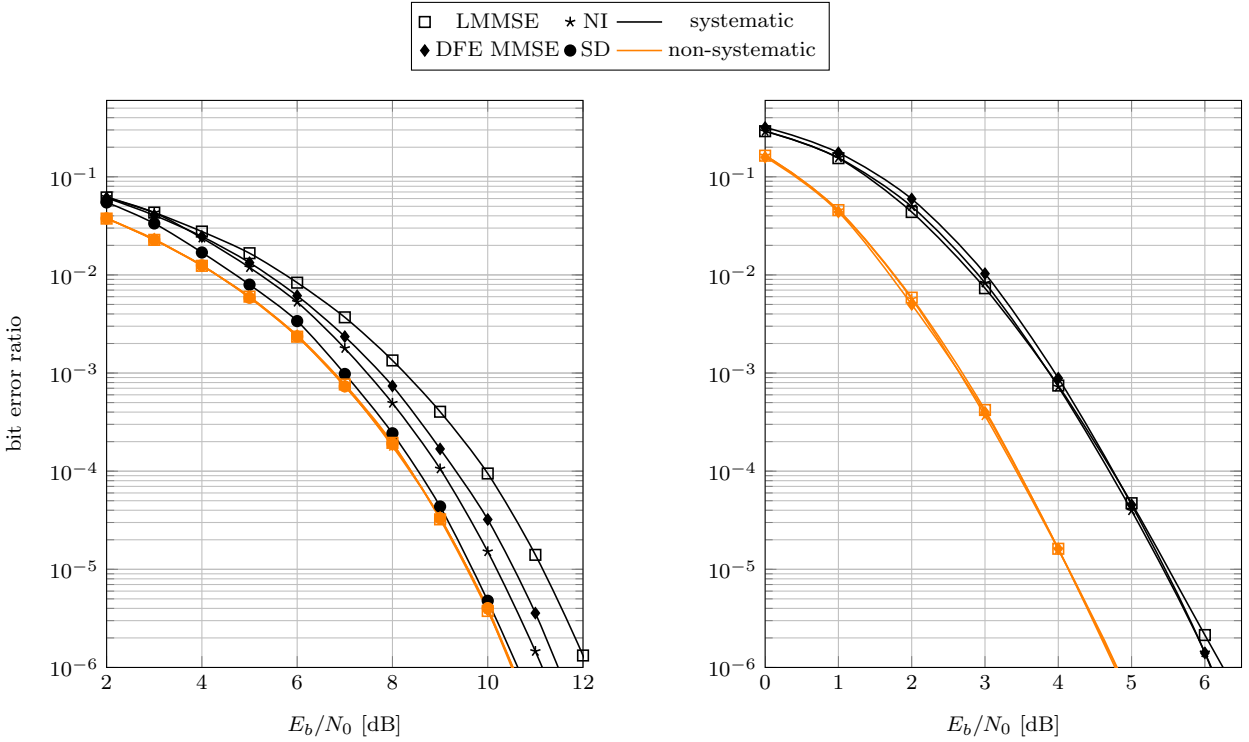
In this section, simulation results that give a reference overview over all nonlinear data estimation methods introduced in this chapter are shown. As a basis, results for the best linear data estimator from Chapter 3 are included, which is the LMMSE data estimator (see Section 3.2). The exemplary UW-OFDM system setups used for the simulations are described in Appendix A.

For the exemplary UW-OFDM system A the results in the AWGN channel are shown in Figure 4.37 and in multipath environment in Figure 4.38. The same simulations have been done for system B and the results are shown in Figure 4.39 and 4.40, respectively. Note that for these system setups, simulation results for the coded soft-output sphere decoder are not available, due to its computational complexity for systems of that size. In the AWGN channel both systems perform almost equivalently for coded transmission (shown in the respective sub-figures (b)) with a small advantage for system B, which is due to its lower average OFDM symbol energy. The lines of the non-systematically generated UW-OFDM systems even match exactly in both systems. In uncoded transmission (sub-figures (a)) system B supersedes system A just by a little bit, and in multipath environment still by not more than 1 dB. For this reason, only comments are made on the results for system B, which mostly apply to system A as well.

In the AWGN channel results, the effect that all receivers in a non-systematic UW-OFDM system perform equivalently with a gain of about 1.7 dB uncoded / 1.5 dB coded over the LMMSE estimator in a systematic UW-OFDM system setup, is repeated. Thus, the LMMSE is already the best receiver for non-systematically generated UW-OFDM, and even sophisticated nonlinear receivers cannot improve the decoding result. The SD for systematic UW-OFDM is able to achieve the performance of non-systematic UW-OFDM asymptotically with increasing SNR, as shown in [HHOH12]. The other receivers for systematic UW-OFDM show different performance in uncoded transmission: At a BER of 10^{-5}, the LMMSE is exceeded by the MMSE DFE by 0.6 dB. The NI even achieves 1 dB gain over the LMMSE or 0.4 dB over the DFE, only superseded by the SD achieving the 1.7 dB gain, as for the non-systematic setup. For coded transmission the systematically generated systems operate far behind the non-systematic UW-OFDM. The LMMSE receiver with its most accurate error statistics achieves the best results for most E_b/N_0-ratios. Only at high SNR the MMSE DFE and the NI receivers outperform the LMMSE, which is barely visible in Figure 4.39b.

In the multipath environment, especially for uncoded transmission, the advantage of the nonlinear receivers increases a lot. Most receivers for systematic UW-OFDM outperform even the LMMSE for the non-systematic system above a medium E_b/N_0-ratio. In the systematic / non-systematic groups, a clear order of good performing UW-OFDM receivers can be identified: the SD, the MMSE DFE receivers, the NI, and the LMMSE estimator. At the 10^{-5} BER point in uncoded transmission, this order is met, all separated by between 1.3 and 2 dB. Only the LMMSE for non-systematic UW-OFDM performs with 4.5 dB far behind the next non-systematic candidate, which is the NI. In coded transmission, the receivers for non-systematic UW-OFDM perform about 1.2 dB better than their systematic counterpart. However, the NI shows a better performance than the MMSE DFE here, although the differences are only minor between 0.03 and 0.15 dB.

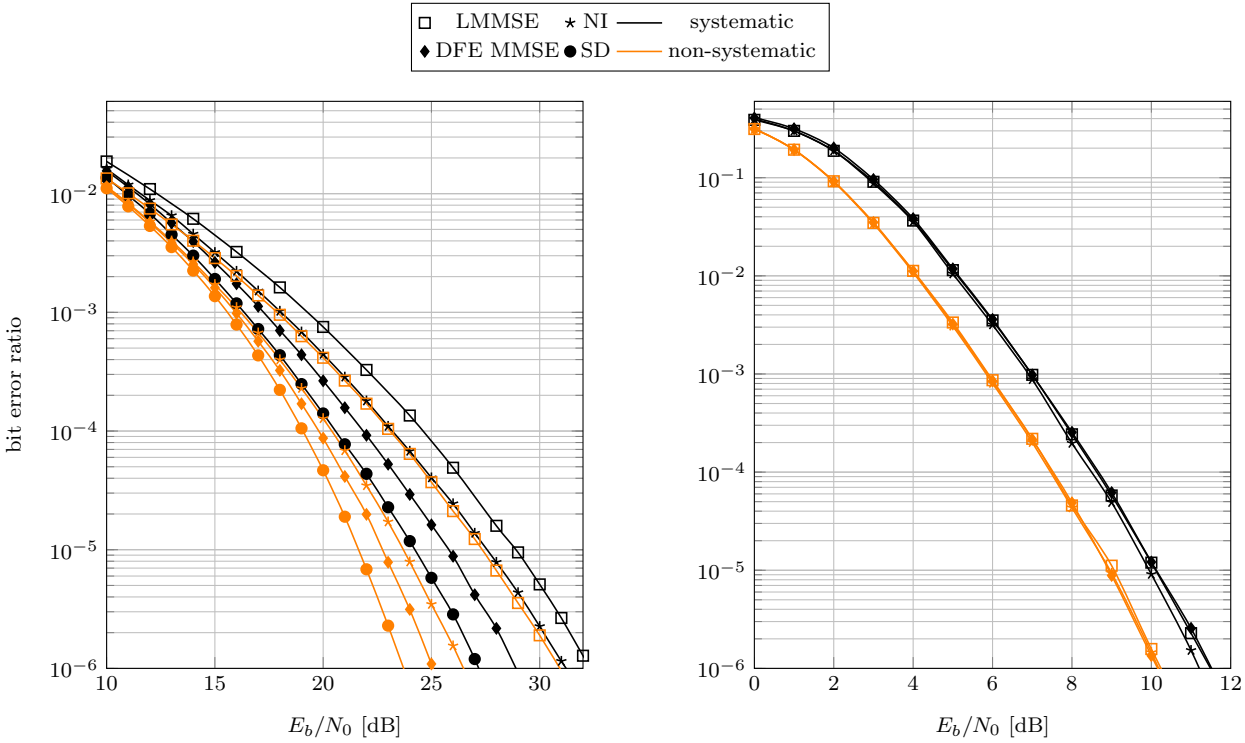
In order to be able to evaluate the coded performance of the SD with soft-output, the AWGN and multipath simulation results for the system SD are included in Figure 4.41. The qualitative characteristics as seen for system B remain the same here. The added curve for the soft-output SD clearly marks the reference performance for UW-OFDM receivers designed for systematically generated UW-OFDM. While the DFE and NI receivers barely supersede the LMMSE performance in coded transmission and show a really distinctive gain only on high SNR, the SD outperforms the LMMSE estimator by 0.31/0.41 dB in AWGN/multipath environment for systematic UW-OFDM. As identified before, for the non-systematic UW-OFDM system, all receivers perform equivalently in AWGN and show a fine forking in multipath environment with the SD in lead, superseding the LMMSE estimator by 0.26 dB.



(a) UW-OFDM system A, uncoded, AWGN

(b) UW-OFDM system A, coded, AWGN

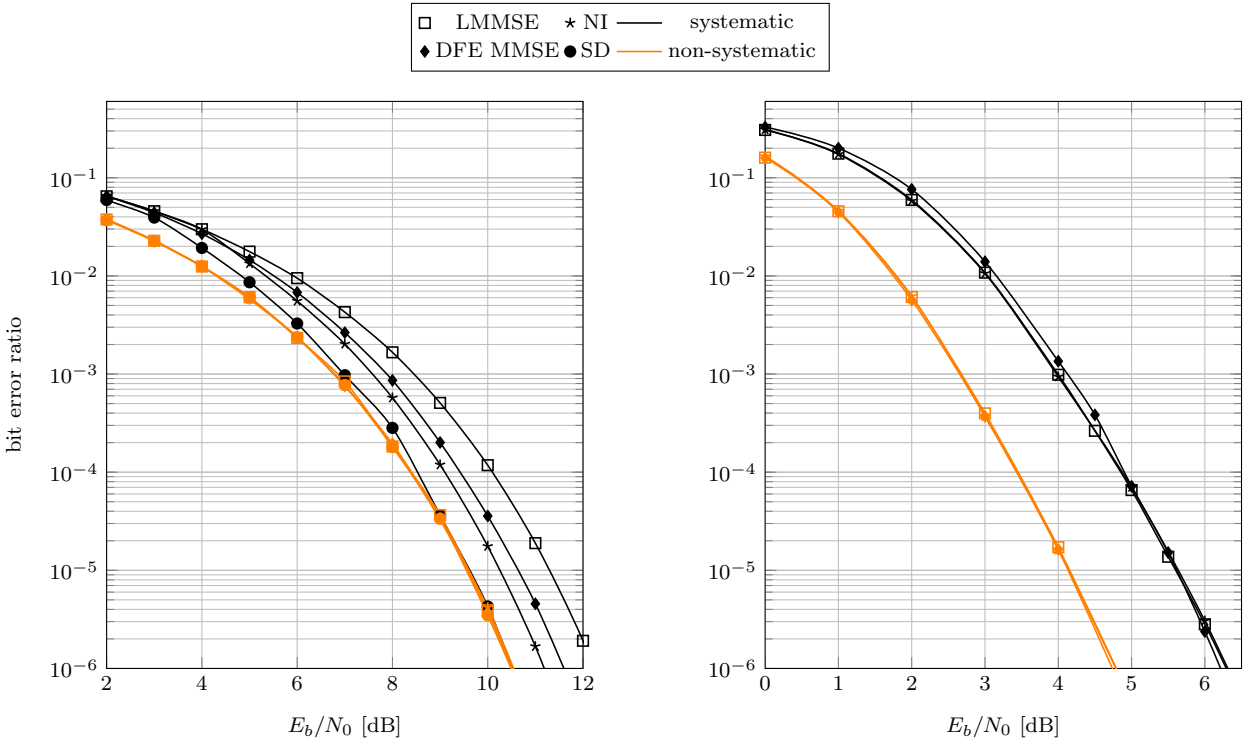
Figure 4.37.: BER performance for system A according to Table A.1 in the AWGN channel.



