Diplomarbeit

Iterative Channel Estimation for UMTS Long Term Evolution

Ausgeführt zum Zwecke der Erlangung des akademischen Grades eines Diplom-Ingenieurs unter Leitung von

Univ. Prof. Dipl.-Ing. Dr.techn. Markus Rupp
Dipl.-Ing. Michal Šimko

E389
Institute of Telecommunications

eingereicht an der Technischen Universität Wien
Fakultät für Elektrotechnik und Informationstechnik

von

Florent Kadrija
Matrikelnr.: 0628074
Hausgrundweg 1/11, 1220 Wien, Österreich

Vienna, November 2012 . . . . . .

Abstract

In this thesis, different iterative channel estimation algorithms for Long Term Evolution (LTE) downlink are investigated. LTE uses coherent detection, which requires channel state information. In order to achieve high data rate transmission over mobile radio channels, it is essential to have accurate channel state information at the receiver side. For channel estimation purpose LTE provides training data known as pilot symbols. The matter of discussion is whether the accuracy of channel estimate based on pilot symbols is satisfactorily sufficient to achieve high data rate transmission. Channel estimate can be further enhanced, if after pilot based channel estimation, additional information such as the hard or soft estimated data symbols from the decoder is utilized by the channel estimator. Using this additional information, different channel estimation algorithms are derived. Their performance is discussed and compared with each other. The impact of processing either extrinsic, a-posteriori or hard feedback information in the channel estimator is investigated. To assess the performance of channel estimators, we exploit the LTE Link Level Simulator, developed at the Institute of Telecommunications (TC), Vienna University of Technology. Channel estimators are compared in terms of Mean Square Error (MSE) and throughput for slowly changing channels. For the SISO (4×4 MIMO) transmission mode using soft feedback information, iterative Least Squares (LS) channel estimator improves about 0.8 dB (0.45 dB), and iterative Linear Minimum Mean Square Error (LMMSE) channel estimator about 0.4 dB (0.7 dB) with respect to the initial channel estimators. The iterative LMMSE estimator loses approximately 0.05 dB with respect to the system with perfect channel knowledge. Although, the performance of iterative LMMSE estimator is superb, its complexity is too high for a real-time implementation. In order to reduce the complexity, meanwhile preserve the performance of the iterative LMMSE estimator, iterative approximate LMMSE (ALMMSE) estimator is investigated. The iterative ALMMSE estimator uses the correlation between the L closest subcarriers. Variation of L allows us to adjust the performance and complexity of the estimator to attain a good trade-off. Accordingly, iterative ALMMSE estimator, for a chosen $L = 12$, gains about 0.75 dB (0.9 dB) with respect to initial ALMMSE estimator, and loses about 0.2 dB (0.3 dB) compared to iterative LMMSE estimator.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

The first mobile communication systems were introduced in the early 1980s, which were developed as independent systems worldwide [1]. The new technology introduced with the Global System for Mobile communications (GSM) in 1991 enabled global roaming. This allowed people, while travelling from one country to another, to have disconnected calls. The successors of GSM, like Universal Mobile Telecommunications System (UMTS) and High Speed Packet Access (HSPA) increased the data transmission rate and allowed people to use internet and browse on web pages from everywhere, make video calls and use different mobile applications. These data intensive applications provide exciting possibilities for customers but also create new bandwidth delivery challenges, since the radio frequency spectrum remains a finite resource [2]. Therefore, standardisation organizations have developed new technologies such as Long Term Evolution (LTE) to allow higher throughput utilizing the existing bandwidth. LTE, as the successor of the GSM and UMTS, increases further the data rate transmission using new modulation techniques and offering scalable bandwidth supporting more than 300 Mbit/s in downlink and up to 75 Mbit/s in uplink peak data rates [3].

To achieve high data rate transmission over mobile radio channels accurate channel state information at the receiver side is necessary. LTE uses coherent detection; therefore, accurate knowledge of channel state information is essential to the detector before detecting the data symbols [4]. Many investigations and plenty of work have been conducted on channel estimation in LTE [3]. The majority of these investigations are based on non-iterative channel estimation algorithms, where only known pilot symbols, provided by the LTE physical layer, are utilized for channel estimation.

In this thesis, iterative channel estimation for LTE downlink is investigated. Iterative channel estimators utilize both pilot symbols and feedback information on data symbols gained from decoder [5], [6]. Iterative approaches are investigated for Single Input Single Output (SISO) and Multiple Input Multiple Output (MIMO) transmission modes. It is assumed that the channel changes slowly, thus the block fading assumption is fulfilled. Furthermore, the impact of processing a-posteriori or extrinsic soft information, and hard information in channel estimation is investigated. The estimators are compared with each other in terms of Mean Square Error (MSE), which shows the performance of the estimator, and in terms of throughput, which shows the influence of the estimator on the entire system.

The thesis is organized as follows:

- In Chapter 2, the important facts about LTE downlink, from channel estimation point of view, are presented. LTE physical layer and the structure of LTE reference signals are discussed.
- In Chapter 3, transmitter is discussed and the new iterative receiver structure is investigated. Then, the functionality of the iterative receiver components is briefly discussed. Furthermore, the soft symbol mapper unit is analysed and finally, a mathematical signal model is introduced.
- In Chapter 4, different iterative channel estimation algorithms are presented and their complexity is discussed.
- In Chapter 5, the performance of the iterative channel estimation algorithms based on simulation results is analysed.
- Finally, Chapter 6 concludes this thesis with a summary of most important points gained in this work.

Chapter 2

LTE Downlink

In this chapter, we discuss the LTE downlink characteristics which are important from channel estimation point of view. A short description of LTE physical layer and the reference signal structure is given.

2.1 LTE Physical Layer

LTE is a project within the 3rd Generation Partnership Project (3GPP), which is introduced in Release 8 [7]. 3GPP developed formerly the 2nd Generation: GSM, General Packet Radio Service (GPRS) and Enhanced Data rates for GSM Evolution (EDGE), which are based on Time and Frequency Division Multiple Access (TDMA/FDMA) schemes, and the 3rd Generation: UMTS and HSPA, which employ Wideband Code Division Multiple Access (W-CDMA) as a transmission scheme [1]. On the other hand, LTE as the newest mobile technology employs Orthogonal Frequency Division Multiplexing (OFDM) for downlink data transmission and supports the use of MIMO and different modulation and coding schemes.

In LTE, the downlink transmission is organized into frames of duration 10 ms. The structure of the LTE frame is depicted in Figure 2.1 [7]. Each frame is further subdivided into ten equally long subframes of duration 1 ms. Each subframe consists of two equally long slots of duration 0.5 ms. Furthermore, each slot can be represented as a rectangular resource grid of dimension $N_K \times N_S$, where N_K is the number of subcarriers used for transmission, and N_S is the number of OFDM symbols. The number of OFDM symbols comprising each slot is equal to 7 in case of normal cyclic prefix or 6 in case of extended cyclic prefix. Throughout this thesis normal cyclic prefix is considered. The smallest unit composing the resource grid is the resource element,

which is identified by the subcarrier index n_k and the OFDM symbol index n_s. Subcarriers are then grouped into resource blocks, where each resource block consists of 12 adjacent subcarriers with 15 kHz spacing between two consecutive subcarriers. Otherwise, a grouping of resource blocks along the frequency dimension defines a slot. LTE supports scalable bandwidths, from 1.4 MHz up to 20 MHz [7]. Figure 2.1 depicts the LTE frame structure for normal cyclic prefix.

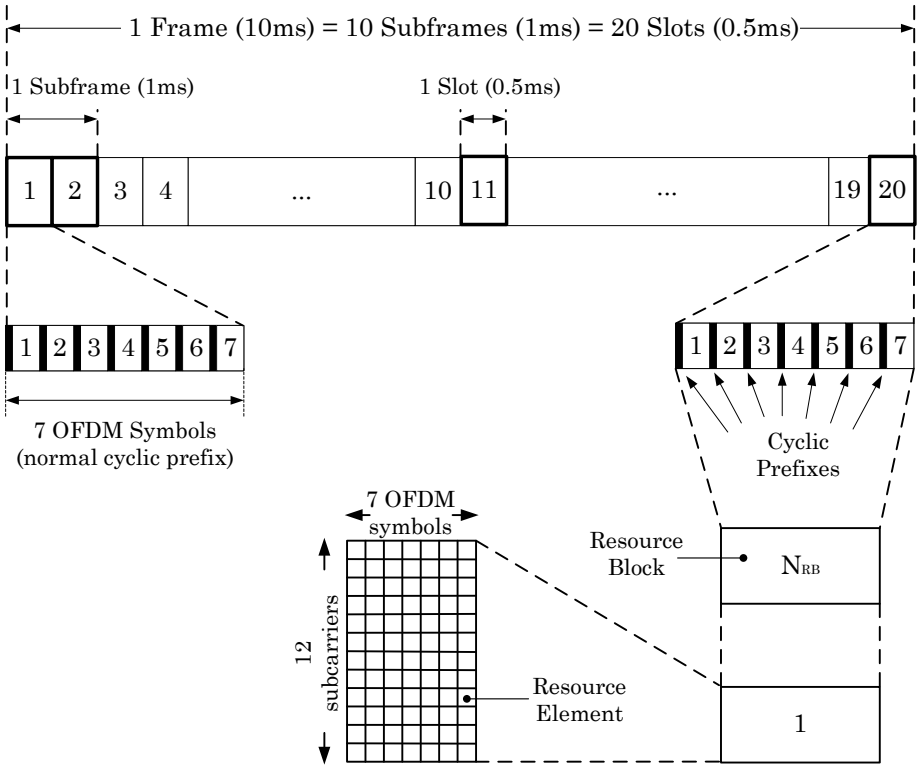

Figure 2.1: LTE frame structure for normal cyclic prefix

2.2 LTE Reference Signal Structure

For channel estimation purpose, LTE specifies training symbols in the downlink known as reference or pilot symbols [7]. Pilot symbols are inserted into each resource block of the transmit signal, which are modulated with 4-Quadrature Amplitude Modulation (QAM). Figure 2.2 depicts the structure

of the pilot symbols allocation in the OFDM time-frequency grid for 1, 2 and 4 transmitting antenna ports. The colored boxes correspond to the pilot symbols at the given antenna port and the crossed boxes to the unused resource elements or zero symbols. Thus, if one antenna port transmits a pilot symbol the other antennas ports at the same position transmit a zero symbol. This structure avoids the interference of the reference symbols transmitted from different antenna ports. It is obvious that increasing the number of antenna ports so increases the number of pilot symbols and zero symbols, too. Because of this fact, the spectral efficiency decreases [3].

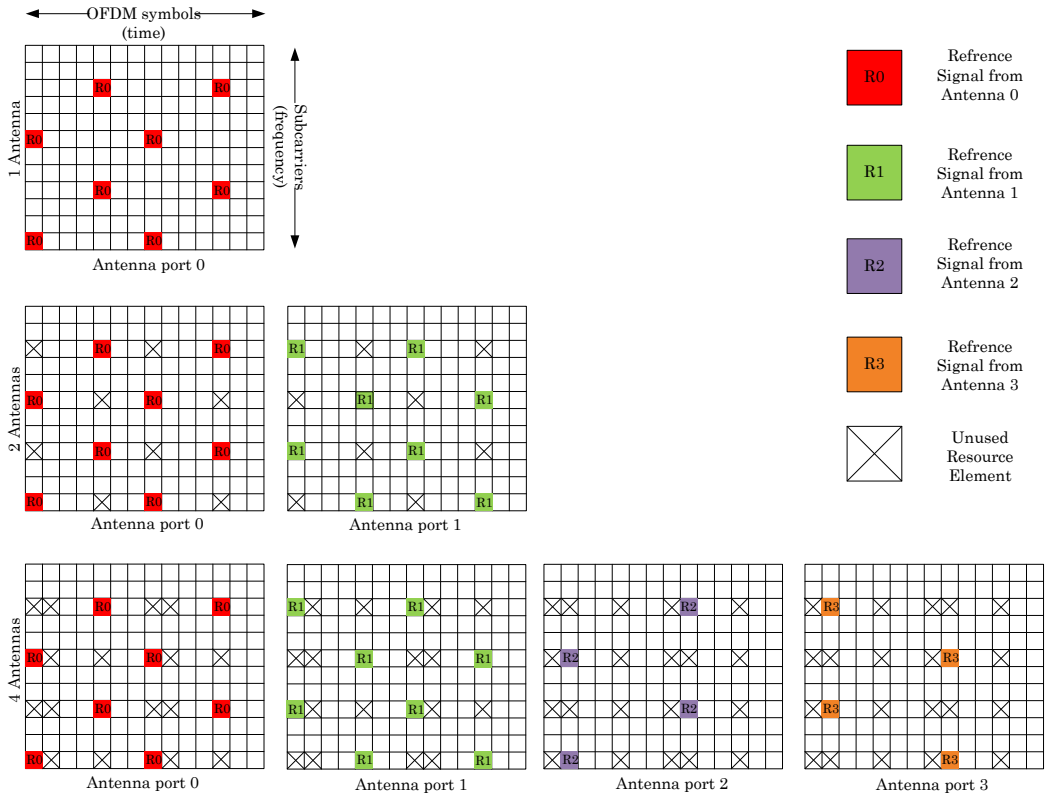

Figure 2.2: Pilot symbols allocation

Chapter 3

System Model

In this chapter, we consider the LTE system model. First, the conventional transmitter and receiver are discussed. The receiver has to be modified and adapted to the iterative receiver. Then, the structure of the iterative receiver and its components, which are important from channel estimation point of view, are discussed. Furthermore, the soft symbol mapper unit is presented and finally, a mathematical signal model is introduced.

3.1 Transmitter

A block diagram of the transmitter is depicted in Figure 3.1 [8]. At the transmitter the information bits $\mathbf{a}_i$ (i =1 or 2) are split into data streams. LTE allows up to two parallel data streams, called codewords. Each codeword is interleaved and rate-matched with a target rate depending on the received Channel Quality Indicator (CQI) user feedback. The coding unit (see Section 3.3.1) involves the following processes: Transport Block (TB) Cyclic Redundancy Check (CRC) attachment, Code Block (CB) segmentation, channel coding and rate-matching [9]. The encoded, interleaved and rate-matched bits $\mathbf{f}_i$ are scrambled (for simplicity not shown in Figure 3.1) and mapped to complex modulated symbols by the symbol mapper. Depending on the CQI, a modulation scheme is selected. Possible modulations are 4-QAM, 16-QAM and 64-QAM, allowing for 2, 4 or 6 bits to be transmitted per symbol. In LTE different multi-antenna schemes are provided like SISO, Transmit Diversity (TxD), Open Loop Spatial Multiplexing (OLSM) or Closed Loop Spatial Multiplexing (CLSM). Next step in the transmitter processing is the layer mapping, respectively mapping codeword symbols to layers. This mapping process depends on the number of codewords and layers. For single antenna transmission there is one-to-one codeword-layer map-

ping. For TxD there is only one codeword and the number of layers equals the number of transmit antenna ports. For Spatial Multiplexing (SM) there is one-to-one, one-to-two, two-to-two, two-to-three or two-to-four codeword-layer mapping [8]. In addition the precoding is applied to the transmit signal, which maps the symbols of the layers to the transmit antenna ports. The MIMO multiplexing component in Figure 3.1 involves layer mapping and precoding process.

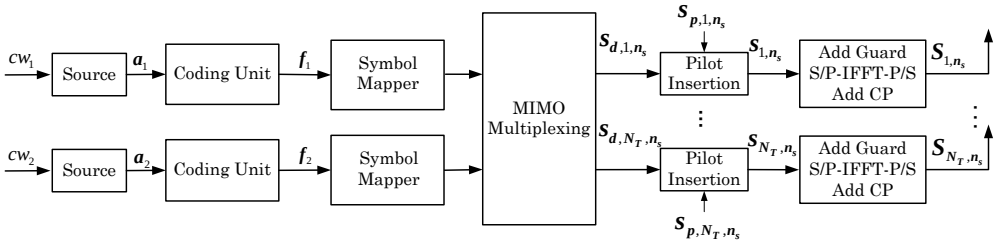

Figure 3.1: Transmitter

Finally, the data symbols to be transmitted on each antenna port are mapped to the resource elements $\mathbf{s}_{d,n_t,n_s}$, with elements s_{d,n_t,n_s,n_k}, where d indicates the data symbols transmitted by the transmit antenna n_t ($n_t = 1, 2 \ldots N_T$) on the corresponding OFDM symbol n_s ($n_s = 0, 1 \ldots N_S - 1$) and on a specific subcarrier n_k ($n_k = 0, 1 \ldots N_K - 1$). Downlink reference symbols $\mathbf{s}_{p,n_t,n_s}$, with elements s_{p,n_t,n_s,n_k}, where p indicates the pilot symbol position, are inserted into the OFDM time-frequency grid. Next, zero subcarriers are padded to eliminate inter-band interference and after Serial-to-Parallel (S/P) conversation, Inverse Fast Fourier Transform (IFFT), and Parallel-to-Serial (P/S) conversation the Cyclic Prefix (CP) is inserted and the transmit signal is generated by a Digital-to-Analog (A/D) (not shown in Figure 3.1) converter to be transmitted through the channel to the receiver [8]. A more detailed mathematical OFDM modulation procedure will be given in Section 3.5.

3.2 Receiver

The complete receiver structure is depicted in Figure 3.2 [5], [6]. The upper part represents the non-iterative receiver, which actually corresponds to the conventional LTE receiver, and the lower one the iterative receiver part. The signal processing procedure for the non-iterative receiver is as follows [8]: the received analog signal is converted into a digital signal through

the Analog-to Digital (A/D) converter (not shown in Figure 3.2), and then the CP is removed and the signal is converted into frequency domain by Fast Fourier Transform (FFT). Finally, the guard symbols are removed. In the non-iterative receiver the channel is estimated only at the pilot symbols (see Figure 2.2). Then, the channel state information knowledge is used for demodulation and soft de-mapping of the OFDM symbols. Finally, the decoding unit (cf. Section 3.3.2) decodes the received coded bits and at the output of the receiver we have the estimated data bits.

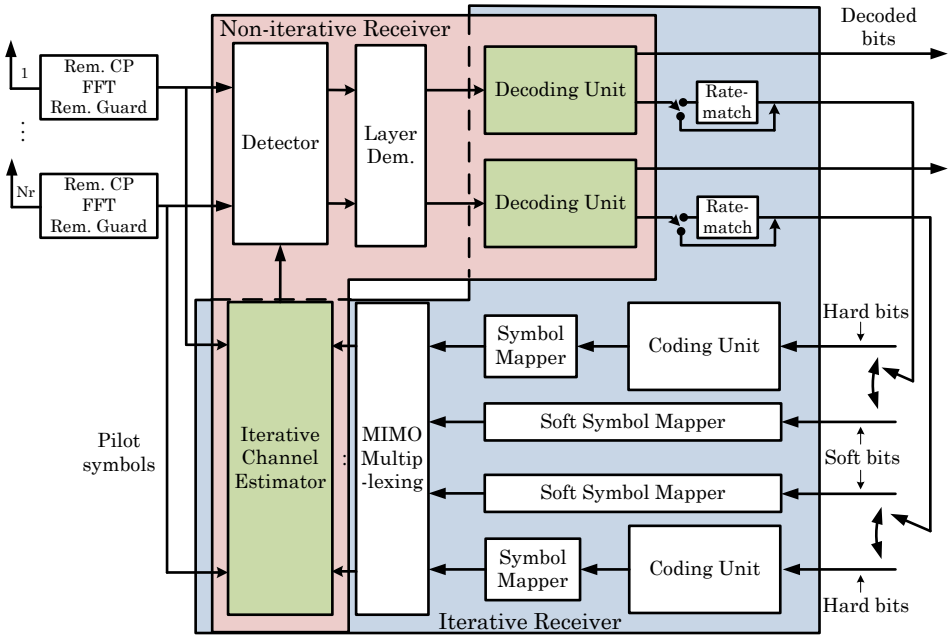

Figure 3.2: Complete Receiver structure

Channel estimator plays a crucial role in the entire system performance, since the knowledge of channel estimation is used by the detector to detect data symbols [4]. On one hand, the more enhanced channel estimate the accurate the detection of data symbols. On the other hand, the more accurate the detection of data symbols the higher the system performance. Channel estimate, besides at the pilot symbols, can be further enhanced by taking the advantage of additional information like the hard decoded data bits or the soft estimated coded bits, which can be obtained from the output of the decoding unit [5], [6]. In order to take advantage of this information, the non-iterative receiver and also the decoding unit, as its component, have to

be modified and adapted to the iterative receiver. As mentioned, the modified decoding unit should be able to output hard and soft information bits. Depending on which feedback information we choose from the decoding unit, two processes are involved in the feedback loop. From Figure 3.2, we see that coding and symbol mapping blocks are used if hard decoded data bits are used as feedback information, and soft symbol mapper block is used if soft coded bits are applied. In the following, the process of coding and decoding units as well as the soft symbol mapper is described in more detail. These components are very important since they constitute the iterative receiver part.