(a) UW-OFDM system A, uncoded, multipath

(b) UW-OFDM system A, coded, multipath

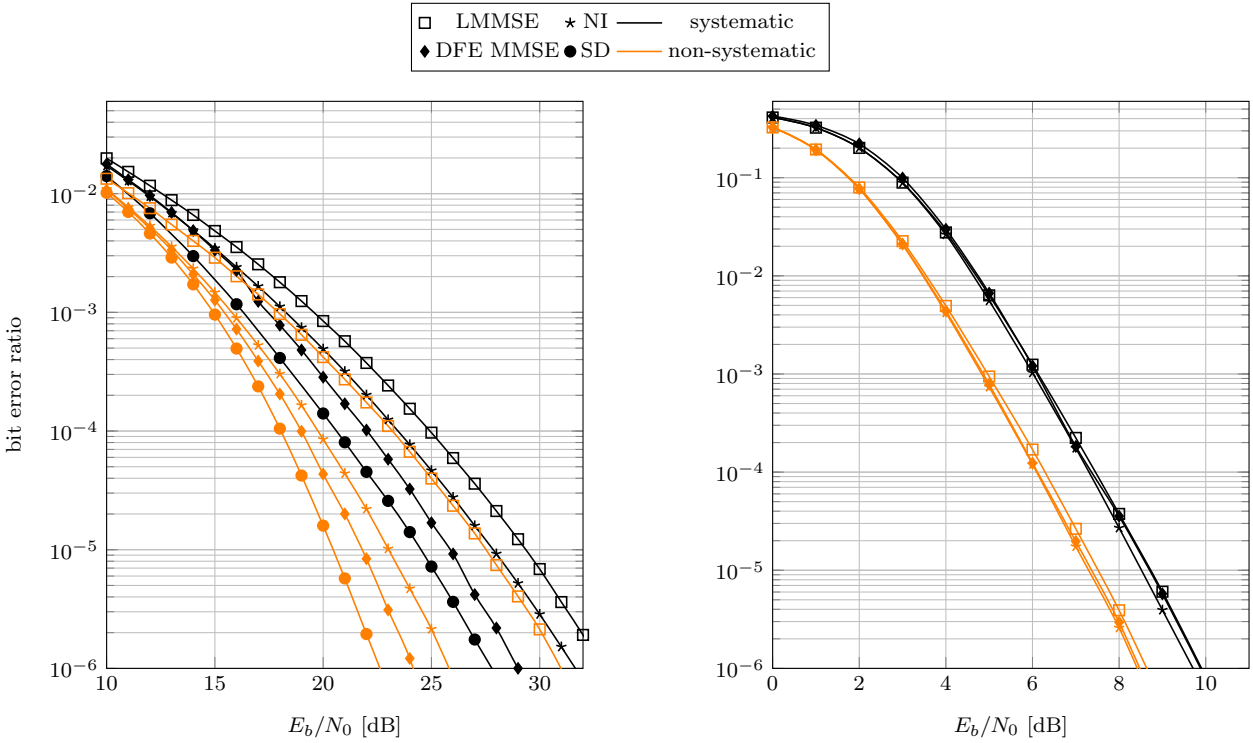
Figure 4.38.: BER performance for system A according to Table A.1 in multipath environment.



(a) UW-OFDM system B, uncoded, AWGN

(b) UW-OFDM system B, coded, AWGN

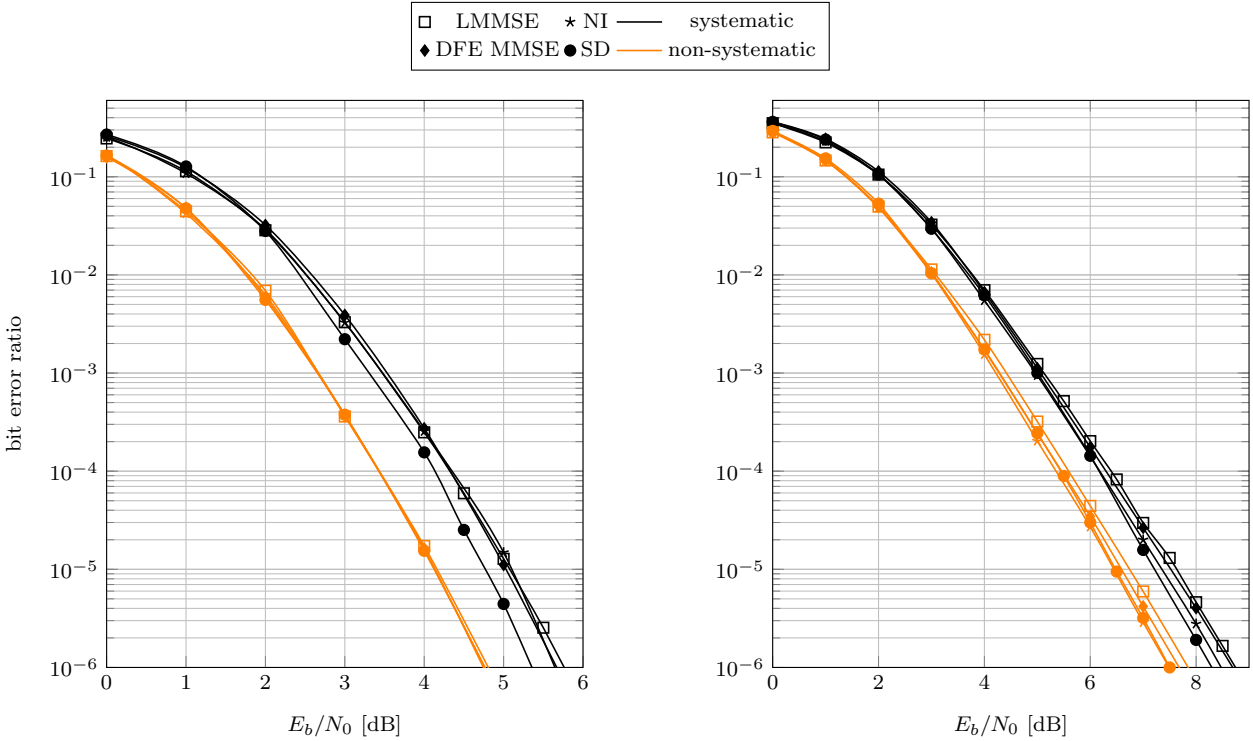
Figure 4.39.: BER performance for system B according to Table A.1 in the AWGN channel.



(a) UW-OFDM system B, uncoded, multipath

(b) UW-OFDM system B, coded, multipath

Figure 4.40.: BER performance for system B according to Table A.1 in multipath environment.



(a) UW-OFDM system SD, coded, AWGN

(b) UW-OFDM system SD, coded, multipath

Figure 4.41.: BER performance for coded system SD according to Table A.1 in the AWGN channel and in multipath environment.

4.6. Summary of Nonlinear Receivers

In this chapter a few nonlinear receivers for UW-OFDM have been discussed and their performance investigated. From Chapter 3 the LMMSE receiver emerged as a quite simple and very well performing receiver in uncoded and coded transmission. In order to compete well against the LMMSE receiver in coded transmission, it is necessary to retrieve reliable soft information for the Viterbi channel decoder, which was investigated for all nonlinear UW-OFDM receivers.

After revising the Wiener filter theory, a vast number of possibilities of how to operate the *noise interpolation* technique is available: Besides several criteria of which noise values to interpolate and which samples to pick to be used for interpolation, noise interpolation can be performed in a batch or an iterative approach. Also the statistics, necessary to compute the Wiener filters, can be derived in various ways. In a rather empirical way, a small set of well performing operation modes was identified that are all able to outperform the LMMSE estimator.

Second, *decision feedback equalization* was revised and shown in the ZF and the MMSE version. This algorithm was reformulated, in order to offer different ways of interpretation.

The optimum decoding method, *maximum likelihood sequence estimation*, has been derived. Due to its computational complexity, it is impractical for simulations. As a much more practical realization producing the same results, the *sphere decoding* algorithm was shown. However, in order to determine reliability information for the detected data, it has to endure some loss over the MLSE solution, due to an approximation that is needed for a complexity reduction. By introducing LLR clipping levels, the still very high complexity can be reduced, at the cost of some detection performance. A complexity analysis and a performance comparison of the soft-output SD with the LMMSE estimator proves the relevance of the shown methods.

In a final evaluation, the performance of the investigated nonlinear receivers for UW-OFDM was shown in terms of the achieved bit error ratio. All receivers are able to outperform the LMMSE estimator, which is nevertheless a simple and very well performing receiver. Especially in coded transmission, it is barely beaten by the nonlinear receivers. But in uncoded transmission, in the AWGN channel or in a multipath environment, the nonlinear receivers clearly show their advantages.

5. Impact of different QR Decompositions on DFE and SD for UW-OFDM

As described earlier, the QR decomposition of the channel matrix $\mathbf{H}$ in (4.1) is enabling the highly efficient sphere decoder in the first place. It also plays an important role in decision feedback equalization, where it yields the variant in Algorithm 7 to perform the DFE. There are quite a few possibilities to compute such a QR decomposition, which has indeed tremendous impact on either the run-time of the SD or the performance of DFE. Hence, in this section the main features of a QR decomposition are shown as well as several methods for its computation. By means of numerical simulation, good QR decomposition methods for DFE and SD can be identified for UW-OFDM systems. A variety of methods that apply to MIMO channels have been analyzed and compared in the recent past, see for example [Fis10, Fis12, WRB$^+$02]. In this work, only a few are picked and taken into consideration. While a QR decomposition of the standard model

$$\mathbf{y} = \mathbf{H}\mathbf{d} + \mathbf{n}$$

yields the ZF DFE, the same can be done with the extended channel matrix and extended vectors

$$\underbrace{\begin{bmatrix}\mathbf{y}\\ \mathbf{0}\end{bmatrix}}_{\breve{\mathbf{y}}} = \underbrace{\begin{bmatrix}\mathbf{H}\\ \mathbf{0}\end{bmatrix}}_{\breve{\mathbf{H}}}\mathbf{d} + \underbrace{\begin{bmatrix}\mathbf{n}\\ \mathbf{0}\end{bmatrix}}_{\breve{\mathbf{n}}},$$

in order to yield the MMSE DFE, as shown in Section 4.2.2. The SD is always executed with the QR decomposed standard channel matrix $\mathbf{H}$.

In summary, a QR decomposition decomposes any matrix $\mathbf{H}$ into a quadratic unitary matrix $\mathbf{Q}'$ and a rectangular matrix of upper triangular structure $\mathbf{R}'$. For this work, it is focused on the case, where $\mathbf{H}$ has more rows than columns, as the Unique Word OFDM symbol generation process yields channel matrices $\mathbf{H} \in \mathbb{C}^{(N_d+N_r)\times N_d}$ with this property. Thus, the dimensions of the components amount to

$$\begin{aligned}
&\mathbf{H} = \mathbf{Q}'\mathbf{R}' && \mathbf{H} \in \mathbb{C}^{(N_d+N_r)\times N_d}\\
&\mathbf{Q}' = \begin{bmatrix}\mathbf{Q} & \mathbf{Q}_0\end{bmatrix} && \mathbf{Q}' \in \mathbb{C}^{(N_d+N_r)\times(N_d+N_r)},\\
& && \mathbf{Q} \in \mathbb{C}^{(N_d+N_r)\times N_d}, \mathbf{Q}_0 \in \mathbb{C}^{(N_d+N_r)\times N_r}\\
&\mathbf{R}' = \begin{bmatrix}\mathbf{R}\\ \mathbf{0}\end{bmatrix} && \mathbf{R}' \in \mathbb{C}^{(N_d+N_r)\times N_d}, \mathbf{R} \in \mathbb{C}^{N_d\times N_d}.
\end{aligned} \tag{5.1}$$

Important parts of the matrices $\mathbf{Q}'$ and $\mathbf{R}'$, which play a special role, can be further identified. This is detailed with the initial idea of the QR decomposition: Orthogonalization.