3.3 Coding and Decoding Units

Coding and especially decoding block are two processing units which play an important role in the complete receiver. The coding unit, besides its use in the transmitter, finds use also in the feedback loop of the iterative receiver. Its role is to re-encode and rate match the decoded informations bits. The decoding unit, as part of both the non-iterative and iterative receiver, plays a double important role. First, in the iterative loop, it is used to provide hard or soft estimated bits (soft bits are represented in terms of Log-Likelihood Ratios (LLRs), as a-posteriori LLRs or extrinsic LLRs) [10]. Second, the decoding unit after some global iterations (the iterations between the channel estimator and decoding unit) outputs hard estimated information bits.

3.3.1 Coding Unit

Coding unit for LTE, as mentioned before, consists of the following blocks: TB CRC attachment, CB segmentation, a rate 1/3 turbo encoder, rate-matcher and CB concatenation [9]. Figure 3.3 depicts the processing procedure of the coding unit from the uncoded data bits $\mathbf{a}_i$ to the coded data bits $\mathbf{f}_i$. The transmitted data bits $\mathbf{a}_i$, for each codeword i of length N_{TB}, arrive at the coding unit in form of one TB every subframe and a CRC is then attached, thus obtaining $\mathbf{b}_i$, of length $N_{TB1} = N_{TB} + N_{CRC}$. For simplicity, the codeword index i will be omitted in the following, because of the fact that each codeword is passed independently through the coding unit. The resulting bit vector $\mathbf{b}$ is then segmented in N_{CB} $\mathbf{c}$ CBs, but here for simplification no segmentation is assumed. The bit vector $\mathbf{c}$ is then inserted in the channel coding component. The channel coding component is a rate 1/3 Parallel Concatenated Convolutional Code (PCCC) with two 8-state constituent encoders and one turbo code interleaver, as shown in Figure 3.4 [9].

The CB $\mathbf{c} = [c_1 \dots c_{N_{TB2}}]$, of length $N_{TB2} = N_{TB1} + F$, represents the bits to be encoded by the turbo encoder, where F is the number of filler bits, which are added to ensure that the CB will fit any of the 188 possible interleaver sizes [11].

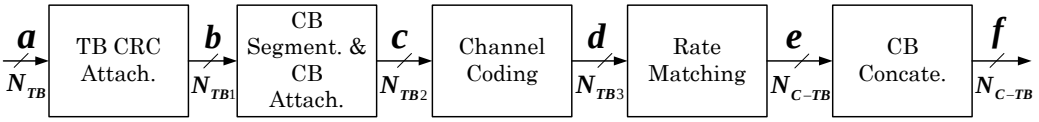

Figure 3.3: Coding unit structure

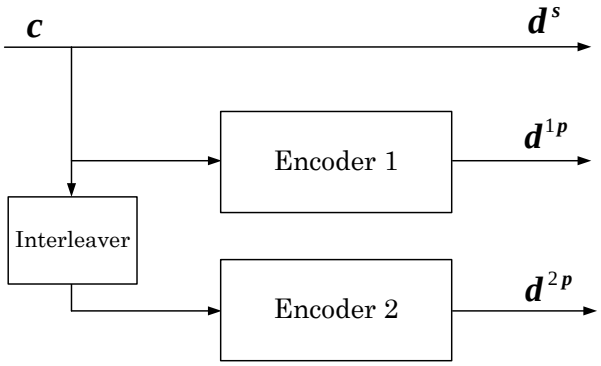

Figure 3.4: Turbo encoder

The code rate of the turbo encoder, called the mother code, is 1/3. This means that for every information bit inserted in the encoder the turbo code outputs three bits, one systematic bit and two parity bits. As can be seen from Figure 3.4, the turbo encoder outputs three encoded bits streams:

$$\mathbf{d}^s = \left[d_1^s \dots d_{N_{TB2}+T}^s\right], \tag{3.1}$$

$$\mathbf{d}^{1p} = \left[d_1^{1p} \dots d_{N_{TB2}+T}^{1p}\right], \tag{3.2}$$

and

$$\mathbf{d}^{2p} = \left[d_1^{2p} \dots d_{N_{TB2}+T}^{2p}\right], \tag{3.3}$$

where $\mathbf{d}^s$, is the vector of $N_{TB2} + T$ systematic bits, and $\mathbf{d}^{1p}$ and $\mathbf{d}^{2p}$ are the bit vectors of length $N_{TB2} + T$ representing the parity bits from first

and second component encoder after turbo coding of $\mathbf{c}$. Termination bits T are distributed across the three outputs. The output bit stream, of length $N_{TB3} = 3(N_{TB2} + T)$, of the turbo encoder is then given by

$$\mathbf{d} = \left[d_1^s d_1^{1p} d_1^{2p} \dots \; d_{N_{TB2}+T}^s d_{N_{TB2}+T}^{1p} d_{N_{TB2}+T}^{2p}\right]. \tag{3.4}$$

In LTE, the bit interleaving is merged in a function called rate matching [11]. After turbo coding, systematic bits and each of the parity are interleaved separately. Furthermore, the LTE rate matcher fits the rate 1/3, and any code rate can be obtained from the initial 1/3 code rate via a process of bit puncturing ($> 1/3$) or repetition ($< 1/3$) [12]. Thus, the code rate of the mother code can be increased by removing some of the systematic or parity bits; such removal of bits is called puncturing or can be decreased by the repetition process. The obtained rate-matched bits $\mathbf{e}$ have the Effective Code Rate (ECR)

$$\text{ECR} = \frac{\mathbf{c}}{\mathbf{e}} \times 1024\,. \tag{3.5}$$

Figure 3.5 depicts visually the overall coding unit process discussed above. The attached and added bits and also the length of each bit stream after each processing block from Figure 3.3 are shown. After rate matching process is performed on each individual CBs (if there is segmentation), all CBs are then concatenated sequentially to from the output of the coding unit, which is the coded TB $\mathbf{f} = \left[f_1 \dots \; f_{N_{C-TB}}\right]$ of length N_{C-TB}.

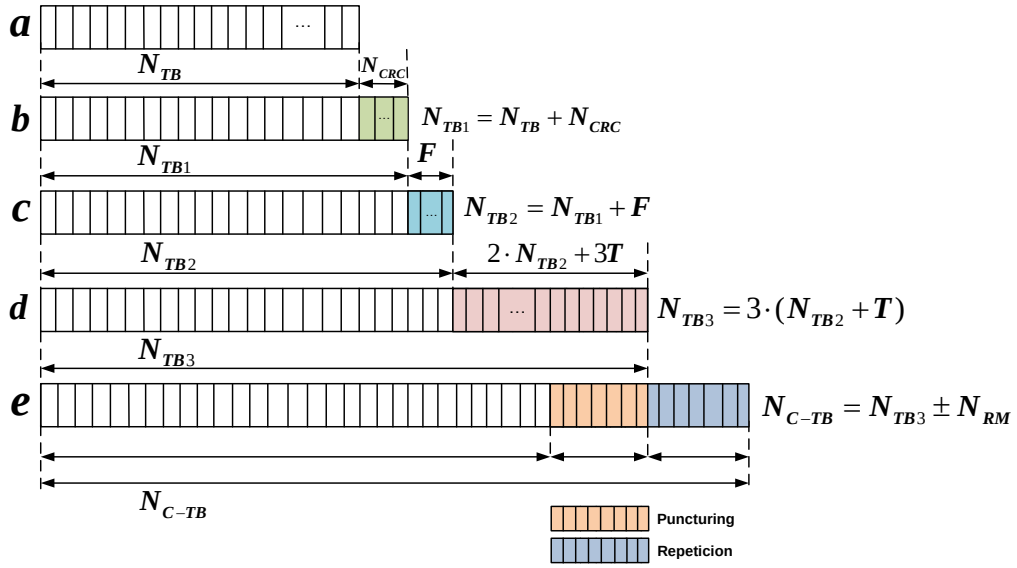

Figure 3.5: Visualized coding unit process

The coded TB is then mapped to a complex modulated symbol sequence (see Figure 3.1). The length N_{C-TB} is actually equal to $N_D N_C$, where N_D is the number of complex modulated data symbols and $N_C = log_2 M$ is the number of bits per symbols. The number of distinct constellation points from the complex constellation set $\mathbb{A}$ (e.g 4-QAM, 16-QAM and 64-QAM) is $M = 2^{N_C}$.

3.3.2 Decoding Unit

Decoding unit is one of the most important components of the complete receiver and especially the turbo decoder as its part [9]. The structure of the complete decoding unit is depicted in Figure 3.6. As mentioned before, the decoding unit of the non-iterative receiver has to be modified in order to be able to output, besides hard decided bits, also soft information of coded bits, which information is further used by the iterative channel estimator to re-estimate the channel. As shown in Figure 3.6, besides the modification of the turbo decoder (see next subsection), the soft rate matcher and the LLR CB concatenation components are added as additional parts to the decoding unit.

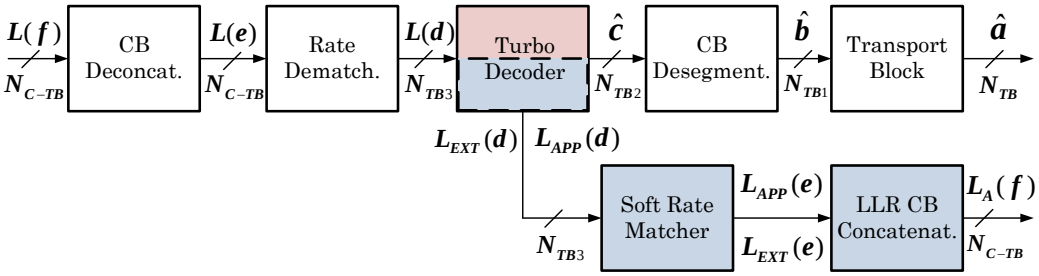

Figure 3.6: Decoding unit structure

The detector (not discussed in this thesis) provides to the decoding unit soft information for each transmitted information coded bit [4]. First, the received soft TB in terms of LLRs $\mathrm{L}(\mathbf{f}) = \left[\mathrm{L}(f_1) \dots \mathrm{L}(f_{N_{C-TB}})\right]$ is de-concatenated into C soft CB $\mathrm{L}(\mathbf{e}) = \left[\mathrm{L}(e_1) \dots \mathrm{L}(e_{N_{C-TB}})\right]$ (if segmentation is assumed). Rate de-matching is then performed on each $\mathrm{L}(\mathbf{e})$ and the output is the soft rate de-matched CB, obtained as

$$\mathrm{L}(\mathbf{d}) = \left[\mathrm{L}(d_1^s)\mathrm{L}(d_1^{1p})\mathrm{L}(d_1^{2p}) \dots \mathrm{L}(d_{N_{TB2}+T}^s)\mathrm{L}(d_{N_{TB2}+T}^{1p})\mathrm{L}(d_{N_{TB2}+T}^{2p})\right], \quad (3.6)$$

which is of length $N_{TB3} = 3(N_{TB2} + T)$, constructed from

$$\mathrm{L}(\mathbf{d}^s) = \left[\mathrm{L}(d_1^s) \dots \mathrm{L}(d_{N_{TB2}+T}^s)\right], \tag{3.7}$$

$$\mathrm{L}(\mathbf{d}^{1p}) = \left[\mathrm{L}(d_1^{1p}) \dots \mathrm{L}(d_{N_{TB2}+T}^{1p})\right], \tag{3.8}$$

and

$$\mathrm{L}(\mathbf{d}^{2p}) = \left[\mathrm{L}(d_1^{2p}) \dots \mathrm{L}(d_{N_{TB2}+T}^{2p})\right], \tag{3.9}$$

where $\mathrm{L}(\mathbf{d}^s)$, $\mathrm{L}(\mathbf{d}^{1p})$ and $\mathrm{L}(\mathbf{d}^{2p})$ represents the LLRs of the systematic and each of the parity bits, that correspond to Equation (3.1), Equation (3.2) and Equation (3.3). Then, the stream of soft bits $\mathrm{L}(\mathbf{d})$ is fed to the turbo decoder. From Figure 3.6 (see also Figure 3.2), we see that the turbo decoder divides the non-iterative part from the iterative part of the complete receiver. In the following, we will discuss in more detail the functionality of the modified turbo decoder.

Turbo Decoder

Figure 3.7 shows the structure of complete turbo decoder [10]. The turbo decoder, as the complete receiver, is divided into two parts. The upper part represents the conventional turbo decoder and the lower part its iterative component. First, we discuss the functionality of the conventional turbo decoder, and then we modify it for the iterative channel estimation purpose. The conventional turbo decoder consists of two component decoders linked by an interleaver at the input, and two additional interleavers at the output of the component decoders [9]. Each component decoder has two inputs, an input of systematic and parity soft bits and an input from the other component decoder. The additional input from the other component decoder is also referred to as a-priori information, and it is the information provided by the other component decoder about the likelihood of the bits being 0 or 1. Only the extrinsic information is transmitted to the other component decoder, in order to avoid the information produced by the one component decoder being fed back to itself. Each component decoder outputs a stream of soft bits. Soft bits are represented by LLRs, which are real numbers that by their sign indicate the estimated value, 0 or 1 of each bit, and by their magnitude the numeric likelihood associated to the value of the bit.

The conventional turbo decoder, thus, operates iteratively [13]. In the first iteration the first component decoder takes as input the stream of soft received bits $\mathrm{L}_{\mathrm{IN}_1}$ (constructed from $\mathrm{L}(\mathbf{d}^s)$ and $\mathrm{L}(\mathbf{d}^{1p})$), which correspond to the bits encoded by the first encoder in the turbo encoder. Since it is the first iteration, the values of the a-priori information $\mathrm{L}_{\mathrm{A}_1}$, not known at the

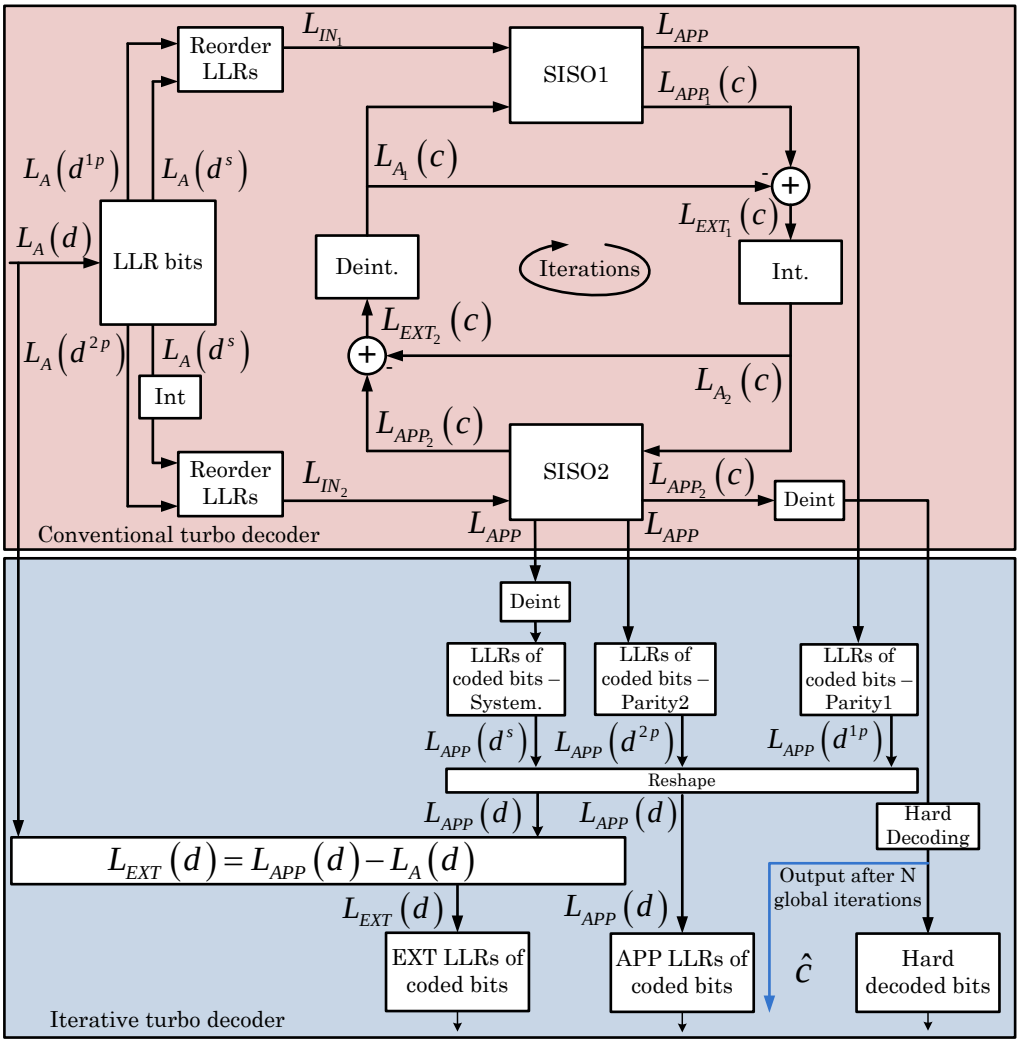

Figure 3.7: Block diagram of the turbo decoder

beginning, are ignored or alternatively set to zero. The soft output provided by the first component decoder L_{APP_1} is then used to calculate the extrinsic information $L_{EXT_1} = L_{APP_1} - L_{A_1}$, which after interleaving is fed as a-priori L_{A_2} information to the second component decoder. Then, the second component decoder estimates the soft bits L_{APP_2} relying now on the information received L_{IN_2} (constructed from Int($L(\mathbf{d}^s)$) and $L(\mathbf{d}^{2p})$), associated to the stream encoded by the second component encoder in the turbo encoder, and the a-priori information L_{A_2} supplied by the first component decoder (after interleaving the L_{EXT_1}). Next, the second iteration begins.

The first component decoder takes again as input the soft received bits L_{IN_1} and in addition the a-priori information L_{A_1} supplied by the second component decoder in the previous iteration, which is obtained after the extrinsic information $L_{EXT_2} = L_{APP_2} - L_{A_2}$ is deinterleaved. As before, the second component decoder takes as input the a-priori information from the first component decoder and the received soft bits. This cycle is repeated until a given criteria is met, like for example after a given number of iterations. When the iterations are completed the estimated sequence is outputted from the second component decoder. The decoding algorithm used in each component decoder is the simplified version of the Maximum A-Posteriori (MAP) algorithm, the Max-Log-MAP algorithm (not discussed here) [10].