5.1. Gram-Schmidt Orthogonalization

In [WRB+02] a Modified Gram-Schmidt orthogonalization is described as a way to decompose a complex flat-fading MIMO channel matrix $\mathbf{H}$, which has the element h_{kl} in the k-th row and l-th column, representing the complex fading gain between transmit antenna l and receive antenna k. However, the same system model applies for any UW-OFDM system, although considering only single-antenna systems.

The Gram-Schmidt orthogonalization, outlined as pseudo-code in Algorithm 12, orthogonalizes the base vectors describing the vector space spanned by $\mathbf{H} = [\mathbf{h}_0, \ldots, \mathbf{h}_{N_d-1}]$ and documents the necessary operations in $\mathbf{R}$.

Algorithm 12 Gram-Schmidt Orthogonalization

```
1:  function [Q, R] =MGS(H)
2:      Q ← H, R ← 0
3:      for l = 0, ..., N_d − 1 do
4:          [R]_{l,l} ← ‖q_l‖_2
5:          q_l ← q_l / [R]_{l,l}
6:          for k = l + 1, ..., N_d − 1 do
7:              [R]_{l,k} ← q_l^H q_k                ▷ part of q_k in direction of q_l
8:              q_k ← q_k − [R]_{l,k} q_l            ▷ subtract projection of q_k to q_l
9:          end for
10:     end for
11: end function
```

It comprises of the following steps:

1. initialize $\mathbf{Q} = \mathbf{H}$, line 2
2. normalize the current base vector by its l2-norm $\|\mathbf{q}_l\|_2 = \sqrt{\sum_{k=0}^{N_d+N_r-1} |[\mathbf{Q}]_{k,l}|^2}$, line 5
3. remove from all remaining base vectors the part in direction of the current base vector, line 8
4. pick next base vector and return to step 2, until the whole $\mathbf{Q}$ is orthogonalized

The modification that distinguishes the *Modified* Gram-Schmidt Orthogonalization (hence MGS) from the original one is a simple adaption in order to increase the numerical stability. Instead of subtracting the parallel part of the current vector $\mathbf{q}_l$ to all already fixed base vectors, the parallel part is subtracted from the future base vectors iteratively (line 8), causing parts of occurring numerical errors to be orthogonalized and removed [HJ90].

The execution of Algorithm 12 yields the matrices $\mathbf{Q}$ and $\mathbf{R}$, as characterized in (5.1). While $\mathbf{Q}$ fulfills the property

$$\mathbf{Q}^{\mathsf{H}}\mathbf{Q} = \mathbf{I}, \tag{5.2}$$

the right part $\mathbf{Q}_0$ of $\mathbf{Q}'$ is missing for full unitarity. $\mathbf{Q}_0$ is a set of orthonormal base vectors that represent the *left null space* or *co-kernel* of $\mathbf{H}$, such that any multiplication

with a vector $\mathbf{v}$ from the left null space yields zero

$$\begin{aligned} \mathbf{v}^{\mathsf{H}}\mathbf{H} &= \mathbf{0}^{\mathsf{H}}, \\ \mathbf{v} &= \mathbf{Q}_0 \left[\nu_0, \nu_1, \ldots, \nu_{N_r-1}\right]^{\mathsf{T}}, \\ \nu_k &\in \mathbb{C}. \end{aligned} \tag{5.3}$$

This can also be observed in

$$\mathbf{R}' = \begin{bmatrix} \mathbf{R} \\ \mathbf{0} \end{bmatrix},$$

where the $\mathbf{0}$-matrix is the part that applies to $\mathbf{Q}_0$. On the other hand, $\mathbf{Q}$ is a set of orthonormal base vectors representing the *co-image* of $\mathbf{H}$.

A valid base vector set of the left null space $\mathbf{Q}_0$ can be determined easily one by one, by choosing a random vector, subtracting the components in direction of all already fixed base vectors of image and null space and final normalization, or by a singular value decomposition of $\mathbf{H} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^{\mathsf{H}}$ and extracting the last N_r columns of $\mathbf{U}$.

5.2. Sorted QR Decomposition

From the description of the Gram-Schmidt Orthogonalization it is obvious that the order, in which the base vectors are orthogonalized, is set by the order they appear in $\mathbf{H}$. For the purpose of DFE however, the order of the orthogonalization is crucial to the probability of error propagation in DFE [BWKK03, WBKK03] and thus its BER performance. For the SD the execution time can also be reduced tremendously by the right permutation of the columns of $\mathbf{H}$ prior to computing the MGS.

For DFE it is beneficial to decide more reliable data symbols first. A data symbol with high reliability is one with a low noise amplification and hence a high SNR, respectively. As noticed in Section 4.2, the data value to be worked on first, corresponds to $\mathbf{q}_{N_d-1}$, which showed a large norm apparent by $[\mathbf{R}]_{N_d-1,N_d-1}$. Since $\mathbf{H}$ cannot be decomposed backwards in order to maximize $[\mathbf{R}]_{N_d-1,N_d-1}$, it should be tried to *minimize* the vector norm (line 5) in each step of the orthogonalization. The idea behind this is that the DFE layers detected last (that are the first ones to result from the decomposition) affect only a few other layers through error propagation and thus may suffer from higher noise, while hoping to get lower noise conditions for later layers that affect the earlier DFE iterations.

This is basically the step that needs to be done, to get from MGS to Sorted QR Decomposition: In each step l, process the vector $\mathbf{q}_k$ that has minimum norm [HL89]. The concept is sketched in Algorithm 13 according to [WBKK03]; operations that are also present in the MGS Algorithm 12 are grayed out, while new operations are in black.

Algorithm 13 Sorted QR Decomposition

function $[\mathbf{Q}, \mathbf{R}, \mathbf{p}] = \text{SQRD}(\mathbf{H})$
  $\mathbf{Q} \leftarrow \mathbf{H}, \mathbf{R} \leftarrow \mathbf{0}, \mathbf{p} \leftarrow (1, \ldots, N_d)$
  $\lambda_l \leftarrow \|\mathbf{q}_l\|_2^2, \quad \forall l = 0, \ldots, N_d - 1$ ▷ calculate norm of base vectors
  for $l = 0, \ldots, N_d - 1$ **do**
    $m = \underset{n=l,\ldots,N_d-1}{\operatorname{argmin}} \lambda_n$
    exchange columns l and m in $\mathbf{Q}$ and $\mathbf{R}$ and elements l and m in $\mathbf{p}$ and $\boldsymbol{\lambda}$
    $[\mathbf{R}]_{l,l} \leftarrow \sqrt{\lambda_l}$
    $\mathbf{q}_l \leftarrow \mathbf{q}_l / [\mathbf{R}]_{l,l}$
    for $k = l + 1, \ldots, N_d - 1$ **do**
      $[\mathbf{R}]_{l,k} \leftarrow \mathbf{q}_l^{\mathrm{H}} \mathbf{q}_k$ ▷ projection of $\mathbf{q}_k$ on $\mathbf{q}_l$
      $\mathbf{q}_k \leftarrow \mathbf{q}_k - [\mathbf{R}]_{l,k} \mathbf{q}_l$ ▷ subtract component of $\mathbf{q}_k$ parallel to $\mathbf{q}_l$
      $\lambda_k \leftarrow \lambda_k - |[\mathbf{R}]_{l,k}|^2$ ▷ update vector norms
    end for
  end for
end function

5.3. SQRD with Post-Sorting

In [WBKK03] an extension of the SQRD with an extended channel matrix (hence the MMSE SQRD) is introduced that is able to improve the initial QR decomposition using a post-sorting algorithm (PSA). This algorithm uses Householder reflectors to restore the triangular structure of a moreover sorted $\mathbf{R}$-matrix and is shown in [WBKK03]. It is provided with the $\mathbf{Q}$-matrix from an initial QR decomposition, an MMSE SQRD, where the SQRD of the extended channel matrix is recommended by the author.

A QR decomposition from the PSA will not yield the best results, but the degradation over the best possible QR decomposition is very small, as shown in Section 5.4.

5.4. Impact of Different QR Decompositions on UW-OFDM Detection

In order to evaluate the impact of all these QR decomposition methods, simulation results are shown that display the achieved bit error performance of the uncoded ZF DFE and the MMSE DFE receiver using the matrices $\mathbf{Q}$ and $\mathbf{R}$, as well as the average number of node visits needed per bit of information to find the MLSE solution with the hard-output SD for the given decomposition. Also the simulation results of all QR decompositions that can be achieved by differently permuted channel matrices and the MGS algorithm are shown, to see how good they perform in the light of the best QR decompositions. Therefore, $\mathbf{H}$ is permuted before execution of the MGS, to get a different order of the data symbols during the DFE processing. All possible permutations of the base vectors in $\mathbf{H}$ are generated, in order to get the full performance range that can be achieved.

In [Knu05, RH84] an algorithm is presented for lexicographic permutation generation. This method, as described in Algorithm 14, generates all existing permutations, if executed

repeatedly and starting with a sequence $\mathbf{p} = [p_0, p_1, \ldots, p_{n-1}]^\mathsf{T}$ that is initially sorted such that

$$p_0 \leq p_1 \leq \cdots \leq p_{n-1}. \tag{5.4}$$

Algorithm 14 Generate next permutation in lexicographic order

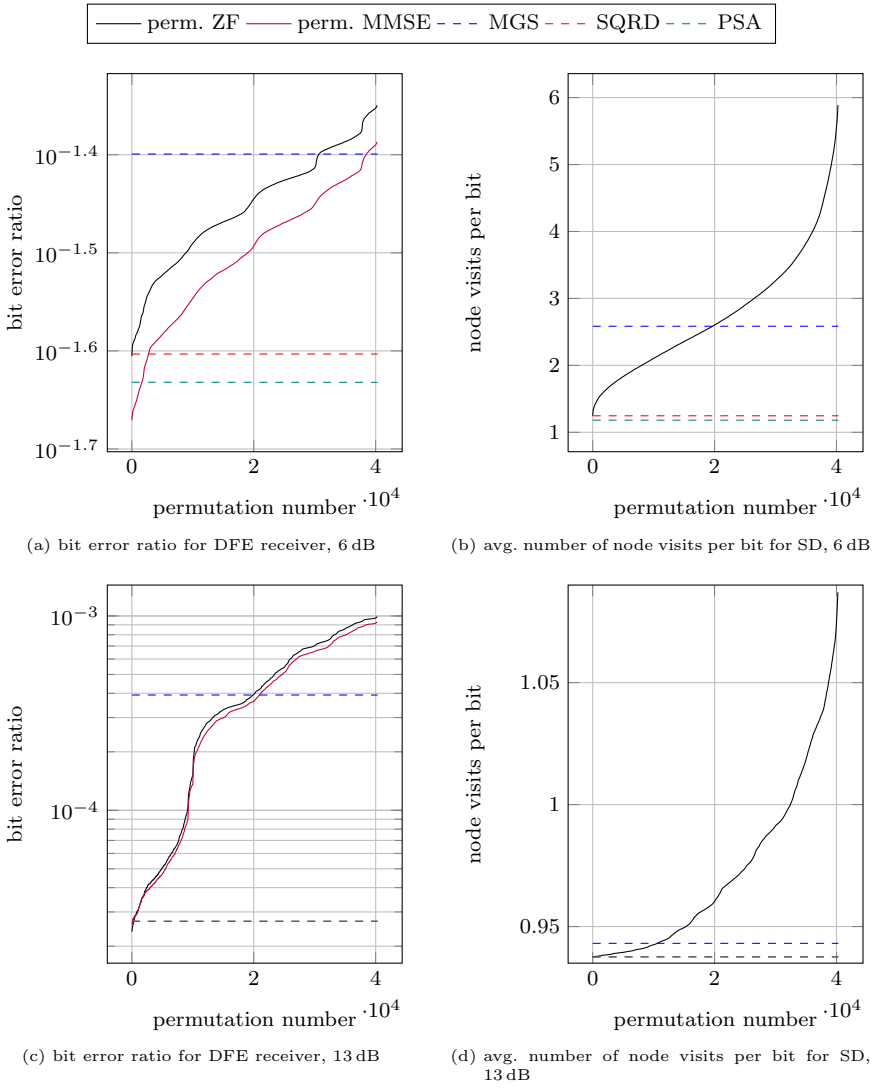
```
1: function p =NEXT_PERMUTATION(p)
2:     k ← N − 1
3:     while p_k ≥ p_{k+1} ∧ k > 0 do
4:         k ← k − 1
5:     end while                                  ▷ 0 ≤ k ≤ N − 1 after this
6:     l ← N
7:     while p_k ≥ p_l do
8:         l ← l − 1
9:     end while
10:    swap p_k and p_l
11:    reverse sequence p_{k+1}, ..., p_N
12: end function
```

With an initial permutation vector $\mathbf{p} = [0, 1, 2, \ldots, N_d - 1]$ the condition for this algorithm is fulfilled, and it will generate all $N_d!$ permutations. The resulting order vector can be utilized by permuting the base vectors in $\mathbf{H}$ according to $\mathbf{p}$ and execute the MGS with the reordered matrix.