The conventional turbo decoder outputs only hard decoded informations bits. To be able to output also soft information of coded bits the turbo decoder has to be modified. Thus, each component decoder should be able to output two different information, namely a-posteriori LLRs of data bits $L_{APP}(\mathbf{c})$ and a-posteriori LLRs of code bits $L_{APP}(\mathbf{d})$ (the extrinsic LLRs are calculated then as $L_{EXT}(\mathbf{d}) = L_{APP}(\mathbf{d}) - L_{A}(\mathbf{d})$). The structure of the component decoder, with its inputs and outputs, implemented for the non-iterative receiver is depicted in Figure 3.8 [13]. This component decoder is implemented in [14]. Referring from [13], the LLRs on data bits are passed to the other component decoder and the LLRs on code bits are not used. From Figure 3.8, the input LLRs of the data bits $L_{A_{IN}}(\mathbf{c})$ represent the information that comes from the other component decoder and LLRs of the code bits $L_{IN}(\mathbf{d})$ are the received soft bits from the detector. The output LLRs of the data bits $L_{APP_{OUT}}(\mathbf{c})$ of the component decoder are passed to the other component decoder and the LLRs of the code bits $L_{OUT}(\mathbf{d})$ in the non-iterative receiver are not used. For iterative channel estimation, these LLRs of the coded bits are used to calculate the soft symbols. Therefore, in additional to the conventional turbo decoder (upper part) the lower part (cf. Figure 3.7) is added to fulfil the requirement for iterative channel estimation.

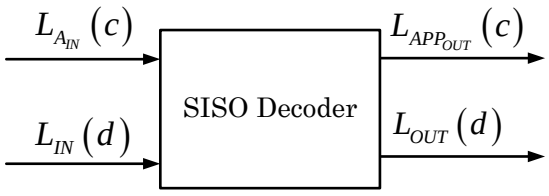

Figure 3.8: Block diagram of the component decoder

The complete turbo decoder now is able, depending on which information we want to use in the feedback loop, to output three different informations:

- A-Posteriori LLRs of data bits $\mathrm{L_{APP}}(\mathbf{c})$
- A-Posteriori LLRs of coded bits $\mathrm{L_{APP}}(\mathbf{d})$
- Extrinsic LLRs of coded bits $\mathrm{L_{EXT}}(\mathbf{d})$

The LLRs of coded bits ($\mathrm{L_{APP}}(\mathbf{d})$ or $\mathrm{L_{EXT}}(\mathbf{d})$), after soft rate matching (cf. Figure 3.6), are considered as a-priori LLRs $\mathrm{L_A}(\mathbf{f})$ and then are used to calculate the soft symbol estimate $\tilde{s}_{d,n_\mathrm{t},n_\mathrm{s},n_k}$ (cf. Section 3.4). On the other hand, the LLRs on data bits $\mathrm{L_{APP}}(\mathbf{c})$, from the second component decoder, after the last internal iteration are utilized to decide on the transmitted information bits along the following rule [5]:

$$\hat{c} = \begin{cases} 0 & \mathrm{L_{APP}}(c) < 0 \\ 1 & \mathrm{L_{APP}}(c) \geq 0 \end{cases} \tag{3.10}$$

The hard decoded bits $\hat{\mathbf{c}}$ are CB de-segmented and at the output of the decoding unit we have the hard decoded TB data bits $\hat{\mathbf{a}}$ (cf. Figure 3.6), which are again encoded, rate-matched and mapped to complex hard symbols $\bar{s}_{d,n_\mathrm{t},n_\mathrm{s},n_k}$. After some global iteration, the decoding unit outputs the hard decoded information bits.

3.4 Soft Symbol Mapper

The a-posteriori LLRs $\mathrm{L_{APP}}(\mathbf{d})$ or extrinsic LLRs $\mathrm{L_{EXT}}(\mathbf{d}) = \mathrm{L_{APP}}(\mathbf{d}) - \mathrm{L_A}(\mathbf{d})$ of coded bits from the output of the turbo decoder, after soft rate matching and CB concatenation (cf. Figure 3.6), are considered as a-priori information ($\mathrm{L}(\mathbf{f}) = \left[\mathrm{L}(f_1) \ldots \mathrm{L}(f_{N_{C-TB}})\right]$) and are utilized by the soft symbol mapper to calculate the soft estimated data symbols as [15]

$$\tilde{s}_{d,n_\mathrm{t},n_\mathrm{s},n_k} = \mathbb{E}\left[s_{d,n_\mathrm{t},n_\mathrm{s},n_k}\right] = \sum_{i=1}^{M} s_{d,n_\mathrm{t},n_\mathrm{s},n_k,i} \cdot \mathrm{P}(s_{d,n_\mathrm{t},n_\mathrm{s},n_k} = s_{d,n_\mathrm{t},n_\mathrm{s},n_k,i}), \tag{3.11}$$

where $s_{d,n_\mathrm{t},n_\mathrm{s},n_k,i}$ is one of the constellation points, that is formed by the information bits $(f_1, f_2, \ldots, f_{N_C})$ and $N_C = log_2 M$ is the number of bits per symbol (M is the number of symbols in the symbol alphabet $\mathbb{A}$, e.g 4-QAM, 16-QAM and 64-QAM). The probability of the symbol being one of

the symbol constellation points equals the product of the corresponding bit probabilities

$$\mathrm{P}(s_{d,n_\mathrm{t},n_\mathrm{s},n_k} = s_{d,n_\mathrm{t},n_\mathrm{s},n_k,i}) = \prod_{n_b=1}^{N_C} \mathrm{P}\big(f_{n_b}(s_{d,n_\mathrm{t},n_\mathrm{s},n_k})\big), \tag{3.12}$$

where $P(f_{n_b}(s_{d,n_\mathrm{t},n_\mathrm{s},n_k}))$ is the probability of the bit that corresponds to the given symbol constellation point. The LLR of the coded bit f_{n_b} is computed by the log-ratio of the probability of the information bit being 1 over the probability of the information bit being 0 [10], i.e

$$\mathrm{L}(f_{n_b}) = \ln\left(\frac{\mathrm{P}(f_{n_b}=1)}{\mathrm{P}(f_{n_b}=0)}\right). \tag{3.13}$$

We know that $\mathrm{P}\big(f_{n_b}(s_{d,n_\mathrm{t},n_\mathrm{s},n_k}) = 1\big) + \mathrm{P}\big(f_{n_b}(s_{d,n_\mathrm{t},n_\mathrm{s},n_k}) = 0\big) = 1$. Using the definitions of LLRs above, we can find the probability of the information bit being 1 or 0 as

$$\mathrm{P}\big(f_{n_b}(s_{d,n_\mathrm{t},n_\mathrm{s},n_k}) = 1\big) = \frac{e^{\mathrm{L}(f_{n_b})}}{1+e^{\mathrm{L}(f_{n_b})}}\,, \tag{3.14}$$

and

$$\mathrm{P}\big(f_{n_b}(s_{d,n_\mathrm{t},n_\mathrm{s},n_k}) = 0\big) = \frac{1}{1+e^{\mathrm{L}(f_{n_b})}}\,. \tag{3.15}$$

Furthermore, Equation (3.14) and Equation (3.15) can be written in terms of tangent hyperbolic functions

$$\mathrm{P}\big(f_{n_b}(s_{d,n_\mathrm{t},n_\mathrm{s},n_k}) = 1\big) = \frac{1}{2}\left(1+\tanh\left(\frac{\mathrm{L}(f_{n_b})}{2}\right)\right), \tag{3.16}$$

and

$$\mathrm{P}\big(f_{n_b}(s_{d,n_\mathrm{t},n_\mathrm{s},n_k}) = 0\big) = \frac{1}{2}\left(1-\tanh\left(\frac{\mathrm{L}(f_{n_b})}{2}\right)\right). \tag{3.17}$$

In practical communication systems, the computation of soft symbols depends on the data modulation scheme. The mathematical derivation of soft symbols for 4-QAM is quite simple, but it gets complicated for higher modulation schemes. Here, the derivation for 4-QAM and 16-QAM is given. The derivation of soft estimated symbols for 64-QAM is given in appendix A.

4-QAM soft symbols

Symbol constellation for 4-QAM according to [7] is depicted in Figure 3.9. There are M=4 symbols, this means that two bits (f_1, f_2) are mapped to a symbol constellation point. Any Gray-mapped QAM constellation symbol

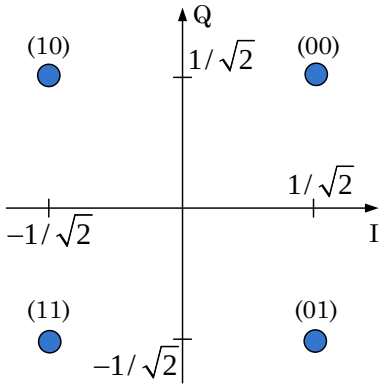

Figure 3.9: 4-QAM signal constellation

can be divided into real and imaginary parts $s_{d,n_t,n_s,n_k,i} = \mathrm{Re}(s_{d,n_t,n_s,n_k,i}) + j\mathrm{Im}(s_{d,n_t,n_s,n_k,i})$ [16]. This implies that the real and imaginary components of each soft symbol are independent $\tilde{s}_{d,n_t,n_s,n_k} = \mathrm{Re}(\tilde{s}_{d,n_t,n_s,n_k}) + j\mathrm{Im}(\tilde{s}_{d,n_t,n_s,n_k})$. Therefore, the first bit is mapped to the real part and the second bit to the imaginary part of a 4-QAM constellation symbol. For the real $\mathrm{Re}(\tilde{s}_{d,n_t,n_s,n_k})$ and imaginary $\mathrm{Im}(\tilde{s}_{d,n_t,n_s,n_k})$ parts, according to Equation (3.11) by applying Equation (3.16) and Equation (3.17), we can find

$$\mathrm{Re}(\tilde{s}_{d,n_t,n_s,n_k}) = 1 \cdot \mathrm{P}(f_1 = 0) + (-1) \cdot \mathrm{P}(f_1 = 1) = -\tanh\left(\frac{\mathrm{L}(f_1)}{2}\right), \quad (3.18)$$

and

$$\mathrm{Im}(\tilde{s}_{d,n_t,n_s,n_k}) = 1 \cdot \mathrm{P}(f_2 = 0) + (-1) \cdot \mathrm{P}(f_2 = 1) = -\tanh\left(\frac{\mathrm{L}(f_2)}{2}\right). \quad (3.19)$$

Then, the soft estimated symbols are calculated as follows

$$\tilde{s}_{d,n_t,n_s,n_k} = \frac{\mathrm{Re}(\tilde{s}_{d,n_t,n_s,n_k}) + j\mathrm{Im}(\tilde{s}_{d,n_t,n_s,n_k})}{\sqrt{2}}. \quad (3.20)$$

Inserting Equation (3.18) and Equation (3.19) into Equation (3.20) we obtain the 4-QAM soft estimated data symbols as

$$\tilde{s}_{d,n_t,n_s,n_k} = \frac{-\tanh\left(\frac{\mathrm{L}(f_1)}{2}\right) - j\tanh\left(\frac{\mathrm{L}(f_2)}{2}\right)}{\sqrt{2}}. \quad (3.21)$$

16-QAM soft symbols

Mathematical derivation of soft estimated symbols for 16-QAM is more complicated than for 4-QAM. Symbol constellation for 16-QAM according to [7] is depicted in Figure 3.10. There are M=16 symbols and four bits (f_1, f_2, f_3, f_4) are mapped to a constellation symbol. As in the 4-QAM case, any Gray-mapped QAM constellation symbol can be divided into real and imaginary parts [16]. Therefore, the first and third bit (f_1, f_3) are mapped to the real part and the second and fourth bit (f_2, f_4) are mapped to the imaginary part of a 16-QAM constellation symbol. This implies that the real and imaginary components of each soft symbol are independent $\tilde{s}_{d,n_\mathrm{t},n_\mathrm{s},n_k} = \mathrm{Re}(\tilde{s}_{d,n_\mathrm{t},n_\mathrm{s},n_k}) + j\mathrm{Im}(\tilde{s}_{d,n_\mathrm{t},n_\mathrm{s},n_k})$. Thus, for the real $\mathrm{Re}(\tilde{s}_{d,n_\mathrm{t},n_\mathrm{s},n_k})$ and the imaginary part $\mathrm{Im}(\tilde{s}_{d,n_\mathrm{t},n_\mathrm{s},n_k})$, according to Equation (3.11), we find

$$\begin{aligned}\mathrm{Re}(\tilde{s}_{d,n_\mathrm{t},n_\mathrm{s},n_k}) = \\ 1 \cdot \mathrm{P}(f_1 = 0)P(f_3 = 0) + 3 \cdot \mathrm{P}(f_1 = 0)P(f_3 = 1) + \\ (-1) \cdot \mathrm{P}(f_1 = 1)P(f_3 = 0) + (-3) \cdot \mathrm{P}(f_1 = 1)P(f_3 = 1),\end{aligned} \tag{3.22}$$

and

$$\begin{aligned}\mathrm{Im}(\tilde{s}_{d,n_\mathrm{t},n_\mathrm{s},n_k}) = \\ 1 \cdot \mathrm{P}(f_2 = 0)P(f_4 = 0) + 3 \cdot \mathrm{P}(f_2 = 0)P(f_4 = 1) + \\ (-1) \cdot \mathrm{P}(f_2 = 1)P(f_4 = 0) + (-3) \cdot \mathrm{P}(f_2 = 1)P(f_4 = 1).\end{aligned} \tag{3.23}$$

Inserting the expressions for bit probabilities from Equation (3.16) and Equation (3.17) into Equation (3.22) and Equation (3.23), and after some calculations for the real and imaginary parts written in terms of tangent hyperbolic functions, we find

$$\mathrm{Re}(\tilde{s}_{d,n_\mathrm{t},n_\mathrm{s},n_k}) = -\tanh\left(\frac{\mathrm{L}(f_1)}{2}\right)\tanh\left(\frac{\mathrm{L}(f_3)}{2}\right) - 2\tanh\left(\frac{\mathrm{L}(f_1)}{2}\right), \tag{3.24}$$

and

$$\mathrm{Im}(\tilde{s}_{d,n_\mathrm{t},n_\mathrm{s},n_k}) = -\tanh\left(\frac{\mathrm{L}(f_2)}{2}\right)\tanh\left(\frac{\mathrm{L}(f_4)}{2}\right) - 2\tanh\left(\frac{\mathrm{L}(f_2)}{2}\right). \tag{3.25}$$

Therefore, the soft estimated symbols for 16-QAM are obtained as

$$\tilde{s}_{d,n_\mathrm{t},n_\mathrm{s},n_k} = \frac{\mathrm{Re}(\tilde{s}_{d,n_\mathrm{t},n_\mathrm{s},n_k}) + j\mathrm{Im}(\tilde{s}_{d,n_\mathrm{t},n_\mathrm{s},n_k})}{\sqrt{10}}, \tag{3.26}$$

where, after inserting the Equation (3.24) and the Equation (3.25) into the Equation (3.26) we obtain the soft estimated symbols for 16-QAM.

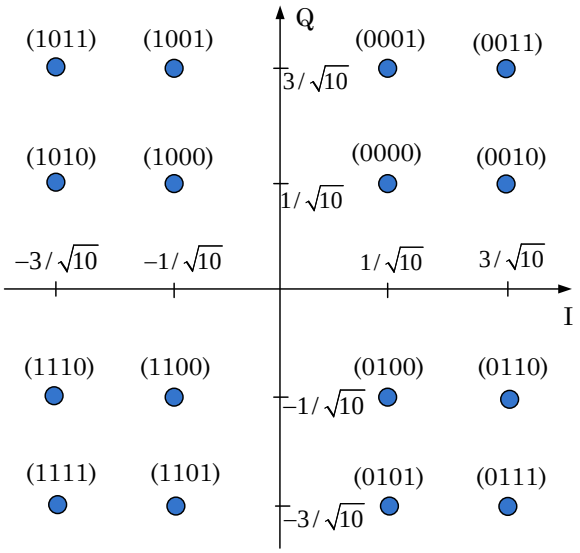

Figure 3.10: 16-QAM signal constellation

3.5 Signal Model

In the following, a signal model is presented. This model is derived based on [17] and extended from [18] for iterative channel estimation.

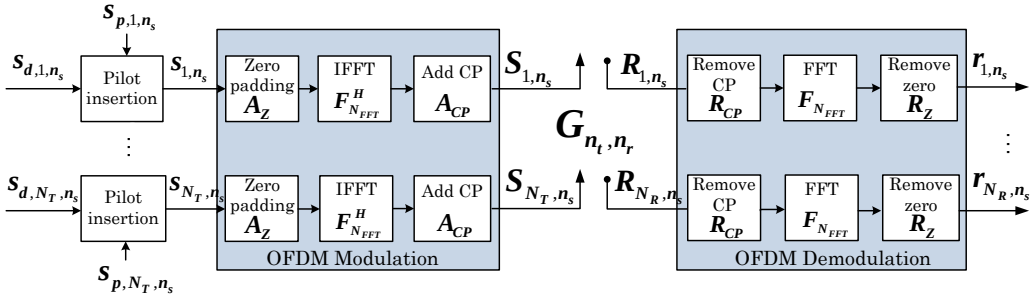

Figure 3.11: Signal Model

Figure 3.11 represents a system with N_T transmit antennas and N_R receive antennas. Let $\mathbf{s}_{d,nt}$ be a length N_D column vector comprising all complex modulated data symbols of one subframe in a frequency domain given as

$$\mathbf{s}_{d,n_t} = \left[s_{d,n_t,0} \dots \; s_{d,n_t,N_D-1}\right]^{\mathrm{T}}, \tag{3.27}$$

where n_{t} is the transmit antenna port ($n_{\mathrm{t}} = 1 \cdots N_T$). Mapping of bits to complex symbols is performed through $s_{d,n_t,n_d} = Q([f_1 \dots \; f_{N_C}])$, where f_{n_b}

are the rate matched and turbo coded bits (see Section 3.3.1) and $Q(\times)$ performs Gray mapping. Next, let $\mathbf{s}_{p,n_t}$ be a length N_P column vector comprising all pilot symbols. If we concatenate the vectors $\mathbf{s}_{d,n_t}$ and $\mathbf{s}_{p,n_t}$ together and then the resource element mapping is perform, where the data and pilot symbols are placed into the frequency-time grid (see Chapter 2), we obtain the resource element mapped signal $\mathbf{s}_{n_t}$ which represents the signal for the entire subframe consisting of N_S OFDM symbols given as

$$\mathbf{s}_{n_\mathrm{t}} = \left[\mathbf{s}^{\mathrm{T}}_{n_\mathrm{t},0} \dots \mathbf{s}^{\mathrm{T}}_{n_\mathrm{t},N_\mathrm{s}-1}\right]^{\mathrm{T}}. \tag{3.28}$$

The n_s-th OFDM symbol is represented by the vector $\mathbf{s}_{n_\mathrm{t},n_\mathrm{s}}$ of length N_K

$$\mathbf{s}_{n_\mathrm{t},n_\mathrm{s}} = \left[s_{n_\mathrm{t},n_\mathrm{s},0} \dots s_{n_\mathrm{t},n_\mathrm{s},N_K-1}\right]^{\mathrm{T}}. \tag{3.29}$$

A detailed mathematical description for OFDM modulation and demodulation blocks is given in [17], therefore we will not go into detail but rather we describe the role of each subblock depicted in Figure 3.11. First, $N_{FFT} - N_K$ zero subcarriers are padded to the transmit signal $\mathbf{s}_{n_\mathrm{t},n_\mathrm{s}}$ by the matrix $\mathbf{A}_\mathrm{Z}$ of size $N_{FFT} \times N_K$. Next, the frequency domain signal is converted into time domain signal through the IFFT operation, by multiplying it with the Discrete Fourier Transform (DFT) matrix $\mathbf{F}_{\mathrm{N_{FFT}}}$, which is of size $N_{FFT} \times N_{FFT}$. Then, the time domain signal is cyclically extended by coping the last N_{CP} samples and pasting them to the front of the signal. This operation is performed by multiplying the OFDM signal in the time domain by the matrix $\mathbf{A}_\mathrm{CP}$ of size $(N_{FFT}+N_{CP}) \times N_{FFT}$. Finally, the signal is transmitted through the channel to the receiver.