To be able to handle the big number of possible permutations the UW-OFDM system shall be restricted to the uncoded exemplary system ML (see Table A.1), with $N_d = 8$ and thus 40 320 permutations. These have an effect on the bit error probability for DFE receivers and the number of node visits for the SD. Both are displayed in Figure 5.1. The results are sorted for visualization. Additionally, the results of the modified Gram-Schmidt Orthogonalization (MGS), the sorted QR decomposition (SQRD) and the SQRD using the Post-Sort algorithm (PSA) are displayed as dashed lines. Both methods using the SQRD are used with the extended channel matrix, while the MGS resembles an unsorted QR decomposition of the regular channel matrix (corresponding to the permutation vector $\mathbf{p} = [0, 1, \ldots, N_d - 1]$). The measurements were executed in the exemplary multipath channel A (see Appendix A) at an E_b/N_0-ratio of 6 and 13 dB in order to represent a low- and a high-SNR environment.

In the low-SNR situation the BER is hardly affected by differently permuted QR decompositions. The MGS and SQRD yield an average bit error ratio here, which is a negligible degradation over the best by permutation achievable BER. The PSA however is able to achieve a very good result, that is only achieved by a few permutations with an MMSE DFE and not at all by a ZF DFE. The reason for this is that the post-sorting algorithm alters the base vectors in $\mathbf{H}$ more fundamentally, in order to determine the $\mathbf{Q}$ and $\mathbf{R}$ matrices.

The number of node visits however, show a much broader range achievable by different permutations at low SNR: The worst permutation needs more than four times longer than the best performing permutation. Almost at the top performing end is the SQRD, which is about 52% faster than with the MGS. The PSA again outperforms all possible permuted QRDs with a gain of 5% over the SQRD. The term '*faster*' actually means that it is able to find the MLSE solution with 5% less node visits on average.



(a) bit error ratio for DFE receiver, 6 dB

(b) avg. number of node visits per bit for SD, 6 dB

(c) bit error ratio for DFE receiver, 13 dB

(d) avg. number of node visits per bit for SD, 13 dB

Figure 5.1.: Effect of differently permuted QR decompositions on the BER for a DFE receiver and the average number of node visits per bit of information for the SD in a low- and high-SNR scenario.

For DFE in the high-SNR scenario, the BER of a bad performing permutation can be improved by more than one and a half orders of magnitude. SQRD and PSA perform equally, as good as the best permutations regarding both, BER and noise visits. However, the dynamic range of the number of node visits has shrunk to the interval from about 0.938 to 1.095 node visits on average, which corresponds to only 14% of achievable gain over the worst permutation. As mentioned several times in this work, ZF and MMSE are equivalent for higher SNR, which is why the MMSE DFE does not yield a significant advantage over the ZF version.

5.5. Summary of QR Decompositions

QR decompositions or detection orderings and their impact on MIMO detection have been studied in many publications, for example [HV05, VH05, WFGV98, BWKK03, WRB+02, WBKK03], while sticking to a few algorithms to decompose the channel matrix. In this section fundamental algorithms were shown in order to analyze their performance for UW-OFDM systems using DFE and SD receivers. Furthermore, all possible base vector permutations of the channel matrix were considered, in order to evaluate their performance to all easily achievable QR decompositions.

Unsurprisingly, the widely used sorted QR decomposition achieves very good results in terms of the bit error probability for the DFE receivers and in terms of average number of node visits per bit of information for the SD receiver. The SQRD with PSA achieves even better results, but is by far more complex and numerically unstable. In a low-SNR scenario, a 'good' QR decomposition method only pays off for the SD, as the average number of node visits can be reduced tremendously. The DFE, however, does not benefit strongly from a good QR decomposition. In the high-SNR scenario the effect is switched: While the average number of node visits are not altered a lot with different decompositions, the DFE is able to gain almost two orders of magnitude in BER. In any case, it should be deliberated whether to employ a more or less complex QR decomposition, depending on the resources, the SNR and the used receiver concept.

6. Conclusion

In this work, Unique Word OFDM as an evolution of the conventional OFDM using cyclic prefixes was considered. The UW-OFDM generation approach that introduces redundancy in the OFDM symbols was introduced and analyzed in detail. The resulting mean OFDM symbol energies exposed a potential pitfall: Based on the analytical as well as the numerical results, the intuitively obvious direct approach of systematic UW-OFDM symbol generation was excluded from further considerations, as it is always inferior compared to the so-called two-step approach. An optimization method to determine good distributions of redundant subcarriers, as they are needed in order to generate UW-OFDM symbols systematically, was introduced. Non-systematically generated UW-OFDM, where the data symbols are no longer visible in frequency domain, was also discussed and its mean OFDM symbol energy derived.

The potential superiority of UW-OFDM over CP-OFDM is shown in other publications [HH12]. In this work, the focus was on linear and nonlinear receivers that are able to exploit the introduced redundancy beneficially for recovery of the transmitted data. As linear receivers, two rather intuitive estimators (CI and TDW) were shown, along with two optimum estimators: From classical estimation theory emerged the BLUE, while considering the system model in the Bayesian sense produced an LMMSE data estimator. In addition to their usual form, both were shown in a version with reduced complexity for systematically generated UW-OFDM. All receivers were analyzed regarding their computational complexity, in terms of complex multiplication equivalents. A final performance comparison in terms of their simulated BER showed that the by far least complex intuitive estimators cannot compete with the BLUE and the LMMSE estimator. Together with reliability information of the estimated data symbols, which was extracted of the statistical information of the remaining error after estimation, the BLUE showed a great performance in coded transmission that was only superseded by the LMMSE estimator. Non-systematically generated UW-OFDM performs in general better than the systematic variant, at the price of higher complexity. When using real transmit symbol alphabets like ASK, another variant, the WLMMSE estimator provides a gain over the plain LMMSE receiver and poses the better choice for this case. It was shown that for higher order QAM and ASK transmit constellations, a symbol scaling effect of the Bayesian estimators applies severely. In these cases, the estimators yield better results, if a symbol scaling compensation is employed. Simulation results proved the excellent performance of all these improved versions and document their benefit.

The distinct system structure of the UW-OFDM signaling scheme also allows for non-linear receivers that are especially known from the MIMO world. In contrast, these methods can still be used for single-antenna UW-OFDM systems. In the linear estimation chapter, the statistics of the remaining error after an LMMSE estimation revealed a correlation of the error values. In the first nonlinear receiver, these correlations are exploited to improve the data estimation. This so-called noise interpolation method was derived from Wiener filter theory and two algorithms were introduced, that perform noise interpolation in a batch approach and in an iterative manner. The main problem turned out to be the

selection of the samples that are used for the estimation of others. Several possibilities, like the noise variance, symbol error probability or correlation criterion, were shown for how to do this selection. In the end, the batch noise interpolator with a well defined selection criterion was picked that performed very well in coded and uncoded transmission.

In decision feedback equalization, the influence of safely decided data symbols is subtracted from the receive vector. That way, following decisions of symbols can be made safer, which results in better data estimates. Here, the processing order is crucial for the BER performance. The algorithm was shown in its ZF and MMSE variant, and translated in a version, where it could be expressed in terms of a QR decomposition and a final iterative processing loop.

The best possible data estimates are obtained by an MLSE. The sphere decoding algorithm is a practical implementation of the MLSE principle. The SD producing hard data decisions was discussed in detail. For the simulation of an uncoded transmission with an SD, this work reached for new records in the considered 48 dimensions. In order to obtain reliability information, a soft-output SD was discussed, which is unfortunately not able to obtain the optimum MLSE results, as it employs an approximation and a clipping threshold to keep the execution time at a practical level. The BER performance as well as the computational complexity in dependence of the SNR was analyzed for several thresholds. Its loss against a soft-output MLSE could only be evaluated for a very small UW-OFDM system.

Next, several ways to do a QR decomposition were discussed, as the realization has a strong impact on the BER performance of DFE and the execution time of the SD. While a wide range is covered by all possible permutations of the channel matrix and an MGS, it could be shown that the SQRD is a quite simple algorithm that produces results near the optimum in both BER performance or execution time, respectively.

A final BER comparison shows that in AWGN, non-systematically generated UW-OFDM achieves the same BER for all discussed receivers. Hence, in this special case, the LMMSE estimator poses already the optimum receiver. For all other situations, the SD has a considerable gain over the LMMSE estimator, at the price of an increased complexity. DFE and NI can be considered as a complexity trade-off, featuring a better performance than the LMMSE estimator but worse than the SD.

A. Parameters of Exemplary UW-OFDM Systems

For simulations, always one of the UW-OFDM systems with the parameters as shown in Table A.1 is used. Bit error simulations are performed with a predetermined set of 10 000 multipath channels. In certain situations one exemplary multipath channel is picked for analysis. The frequency response of this channel constitutes a quite severe fading environment by its deep spectral notches and is displayed in Figure A.1.

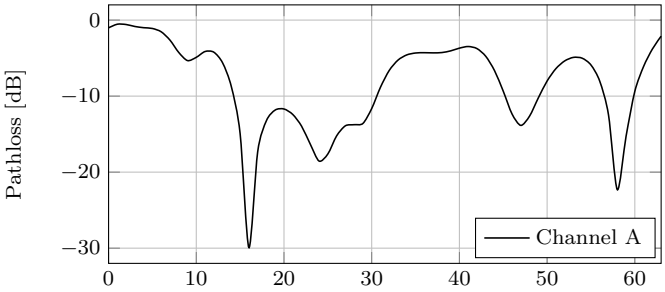


Figure A.1.: Frequency response of exemplary multipath channel A.

For the channel simulation the transmission over various time-invariant multipath channels is realized by convolution of the transmit signal with a channel impulse response vector $\mathbf{h}$ with a maximum length of $N_\mathrm{G}+1$ taps. In any case additive white Gaussian noise is added after this. For the simulations of the multipath environment, channel impulse responses (CIRs) are generated according to the model presented in [Fak97], which has also been used during the IEEE 802.11a standardization process. A fixed set of 10 000 CIRs is used as realization of the channel model. The CIR is assumed to be constant during the transmission of many OFDM symbols. The BER values are the averaged results of the transmissions over all these channel realizations. The choice of interleaving factors (number of positions, neighboring code bits are spread apart) is given for each system setup in Table A.1. For the special case, when the transmission over an AWGN channel is simulated, the CIR is simply set to $\mathbf{h} = [1]$. It is always assumed that the receiver has perfect knowledge of the current channel statistics, which is the CIR and the E_b/N_0-ratio. Since the focus lies on data estimation procedures in this work rather than on synchronization or channel estimation approaches, the zero UW is chosen for the BER simulations.

UW-OFDM system		A	B	SD	ML
DFT size	N	64	64	24	12
data subcarriers	N_d	36	48	16	8
redundant subcarriers	N_r	16	16	8	4
red. subcarrier indices	$\mathcal{I}_r$	$\{2, 6, 10, 14, 17, 21$ $24, 26, 38, 40, 43,$ $47, 50, 54, 58, 62\}$	$\{1, 5, \ldots, 61\}$	$\{1, 4, \ldots, 22\}$	$\{1, 4, 7, 10\}$
zero subcarriers	N_z	12	0	0	0
zero subcarrier indices	$\mathcal{I}_z$	$\{0, 27, 28, \ldots, 37\}$	$\{\}$	$\{\}$	$\{\}$
unique word length	N_u	16	16	8	4
interleaving factor		15	15	9	9

Table A.1.: Parameters of exemplary UW-OFDM system.