At the receiver, the time domain received signal $\mathbf{R}_{n_\mathrm{r},n_\mathrm{s}}$ on the n_s-th OFDM symbol at receive antenna port n_r, of length $N_{FFT} + N_{CP}$, is obtained. The channel matrix $\mathbf{G}_{n_\mathrm{t},n_\mathrm{r}}$ is a Toeplitz matrix of dimension $(N_{FFT} + N_{CP}) \times (N_{FFT} + N_{CP})$ between the n_t-th transmit and n_r-th receive antenna port. The channel impulse response $\mathbf{g}_{n_\mathrm{t},n_\mathrm{r}} = \left[g_{n_\mathrm{t},n_\mathrm{r},0} \dots g_{n_\mathrm{t},n_\mathrm{r},N_g-1}\right]^{\mathrm{T}}$ has at most N_g taps and is assumed to be constant over the duration of a subframe. The first processing step at the receiver is the cyclic prefix removal, which is performed by multiplying the time domain signal with the matrix $\mathbf{R}_\mathrm{CP}$ of dimension $N_{FFT} \times (N_{FFT}+N_{CP})$. Next, the time domain signal is converted into frequency domain signal through the FFT operation by multiplying it with the matrix $\mathbf{F}_{\mathrm{N_{FFT}}}$ and finally, the guard symbols are removed by multiplying the signal with the matrix $\mathbf{R}_\mathrm{Z}$ of dimension $N_K \times N_{FFT}$. The received frequency domain signal on the n_s-th OFDM symbol at receive antenna port

n_r is obtained as

$$\mathbf{r}_{n_\mathrm{r},n_\mathrm{s}} = \sum_{n_\mathrm{t}=1}^{N_\mathrm{t}} \mathbf{H}_{n_\mathrm{t},n_\mathrm{r}}\, \mathbf{s}_{n_\mathrm{t},n_\mathrm{s}} + \mathbf{n}_{n_\mathrm{r},n_\mathrm{s}}, \tag{3.30}$$

where $\mathbf{r}_{n_\mathrm{r},n_\mathrm{s}}$ is the n_s-th OFDM symbol in the frequency domain, $\mathbf{n}_{n_\mathrm{r},n_\mathrm{s}}$ is the noise vector, which elements are considered to be white Gaussian zero mean random variables with variance σ_n^2 and $\mathbf{H}_{n_\mathrm{t},n_\mathrm{r}}$ is an $N_K \times N_K$ channel matrix between the n_t-th and n_r-th antenna port, which is obtained as

$$\mathbf{H}_{n_\mathrm{t},n_\mathrm{r}} = \mathbf{R}_\mathrm{Z}\, \mathbf{F}_{\mathrm{N_{FFT}}}\, \mathbf{R}_\mathrm{CP}\, \mathbf{G}_{n_\mathrm{t},n_\mathrm{r}} \mathbf{A}_\mathrm{CP}\, \mathbf{F}^\mathrm{H}_{\mathrm{N_{FFT}}}\, \mathbf{A}_\mathrm{Z}. \tag{3.31}$$

If the channel is not changing during the duration of one subframe, the matrix $\mathbf{H}_{n_\mathrm{t},n_\mathrm{r}}$ is a diagonal matrix. The elements of the main diagonal of $\mathbf{H}_{n_\mathrm{t},n_\mathrm{r}}$ matrix represent the channel vector $\mathbf{h}_{n_\mathrm{t},n_\mathrm{r}} = [h_{n_\mathrm{t},n_\mathrm{r},0} \dots\ h_{n_\mathrm{t},n_\mathrm{r},N_K-1}]^\mathrm{T}$ in the frequency domain. Therefore, Equation (3.30) can equivalently be written as

$$\mathbf{r}_{n_\mathrm{r},n_\mathrm{s}} = \sum_{n_\mathrm{t}=1}^{N_\mathrm{t}} \mathbf{D}_{n_\mathrm{t},n_\mathrm{s}}\, \mathbf{h}_{n_\mathrm{t},n_\mathrm{r}} + \mathbf{n}_{n_\mathrm{r},n_\mathrm{s}}, \tag{3.32}$$

where $\mathbf{D}_{n_\mathrm{t},n_\mathrm{s}} = \mathrm{diag}(\mathbf{s}_{n_\mathrm{t},n_\mathrm{s}})$ is a diagonal matrix with the elements of $\mathbf{s}_{n_\mathrm{t},n_\mathrm{s}} = [s_{n_\mathrm{t},n_\mathrm{s},0} \dots\ s_{n_\mathrm{t},n_\mathrm{s},N_K-1}]^\mathrm{T}$ on the main diagonal. At this point, we have to distinguish between two signal models; the signal model for initial channel estimation and the one for iterative channel estimation derivation. As mentioned before, initial channel estimation is based only on the pilot symbols inserted in the time-frequency grid. Iterative channel estimation, on the other side, is based on the pilot symbols plus the estimated data symbols. If we concatenate Equation (3.32) for N_S OFDM symbols, we have

$$\mathbf{r}_{n_\mathrm{r}} = \sum_{n_\mathrm{t}=1}^{N_\mathrm{t}} \mathbf{D}_{n_\mathrm{t}}\, \mathbf{h}_{n_\mathrm{t},n_\mathrm{r}} + \mathbf{n}_{n_\mathrm{r}}, \tag{3.33}$$

where $\mathbf{D}_{n_\mathrm{t}}$ is of size $(N_K\, N_S) \times N_K$, and $\mathbf{r}_{n_\mathrm{r}}$ and $\mathbf{n}_{n_\mathrm{r}}$ are of size $(N_K\, N_S) \times 1$, which are defined as

$$\mathbf{D}_{n_\mathrm{t}} = [\mathbf{D}_{n_\mathrm{t},0} \dots\ \mathbf{D}_{n_\mathrm{t},N_S-1}]^\mathrm{T}, \tag{3.34}$$

$$\mathbf{r}_{n_\mathrm{r}} = \left[\mathbf{r}^\mathrm{T}_{n_\mathrm{r},0} \dots\ \mathbf{r}^\mathrm{T}_{n_\mathrm{r},N_S-1}\right]^\mathrm{T}, \tag{3.35}$$

and

$$\mathbf{n}_{n_\mathrm{r}} = \left[\mathbf{n}^\mathrm{T}_{n_\mathrm{r},0} \dots\ \mathbf{n}^\mathrm{T}_{n_\mathrm{r},N_S-1}\right]^\mathrm{T}. \tag{3.36}$$

For the initial channel estimation the sum in Equation (3.33) disappears because if there is a pilot symbol located at the n_{t}-th transmit antenna port within the n_{s}-th OFDM symbol on the n_k-th subcarrier, symbols at the remaining transmit antenna ports at the same positions are zero. Therefore, the extracted signal on pilot symbols is obtained as

$$\mathbf{r}_{p,n_{\mathrm{r}}} = \mathbf{D}_{p,n_{\mathrm{t}}}\, \mathbf{h}_{p,n_{\mathrm{t}},n_{\mathrm{r}}} + \mathbf{n}_{p,n_{\mathrm{r}}}, \tag{3.37}$$

where $\mathbf{r}_{p,n_{\mathrm{r}}}$, $\mathbf{h}_{p,n_{\mathrm{t}},n_{\mathrm{r}}}$ and $\mathbf{n}_{p,n_{\mathrm{r}}}$ represent the received signal, channel vector and the noise vector on pilot positions, all of length N_P. The diagonal matrix $\mathbf{D}_{p,n_{\mathrm{t}}}$, of size $N_P \times N_P$, contains pilot symbols on its diagonal. Therefore, Equation (3.37) is used to derive the initial channel estimation algorithms.

For the iterative channel estimation case, the matrix $\mathbf{D}_{n_{\mathrm{t}}}$ (see Equation (3.33)) has to be replaced by the matrix $\tilde{\mathbf{D}}_{n_{\mathrm{t}}}$, in which the data symbols are replaced by the estimated data symbols calculated from the feedback information from the decoder. Data symbols are replaced either by hard estimated symbols $\bar{s}_{d,n_{\mathrm{t}},n_{\mathrm{s}},n_k}$ or soft estimated symbols $\tilde{s}_{d,n_{\mathrm{t}},n_{\mathrm{s}},n_k}$. Therefore, Equation (3.33) for the iterative case is written as

$$\mathbf{r}_{n_{\mathrm{r}}} = \sum_{n_{\mathrm{t}}=1}^{N_{\mathrm{t}}} \tilde{\mathbf{D}}_{n_{\mathrm{t}}}\, \mathbf{h}_{n_{\mathrm{t}},n_{\mathrm{r}}} + \mathbf{n}_{n_{\mathrm{r}}}, \tag{3.38}$$

and can equivalently be rewritten as

$$\mathbf{r}_{n_{\mathrm{r}}} = \tilde{\mathbf{D}}\mathbf{h}_{n_{\mathrm{r}}} + \mathbf{n}_{n_{\mathrm{r}}}, \tag{3.39}$$

where $\mathbf{r}_{n_{\mathrm{r}}}$ and $\mathbf{n}_{n_{\mathrm{r}}}$ are defined in Equation (3.35) and Equation (3.36). The matrix $\tilde{\mathbf{D}}$ of size $N_K\, N_S \times N_K\, N_T$ is defined as

$$\tilde{\mathbf{D}} = \begin{bmatrix} \tilde{\mathbf{D}}_{1,0} & \tilde{\mathbf{D}}_{2,0} & \dots & \tilde{\mathbf{D}}_{N_T,0} \\ \tilde{\mathbf{D}}_{1,1} & \tilde{\mathbf{D}}_{2,1} & \dots & \tilde{\mathbf{D}}_{N_T,1} \\ \vdots & \vdots & \ddots & \vdots \\ \tilde{\mathbf{D}}_{1,N_S-1} & \tilde{\mathbf{D}}_{2,N_S-1} & \dots & \tilde{\mathbf{D}}_{N_T,N_S-1} \end{bmatrix}, \tag{3.40}$$

where $\tilde{\mathbf{D}}_{n_{\mathrm{t}},n_{\mathrm{s}}} = \mathrm{diag}(\tilde{\mathbf{s}}_{n_{\mathrm{t}},n_{\mathrm{s}}})$ or $\tilde{\mathbf{D}}_{n_{\mathrm{t}},n_{\mathrm{s}}} = \mathrm{diag}(\bar{\mathbf{s}}_{n_{\mathrm{t}},n_{\mathrm{s}}})$ is a diagonal matrix with pilot symbols and soft or hard estimated symbols on the main diagonal and $\mathbf{h}_{n_{\mathrm{r}}}$ is the channel vector of size $N_K\, N_T \times 1$ defined as

$$\mathbf{h}_{n_{\mathrm{r}}} = \left[\mathbf{h}_{1,n_{\mathrm{r}}}^{\mathrm{T}} \dots\, \mathbf{h}_{N_T,n_{\mathrm{r}}}^{\mathrm{T}}\right]^{\mathrm{T}}. \tag{3.41}$$

Thus, Equation (3.37) is utilized for derivation of initial channel estimation algorithms and Equation (3.39) for iterative channel estimation algorithms.

Chapter 4

Iterative Channel Estimation

In this chapter, iterative channel estimation algorithms assuming block fading are derived. As mentioned before, block fading assumes that the channel impulse response $\mathbf{g}_{n_\mathrm{t},n_\mathrm{r}}$ is approximately constant during the duration of one subframe. Therefore, the output of the iterative block fading estimator is the estimated channel vector

$$\hat{\mathbf{h}}_{n_\mathrm{r}}^{(i)} = \left[\hat{\mathbf{h}}_{1,n_\mathrm{r}}^{(i)\ \mathrm{T}} \dots\ \hat{\mathbf{h}}_{N_T,n_\mathrm{r}}^{(i)\ \mathrm{T}}\right]^\mathrm{T}, \tag{4.1}$$

where $\hat{\mathbf{h}}_{n_\mathrm{t},n_\mathrm{r}}^{(i)} = \left[\hat{h}_{n_\mathrm{t},n_\mathrm{r},0}^{(i)} \dots\ \hat{h}_{n_\mathrm{t},n_\mathrm{r},N_K-1}^{(i)}\right]^\mathrm{T}$ is the channel vector for the transmit-receive antenna pair and the upper index i is the iteration number. Thus, i=1 represent the initial channel estimation and i>1 the iterative channel estimation. As already discussed, iterative channel estimation is based not only on the pilot symbols but also on the estimated data symbols. Soft estimated symbols $\tilde{s}_{n_\mathrm{t},n_\mathrm{s},n_k}$ or hard estimated symbols $\bar{s}_{n_\mathrm{t},n_\mathrm{s},n_k}$ are used as feedback information. To be consistent with processing carried out for the derivation of the iterative channel estimators, we shortly discuss again the procedure of the iterative part of the complete receiver. Figure 4.1 depicts a kindly different view of the iterative receiver part, where the channel estimator is represented as the last part of the iterative receiver and the decoding unit as the part which provides the channel estimator with the soft or hard information. As mentioned before, if hard bits are chosen to be used as feedback information, then they have to be again encoded, rate-matched and mapped to hard complex symbols $\bar{s}_{n_\mathrm{t},n_\mathrm{s},n_k}$. On the other hand, if soft bits are chosen then the soft symbol mapper calculates the soft symbols $\tilde{s}_{n_\mathrm{t},n_\mathrm{s},n_k}$. The hard or soft symbols are then used by the channel estimator as additional known symbols to re-estimate the channel. The interaction between the decoding unit and channel estimator is repeated i times i.e until the performance of

the estimator and the overall system is not improved any more. In the following, regarding the derivation of the channel estimation algorithms and to simplify the notation the iteration index number i is omitted, and the iterative algorithms are distinguish from the initial algorithms by indexing them with I in front of their abbreviations i.e ILS, ILMMSE and IALMMSE.

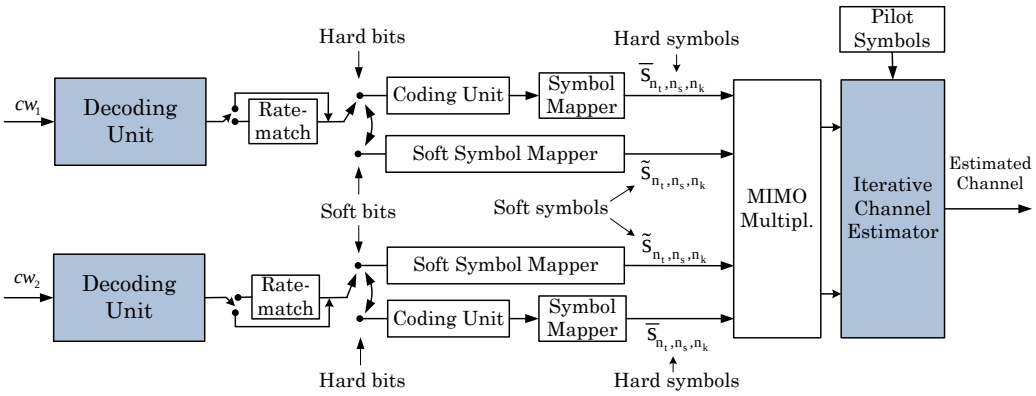

Figure 4.1: Iterative part of the complete receiver

Throughout this chapter, initial as well as iterative channel estimation algorithms are derived. Regarding the derivation of iterative channel estimation algorithms the following assumption is considered:

- The turbo decoder computes hard decoded data bits or soft coded bits in terms of LLRs. Then, the hard or soft bits are mapped to a QAM hard symbol $\bar{s}_{n_t,n_s,n_k}$ or soft symbol $\tilde{s}_{n_t,n_s,n_k}$. The channel estimator assumes that the estimated data symbols are error free and uses them as additional known symbols, thus neglecting the variance of the estimated symbols.

Because of the above assumption, the matrix $\tilde{\mathbf{D}}$ in Equation (3.39) for both cases (hard or soft) is considered as deterministic matrix known to the channel estimator. Under this assumption, it is clear that the channel estimator can yield accurate channel estimate if the soft or hard estimated symbols are close to the transmitted data symbols. In both cases, error propagation can degrade the channel estimation performance and thus the overall system performance.

4.1 Iterative Least Squares Channel Estimation

In this section, first initial channel estimation based on Equation (3.37) ([3]) is derived and then the result is extended to iterative case, that is based on Equation (3.39).

Initial Channel Estimation

Initial Least Squares (LS) channel estimation tries to find a solution $\hat{\mathbf{h}}^{\mathrm{LS}}_{p,n_\mathrm{t},n_\mathrm{r}}$ by minimizing the Euclidean norm squared for $\mathbf{D}_{p,n_\mathrm{t}}\,\hat{\mathbf{h}}^{\mathrm{LS}}_{p,n_\mathrm{t},n_\mathrm{r}} - \mathbf{r}_{p,n_\mathrm{r}}$ [19], that is

$$\begin{aligned}
\underset{\hat{\mathbf{h}}^{\mathrm{LS}}_{p,n_\mathrm{t},n_\mathrm{r}}}{\operatorname{argmin}} J(\hat{\mathbf{h}}^{\mathrm{LS}}_{p,n_\mathrm{t},n_\mathrm{r}}) &= \underset{\hat{\mathbf{h}}^{\mathrm{LS}}_{p,n_\mathrm{t},n_\mathrm{r}}}{\operatorname{argmin}} \left\| \mathbf{D}_{p,n_\mathrm{t}}\hat{\mathbf{h}}^{\mathrm{LS}}_{p,n_\mathrm{t},n_\mathrm{r}} - \mathbf{r}_{p,n_\mathrm{r}} \right\|_2^2 \\
&= \underset{\hat{\mathbf{h}}^{\mathrm{LS}}_{p,n_\mathrm{t},n_\mathrm{r}}}{\operatorname{argmin}} \Big((\mathbf{D}_{p,n_\mathrm{t}}\hat{\mathbf{h}}^{\mathrm{LS}}_{p,n_\mathrm{t},n_\mathrm{r}} - \mathbf{r}_{p,n_\mathrm{r}})^{\mathrm{H}} (\mathbf{D}_{p,n_\mathrm{t}}\hat{\mathbf{h}}^{\mathrm{LS}}_{,p,n_\mathrm{t},n_\mathrm{r}} - \mathbf{r}_{p,n_\mathrm{r}}) \Big) \\
&= \underset{\hat{\mathbf{h}}^{\mathrm{LS}}_{p,n_\mathrm{t},n_\mathrm{r}}}{\operatorname{argmin}} \Big(\mathbf{r}^{\mathrm{H}}_{p,n_\mathrm{r}}\mathbf{r}_{p,n_\mathrm{r}} + (\mathbf{D}_{p,n_\mathrm{t}}\hat{\mathbf{h}}^{\mathrm{LS}}_{p,n_\mathrm{t},n_\mathrm{r}})^{\mathrm{H}} (\mathbf{D}_{p,n_\mathrm{t}}\hat{\mathbf{h}}^{\mathrm{LS}}_{p,n_\mathrm{t},n_\mathrm{r}}) \\
&\quad - \mathbf{r}^{\mathrm{H}}_{p,n_\mathrm{r}} (\mathbf{D}_{p,n_\mathrm{t}}\hat{\mathbf{h}}^{\mathrm{LS}}_{p,n_\mathrm{t},n_\mathrm{r}}) - (\mathbf{D}_{p,n_\mathrm{t}}\hat{\mathbf{h}}^{\mathrm{LS}}_{p,n_\mathrm{t},n_\mathrm{r}})^{\mathrm{H}} \mathbf{r}_{p,n_\mathrm{r}} \Big).
\end{aligned} \tag{4.2}$$

The minimum is found by computing the gradient of the cost function $J(\hat{\mathbf{h}}^{\mathrm{LS}}_{p,n_\mathrm{t},n_\mathrm{r}})$ and equating it to the zero vector

$$\nabla J(\hat{\mathbf{h}}^{\mathrm{LS}}_{p,n_\mathrm{t},n_\mathrm{r}}) = 2\mathbf{D}^{\mathrm{H}}_{p,n_\mathrm{t}}\mathbf{D}_{p,n_\mathrm{t}}\hat{\mathbf{h}}^{\mathrm{LS}}_{p,n_\mathrm{t},n_\mathrm{r}} - 2\mathbf{D}^{\mathrm{H}}_{p,n_\mathrm{t}}\mathbf{r}_{p,n_\mathrm{r}} = \mathbf{0}. \tag{4.3}$$

Therefore, the initial LS channel estimation is obtained as

$$\hat{\mathbf{h}}^{\mathrm{LS}}_{p,n_\mathrm{t},n_\mathrm{r}} = (\mathbf{D}^{\mathrm{H}}_{p,n_\mathrm{t}}\mathbf{D}_{p,n_\mathrm{t}})^{-1}\mathbf{D}^{\mathrm{H}}_{p,n_\mathrm{t}}\mathbf{r}_{p,n_\mathrm{r}}, \tag{4.4}$$

under the condition that the matrix $\mathbf{D}_{p,n_\mathrm{t}}$ has full column rank. Since LTE pilot symbols are with unit power, the inverse matrix $(\mathbf{D}^{\mathrm{H}}_{p,n_\mathrm{t}}\mathbf{D}_{p,n_\mathrm{t}})^{-1}$ is an identity matrix given as

$$(\mathbf{D}^{\mathrm{H}}_{p,n_\mathrm{t}}\mathbf{D}_{p,n_\mathrm{t}})^{-1} = \mathbb{I}_{N_P}. \tag{4.5}$$

Inserting Equation (4.5) into Equation (4.4), the initial LS channel estimation yields

$$\hat{\mathbf{h}}^{\mathrm{LS}}_{p,n_\mathrm{t},n_\mathrm{r}} = \mathbf{D}^{\mathrm{H}}_{p,n_\mathrm{t}}\mathbf{r}_{p,n_\mathrm{r}}. \tag{4.6}$$

Initial LS channel estimator estimates the channel only over the subcarriers carrying pilot symbols. Therefore, channel estimation on the other positions

has to be obtained by interpolation. Although, there exist a number of interpolation techniques ([20],[21]), among them linear interpolation is the less complex one, which delivers very good results. Therefore, in this thesis we consider linear interpolation.