B. Proof of Cyclicity of UW-OFDM Symbols

One of the most important underlying properties of OFDM transmission is that the alignment creates some kind of cyclicity of the OFDM symbols. In this context, the cyclicity property is synonymic to the transformation of the linear convolution, as it appears in multipath channel propagation, to a cyclic convolution. Only then, the convolution in time domain can be expressed as a multiplication in frequency domain after transformation with the DFT. Furthermore, this property also ensures that each OFDM symbol is free of inter-symbol interference and can be regarded on its own and independent of any previous transmission.

For CP-OFDM most text books about the topic show the cyclicity proof [NP00]. When showing the cyclicity of a UW-OFDM transmission, the two succeeding UW-OFDM symbols number $l-1$ and l, as defined in (2.16), are given by

$$\begin{bmatrix} \mathbf{x}'^{(l-1)} \\ \mathbf{x}'^{(l)} \end{bmatrix} = \begin{bmatrix} \mathbf{x}_{\mathrm{p}}^{(l-1)} \\ \mathbf{x}_{\mathrm{u}} \\ \mathbf{x}_{\mathrm{p}}^{(l)} \\ \mathbf{x}_{\mathrm{u}} \end{bmatrix},$$

split in the payload part of length $N - N_u$ and the unique word $\mathbf{x}_{\mathrm{u}}$ of length N_u. After propagation over a multipath channel with a CIR not longer than $N_u + 1$ and disregarding AWGN, the payload of symbol l is clearly not affected by the payload of symbol $l-1$, such that $\mathbf{x}_{\mathrm{p}}^{(l-1)}$ can be omitted. Then, the receive vector, omitting the channel transient of the last UW, can be written as

$$\mathbf{y} = \mathbf{H}_c \begin{bmatrix} \mathbf{x}_{\mathrm{u}} \\ \mathbf{x}_{\mathrm{p}}^{(l)} \\ \mathbf{x}_{\mathrm{u}} \end{bmatrix}, \tag{B.1}$$

with the linear convolution matrix of size $(N+N_u)\times(N+N_u)$

$$\mathbf{H}_c = \begin{bmatrix} h_0 & & & \mathbf{0} & & \\ h_1 & h_0 & & & & \\ \vdots & h_1 & \ddots & & & \\ h_{N_u} & \vdots & & \ddots & & \\ & & & & \ddots & \\ \mathbf{0} & & h_{N_u} & \cdots & h_1 & h_0 \end{bmatrix}. \tag{B.2}$$

The fact that $\mathbf{x}_\mathrm{u}$ is included twice in the transmit vector can be modeled by defining a copy matrix

$$\Theta_\mathrm{c} = \left[\begin{array}{c|c} \mathbf{0}^{(N_u \times (N-N_u))} & \mathbf{I}^{(N_u)} \\ \hline \multicolumn{2}{c}{\mathbf{I}^{(N)}} \end{array}\right]. \tag{B.3}$$

Furthermore, the part of the receive vector that is relevant for UW-OFDM receiver processing for symbol number l can be extracted by an extraction matrix

$$\Theta_\mathrm{x} = \left[\mathbf{0}^{(N \times N_u)} \quad \mathbf{I}^{(N)}\right]. \tag{B.4}$$

Then, the relevant receive vector can be expressed as

$$\mathbf{y}_x = \Theta_x \mathbf{H}_c \Theta_c \begin{bmatrix} \mathbf{x}_\mathrm{p}^{(l)} \\ \mathbf{x}_\mathrm{u} \end{bmatrix}. \tag{B.5}$$

It can be easily shown that the overall channel matrix

$$\mathbf{H}_\mathrm{O} = \Theta_x \mathbf{H}_c \Theta_c \tag{B.6}$$

$$= \begin{bmatrix} h_0 & \mathbf{0} & & h_{N_u} & \cdots & h_1 \\ h_1 & h_0 & & & \ddots & \vdots \\ \vdots & \ddots & \ddots & & & h_{N_u} \\ h_{N_u} & & & & & \mathbf{0} \\ & \ddots & & & \ddots & \\ \mathbf{0} & & h_{N_u} & \cdots & h_1 & h_0 \end{bmatrix} \tag{B.7}$$

is circulant, describing a cyclic convolution. Any circulant matrix is diagonalized by the Fourier matrix. This way,

$$\widetilde{\mathbf{H}} = \mathbf{F}_N \mathbf{H}_\mathrm{O} \mathbf{F}_N^{-1} \tag{B.8}$$

is a diagonal.

C. Proof of Inequality (2.48)

In the following it will be shown that the OFDM symbol energy (2.41) is always smaller or equal to (2.47), such that

$$\frac{1}{N}\mathbf{x}_{\mathrm{u}}^{\mathsf{H}}\mathbf{M}_{22}^{-\mathsf{H}}\mathbf{M}_{22}^{-1}\mathbf{x}_{\mathrm{u}} \geq \mathbf{x}_{\mathrm{u}}^{\mathsf{H}}\mathbf{x}_{\mathrm{u}} \tag{C.1}$$

holds.

As the zero UW clearly yields equality, only the case $\mathbf{x}_{\mathrm{u}} \neq \mathbf{0}$ will be considered here. Some prerequisites are stated in the following, which will be used throughout the proof:

- For any matrix $\mathbf{C}$, the Gramian matrix $\mathbf{D} = \mathbf{C}^{\mathsf{H}}\mathbf{C}$ is a positive-semidefinite Hermitian matrix, for which all eigenvalues λ_k are real and non-negative. Furthermore, if $\mathbf{C}$ is invertible, then $\mathbf{D}$ is also invertible and positive-definite with all eigenvalues $\lambda_k > 0$. The definition $\mathbf{D} = \frac{1}{N}\mathbf{M}_{22}^{-\mathsf{H}}\mathbf{M}_{22}^{-1}$ fulfills all the stated properties.
- For any vector $\mathbf{x}_{\mathrm{u}} \neq \mathbf{0}$ the Rayleigh quotient of a Hermitian matrix $\mathbf{D}$ is defined as

$$R(\mathbf{x}_{\mathrm{u}}) = \frac{\mathbf{x}_{\mathrm{u}}^{\mathsf{H}}\mathbf{D}\mathbf{x}_{\mathrm{u}}}{\mathbf{x}_{\mathrm{u}}^{\mathsf{H}}\mathbf{x}_{\mathrm{u}}}. \tag{C.2}$$

  The following inequalities hold for the Rayleigh quotient:

$$\lambda_{\min}(\mathbf{D}) \leq R(\mathbf{x}_{\mathrm{u}}) \leq \lambda_{\max}(\mathbf{D}), \tag{C.3}$$

  where $\lambda_{\min}(\mathbf{D})$ and $\lambda_{\max}(\mathbf{D})$ denote the minimum and maximum eigenvalue of $\mathbf{D}$, respectively.

Considering (C.2) and (C.3), it can be stated that

$$\mathbf{x}_{\mathrm{u}}^{\mathsf{H}} \underbrace{\left[\frac{1}{N}\mathbf{M}_{22}^{-\mathsf{H}}\mathbf{M}_{22}^{-1}\right]}_{\mathbf{D}} \mathbf{x}_{\mathrm{u}} \geq \underbrace{\lambda_{\min}(\mathbf{D})\mathbf{x}_{\mathrm{u}}^{\mathsf{H}}\mathbf{x}_{\mathrm{u}}}_{z}. \tag{C.4}$$

In the following the right hand side of (C.4) comes into consideration, which can be written as

$$\begin{aligned} z &= \lambda_{\min}\left(\frac{1}{N}\mathbf{M}_{22}^{-\mathsf{H}}\mathbf{M}_{22}^{-1}\right)\mathbf{x}_{\mathrm{u}}^{\mathsf{H}}\mathbf{x}_{\mathrm{u}} \\ &= \frac{1}{N}\lambda_{\min}\left(\mathbf{M}_{22}^{-\mathsf{H}}\mathbf{M}_{22}^{-1}\right)\mathbf{x}_{\mathrm{u}}^{\mathsf{H}}\mathbf{x}_{\mathrm{u}} \end{aligned}$$

For any invertible matrix $\mathbf{A}$ with an eigenvector $\mathbf{v}$ and the associated eigenvalue λ, it holds true that $\mathbf{v}$ is also an eigenvector of the inverse matrix $\mathbf{A}^{-1}$ to the associated eigenvalue $\frac{1}{\lambda}$. Hence, it can be written

$$\begin{aligned} z &= \frac{1}{N} \frac{\mathbf{x}_\mathrm{u}^\mathsf{H}\mathbf{x}_\mathrm{u}}{\lambda_{\max}\left(\left[\mathbf{M}_{22}^{-\mathsf{H}}\mathbf{M}_{22}^{-1}\right]^{-1}\right)} \\ &= \frac{1}{N} \frac{\mathbf{x}_\mathrm{u}^\mathsf{H}\mathbf{x}_\mathrm{u}}{\lambda_{\max}\left(\mathbf{M}_{22}\mathbf{M}_{22}^\mathsf{H}\right)} \\ &= \frac{1}{N} \frac{\mathbf{x}_\mathrm{u}^\mathsf{H}\mathbf{x}_\mathrm{u}}{\left\|\mathbf{M}_{22}^\mathsf{H}\right\|^2} \\ &= \frac{1}{N} \frac{\mathbf{x}_\mathrm{u}^\mathsf{H}\mathbf{x}_\mathrm{u}}{\left\|\mathbf{M}_{22}\right\|^2} \end{aligned} \tag{C.5}$$

Here, $\|\cdot\|$ denotes the spectral norm defined as $\|\mathbf{M}_{22}\| = \sqrt{\lambda_{\max}\left(\mathbf{M}_{22}^\mathsf{H}\mathbf{M}_{22}\right)}$. For the IDFT matrix $\mathbf{F}_N^{-1}$, the sprectral norm is $\left\|\mathbf{F}_N^{-1}\right\| = \frac{1}{\sqrt{N}}$. The spectral norm of any submatrix cannot exceed the spectral norm of the matrix it has been extracted from. Since $\mathbf{M}_{22}$ is a submatrix of $\mathbf{F}_N^{-1}$, it can be stated that

$$\|\mathbf{M}_{22}\| \leq \left\|\mathbf{F}_N^{-1}\right\| = \frac{1}{\sqrt{N}} \qquad \text{or} \qquad \frac{1}{\|\mathbf{M}_{22}\|} \geq \sqrt{N}. \tag{C.6}$$

Taking this into account, it immediately follows for $N > 1$ that

$$z \geq \mathbf{x}_\mathrm{u}^\mathsf{H}\mathbf{x}_\mathrm{u}. \tag{C.7}$$

Linking this to (C.4), it can be further summarized as

$$\mathbf{x}_\mathrm{u}^\mathsf{H}\left[\frac{1}{N}\mathbf{M}_{22}^{-\mathsf{H}}\mathbf{M}_{22}^{-1}\right]\mathbf{x}_\mathrm{u} \geq \mathbf{x}_\mathrm{u}^\mathsf{H}\mathbf{x}_\mathrm{u}, \tag{C.8}$$

which concludes the proof.

D. Analytical Determination of the Symbol Error Probability

For the case of zero-mean additive Gaussian noise v on a complex symbol d from a QAM alphabet $\mathcal{A}$

$$\begin{aligned} &\tilde{d} = d + v, \\ &d \in \mathcal{A}, \\ &v \sim \mathcal{N}_{\mathrm{C}}(0; \sigma_v^2), \end{aligned} \tag{D.1}$$

the probability of a false detection can be approximated analytically [Pro00]. For hard-decision data detection, this symbol error probability is the important measure to indicate the probability of slicing to the wrong data symbol. The assumption of zero-mean Gaussian noise after linear data estimation is valid, for example, for the BLUE. However, it is only an approximations for LMMSE estimation.

At first the symbol error probability (also symbol error ratio, SER) for the case of a binary symbol alphabet $\mathcal{A}$ as apparent in Figure D.1 is derived. An incorrect decision happens, if the added noise pushes the receive value over the decision boundary, which is for equiprobable source bits in the middle between the two transmit symbols that are separated by the distance $d_{\min}$. This is the case, if the part of the noise v in direction of the neighboring symbol is larger than $d_{\min}/2$.

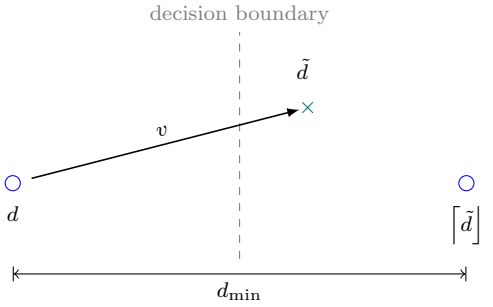


Figure D.1.: Binary transmit symbol alphabet with Gaussian noise.

As the statistical properties of the Gaussian random variable v are known, the exact

symbol error probability can be determined as

$$\chi = \Pr\left(\left\lceil \tilde{d} \right\rfloor \neq d\right) \tag{D.2}$$

$$= \Pr\left(\Re\left\{v\right\} \geq d_{\min}/2\right) \tag{D.3}$$

$$= \frac{1}{\sqrt{2\pi\sigma_{v_\Re}^2}} \int_{d_{\min}/2}^{\infty} \mathrm{e}^{-\frac{t^2}{2\sigma_{v_\Re}^2}} \,\mathrm{d}t. \tag{D.4}$$

The variance of the real part only, assuming properness of the complex Gaussian noise, is $\sigma_{v_\Re}^2 = \sigma_v^2/2$. This yields

$$\chi = \frac{1}{\sqrt{\pi\sigma_v^2}} \int_{d_{\min}/2}^{\infty} \mathrm{e}^{-\frac{t^2}{\sigma_v^2}} \,\mathrm{d}t, \tag{D.5}$$

and after introducing the substitution $\tau = \sqrt{2}t/\sigma_v$ and thus $t = \sigma_v\tau/\sqrt{2}$

$$\chi = \frac{1}{\sqrt{2\pi}} \int_{\frac{d_{\min}}{\sqrt{2}\sigma_v}}^{\infty} \mathrm{e}^{-\frac{\tau^2}{2}} \,\mathrm{d}\tau, \tag{D.6}$$

which can be easily evaluated numerically, for example using the Q-function or the complementary error function

$$\chi = \mathrm{Q}\left(\sqrt{\frac{d_{\min}^2}{2\sigma_v^2}}\right), \tag{D.7}$$

$$\mathrm{Q}\left(x\right) = \frac{1}{\sqrt{2\pi}} \int_{x}^{\infty} \mathrm{e}^{-t^2/2} \mathrm{d}t = \frac{1}{2}\mathrm{erfc}\left(\frac{x}{\sqrt{2}}\right). \tag{D.8}$$

An interpretation of the Q-function is shown in Figure D.2, where the hatched area is the result of $\mathrm{Q}\left(x\right)$.

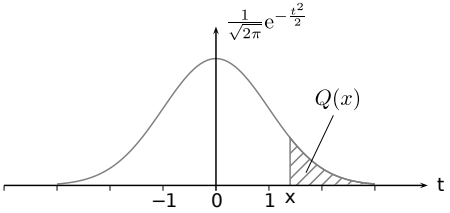


Figure D.2.: Interpretation of the Q-function.

For higher order constellations and in particular QAM, however, a closer look needs to be taken. Figure D.3 shows an extract of a big QAM constellation and identifies neighboring detection candidates. A point not at an edge has two borders of the decision region at distance $d_{\min}/2$ in the real and imaginary part, respectively, which yield a symbol error if exceeded. Then, the probability of these four events considered separately add up to

$$\chi \approx \Pr\left(\Re\left\{v\right\} \geq d_{\min}/2\right) + \Pr\left(\Re\left\{v\right\} \leq -d_{\min}/2\right) + \tag{D.9}$$

$$+ \Pr\left(\Im\left\{v\right\} \geq d_{\min}/2\right) + \Pr\left(\Im\left\{v\right\} \leq -d_{\min}/2\right) \tag{D.10}$$

$$= 4\mathrm{Q}\left(\sqrt{\frac{d_{\min}^2}{2\sigma_v^2}}\right). \tag{D.11}$$

Proceeding like this introduces the issue that this probability is too high for the following reason: Considering the addition of the two probabilities $\Pr\left(\Re\left\{v\right\} \geq d_{\min}/2\right)$ and $\Pr\left(\Im\left\{v\right\} \geq d_{\min}/2\right)$, the respective other component is ignored, such that some possible outcomes of v are counted twice. This corresponds to the hatched area in Figure D.3, where the probabilities are doubly integrated.

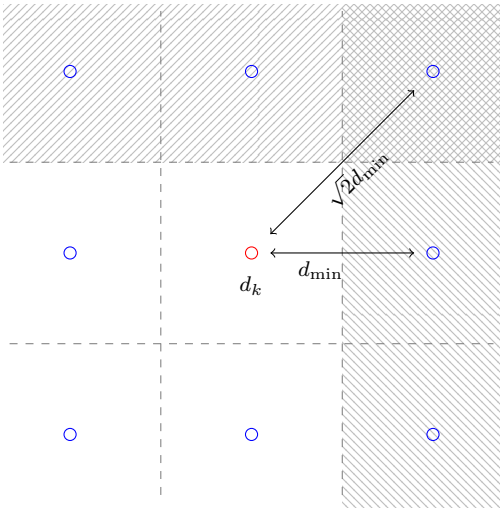


Figure D.3.: Error regions in QAM constellations.

However, the probability contribution at this distance has strongly decayed already and only minor influence on the final result. In Figure D.4 the result of the Q-function is displayed for rather high noise with variance $\sigma_v^2 = 0.1$, where the doubly charged part at $\sqrt{2}d_{\min}$ is more than one order of magnitude smaller than the relevant probability and can be neglected. Still this is a source of a small uncertainty about the actual symbol error probability.

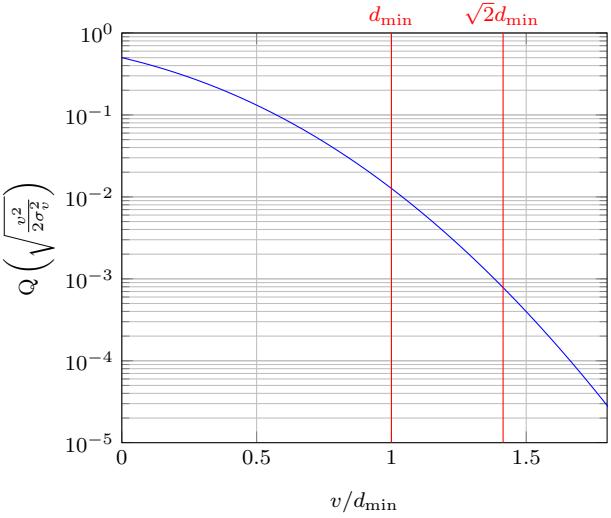


Figure D.4.: Q-function for Gaussian noise with variance $\sigma_v^2 = 0.1$.

While this derivation is true for QAM constellation points with four direct neighbors only, it is clear that practical QAM constellations have most points at an edge or corner. If all constellation points are transmitted equally probable, the average number of direct neighbors in minimum distance $N_{\min}$ can be determined for every constellation. For 4-QAM this is clearly $N_{\min} = 2$, since all points are in corners, for other important constellations this value is shown in Table D.1.

R	$M = 2^R$	$N_{\min}$
1	2	1.0
2	4	2.0
4	16	3.0
6	64	3.5
8	256	3.75
10	1024	3.875
12	4096	3.9375

Table D.1.: Properties of constellations with M points and R bits per symbol.

Hence, this section can be concluded with the analytical expression for determining the symbol error probability

$$\chi = N_{\min} \mathrm{Q}\left(\sqrt{\frac{d_{\min}^2}{2\sigma_v^2}}\right). \tag{D.12}$$

List of Abbreviations

ASK	amplitude shift keying
BER	bit error ratio
BICM	bit-interleaved coded modulation
BLUE	best linear unbiased estimator
CI	channel inversion
CIR	channel impulse response
CME	complex multiplication equivalents
CP	cyclic prefix
DFE	decision feedback equalization
DFT	discrete Fourier transform
ECB	equivalent complex baseband
FFT	fast Fourier transform
IDFT	inverse discrete Fourier transform
iid	independent and identically distributed
ISI	inter-symbol inteference
LLR	log-likelihood ratio
LMMSE	linear MMSE
MGS	modified Gram-Schmidt orthogonalization
MIMO	multiple input multiple output
ML	maximum-likelihood
MLSE	maximum-likelihood sequence estimation
MMSE	minimum mean square error
MVU	minimum variance unbiased
NI	noise interpolation
OFDM	orthogonal frequency division multiplexing
pdf	probability density function
PSA	post sorting algorithm
PSK	phase shift keying

QAM	quadrature amplitude modulation
SD	sphere decoding
SER	symbol error ratio
SIC	successive interference cancellation
SNR	signal-to-noise ratio
SQRD	sorted QR decomposition
ssc	symbol scaling compensation
TDW	time domain windowing
UW	unique word
WLMMSE	widely linear MMSE
ZF	zero forcing

Bibliography

[AA08] Luay Azzam and Ender Ayanoglu. Reduction of ML decoding complexity for MIMO Sphere Decoding, QOSTBC, and OSTBC. In *Proceedings of the Information Theory and Applications Workshop (ITA)*, pages 18–25, San Diego, CA, USA, 2008.

[AEVZ02] Erik Agrell, Thomas Eriksson, Alexander Vardy, and Kenneth Zeger. Closest Point Search in Lattices. *IEEE Transactions on Information Theory*, 48(8):2201–2214, August 2002.

[ANS95] Network and Customer Installation Interfaces — Asymmetric Digital Subscriber Line (ADSL) Metallic Interface. ANSI standard T1.413-1995, 1995.

[ASS11] Tülay Adalı, Peter J. Schreier, and Louis L. Scharf. Complex-Valued Signal Processing: The Proper Way to Deal with Impropriety. *IEEE Transactions on Signal Processing*, 59(11):5101–5125, November 2011.

[BBW+05] Andreas Burg, Moritz Borgmann, Markus Wenk, Martin Zellweger, Wolfgang Fichtner, and Helmut Bölcskei. VLSI Implementation of MIMO Detection using the Sphere Decoding Algorithm. *IEEE Journal of Solid-State Circuits*, 40(7):1566–1577, July 2005.

[Bla03] Richard E. Blahut. *Algebraic Codes for Data Transmission*. Cambridge University Press, 1st edition, March 2003.

[BWKK03] Ronald Böhnke, Dirk Wübben, Volker Kühn, and Karl-Dirk Kammeyer. Reduced Complexity MMSE Detection for BLAST Architectures. In *Proceedings of the IEEE Global Telecommunications Conference, (GLOBECOM)*, volume 4, pages 2258–2262, San Francisco, CA, USA, December 2003.

[CCC00] Won-Joon Choi, Kok-Wui Cheong, and John M. Cioffi. Iterative Soft Interference Cancellation for Multiple Antenna Systems. In *Proceedings of the Wireless Communications and Networking Confernce (WCNC)*, volume 1, pages 304–309, Chicago, IL, USA, 2000.

[CDCY11] Todor Cooklev, Hakan Doğan, Renato J. Cintra, and Hakan Yıldız. A Generalized Prefix Construction for OFDM Systems Over Quasi-Static Channels. *IEEE Transactions on Vehicular Technology*, 60(8):3684–3693, October 2011.

[CG68] Robert W. Chang and Richard A. Gibby. A Theoretical Study of Performance of an Orthogonal Multiplexing Data Transmission Scheme. *IEEE Transactions on Communication Technology*, 16(4):529–540, August 1968.

[Cha66] Robert W. Chang. Synthesis of Band-Limited Orthogonal Signals for Multichannel Data Transmission. *Bell System Technical Journal*, 45:1775–1796, December 1966.

[CM01] Raphael Cendrillon and Marc Moonen. Efficient Equalizers for Single and Multi-Carrier Environments with Known Symbol Padding. In *Proceedings of the 6th International Symposium on Signal Processing and its Applications*, volume 2, pages 607–610, Kuala Lumpur, Malaysia, August 2001.

[CNS06] Framing Structure, Channel Coding and Modulation for Digital Television Terrestrial Broadcasting System. Chinese National Standard, Std. GB 20 600-2006, 2006.

[Coo12] Todor Cooklev. Improved Prefix for OFDM-Based Cognitive Radios. *IET Electronics Letters*, 48(4):240–241, February 2012.

[CTB98] Giuseppe Caire, Giorgio Taricco, and Ezio Biglieri. Bit-Interleaved Coded Modulation. *IEEE Transactions on Information Theory*, 44(3):927–946, May 1998.