Iterative Channel Estimation

Iterative LS channel estimation is based on the pilot symbols and on the estimated hard or soft data symbols, and therefore no interpolation is needed. The matrix $\tilde{\mathbf{D}}$ form Equation (3.39) contains pilot symbols and either hard or soft estimated symbols. The derivation of iterative LS channel estimation follows the same steps as the initial case, but now based on Equation (3.39). Iterative LS channel estimate also tries to find a solution $\hat{\mathbf{h}}_{n_\mathrm{r}}^{\mathrm{ILS}}$ by minimizing the Euclidean norm squared $J(\hat{\mathbf{h}}_{n_\mathrm{r}}^{\mathrm{ILS}}) = \left\| \tilde{\mathbf{D}}\, \hat{\mathbf{h}}_{n_\mathrm{r}}^{\mathrm{ILS}} - \mathbf{r}_{n_\mathrm{r}} \right\|_2^2$. The minimum is found by computing the gradient of the cost function $\nabla J(\hat{\mathbf{h}}_{n_\mathrm{r}}^{\mathrm{ILS}})$ and equating it to the zero vector. Therefore, the iterative LS channel estimation reads

$$\hat{\mathbf{h}}_{n_\mathrm{r}}^{\mathrm{ILS}} = (\tilde{\mathbf{D}}^{\mathrm{H}}\tilde{\mathbf{D}})^{-1}\tilde{\mathbf{D}}^{\mathrm{H}}\mathbf{r}_{n_\mathrm{r}}. \tag{4.7}$$

The term $(\tilde{\mathbf{D}}^{\mathrm{H}}\tilde{\mathbf{D}})^{-1}\tilde{\mathbf{D}}^{\mathrm{H}}$ is called the pseudo inverse of the matrix $\tilde{\mathbf{D}}$. Equation (4.7) can be applied under the following condition [18]

$$rank(\tilde{\mathbf{D}}) = N_K N_T, \tag{4.8}$$

that is, the matrix $\tilde{\mathbf{D}}$ should have full column rank. If the matrix $\tilde{\mathbf{D}}$ does not have full column rank the matrix inversion involved in Equation (4.7) turns to be ill-conditioned. Therefore, for the ill-conditioned problem we reformulate the iterative LS channel estimate on per subcarrier basis. Equation (3.39) for each subcarrier, for all OFDM symbols and each receive antenna can be written as

$$\mathbf{r}_{n_r,n_k} = \tilde{\mathbf{D}}_{n_k}\mathbf{h}_{n_r,n_k} + \mathbf{n}_{n_r,n_k}, \tag{4.9}$$

where $\mathbf{r}_{n_r,n_k}$ and $\mathbf{n}_{n_r,n_k}$ are of size $N_S \times 1$. Matrix $\tilde{\mathbf{D}}_{n_k}$, of size $N_S \times N_T$, now has the form

$$\tilde{\mathbf{D}}_{n_k} = \begin{bmatrix} \tilde{s}_{1,0,n_k} & \tilde{s}_{2,0,n_k} & \cdots & \tilde{s}_{N_T,0,n_k} \\ \tilde{s}_{1,1,n_k} & \tilde{s}_{2,1,n_k} & \cdots & \tilde{s}_{N_T,1,n_k} \\ \vdots & \vdots & \ddots & \vdots \\ \tilde{s}_{1,N_S-1,n_k} & \tilde{s}_{2,N_S-1,n_k} & \cdots & \tilde{s}_{N_T,N_S-1,n_k} \end{bmatrix}, \tag{4.10}$$

where $\tilde{s}_{n_t,n_s,n_k}$ represent the soft or hard estimated symbol. This method solves the ill-conditioned problem by extrapolating the values at those subcarriers where the matrix $\tilde{\mathbf{D}}_{n_k}$ does not have full column rank. If we concatenate Equation (4.9) for all receive antennas, we have

$$\mathbf{r}_{n_k} = \tilde{\mathbf{D}}\mathbf{h}_{n_k} + \mathbf{n}_{n_k}, \tag{4.11}$$

where $\mathbf{r}_{n_k}$ and $\mathbf{n}_{n_k}$ are of size $N_R N_S \times 1$, $\mathbf{h}_{n_k}$ is of size $N_R N_T \times 1$ and $\tilde{\mathbf{D}} = \mathrm{I}_{N_R} \bigotimes \tilde{\mathbf{D}}_{n_k}$ of size $N_R N_S \times N_R N_T$. The iterative LS channel estimate is then found by minimizing $J(\hat{\mathbf{h}}_{n_k}^{\mathrm{ILS}}) = \left\| \tilde{\mathbf{D}}\, \hat{\mathbf{h}}_{n_k}^{\mathrm{ILS}} - \mathbf{r}_{n_k} \right\|_2^2$ and is given by

$$\hat{\mathbf{h}}_{n_k}^{\mathrm{ILS}} = (\tilde{\mathbf{D}}^{\mathrm{H}}\tilde{\mathbf{D}})^{-1}\tilde{\mathbf{D}}^{\mathrm{H}}\mathbf{r}_{n_k}. \tag{4.12}$$

As mentioned above, if the matrix $\tilde{\mathbf{D}}_{n_k}$ does not have full column rank for a particular subcarrier, then we discard these values and we extrapolate the estimate at these positions. Both iterative methods deliver basically the same result if the matrices have full column rank.

Complexity

Complexity of the initial and iterative LS channel estimator, in terms of matrix inversion and matrix multiplication, calculated for each receive antenna is given in Table 4.1. It is obvious that the iterative LS is more complex because it requires a larger matrix inversion and a matrix multiplication for each iteration.

Initial LS estimation: $\hat{\mathbf{h}}_{p,n_\mathrm{t},n_\mathrm{r}}^{\mathrm{LS}} = \mathbf{D}_{p,n_\mathrm{t}}^{\mathrm{H}}\mathbf{r}_{p,n_\mathrm{r}}$		
Function	Size	Counts
Matrix inversion	$-$	$-$
Matrix multiplication	$(N_P \times N_P) \times (N_P \times 1)$	N_T
Iterative LS estimation: $\hat{\mathbf{h}}_{n_\mathrm{r}}^{\mathrm{ILS}} = (\tilde{\mathbf{D}}^{\mathrm{H}}\tilde{\mathbf{D}})^{-1}\tilde{\mathbf{D}}^{\mathrm{H}}\mathbf{r}_{n_\mathrm{r}}$		
Function	Size	Counts
Matrix inversion	$(N_K N_T \times N_K N_T)$	N_{ITER}
Matrix multiplication	$(N_K N_T \times N_K N_T) \times (N_K N_T \times N_K N_S)$ $(N_K N_T \times N_K N_S) \times (N_K N_S \times 1)$	N_{ITER}

Table 4.1: Complexity of the initial and iterative LS estimator

4.2 Iterative Linear Minimum Mean Square Error Channel Estimation

LS channel estimation does not make any assumption about statistical channel properties or the noise variance. Taking under the consideration the statistical channel properties, LS channel estimate can be further enhanced. We assume that the first and the second order fading statistics and the noise variance are perfectly known. As in the case of LS channel estimation, first initial channel estimation is derived and then the result is extended for the iterative case.

Initial Channel Estimation

The initial Linear Minimum Mean Square Error (LMMSE) channel estimator provides estimated channel coefficients by minimizing the MSE

$$\epsilon = \mathbb{E}\left\{\left\|\mathbf{h}_{n_\mathrm{t},n_\mathrm{r}} - \hat{\mathbf{h}}^{\mathrm{LMMSE}}_{n_\mathrm{t},n_\mathrm{r}}\right\|_2^2\right\}. \tag{4.13}$$

The LMMSE channel estimation can be obtained, according to [19], as filtering the received signal $\mathbf{r}_{p,n_\mathrm{r}}$ by a matrix $\bar{\mathbf{A}}_{\mathrm{LMMSE}}$

$$\hat{\mathbf{h}}^{\mathrm{LMMSE}}_{n_\mathrm{t},n_\mathrm{r}} = \bar{\mathbf{A}}_{\mathrm{LMMSE}}\mathbf{r}_{p,n_\mathrm{r}}, \tag{4.14}$$

where the filtering matrix $\bar{\mathbf{A}}_{\mathrm{LMMSE}}$ is the solution to the linear equation $\bar{\mathbf{A}}_{\mathrm{LMMSE}}\mathbf{R}_{\mathbf{r}_{p,n_\mathrm{r}},\mathbf{r}_{p,n_\mathrm{r}}} = \mathbf{R}_{\mathbf{h}_{n_\mathrm{t},n_\mathrm{r}},\mathbf{r}_{p,n_\mathrm{r}}}$ (known as Wiener-Hopf equation) given as

$$\bar{\mathbf{A}}_{\mathrm{LMMSE}} = \mathbf{R}_{\mathbf{h}_{n_\mathrm{t},n_\mathrm{r}},\mathbf{r}_{p,n_\mathrm{r}}}\mathbf{R}^{-1}_{\mathbf{r}_{p,n_\mathrm{r}},\mathbf{r}_{p,n_\mathrm{r}}}. \tag{4.15}$$

The cross-correlation matrix $\mathbf{R}_{\mathbf{h}_{n_\mathrm{t},n_\mathrm{r}},\mathbf{r}_{p,n_\mathrm{r}}}$ is derived as

$$\begin{aligned}\mathbf{R}_{\mathbf{h}_{n_\mathrm{t},n_\mathrm{r}},\mathbf{r}_{p,n_\mathrm{r}}} &= \mathbb{E}\left\{\mathbf{h}_{n_\mathrm{t},n_\mathrm{r}}\mathbf{r}^{\mathrm{H}}_{p,n_\mathrm{r}}\right\} \\ &= \mathbb{E}\left\{\mathbf{h}_{n_\mathrm{t},n_\mathrm{r}}(\mathbf{D}_{p,n_\mathrm{t}}\,\mathbf{h}_{p,n_\mathrm{t},n_\mathrm{r}} + \mathbf{n}_{p,n_\mathrm{r}})^{\mathrm{H}}\right\} \\ &= \mathbf{R}_{\mathbf{h}_{n_\mathrm{t},n_\mathrm{r}},\mathbf{h}_{p,n_\mathrm{t},n_\mathrm{r}}}\mathbf{D}^{\mathrm{H}}_{p,n_\mathrm{t}},\end{aligned} \tag{4.16}$$

where $\mathbf{R}_{\mathbf{h}_{n_\mathrm{t},n_\mathrm{r}},\mathbf{h}_{p,n_\mathrm{t},n_\mathrm{r}}} = \mathbb{E}\left\{\mathbf{h}_{n_\mathrm{t},n_\mathrm{r}}\mathbf{h}^{\mathrm{H}}_{p,n_\mathrm{t},n_\mathrm{r}}\right\}$ is the cross-correlation matrix between $\mathbf{h}_{n_\mathrm{t},n_\mathrm{r}}$ and $\mathbf{h}_{p,n_\mathrm{t},n_\mathrm{r}}$. The autocorrelation matrix $\mathbf{R}_{\mathbf{r}_{p,n_\mathrm{r}},\mathbf{r}_{p,n_\mathrm{r}}}$ is given by

$$\begin{aligned}\mathbf{R}_{\mathbf{r}_{p,n_\mathrm{r}},\mathbf{r}_{p,n_\mathrm{r}}} &= \mathbb{E}\left\{\mathbf{r}_{p,n_\mathrm{r}}\mathbf{r}^{\mathrm{H}}_{p,n_\mathrm{r}}\right\} \\ &= \mathbb{E}\left\{(\mathbf{D}_{p,n_\mathrm{t}}\mathbf{h}_{p,n_\mathrm{t},n_\mathrm{r}} + \mathbf{n}_{p,n_\mathrm{r}})(\mathbf{D}_{p,n_\mathrm{t}}\mathbf{h}_{p,n_\mathrm{t},n_\mathrm{r}} + \mathbf{n}_{p,n_\mathrm{r}})^{\mathrm{H}}\right\} \\ &= \mathbf{D}_{p,n_\mathrm{t}}\mathbf{R}_{\mathbf{h}_{p,n_\mathrm{t},n_\mathrm{r}},\mathbf{h}_{p,n_\mathrm{t},n_\mathrm{r}}}\mathbf{D}^{\mathrm{H}}_{p,n_\mathrm{t}} + \sigma_n^2\mathbf{I},\end{aligned} \tag{4.17}$$

where $\mathbf{R}_{\mathbf{h}_{p,n_\mathrm{t},n_\mathrm{r}},\mathbf{h}_{p,n_\mathrm{t},n_\mathrm{r}}} = \mathbb{E}\left\{\mathbf{h}_{p,n_\mathrm{t},n_\mathrm{r}}\mathbf{h}_{p,n_\mathrm{t},n_\mathrm{r}}^{\mathrm{H}}\right\}$ is the autocorrelation matrix of $\mathbf{h}_{p,n_\mathrm{t},n_\mathrm{r}}$, and σ_n^2 is the noise variance. The cross-correlation and correlation matrices ($\mathbf{R}_{\mathbf{h}_{n_\mathrm{t},n_\mathrm{r}},\mathbf{h}_{p,n_\mathrm{t},n_\mathrm{r}}}$ and $\mathbf{R}_{\mathbf{h}_{p,n_\mathrm{t},n_\mathrm{r}},\mathbf{h}_{p,n_\mathrm{t},n_\mathrm{r}}}$) are derived as submatrices of the autocorrelation matrix $\mathbf{R}_{\mathbf{h}_{n_\mathrm{t},n_\mathrm{r}},\mathbf{h}_{n_\mathrm{t},n_\mathrm{r}}} = \mathbb{E}\left\{\mathbf{h}_{n_\mathrm{t},n_\mathrm{r}}\mathbf{h}_{n_\mathrm{t},n_\mathrm{r}}^{\mathrm{H}}\right\}$ of the channel vector $\mathbf{h}_{n_\mathrm{t},n_\mathrm{r}}$. Inserting the Equation (4.16) and Equation (4.17) into Equation (4.15) the filtering matrix yield

$$\bar{\mathbf{A}}_{\mathrm{LMMSE}} = \mathbf{R}_{\mathbf{h}_{n_\mathrm{t},n_\mathrm{r}},\mathbf{h}_{p,n_\mathrm{t},n_\mathrm{r}}}\mathbf{D}_{p,n_\mathrm{t}}^{\mathrm{H}}(\mathbf{D}_{p,n_\mathrm{t}}\mathbf{R}_{\mathbf{h}_{p,n_\mathrm{t},n_\mathrm{r}},\mathbf{h}_{p,n_\mathrm{t},n_\mathrm{r}}}\mathbf{D}_{p,n_\mathrm{t}}^{\mathrm{H}} + \sigma_n^2\mathbf{I})^{-1}, \quad (4.18)$$

and the initial LMMSE from Equation (4.14) reads

$$\hat{\mathbf{h}}_{n_\mathrm{t},n_\mathrm{r}}^{\mathrm{LMMSE}} = \mathbf{R}_{\mathbf{h}_{n_\mathrm{t},n_\mathrm{r}},\mathbf{h}_{p,n_\mathrm{t},n_\mathrm{r}}}\mathbf{D}_{p,n_\mathrm{t}}^{\mathrm{H}}(\mathbf{D}_{p,n_\mathrm{t}}\mathbf{R}_{\mathbf{h}_{p,n_\mathrm{t},n_\mathrm{r}},\mathbf{h}_{p,n_\mathrm{t},n_\mathrm{r}}}\mathbf{D}_{p,n_\mathrm{t}}^{\mathrm{H}} + \sigma_n^2\mathbf{I})^{-1}\mathbf{r}_{p,n_\mathrm{r}}. \quad (4.19)$$

Equation (4.19) can be equivalently written as [22]

$$\hat{\mathbf{h}}_{n_\mathrm{t},n_\mathrm{r}}^{\mathrm{LMMSE}} = \mathbf{R}_{\mathbf{h}_{n_\mathrm{t},n_\mathrm{r}},\mathbf{h}_{p,n_\mathrm{t},n_\mathrm{r}}}(\mathbf{R}_{\mathbf{h}_{p,n_\mathrm{t},n_\mathrm{r}},\mathbf{h}_{p,n_\mathrm{t},n_\mathrm{r}}} + \sigma_n^2(\mathbf{D}_{p,n_\mathrm{t}}^{\mathrm{H}}\mathbf{D}_{p,n_\mathrm{t}})^{-1})^{-1}\hat{\mathbf{h}}_{p,n_\mathrm{t},n_\mathrm{r}}^{\mathrm{LS}}, \quad (4.20)$$

where from Equation (4.5), it can be recognized that $(\mathbf{D}_{p,n_\mathrm{t}}^{\mathrm{H}}\mathbf{D}_{p,n_\mathrm{t}})^{-1} = \mathbb{I}_{N_P}$, and $\hat{\mathbf{h}}_{p,n_\mathrm{t},n_\mathrm{r}}^{\mathrm{LS}}$ is the initial LS channel estimate. Therefore, Equation (4.20) can be rewritten as

$$\hat{\mathbf{h}}_{n_\mathrm{t},n_\mathrm{r}}^{\mathrm{LMMSE}} = \mathbf{R}_{\mathbf{h}_{n_\mathrm{t},n_\mathrm{r}},\mathbf{h}_{p,n_\mathrm{t},n_\mathrm{r}}}(\mathbf{R}_{\mathbf{h}_{p,n_\mathrm{t},n_\mathrm{r}},\mathbf{h}_{p,n_\mathrm{t},n_\mathrm{r}}} + \sigma_n^2\mathbf{I})^{-1}\hat{\mathbf{h}}_{p,n_\mathrm{t},n_\mathrm{r}}^{\mathrm{LS}}. \quad (4.21)$$

As opposed to the Equation (4.14), the initial LMMSE channel estimation can otherwise be obtained as filtering of the initial LS channel estimate by a matrix $\mathbf{A}_{\mathrm{LMMSE}}$

$$\hat{\mathbf{h}}_{n_\mathrm{t},n_\mathrm{r}}^{\mathrm{LMMSE}} = \mathbf{A}_{\mathrm{LMMSE}}\hat{\mathbf{h}}_{p,n_\mathrm{t},n_\mathrm{r}}^{\mathrm{LS}}, \quad (4.22)$$

with

$$\mathbf{A}_{\mathrm{LMMSE}} = \mathbf{R}_{\mathbf{h}_{n_\mathrm{t},n_\mathrm{r}},\mathbf{h}_{p,n_\mathrm{t},n_\mathrm{r}}}(\mathbf{R}_{\mathbf{h}_{p,n_\mathrm{t},n_\mathrm{r}},\mathbf{h}_{p,n_\mathrm{t},n_\mathrm{r}}} + \sigma_n^2\mathbf{I})^{-1}. \quad (4.23)$$

The matrix to be inverted in case of initial LMMSE channel estimator is of size $N_P \times N_P$, with N_P the number of pilot symbols in one subframe.

Iterative Channel Estimation

Iterative LMMSE channel estimation is derived based on the Equation (3.39) and follows the same steps as the initial LMMSE case. Equation (4.14) for the iterative case is written as

$$\hat{\mathbf{h}}_{n_\mathrm{r}}^{\mathrm{ILMMSE}} = \bar{\mathbf{A}}_{\mathrm{ILMMSE}}\mathbf{r}_{n_\mathrm{r}}, \quad (4.24)$$

where the filtering matrix $\bar{\mathbf{A}}_{\text{ILMMSE}}$ is obtained as

$$\bar{\mathbf{A}}_{\text{ILMMSE}} = \mathbf{R}_{\mathbf{h}_{n_\text{r}},\mathbf{h}_{n_\text{r}}}\tilde{\mathbf{D}}^{\text{H}}(\tilde{\mathbf{D}}\mathbf{R}_{\mathbf{h}_{n_\text{r}},\mathbf{h}_{n_\text{r}}}\tilde{\mathbf{D}}^{\text{H}} + \sigma_n^2\mathbf{I}_{N_K N_S})^{-1}. \tag{4.25}$$

where $\mathbf{R}_{\mathbf{h}_{n_\text{r}},\mathbf{h}_{n_\text{r}}} = \mathbb{E}\left\{\mathbf{h}_{n_r}\mathbf{h}_{n_r}^{\text{H}}\right\}$ is the autocorrelation matrix of $\mathbf{h}_{n_r}$ of size $(N_K N_T) \times (N_K N_T)$ and σ_n^2 is the noise variance. $\mathbf{I}_{N_K N_S}$ is an identity matrix of size $(N_K N_S) \times (N_K N_S)$. Otherwise, iterative LMMSE (Equation (4.24)) can equivalently be obtained as filtering the iterative LS channel estimate $\hat{\mathbf{h}}_{n_\text{r}}^{\text{ILS}}$ with the filtering matrix $\mathbf{A}_{\text{ILMMSE}}$ [22]

$$\hat{\mathbf{h}}_{n_r}^{\text{ILMMSE}} = \mathbf{A}_{\text{ILMMSE}}\hat{\mathbf{h}}_{n_\text{r}}^{\text{ILS}} = \mathbf{R}_{\mathbf{h}_{n_r},\mathbf{h}_{n_r}}(\mathbf{R}_{\mathbf{h}_{n_r},\mathbf{h}_{n_r}} + \sigma_n^2(\tilde{\mathbf{D}}^{\text{H}}\tilde{\mathbf{D}})^{-1})^{-1}\hat{\mathbf{h}}_{n_\text{r}}^{\text{ILS}}. \tag{4.26}$$

The matrix to be inverted in case of iterative LMMSE channel estimator is of size $N_K N_T \times N_K N_T$.

Complexity

Complexity of the initial and iterative LMMSE channel estimator, in terms of matrix inversion and matrix multiplication, calculated for each receive antenna is illustrated in Table 4.2. It is obvious that the iterative LMMSE has higher complexity than the initial case, since matrices of higher sizes have to be inverted for every iteration. Thus, for every iteration, the iterative LMMSE can be considered to have the same complexity as the initial case.

Initial LMMSE: $\hat{\mathbf{h}}_{n_\text{t},n_\text{r}}^{\text{LMMSE}} = \mathbf{R}_{\mathbf{h}_{n_\text{t},n_\text{r}},\mathbf{h}_{p,n_\text{t},n_\text{r}}}(\mathbf{R}_{\mathbf{h}_{p,n_\text{t},n_\text{r}},\mathbf{h}_{p,n_\text{t},n_\text{r}}} + \sigma_n^2\mathbf{I})^{-1}\hat{\mathbf{h}}_{p,n_\text{t},n_\text{r}}^{\text{LS}}$		
Function	Size	Count
Matrix inversion	$N_P \times N_P$	N_T
Matrix multiplication	$(N_P \times N_P) \times (N_P \times 1)$ $(N_K \times N_P) \times (N_P \times 1)$	 N_T
Iterative LMMSE: $\hat{\mathbf{h}}_{n_\text{r}}^{\text{ILMMSE}} = \mathbf{R}_{\mathbf{h}_{n_\text{r}},\mathbf{h}_{n_\text{r}}}(\mathbf{R}_{\mathbf{h}_{n_\text{r}},\mathbf{h}_{n_\text{r}}} + \sigma_n^2(\tilde{\mathbf{D}}^{\text{H}}\tilde{\mathbf{D}})^{-1})^{-1}\hat{\mathbf{h}}_{n_\text{r}}^{\text{ILS}}$		
Function	Size	Count
Matrix inversion	$(N_K N_T \times N_K N_T)$	$2N_{ITER}$
Matrix multiplication	$(N_K N_T \times N_K N_T) \times (N_K N_T \times 1)$	$2N_{ITER}$

Table 4.2: Complexity of the initial and iterative LMMSE estimator

4.3 Iterative Approximate LMMSE Channel Estimation

The computational complexity that comes with initial and iterative LMMSE is very high because of the matrix inversion involved in the Equation (4.21)

and Equation (4.26). Therefore, their use in a real-time implementation is limited and restricted until their complexity is reduced. The goal of this section is to investigate an approximate iterative channel estimation algorithm which reduces the complexity of the iterative LMMSE and preserves its performance. The estimator discussed in this section is presented in [23] for Worldwide Interoperability for Microwave Access (WiMAX) and adapted for LTE system in [3]. The estimator presented in [3] and [23] is derived for initial channel estimation based on training and pilot symbols. In this section, we modify the estimator presented in [3] for the iterative channel estimation case. Therefore, only the derivation for the iterative channel estimation is shown, since the derivation for initial channel estimation can be found in [3].

The matrix to be inverted in case of iterative LMMSE estimator (Equation (4.26)) is of size $N_K N_T \times N_K N_T$. Therefore, reducing the size of the correlation matrix the overall complexity of the iterative LMMSE estimator is reduced, too. The channel vector to be estimated in the iterative case is of size $N_K N_T \times 1$, and given as

$$\hat{\mathbf{h}}_{n_\mathrm{r}} = \left[\hat{\mathbf{h}}^{\mathrm{T}}_{1,n_\mathrm{r}} \dots \hat{\mathbf{h}}^{\mathrm{T}}_{N_T,n_\mathrm{r}}\right]^{\mathrm{T}}. \tag{4.27}$$

As presented in [3] and [23], the main idea of iterative Approximate Linear Minimum Mean Square Error (ALMMSE) estimator is to approximate the correlation matrix $\mathbf{R_h} = \mathbf{R}_{\mathbf{h}_{n_t,n_r}}$ by utilizing only the correlation of the L closest subcarriers and to average over all $L \times L$ matrices to get just one correlation matrix $\hat{\mathbf{R}}^{(L)}_{\mathbf{h}} = \hat{\mathbf{R}}^{(L)}_{\mathbf{h}_{n_t,n_r}}$, which is of smaller dimension $(L \times L)$ than the full correlation matrix $(N_K \times N_K)$. The approximated correlation matrix, as in [3], is found

$$\hat{\mathbf{R}}^{(L)}_{\mathbf{h}} = \frac{1}{\lfloor \frac{N_K}{L} \rfloor} \sum_{i=0}^{\lfloor \frac{N_K}{L} \rfloor - 1} (\mathbf{R_h})_{iL+1:(i+1)L, iL+1:(i+1)L}\,, \tag{4.28}$$

where the full correlation matrix is then given by $\hat{\mathbf{R}}^{(L)}_{\mathbf{h}_{nr}} = \mathrm{I}_{\mathbf{N_T}} \otimes \hat{\mathbf{R}}^{(L)}_{\mathbf{h}}$, which is of size $LN_T \times LN_T$ and $\otimes$ denotes the Kronecker product. As in [3] and [23], first the number of L closest subcarriers has to be defined, over which the correlation matrix $\hat{\mathbf{R}}^{(L)}_{\mathbf{h}_{n_r}}$ is approximated. Then, the channel vector $\hat{\mathbf{h}}_{n_\mathrm{r}}$

has to be defined over the interval of these L consecutive subcarriers as

$$\hat{\mathbf{h}}_{n_\mathrm{r}} = \begin{cases} \left[[\hat{\mathbf{h}}_{1,n_\mathrm{r}}]_{1:L}^{\mathrm{T}} \cdots [\hat{\mathbf{h}}_{N_T,n_\mathrm{r}}]_{1:L}^{\mathrm{T}} \right]^{\mathrm{T}} & ; k \leq \frac{L+1}{2} \\ \left[[\hat{\mathbf{h}}_{1,n_\mathrm{r}}]_{k-\lfloor \frac{L-1}{2} \rfloor : k+\lceil \frac{L-1}{2} \rceil}^{\mathrm{T}} \cdots [\hat{\mathbf{h}}_{N_T,n_\mathrm{r}}]_{k-\lfloor \frac{L-1}{2} \rfloor : k+\lceil \frac{L-1}{2} \rceil}^{\mathrm{T}} \right]^{\mathrm{T}} & ; \text{otherwise} \\ \left[[\hat{\mathbf{h}}_{1,n_\mathrm{r}}]_{N_K-L+1:N_K}^{\mathrm{T}} \cdots [\hat{\mathbf{h}}_{N_T,n_\mathrm{r}}]_{N_K-L+1:N_K}^{\mathrm{T}} \right]^{\mathrm{T}} & ; k \geq N_K - \frac{L-1}{2} \end{cases} \tag{4.29}$$

so that the output of the iterative channel estimator $\hat{\mathbf{h}}_{n_\mathrm{r}}$ (Equation (4.27)) for the given interval of L subcarriers becomes

$$\hat{\mathbf{h}}_{n_\mathrm{r}}^{(\mathrm{L})} = \left[\hat{\mathbf{h}}_{1,n_\mathrm{r}}^{(\mathrm{L})\ \mathrm{T}} \cdots \hat{\mathbf{h}}_{N_T,n_\mathrm{r}}^{(\mathrm{L})\ \mathrm{T}} \right]^{\mathrm{T}}, \tag{4.30}$$

which is of size $LN_T \times 1$. Next, we derive the iterative LS channel estimate $\hat{\mathbf{h}}_{\mathrm{ILS},n_\mathrm{r}}$. As opposed to the derivation of the initial channel estimation, where the LS channel estimate is derived only at the pilot subcarriers, in the iterative case the LS estimate is derived over all subcarriers (see Equation (4.7)). Therefore, the derived iterative LS estimate is defined also over the interval of L consecutive subacarriers, as $\hat{\mathbf{h}}_{n_\mathrm{r}}^{(\mathrm{L})}$ in Equation (4.29), by

$$\hat{\mathbf{h}}_{n_\mathrm{r}}^{\mathrm{ILS}} = \begin{cases} \left[\hat{h}_{1,n_\mathrm{r},1}^{\mathrm{ILS}} \cdots \hat{h}_{1,n_\mathrm{r},L}^{\mathrm{ILS}} \cdots \hat{h}_{N_T,n_\mathrm{r},1}^{\mathrm{ILS}} \cdots \hat{h}_{N_T,n_\mathrm{r},L}^{\mathrm{ILS}} \right]^{\mathrm{T}} \\ \left[\hat{h}_{1,n_\mathrm{r},k-\lfloor \frac{L-1}{2} \rfloor}^{\mathrm{ILS}} \cdots \hat{h}_{1,n_\mathrm{r},k+\lceil \frac{L-1}{2} \rceil}^{\mathrm{ILS}} \cdots \hat{h}_{N_T,n_\mathrm{r},k-\lfloor \frac{L-1}{2} \rfloor}^{\mathrm{ILS}} \cdots \hat{h}_{N_T,n_\mathrm{r},k+\lceil \frac{L-1}{2} \rceil}^{\mathrm{ILS}} \right]^{\mathrm{T}} \\ \left[\hat{h}_{1,n_\mathrm{r},N_K-L+1}^{\mathrm{ILS}} \cdots \hat{h}_{1,n_\mathrm{r},N_K}^{\mathrm{ILS}} \cdots \hat{h}_{N_T,n_\mathrm{r},N_K-L+1}^{\mathrm{ILS}} \cdots \hat{h}_{N_T,n_\mathrm{r},N_K}^{\mathrm{ILS}} \right]^{\mathrm{T}} \end{cases} \tag{4.31}$$

Next, for the defined interval above, we calculate the iterative ALMMSE channel estimate by filtering the iterative LS estimate $\hat{\mathbf{h}}_{\mathrm{ILS},n_\mathrm{r}}^{(\mathrm{L})}$ with the filtering matrix $\mathbf{A}_{\mathrm{IALMMSE}}^{(L)}$ as follows

$$\hat{\mathbf{h}}_{n_r}^{(\mathrm{L})} = \mathbf{A}_{\mathrm{IALMMSE}}^{(L)} \hat{\mathbf{h}}_{\mathrm{ILS},n_\mathrm{r}}^{(\mathrm{L})}, \tag{4.32}$$

where the filtering matrix $\mathbf{A}_{\mathrm{IALMMSE}}^{(L)}$ is given by

$$\mathbf{A}_{\mathrm{IALMMSE}}^{(L)} = \hat{\mathbf{R}}_{\mathbf{h}_{n_r}}^{(L)} \left(\hat{\mathbf{R}}_{\mathbf{h}_{n_r}}^{(L)} + \sigma_n^2 \Big((\tilde{\mathbf{D}}^{\mathrm{H}} \tilde{\mathbf{D}})^{-1} \Big)^{(L)} \right)^{-1}. \tag{4.33}$$

The matrix $\hat{\mathbf{R}}_{\mathbf{h}_{n_r}}^{(L)}$, of size $LN_T \times LN_T$, is the correlation matrix of the channel vector $\hat{\mathbf{h}}_{n_\mathrm{r}}^{(\mathrm{L})}$ defined in Equation (4.30). Thus, the matrix to be inverted in case of iterative ALMMSE estimator is of size $LN_T \times LN_T$ and compared to Equation (4.26) (where the matrix to be inverted is of size $N_K N_T \times N_K N_T$)

we have a reduction of the complexity by the factor N_K/L, where L can be $< N_K$ (or even $\ll N_K$), but at the same time the performance of the iterative LMMSE channel estimator is preserved, with a small performance loss (see Section 5.1.3). Finally, we select the k-th subcarrier, as in [3], but now for the estimated channel vector $\hat{\mathbf{h}}_{n_\mathrm{r}}^{(\mathrm{L})}$, derived in Equation (4.32)

$$\hat{h}_{\mathrm{IALMMSE},n_\mathrm{r},k} = \begin{cases} \left[[\hat{\mathbf{h}}_{1,n_\mathrm{r}}^{(\mathrm{L})}]_k^{\mathrm{T}} \ldots [\hat{\mathbf{h}}_{N_T,n_\mathrm{r}}^{(\mathrm{L})}]_k^{\mathrm{T}} \right]^{\mathrm{T}} & ; k \leq \frac{L+1}{2} \\ \left[[\hat{\mathbf{h}}_{1,n_\mathrm{r}}^{(\mathrm{L})}]_{\lceil \frac{L+1}{2} \rceil}^{\mathrm{T}} \ldots [\hat{\mathbf{h}}_{N_T,n_\mathrm{r}}^{(\mathrm{L})}]_{\lceil \frac{L+1}{2} \rceil}^{\mathrm{T}} \right]^{\mathrm{T}} & ; \text{otherwise} \\ \left[[\hat{\mathbf{h}}_{1,n_\mathrm{r}}^{(\mathrm{L})}]_{L+k-N_K}^{\mathrm{T}} \ldots [\hat{\mathbf{h}}_{N_T,n_\mathrm{r}}^{(\mathrm{L})}]_{L+k-N_K}^{\mathrm{T}} \right]^{\mathrm{T}} & ; k \geq N_K - \frac{L-1}{2} \end{cases} \tag{4.34}$$

Chapter 5

Simulation Results

In this chapter, we discuss the performance of the iterative channel estimation approaches based on simulation results. We compare the iterative approaches with the pilot based approaches studied in Chapter 4. Naturally, the question whether to apply a-posteriori, extrinsic or hard mapped symbols in the feedback loop arises. In the following, we investigate the impact of processing of this feedback information for channel estimation. We distinguish between initial, perfect, a-posteriori, extrinsic, soft and hard cases. In the initial case, denoted by It.1 in the plots, the channel is estimated based only on pilot symbols. Perfect case, denoted by PERFECT, is the case where the system knows the channel state information perfectly. For the soft case either a-posteriori (app) or extrinsic (ext) information is utilized. The estimators are compared with each other in terms of MSE and throughput over Signal to Noise Ratio (SNR). Furthermore, different antennas setups are considered, namely SISO and MIMO case. For the SISO case a detailed analysis for each channel estimation approach is given, whereas for the MIMO case only the comparison between them is given. All results are obtained with the LTE Link Level Simulator, developed at the Institute of Telecommunications (TC), Vienna University of Technology [14]. The simulator is implemented according to [9] in the complex base band. Table 5.1 presents the most important simulator settings.

5.1 SISO case

We start by considering the case of a simple SISO system. It is observed that increasing the number of iterations to more than three no more performance gain can be achieved. Figure 5.1 depicts the MSE over the number of iterations for SNR=10 dB using a-posteriori feedback information. MSE

Parameter	Value
Bandwidth	1.4 MHz
Number of subframes	1000
Number of iterations	3
Number of transmit antennas	1, 4
Number of receive antennas	1, 4
Channel Quality Indicator (CQI)	4
Modulation order	4-QAM
Transmission mode	SISO, MIMO
Channel type	ITU PedB [24]
Detector	SSD [4]

Table 5.1: Simulation settings

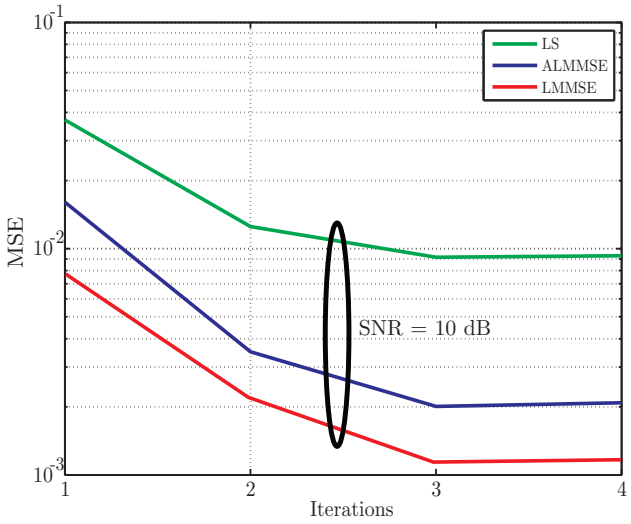

Figure 5.1: MSE versus the number of iterations for SISO case

decreases most rapidly for the LMMSE, followed by the ALMMSE, and then the LS channel estimate. Therefore, in the following simulations only three iterations between the channel estimator and the decoder are considered.