[CWK06] Donghua Chen, Mei Wang, and Quan Kuang. Doubly Iterative Soft Interference Cancellation for MIMO-OFDM Systems. In *Proceedings of the International Conference on Communications, Circuits and Systems*, volume 2, pages 645–649, Guilin, China, June 2006.

[Czy97] Andreas Czylwik. Comparison Between Adaptive OFDM and Single Carrier Modulation with Frequency Domain Equalization. In *Proceedings of the 47th IEEE Vehicular Technology Conference (VTC Spring)*, volume 2, pages 865–869, Phoenix, AZ, USA, May 1997.

[DCB00] Oussama Damen, Ammar Chkeif, and Jean-Claude Belfiore. Lattice Code Decoder for Space-Time Codes. *IEEE Communications Letters*, 4(5):161–163, May 2000.

[DGC03] Mohamed Oussama Damen, Hesham El Gamal, and Guiseppe Caire. On Maximum-Likelihood Detection and the Search for the Closest Lattice Point. *IEEE Transactions on Information Theory*, 49(10):2389–2402, October 2003.

[DGE01] Luc Deneire, Bert Gyselinckx, and Marc Engels. Training Sequence versus Cyclic Prefix—A New Look on Single Carrier Communication. *IEEE Communications Letters*, 5(7):292–294, July 2001.

[DPS11] Erik Dahlman, Stefan Parkvall, and Johan Skold. *4G: LTE/LTE-Advanced for Mobile Broadband: LTE/LTE-Advanced for Mobile Broadband*. Elsevier Science, 2011.

[DWC11] Linglong Dai, Zhaocheng Wang, and Sheng Chen. A Novel Uplink Multiple Access Scheme Based on TDS-FDMA. *IEEE Transactions on Wireless Communications*, 10(3):757–761, March 2011.

[Eur95] European Telecommunications Standards Institute (ETSI). Radio Broadcasting Systems; Digital Audio Broadcasting (DAB) to Mobile, Portable, and Fixed Receivers. European Telecommunication Standard ETS 300 401, February 1995. 1st edition, reference DE/JTC-DAB.

[Eur97] European Telecommunications Standards Institute (ETSI). Digital Video Broadcasting (DVB); Framing Structure, Channel Coding and Modulation for Digital Terrestrial Television. European Telecommunication Standard ETS 300 744, March 1997. 1st edition, reference DE/JTC-DVB-8.

[Fak97] John Fakatselis. *Criteria for 2.4 GHz PHY Comparison of Modulation*. IEEE Document, 1997. P802.11-97/157r1.

[Fis02] Robert F. H. Fischer. *Precoding and Signal Shaping for Digital Transmission*. John Wiley & Sons, New York, 2002.

[Fis10] Robert F. H. Fischer. From Gram-Schmidt Orthogonalization via Sorting and Quantization to Lattice Reduction. In *Proceedings of the Joint Workshop on Coding and Communications (JWCC)*, Santo Stefano Belbo, Italy, October 2010. Invited Talk.

[Fis12] Robert F. H. Fischer. Complexity-Performance Trade-Off of Algorithms for Combined Lattice Reduction and QR Decomposition. *AEU - International Journal of Electronics and Communications*, 66(11):871–879, November 2012.

[Fos96] Gerard J. Foschini. Layered Space-Time Architecture for Wireless Communication in a Fading Environment When Using Multi-Element Antennas. *Bell Labs Technical Journal*, 1(2):41–59, 1996.

[FP85] Ulrich Fincke and Michael Pohst. Improved Methods for Calculating Vectors of Short Length in a Lattice, Including a Complexity Analysis. *Mathematics of Computation*, 44:463–463, 1985.

[GL96] Gene H. Golub and Charles F. Van Loan. *Matrix Computations*. Johns Hopkins Studies in the Mathematical Sciences. Johns Hopkins University Press, 1996.

[GL08] Stefano Galli and Oleg Logvinov. Recent Developments in the Standardization of Power Line Communications within the IEEE. *IEEE Communications Magazine*, 46(7):64–71, July 2008.

[GRS01] Bernd Girod, Rudolf Rabenstein, and Alexander Stenger. *Signals & Systems*. John Wiley & Sons, 1st edition, 2001.

[GS65] Solomon W. Golomb and Robert A. Scholtz. Generalized Barker Sequences. *IEEE Transactions on Information Theory*, 11(4):533–537, October 1965.

[GSL03] Wolfgang H. Gerstacker, Robert Schober, and Alexander Lampe. Receivers with Widely Linear Processing for Frequency-Selective Channels. *IEEE Transactions on Communications*, 51(9):1512–1523, September 2003.

[Has00] Babak Hassibi. An Efficient Square-Root Algorithm for BLAST. In *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, volume 2, pages II737–II740, Istanbul, Turkey, 2000.

[Hay96] Simon Haykin. *Adaptive Filter Theory*. Prentice-Hall, Upper Saddle River, NJ, USA, 3rd edition, 1996.

[HH12] Christian Hofbauer and Mario Huemer. A Study of Data Rate Equivalent UW-OFDM and CP-OFDM Concepts. In *Conference Record of the 46th Asilomar Conference on Signals, Systems and Computers*, pages 173–177, Pacific Grove, CA, USA, November 2012.

[HHH10a] Christian Hofbauer, Mario Huemer, and Johannes B. Huber. Coded OFDM by Unique Word Prefix. In *Proceedings of the IEEE International Conference on Communication Systems (ICCS)*, pages 426–430, Singapore, November 2010.

[HHH10b] Christian Hofbauer, Mario Huemer, and Johannes B. Huber. On the Impact of Redundant Subcarrier Energy Optimization in UW-OFDM. In *Proceedings of the 4th International Conference on Signal Processing and Communication Systems (ICSPCS)*, page 6, Gold Coast, Australia, December 2010.

[HHH10c] Mario Huemer, Christian Hofbauer, and Johannes B. Huber. The Potential of Unique Words in OFDM. In *Proceedings of the 15th International OFDM Workshop*, pages 140–144, Hamburg, September 2010.

[HHH10d] Mario Huemer, Christian Hofbauer, and Johannes B. Huber. Unique Word Prefix in SC/FDE and OFDM: A Comparison. In *Proceedings of the IEEE GLOBECOM Workshops*, pages 1296–1301, Miami, FL, USA, December 2010.

[HHH12] Mario Huemer, Christian Hofbauer, and Johannes B. Huber. Non-Systematic Complex Number RS Coded OFDM by Unique Word Prefix. *IEEE Transactions on Signal Processing*, 60(1):285–299, January 2012.

[HHK09] Werner Henkel, Fangning Hu, and Ina Kodrasi. Inherent Time-Frequency Coding in OFDM and ISI Correction without a Cyclic Prefix. In *Proceedings of the 14th International OFDM Workshop*, Hamburg, Germany, September 2009.

[HHOH12] Mario Huemer, Christian Hofbauer, Alexander Onic, and Johannes B. Huber. On the Exploitation of the Redundant Energy in UW-OFDM: LMMSE Versus Sphere Detection. *IEEE Signal Processing Letters*, 19(6):340–343, June 2012.

[Hir81] Botaro Hirosaki. An Orthogonally Multiplexed QAM System Using the Discrete Fourier Transform. *IEEE Transactions on Communications*, 29(7):982–989, July 1981.

[HJ90] Roger A. Horn and Charles R. Johnson. *Matrix Analysis*. Cambridge University Press, 1st edition, 1990.

[HKWR03] Mario Huemer, Alois Koppler, Robert Weigel, and Leonhard Reindl. A Review of Cyclically Extended Single Carrier Transmission with Frequency Domain Equalization for Broadband Wireless Transmission. *European Transactions on Telecommunications*, 14(4):329–341, 2003.

[HL89] Johannes B. Huber and Weiling Liu. An Alternative Approach to Reduced-Complexity CPM-Receivers. *IEEE Journal on Selected Areas in Communications*, SAC-7:1437–1449, December 1989.

[HM84] Michael L. Honig and David G. Messerschmitt. *Adaptive Filters: Structures, Algorithms and Applications*. Kluwer Academic Publishers, 1984.

[Hof] Christian Hofbauer. *Design and Analysis of Unique Word OFDM*. Dissertation in preparation, Institute of Networked and Embedded Systems, Alpen-Adria-Universität Klagenfurt.

[HOH11] Mario Huemer, Alexander Onic, and Christian Hofbauer. Classical and Bayesian Linear Data Estimators for Unique Word OFDM. *IEEE Transactions on Signal Processing*, 59(12):6073–6085, December 2011.

[HtB03] Bertrand N. Hochwald and Stephan ten Brink. Achieving Near-Capacity on a Multiple-Antenna Channel. *IEEE Transactions on Communications*, 51(3):389–399, March 2003.

[HTO+02] Werner Henkel, Georg Tauböck, Per Ödling, Per O. Börjesson, and Niklas Petersson. The Cyclic Prefix of OFDM/DMT - An Analysis. In *Proceedings of the International Zurich Seminar on Broadband Communications*, pages 22–1 – 22–3, Zürich, Switzerland, February 2002.

[Hub02] Johannes B. Huber. Grundlagen der Wahrscheinlichkeitsrechnung für iterative Decodierverfahren. *Elektrotechnik und Informationstechnik*, 119(11):386ff, November 2002.

[HV05] Babak Hassibi and Haris Vikalo. On the Sphere-Decoding Algorithm I. Expected Complexity. *IEEE Transactions on Signal Processing*, 53(8):2806–2818, 2005.

[HWH03] Mario Huemer, Harald Witschnig, and Josef Hausner. Unique Word Based Phase Tracking Algorithms for SC/FDE-Systems. In *Proceedings of the IEEE Global Telecommunications Conference (GLOBECOM)*, volume 1, pages 70–74, San Francisco, CA, USA, December 2003.

[IEE99] IEEE Std 802.11a-1999, Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) specifications: High-Speed Physical Layer in the 5 GHz Band, 1999.

[IEE04] IEEE Std 802.16-2004, Part 16: Air Interface for Fixed Broadband Wireless Access Systems, 2004.

[IEE06] ISO/IEC 8802-11:2005/AMD4 [IEEE Std 802.11g-2003] Information technology– Local and metropolitan area networks– Part 11: Wireless LAN Medium Access Control (Mac) and Physical Layer (PHY) Specifications– Amendment 4: Further Higher Data Rate Extension in the 2.4 GHz Band, 2006.

[IEE12] IEEE Standard for Information Technology–Telecommunications and Information Exchange between Systems; Local and Metropolitan Area Networks– Specific Requirements Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications, 2012.

[Jan04] Mohinder Jankiraman. *Space-Time Codes and MIMO Systems*. Artech House Universal Personal Communications Series. Artech House, 2004.

[JHJR02] Liao Jingyi, Wang Hai, Panyuh Joo, and Jungmin Ro. The Effect of Filling Unique Words to Guard Interval for OFDM System. C802.16a-02/87, IEEE 802.16 Broadband Wireless Access Working Group, September 2002.

[JO05] Joakim Jaldèn and Björn Ottersten. Parallel Implementation of a Soft Output Sphere Decoder. In *Conference Record of the 39th Asilomar Conference on Signals, Systems and Computers*, pages 581–585, Pacific Grove, CA, USA, November 2005.

[Kad97] Gerhard Kadel. Diversity and Equalization in Frequency Domain – A Robust and Flexible Receiver Technology for Broadband Mobile Communication Systems. In *Proceedings of the 47th IEEE Vehicular Technology Conference (VTC Spring)*, volume 2, pages 894–898, Phoenix, AZ, USA, May 1997.

[Kam04] Karl-Dirk Kammeyer. *Nachrichtenübertragung*. Vieweg+Teubner, Reihe Informations-/Kommunikationstechnik, Wiesbaden, Germany, 3rd edition, November 2004.

[Kay93] Steven M. Kay. *Fundamentals of Statistical Signal Processing, Volume I: Estimation Theory (v. 1)*. Prentice Hall, 1st edition, April 1993.

[Knu05] Donald E. Knuth. *The Art of Computer Programming, Volume 4, Fascicle 2: Generating All Tuples and Permutations*. The Art of Computer Programming. Addison-Wesley Professional, 1st edition, February 2005.

[LPV10] Yuan-Pei Lin, See-May Phoong, and P. P. Vaidyanathan. *Filter Bank Transceivers for OFDM and DMT Systems*. Cambridge University Press, 2010.

[MdCD06] Markus Muck, Marc de Courville, and Pierre Duhamel. A Pseudorandom Postfix OFDM Modulator—Semi-Blind Channel Estimation and Equalization. *IEEE Transactions on Signal Processing*, 54(3):1005–1017, March 2006.

[MF04] Zbigniew Michalewicz and David B. Fogel. *How to Solve It: Modern Heuristics*. Springer, enlarged 2nd edition, December 2004.

[MMM09] Geert Van Meerbergen, Marc Moonen, and Hugo De Man. Reed-Solomon Codes Implementing a Coded Single-Carrier with Cyclic Prefix Scheme. *IEEE Transactions on Communications*, 57(4):1031–1038, April 2009.

[Mow94] Wai-Ho Mow. Maximum Likelihood Sequence Estimation from the Lattice Viewpoint. *IEEE Transactions on Information Theory*, 40(5):1591–1600, September 1994.

[NM93] Fredy D. Neeser and James L. Massey. Proper Complex Random Processes with Applications to Information Theory. *IEEE Transactions on Information Theory*, 39(4):1293–1302, July 1993.

[NP00] Richard van Nee and Ramjee Prasad. *OFDM for Wireless Multimedia Communications*. Artech House, Inc., Norwood, MA, USA, 1st edition, 2000.

[OH10] Alexander Onic and Mario Huemer. Direct vs. Two-Step Approach for Unique Word Generation in UW-OFDM. In *Proceedings of the 15th International OFDM Workshop*, pages 145–149, Hamburg, September 2010.

[OH11a] Alexander Onic and Mario Huemer. Limiting the Complexity of Sphere Decoding for UW-OFDM. In *Proceedings of the 16th International OFDM Workshop*, pages 135–139, Hamburg, September 2011.

[OH11b] Alexander Onic and Mario Huemer. Sphere Decoding for Unique Word OFDM. In *Proceedings of the IEEE Global Telecommunications Conference (GLOBECOM)*, Houston, TX, USA, December 2011.

[Orf88] Sophocles J. Orfanidis. *Optimum Signal Processing: An Introduction*. Macmillan, 1988.

[OSHH12] Alexander Onic, Andreas Schenk, Mario Huemer, and Johannes B. Huber. Soft-Output Sphere Detection for Coded Unique Word OFDM. In *Conference Record of the 46th Asilomar Conference on Signals, Systems and Computers*, pages 138–142, Asilomar, CA, USA, November 2012.

[PC95] Bernard Picinbono and Pascal Chevalier. Widely Linear Estimation with Complex Data. *IEEE Transactions on Signal Processing*, 43(8):2030–2033, August 1995.

[Poh81] Michael Pohst. On the Computation of Lattice Vectors of Minimal Length, Successive Minima and Reduced Bases with Applications. *SIGSAM Bull.*, 15(1):37–44, February 1981.

[Pop92] Branislav M. Popovic̀. Generalized Chirp-Like Polyphase Sequences with Optimum Correlation Properties. *IEEE Transactions on Information Theory*, 38(4):1406–1409, July 1992.

[PR80] Abraham Peled and Antonio Ruiz. Frequency Domain Data Transmission using Reduced Computational Complexity Algorithms. In *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, volume 5, pages 964–967, Denver, CO, USA, April 1980.

[Pro00] John Proakis. *Digital Communications*. McGraw-Hill Science/Engineering/Math, 4th edition, August 2000.

[RBH05] Steffen Reinhardt, Tufik Buzid, and Mario Huemer. MIMO Extensions for SC/FDE Systems. In *Proceedings of the European Conference on Wireless Technology*, pages 109–112, Paris, France, October 2005.

[Ret10] Jakob Rettelbach. *Detektionsverfahren für Unique-Word OFDM*. Studienarbeit, Lehrstuhl für Informationsübertragung, Friedrich-Alexander-Universität Erlangen-Nürnberg, Germany, September 2010.

[RH84] Christian F. Rüdiger and Karl F. Hindenburg. *Specimen Analyticum de Lineis Curvis*. 1784. `http://books.google.at/books?id=QYY_AAAAcAAJ`, last access in November 2013.

[RH13] Jakob Rettelbach and Johannes B. Huber. private communications. 2013.

[RVH95] Patrick Robertson, Emmanuelle Villebrun, and Peter Hoeher. A Comparison of Optimal and Sub-Optimal MAP Decoding Algorithms Operating in the Log Domain. In *Proceedings of the IEEE International Conference on Communications (ICC)*, volume 2, pages 1009–1013, Seattle, June 1995.

[Sal67] Burton R. Saltzberg. Performance of an Efficient Parallel Data Transmission System. *IEEE Transactions on Communication Technology*, 15(6):805–811, December 1967.

[SBB08] Christoph Studer, Andreas Burg, and Helmut Bölcskei. Soft-Output Sphere Decoding: Algorithms and VLSI Implementation. *IEEE Journal on Selected Areas in Communications*, 26(2):290–300, February 2008.

[Sch] Christoph Schwab. Numerische Methoden. Lecture Notes (in German), ETH Zürich, 2006.

[SE94] Claus-Peter Schnorr and M. Euchner. Lattice Basis Reduction: Improved Practical Algorithms and Solving Subset Sum Problems. *Mathematical Programming*, 66(1-3):181–199, 1994.

[SF10] Andreas Schenk and Robert F. H. Fischer. Soft-Output Sphere Decoder for Multiple-Symbol Differential Detection of Impulse-Radio Ultra-Wideband. In *Proceedings of the IEEE International Symposium on Information Theory (ISIT)*, pages 2258–2262, Austin, TX, USA, June 2010.

[SKJ94] Hikmet Sari, Georges Karam, and I. Jeanclaud. Frequency-Domain Equalization of Mobile Radio and Terrestrial Broadcast Channels. In *Proceedings of the IEEE Global Telecommunications Conference (GLOBECOM)*, San Francisco, CA, USA, November 1994.

[SM07] Heidi Steendam and Marc Moeneclaey. Different Guard Interval Techniques for OFDM: Performance Comparison. In Simon Plass, Armin Dammann, Stefan Kaiser, and Khaled Fazel, editors, *Multi-Carrier Spread Spectrum 2007*, volume 1 of *Lecture Notes Electrical Engineering*, pages 11–24. Springer Netherlands, 2007.

[Ste12] Heidi Steendam. Analysis of the Redundant Energy in UW-OFDM. *IEEE Transactions on Communications*, 60(6):1692–1701, June 2012.

[Ste13] Heidi Steendam. On the Selection of the Redundant Carrier Positions in UW-OFDM. *IEEE Transactions on Signal Processing*, 61(5):1112–1120, March 2013.

[Tec09] Agilent Technologies. *LTE and the Evolution to 4G Wireless: Design and Measurement Challenges*. Agilent Technologies, 2009.

[Tra13] Stefan Trampitsch. *Complex-Valued Data Estimation – Second-Order Statistics and Widely Linear Estimators*. Master thesis, Institute of Networked and Embedded Systems, Alpen-Adria-Universität Klagenfurt, Austria, March 2013.

[TYP+07] Shigang Tang, Fang Yang, Kewu Peng, Changyong Pan, Ke Gong, and Zhixing Yang. Iterative Channel Estimation for Block Transmission with Known Symbol Padding — A New Look at TDS-OFDM. In *Proceedings of the IEEE Global Telecommunications Conference (GLOBECOM)*, pages 4269–4273, Washington, DC, USA, November 2007.

[VB99] Emanuele Viterbo and Joseph Boutros. A Universal Lattice Code Decoder for Fading Channels. *IEEE Transactions on Information Theory*, 45(5):1639–1642, July 1999.

[VH05] Haris Vikalo and Babak Hassibi. On the Sphere-Decoding Algorithm II. Generalizations, Second-Order Statistics, and Applications to Communications. *IEEE Transactions on Signal Processing*, 53(8):2819–2834, 2005.

[WBKK03] Dirk Wübben, Ronald Böhnke, Volker Kühn, and Karl-Dirk Kammeyer. MMSE Extension of V-BLAST based on Sorted QR Decomposition. In *Proceedings of the 58th IEEE Vehicular Technology Conference (VTC Fall)*, volume 1, pages 508–512, Orlando, FL, USA, October 2003.

[WBR+01] Dirk Wübben, Ronald Böhnke, Jürgen Rinas, Volker Kühn, and Karl-Dirk Kammeyer. Efficient Algorithm for Decoding Layered Space-Time Codes. *IET Electronics Letters*, 37(22):1348–1350, October 2001.

[WE71] Stephen B. Weinstein and Paul M. Ebert. Data Transmission by Frequency-Division Multiplexing Using the Discrete Fourier Transform. *IEEE Transactions on Communication Technology*, 19(5):628–634, October 1971.

[Web12] UW-OFDM: Code Examples for Matlab, 2012. `http://uwofdm.jku.at`, last access in November 2013.

[Wei09] Stephen B. Weinstein. The History of Orthogonal Frequency-Division Multiplexing. *IEEE Communications Magazine*, 47(11):26–35, November 2009.

[WFGV98] Peter W. Wolniansky, Gerard J. Foschini, Glen D. Golden, and Reinaldo A. Valenzuela. V-BLAST: An Architecture for Realizing Very High Data Rates Over the Rich-Scattering Wireless Channel. In *Proceedings of the URSI International Symposium on Signals, Systems, and Electronics, (ISSSE)*, pages 295–300, Pisa, Italy, September 1998.

[Wie49] Norbert Wiener. *Extrapolation, Interpolation and Smoothing of Stationary Time Series with Engineering Applications*. Technology Press and John Wiley & Sons, Inc., New York, 1949.

[Win04] Christoph Windpassinger. *Detection and Precoding for Multiple Input Multiple Output Channels*. Dissertation, Technische Fakultät, Friedrich-Alexander-Universität Erlangen-Nürnberg, January 2004.

[Wit04] Harald Witschnig. *Frequency Domain Equalization for Broadband Wireless Communication — With Special Reference to Single Carrier Transmission Based on Known Pilot Sequences*. Dissertation, Institute for Communications and Information Engineering, University of Linz, 2004.

[WLL11] Xianbin Wang, Hao Li, and Hai Lin. A New Adaptive OFDM System with Precoded Cyclic Prefix for Dynamic Cognitive Radio Communications. *IEEE Journal on Selected Areas in Communications*, 29(2):431–442, February 2011.

[WMS+02a] Harald Witschnig, Thomas Mayer, Andreas Springer, A. Koppler, Linus Maurer, Mario Huemer, and Robert Weigel. A Different Look on Cyclic Prefix for SC/FDE. In *Proceedings of the 13th IEEE International Symposium on Personal, Indoor and Mobile Radio Communications*, volume 2, pages 824–828, Lisbon, Portugal, September 2002.

[WMS+02b] Harald Witschnig, Thomas Mayer, Andreas Springer, Linus Maurer, Mario Huemer, and Robert Weigel. The Advantages of a Known Sequence versus Cyclic Prefix in a SC/FDE System. In *Proceedings of the 5th International Symposium on Wireless Personal Multimedia Communications*, volume 3, pages 1328–1332, Honolulu, HI, USA, October 2002.

[WNH+06] Xinglin Wang, Kai Niu, Zhiqiang He, Weiling Wu, and Xin Zhang. List Sphere Decoding Combined with Linear Detection-Based Iterative Soft Interference Cancellation Via EXIT Chart. In *Proceedings of the 17th IEEE International Symposium on Personal, Indoor and Mobile Radio Communications*, Helsinki, Finland, September 2006.

[WRB+02] Dirk Wübben, Jürgen Rinas, Ronald Böhnke, Volker Kühn, and Karl-Dirk Kammeyer. Efficient Algorithm for Detecting Layered Space-Time Codes. In *Proceedings of the 4th International ITG Conference on Source and Channel Coding (SCC)*, Berlin, Germany, January 2002.

[WS10] Dieter Van Welden and Heidi Steendam. Near Optimal Iterative Channel Estimation for KSP-OFDM. *IEEE Transactions on Signal Processing*, 58(9):4948–4954, September 2010.

[WSM08] Dieter Van Welden, Heidi Steendam, and Marc Moeneclaey. Iterative DA/DD Channel Estimation for KSP-OFDM. In *Proceedings of the IEEE International Conference on Communications (ICC)*, pages 693–697, Beijing, China, May 2008.

[Zak05] Anthony Zaknich. *Principles of Adaptive Filters and Self-learning Systems*. Advanced Textbooks in Control and Signal Processing. Springer-Verlag London Limited, 2005.