5.1.1 Iterative LS Channel Estimation

Figure 5.2 depicts the performance of the iterative LS channel estimator in terms of MSE over SNR. For very low SNR values the extrinsic case performs slightly better than the initial and a-posteriori case. As the SNR values increase, the MSE for both a-posteriori and extrinsic cases decreases consid-

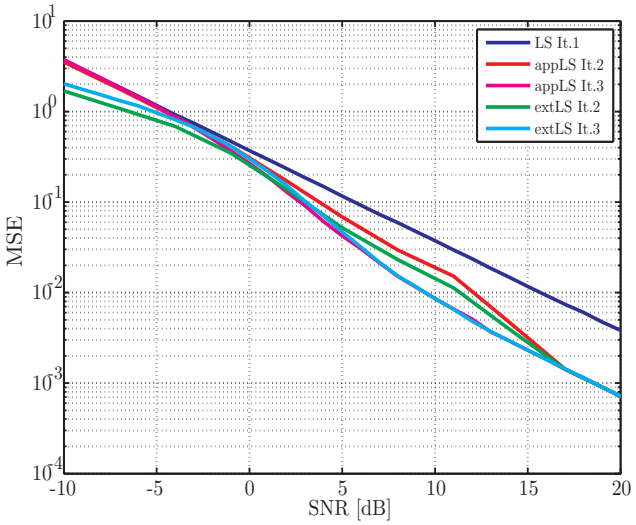

Figure 5.2: MSE for soft iterative versus initial LS estimate

erably compared to the initial case. The a-posteriori case from the 2nd to the 3rd iteration at MSE 10^{-2} improves approximately 2.6 dB, whereas the extrinsic case improves about 1.8 dB. At high SNR values, as the soft symbols become more reliable, the a-posteriori and the extrinsic case perform equally likely. Moreover, at these SNR values no gain from the 2nd to the 3rd iteration is observed. In terms of throughput, shown in Figure 5.3, the SNR loss of the initial LS with respect to the system with perfect channel knowledge is approximately 2 dB. Applying iterative LS channel estimation we gain approximately 0.8 dB with respect to the initial case, and loose approximately 1.2 dB compared to the perfect case. There is observed a gain of approximately 0.2 dB of the a-posteriori case compared to the extrinsic case for the 3rd iteration at throughput 0.4 Mbit/s. Thus, using a-posteriori feedback information the performance of the system improves more than using extrinsic information and the throughput converges faster to the system with perfect channel knowledge.

Finally, hard feedback information is considered. In Figure 5.4, the MSE versus SNR is depicted. The factor that mostly influences the performance of the channel estimator and therefore of the overall system, especially for low SNR values, is the propagation error. At low SNR values, the hard case from 2nd to the 3rd iteration does not improve. For higher SNR values the MSE decreases considerably compared to the initial case. The complete system performance has slightly been improved. Figure 5.5 depicts the throughput

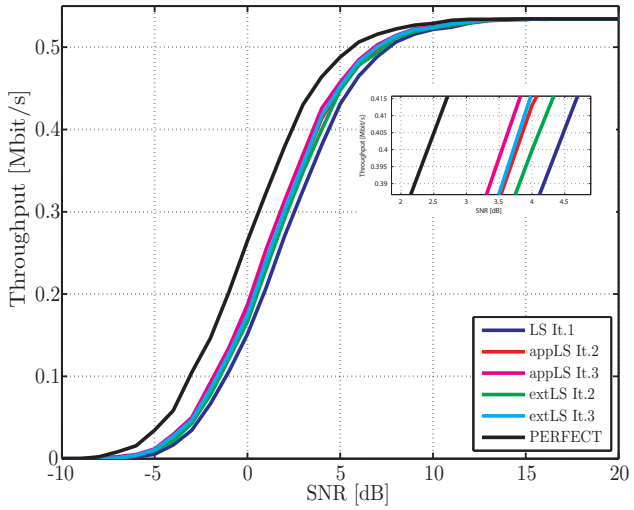

Figure 5.3: Throughput for soft iterative versus initial LS estimate

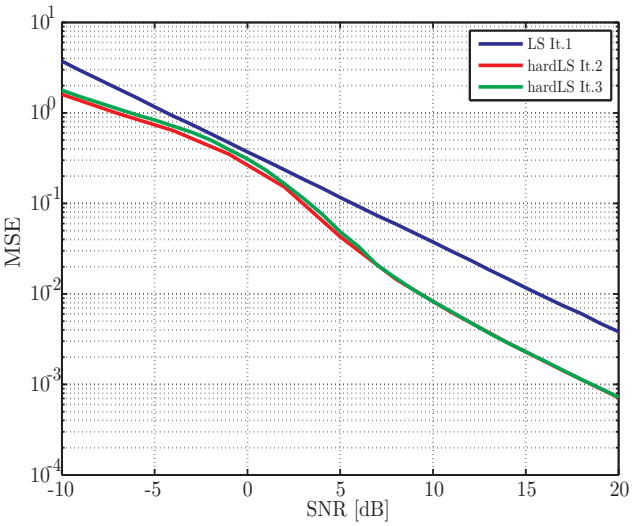

Figure 5.4: MSE for hard iterative versus initial LS estimate

for the hard case. Although, the MSE for the hard case is decreased with respect to the initial case, the overall system performance has slightly been improved. Therefore, it is not always clear if the decrease of MSE will result in an increased performance of the system.

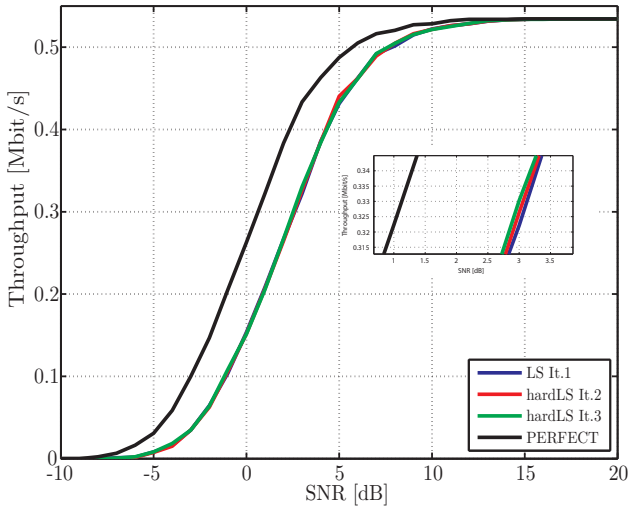

Figure 5.5: Throughput for hard iterative versus initial LS estimate

5.1.2 Iterative LMMSE Channel Estimation

Figure 5.6 depicts the performance of the iterative LMMSE channel estimator in terms of MSE over SNR for the soft feedback information. It is assumed that the second order statistics are perfectly known. As we can see, initial LMMSE for very low SNR values performs better than both soft cases. This is due to the fact, that for low SNR values the reliability of soft symbols is low. The a-posteriori case from the 2nd to the 3rd iteration at MSE 10^{-3} improves approximately 1.6 dB, whereas the extrinsic case improves about 0.9 dB. For low to intermediate SNR values, after the 3rd iteration, the extrinsic case shows a lower MSE than the a-posteriori case. The MSE, for both cases, is considerably decreased compared to the initial case. At high SNR values both cases show similar performance behaviour, since at high SNR values the reliability of soft symbols is high. Figure 5.7 depicts the throughput over SNR. Initial LMMSE is approximately 0.5 dB worse than the system with perfect channel knowledge. The performance of the iterative LMMSE, on the other hand, is almost equal to the system with perfect channel knowledge. Thus, there is approximately 0.45 dB gain compared to the initial case, and approximately 0.05 dB SNR loss compared to the system with perfect channel knowledge. From the MSE plot, the extrinsic case outperforms the a-posteriori case but in terms of throughput, especially at low to intermediate SNR values, it is hard to see any difference between them, since the a-posteriori and extrinsic cases during this SNR region inter-

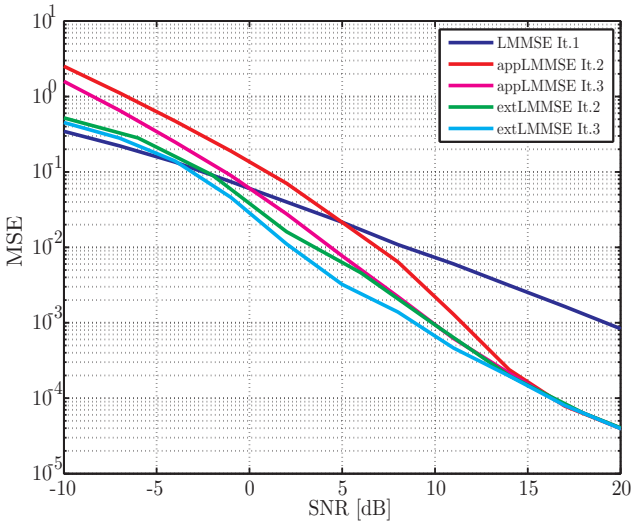

Figure 5.6: MSE for soft iterative versus initial LMMSE estimate

change their role in the performance behaviour. For higher SNR values, it is observed that the extrinsic case converges slightly faster to the system with perfect channel knowledge. There is an out-performance of approximately 0.05 dB, which is actually negligible. Thus, the error propagation is slightly more severe for a-posteriori information than for extrinsic information.

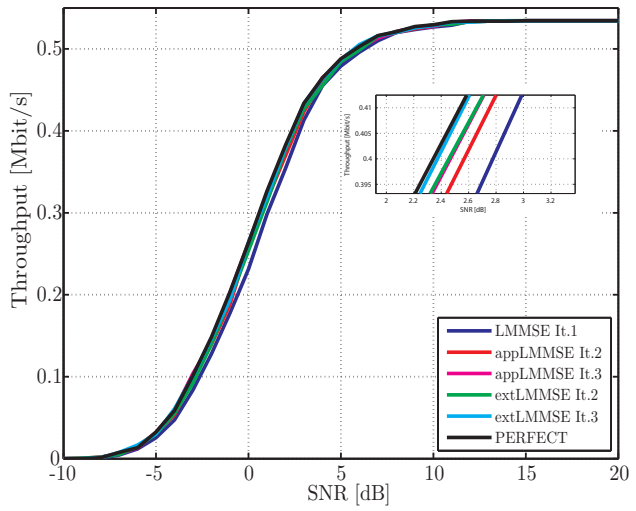

Figure 5.7: Throughput for soft iterative versus initial LMMSE estimate

The performance of the hard case is considered in Figure 5.8 and Figure 5.9. The MSE for hard case is negligibly decreased from 2nd to the 3rd iteration but is considerably decreased with respect to the initial case. The system performance has only slightly improved compared to the initial case.

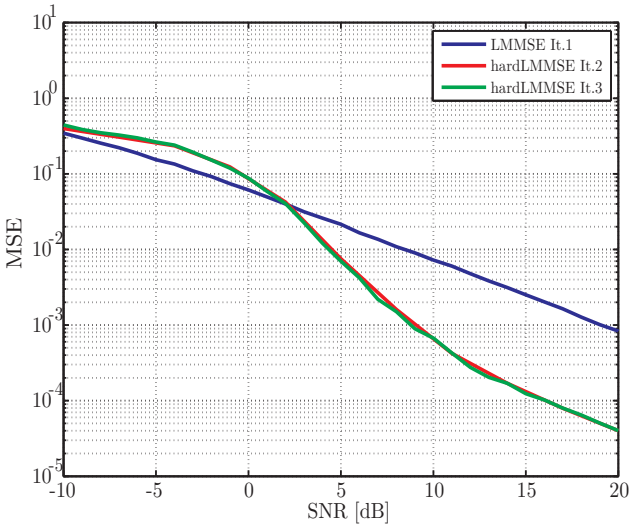

Figure 5.8: MSE for hard iterative versus initial LMMSE estimate

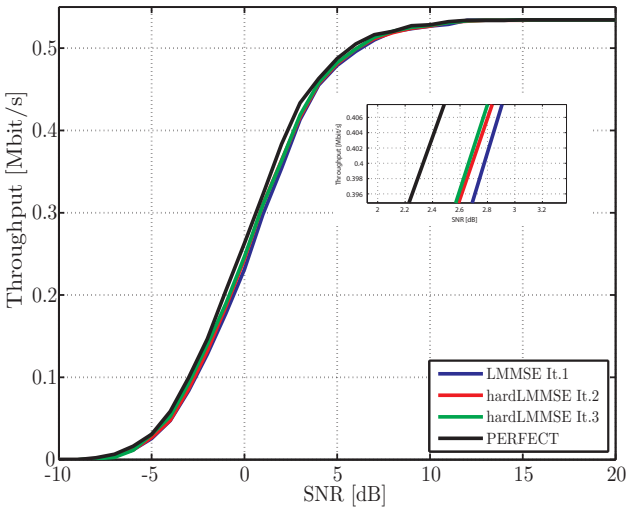

Figure 5.9: Throughput for hard iterative versus initial LMMSE estimate

5.1.3 Iterative ALMMSE Channel Estimation

In this subsection, we will present results for the iterative ALMMSE channel estimator performance. In Figure 5.10 and Figure 5.11, the performance of the iterative ALMMSE estimator for different values of L, for a-posteriori case after the 3rd iteration, is plotted. It can be seen that with increasing L the MSE decreases while the throughput increases. For $L = N_K$, the iterative ALMMSE estimator is equal to the iterative LMMSE estimator. It is obvious that with increasing L also the complexity of the estimator increases. This fact allows adjusting the performance and complexity of the estimator to achieve a good trade-off. The simulations have shown (not given here) that the performance of the initial ALMMSE estimator for L=3 is poorer than the performance of the initial LS estimator. However, this issue is not observed for the iterative case. We see that the iterative ALMMSE for L=3 performs better than the iterative LS. Although, the MSE for L=2 seems to be smaller than the iterative LS, the performance is poorer or at least equal.

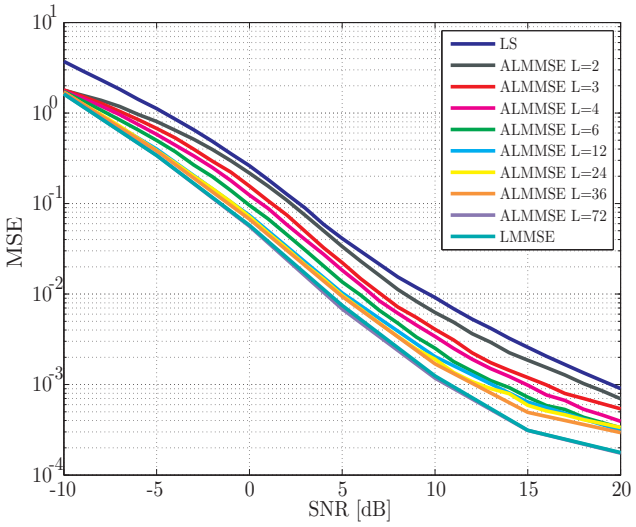

Figure 5.10: MSE for soft iterative ALMMSE for different L

Figure 5.12 and Figure 5.13 depict the MSE and throughput over SNR for the iterative ALMMSE channel estimator using soft feedback information for $L = 12$. The choice of $L = 12$ is a good trade-off between the complexity and performance. For the chosen L, the achieved gain is approximately 0.75 dB compared to the initial case and an SNR loss of approximately 0.3 dB is observed with respect to the system with perfect channel knowledge. Again, as in the iterative LMMSE case, it is hard to say whether a-posteriori

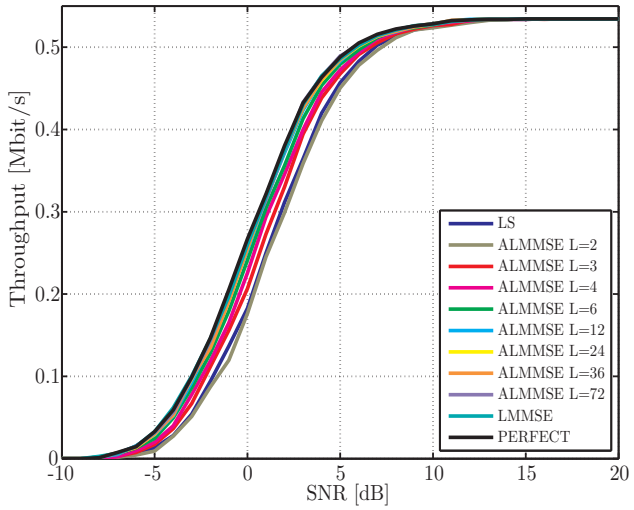

Figure 5.11: Throughput for soft iterative ALMMSE for different L

information or extrinsic information perform better, although the MSE for extrinsic case is slightly lower. The system performance for hard case shows the same patterns as in the LMMSE case and therefore, the plots are not shown here. A comparison between the three channel estimation algorithms is given in the next subsection.

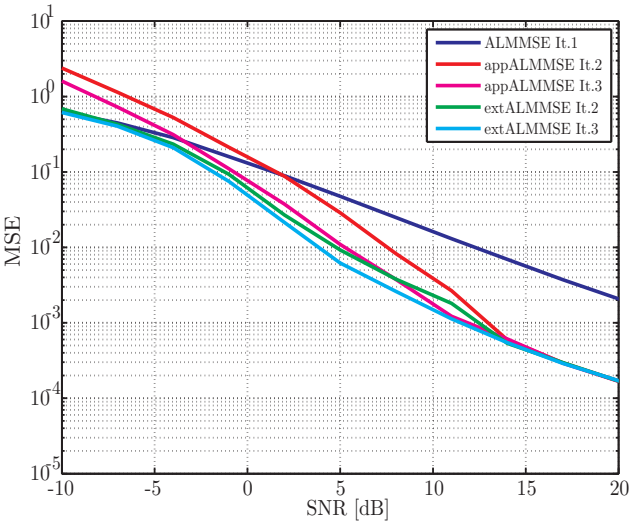

Figure 5.12: MSE for soft iterative versus initial ALMMSE estimate

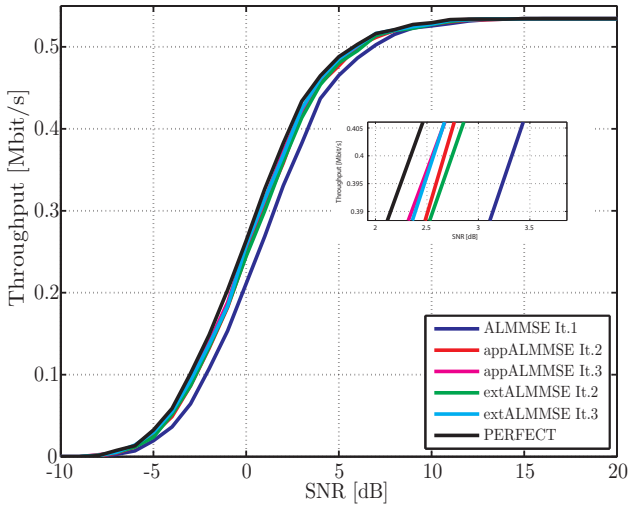

Figure 5.13: Throughput for soft iterative versus initial ALMMSE estimate

5.1.4 Comparison of Channel Estimation Algorithms

In this subsection, we will compare the channel estimation algorithms with each other. For iterative ALMMSE channel estimate $L = 12$ is chosen. For iterative LS channel estimate a-posteriori information is chosen as soft feedback information, whereas for iterative ALMMSE and LMMSE the extrinsic information, since for these estimators the MSE for extrinsic case decreases faster and the throughput converges slightly faster to the system with perfect channel knowledge. In the following, we will compare the estimators for the soft and hard case. Moreover, we will present gain and loss values in a tabulated form, which show in more detail the obtained results.

Soft Feedback Information

Figure 5.14 shows the MSE over SNR for the three algorithms discussed above. Iterative approaches after the 3rd iteration are compared with the pilot approaches. It can be observed that the MSE of three iterative approaches considerably decreases compared to their initial approaches. The MSE decreases most rapidly for the LMMSE, followed by the ALMMSE, and then the LS. Figure 5.15 depicts the throughput over SNR. For iterative LS we have approximately 0.8 dB gain, for iterative ALMMSE 0.75 dB and for iterative LMMSE approximately 0.4 dB with respect to their initial approaches. Iterative ALMMSE after the 3rd iteration outperforms the initial LMMSE for about 0.2 dB and loses about 0.2 dB compared to iterative

LMMSE. Iterative LMMSE channel estimator, after only three iterations, performs approximately equal as the system with perfect channel knowledge, with an SNR loss of 0.05 dB. Table 5.2 and Table 5.3 summarize in more detail the above SNR gain and loss values for the soft case, measured at throughput of 0.4 Mbit/s. Each iterative estimator is compared with the other iterative estimators and with its initial estimator and as well as with the system with perfect channel knowledge.

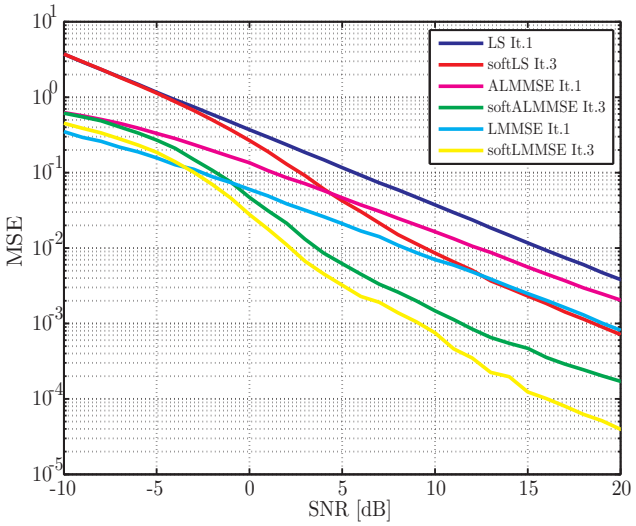

Figure 5.14: MSE comparison for soft information for SISO case

SNR gain values for soft case				
Iterative vs. initial	Init. LS	Init. ALMMSE	Init. LMMSE	Perfect
Iter. LS	0.8 dB	-	-	-
Iter. ALMMSE	1.7 dB	0.75 dB	0.2 dB	-
Iter. LMMSE	1.9 dB	1 dB	0.4 dB	-

Table 5.2: Summary of SNR gain values for soft case (SISO case)

SNR loss values for soft case				
Iterative vs. initial	Init. LS	Init. ALMMSE	Init. LMMSE	Perfect
Iter. LS	-	0.2 dB	0.7 dB	1.2 dB
Iter. ALMMSE	-	-	-	0.4 dB
Iter. LMMSE	-	-	-	0.05 dB

Table 5.3: Summary of SNR loss values for soft case (SISO case)

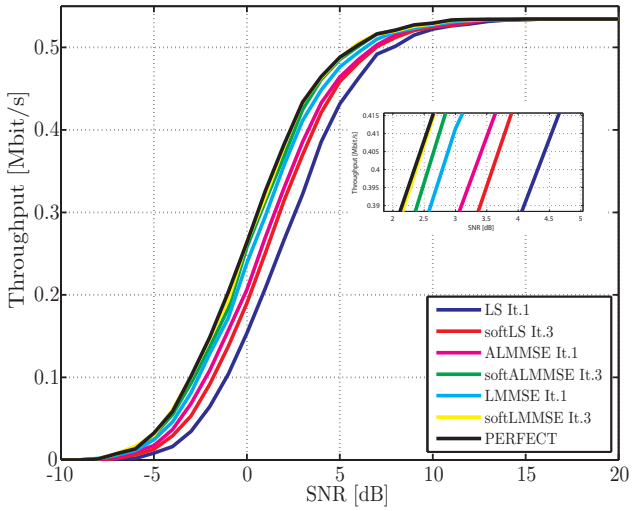

Figure 5.15: Throughput comparison for soft information for SISO case

Hard Feedback Information

Figure 5.16 depicts the MSE over SNR for the hard case. We can see that the MSE for iterative approaches is equal or worse for low SNR values compared to initial approaches, this is due to wrong bit decisions at the decoder that may occur at low SNR values. For high SNR values the MSE decreases considerably with respect to the initial case. In terms of throughput, shown in Figure 5.17, for iterative LS we have approximately 0.05 dB gain, for ALMMSE approximately 0.3 dB and LMMSE only 0.1 dB with respect to their initial channel estimation. Table 5.4 and Table 5.5 summarize in more detail the above SNR gain and loss values for the hard case.

SNR gain values for hard case				
Channel estimator	Init. LS	Init. ALMMSE	Init. LMMSE	Perfect
Iter. LS	0.05 dB	-	-	-
Iter. ALMMSE	1.2 dB	0.3 dB	-	-
Iter. LMMSE	1.7 dB	0.6 dB	0.1 dB	-

Table 5.4: Summary of SNR gain values for hard case (SISO case)

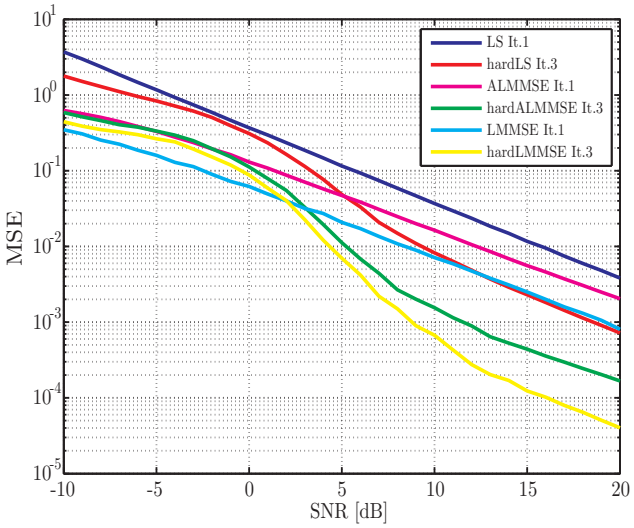

Figure 5.16: MSE comparison for hard information for SISO case

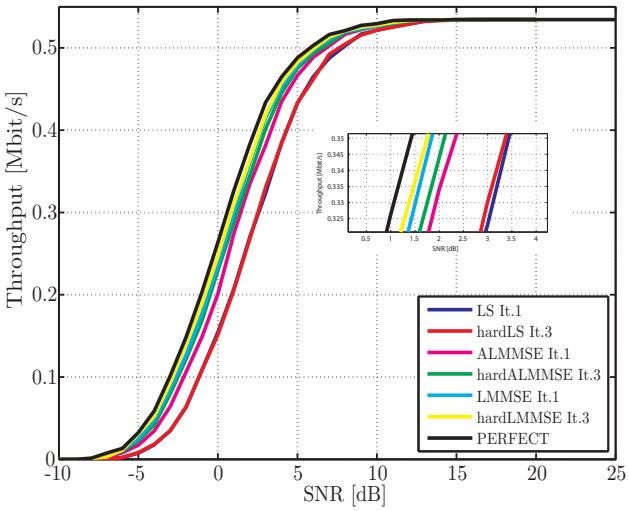

Figure 5.17: Throughput comparison for hard information for SISO case

SNR loss values for hard case				
Channel estimator	Init. LS	Init. ALMMSE	Init. LMMSE	Perfect
Iter. LS	-	1 dB	1.6 dB	1.95 dB
Iter. ALMMSE	-	-	0.25 dB	0.7 dB
Iter. LMMSE	-	-	-	0.3 dB

Table 5.5: Summary of SNR loss values for hard case (SISO case)

Hard versus Soft Feedback Information

Figure 5.18 compares the throughput over SNR for hard and soft cases for the 3rd iteration. It is observed that the improvement of soft case over the hard case for iterative LS is about 0.7 dB, soft iterative ALMMSE improves about 0.4 dB and soft iterative LMMSE approximately 0.3 dB. Therefore, using soft feedback information the overall system performance is more improved than using hard feedback information. The reason is that for hard feedback information the error propagation is more severe and therefore, it degrades the estimator performance and thus the overall system performance.

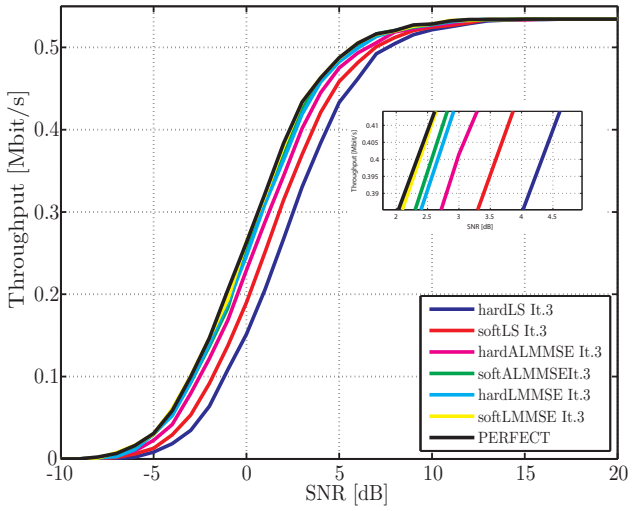

Figure 5.18: Throughput for hard versus soft information for SISO case

5.2 MIMO case

In this section, we consider a system which consist of four transmit and four receive antennas (for comparison purposes we also consider the 2×2 MIMO case). We will not investigate each channel estimator algorithm separately, but rather we compare them with each other, as carried out in Section 5.1.4. As for the SISO case, a comparison is carried out in terms of MSE and throughput over SNR. In the SISO case, except for iterative LS, it was not quite clear which soft information to use in the feedback loop, since in general, for the given simulations settings (Table 5.1) the a-posteriori and extrinsic case perform nearly equally for the iterative LMMSE and ALMMSE;

although MSE plots have shown that the extrinsic case outperforms the a-posteriori case. Here, we directly compare the extrinsic and a-posteriori case.

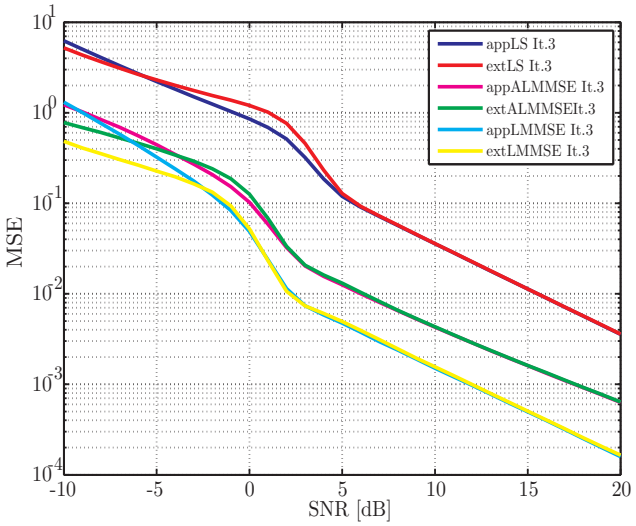

Figure 5.19: MSE comparison for a-posteriori and extrinsic information for 4×4 MIMO case

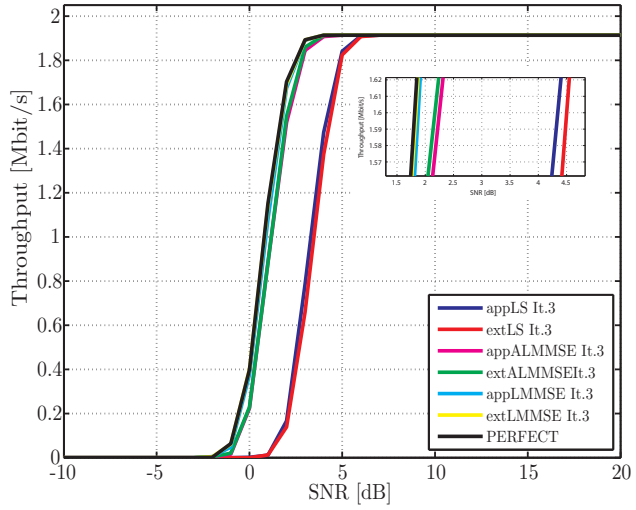

Figure 5.20: Throughput comparison for a-posteriori and extrinsic information for 4×4 MIMO case

Figure 5.19 and Figure 5.20 compare the MSE and throughput over SNR for a-posteriori and extrinsic feedback information for the 3rd iteration. For low SNR values the MSE of a-posteriori case for iterative LS outperforms the extrinsic case. This is also verified in the throughput plot, where the a-posteriori case, for instance at throughput 1.6 Mbit/s, gains about 0.15 dB over the extrinsic case. The MSE plot for iterative LMMSE and ALMMSE shows that the two soft cases differentiate slightly from each other only in the low SNR region, where the MSE for extrinsic information shows a lower value than the a-posteriori information. For higher SNR values both cases for all estimators perform equally. In terms of throughput, for iterative LMMSE and ALMMSE the extrinsic case slightly outperforms the a-posteriori case, although this gain is approximately 0.05 dB for both iterative approaches. To be consistent with the above analysis, for iterative soft LS channel estimate we will subsequently use a-posteriori feedback information, whereas for iterative soft ALMMSE and LMMSE estimate extrinsic feedback information is utilized. In the following, iterative approaches for the 3rd iteration are compared with the pilot approaches and with the system with perfect channel knowledge.

5.2.1 Soft Feedback Information

Figure 5.21 depicts the MSE over SNR for the three algorithms. Moreover, for comparison purposes, in the MSE plot we add the curves for the 2×2 MIMO case (dashed curves with the same colours). As expected, the MSE for the 4×4 MIMO case increases with respect to the 2×2 MIMO case. The reason is that the number of pilot symbols with respect to transmitting antenna ports does not double in the case of 4×4 MIMO, as it is doubled from SISO to the 2×2 MIMO case. At the 3rd and 4th transmit antenna ports less pilot symbols than at the 1st and 2nd transmit antenna ports are located. It is obvious that with increasing of the number of antennas, the number of pilot symbols and symbols which are 0 increases as well. Therefore, in general the quality of the channel estimate from 3rd and 4th transmit antenna ports will be poorer than the quality of channel estimate from 1st and 2nd transmit antenna port. For low SNR values, where the reliability of soft symbols is low, the MSE for the three iterative approaches is worse or at least equal to the initial approaches. As the SNR values increase the MSE for all iterative approaches decreases considerably compared to their initial approaches. The same pattern are observed also for 2×2 MIMO case. Considering the throughput in Figure 5.22, we see that the entire system performance has been considerably improved using iterative channel estimation. For comparison purposes we also depict the throughput for the

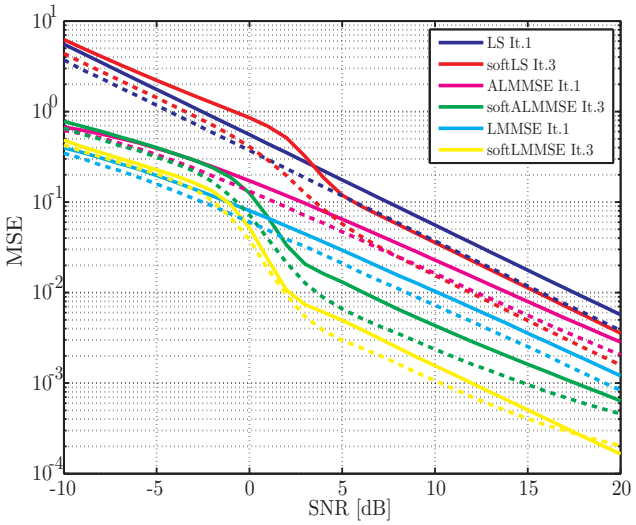

Figure 5.21: MSE comparison for soft information for 4×4 MIMO case (2×2 MIMO case with dashed curves)

2×2 MIMO case, but only the 4×4 MIMO case is analysed. With respect to their initial approaches, iterative LS gains approximately 0.45 dB, iterative ALMMSE 0.9 dB and iterative LMMSE 0.7 dB. Iterative ALMMSE after the 3rd iteration outperforms the initial LMMSE for approximately 0.35 dB and loses about 0.35 dB compared to the system with perfect channel knowledge. Iterative LMMSE after the 3rd iteration is approximately equal to the system with perfect channel knowledge, and an SNR loss of approximately 0.05 dB is observed. The SNR gain and loss values, measured at throughput 1.6 Mbit/s, are summarized in more detail in Table 5.6 and Table 5.7. As in the SISO case, each iterative approach is compared with the other approaches and with their initial approaches as well as with the system with perfect channel knowledge.

SNR gain values for soft case				
Channel estimator	Init. LS	Init. ALMMSE	Init. LMMSE	Perfect
Iter. LS	0.45 dB	-	-	-
Iter. ALMMSE	2.6 dB	0.9 dB	0.35 dB	-
Iter. LMMSE	2.9 dB	1.2 dB	0.7 dB	-

Table 5.6: Summary of SNR gain values for soft case (4×4 MIMO case)

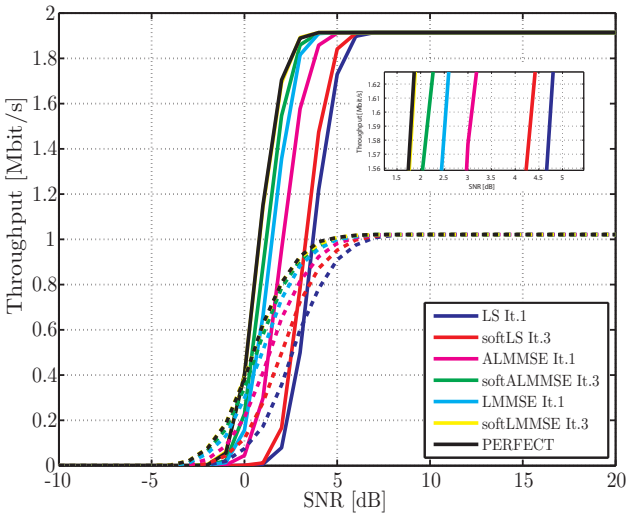

Figure 5.22: Throughput comparison for soft information for 4×4 MIMO case (2×2 MIMO case with dashed curves)

SNR loss values for soft case				
Channel estimator	Init. LS	Init. ALMMSE	Init. LMMSE	Perfect
Iter. LS	-	1.3 dB	1.85 dB	2.5 dB
Iter. ALMMSE	-	-	-	0.35 dB
Iter. LMMSE	-	-	-	0.05 dB

Table 5.7: Summary of SNR loss values for soft case (4×4 MIMO case)

5.2.2 Hard Feedback Information

Figure 5.23 depicts the MSE over SNR for the hard 4×4 MIMO case (also 2×2 MIMO case with dashed curves is depicted). For higher SNR values, same performance patterns are observed as in the soft case analysed before. At low SNR values the bit decision at the decoder may be wrong and therefore the error propagation influences the estimators performance. Figure 5.24 depicts the throughput for the hard case for all iterative approaches in comparison to the initial approaches. Iterative LS, as in the SISO case, improves approximately 0.05 dB with respect to the initial channel estimation, and iterative LMMSE and ALMMSE approximately 0.25 dB. A detailed representation of the SNR gain and loss values is presented in Table 5.8 and Table 5.9.

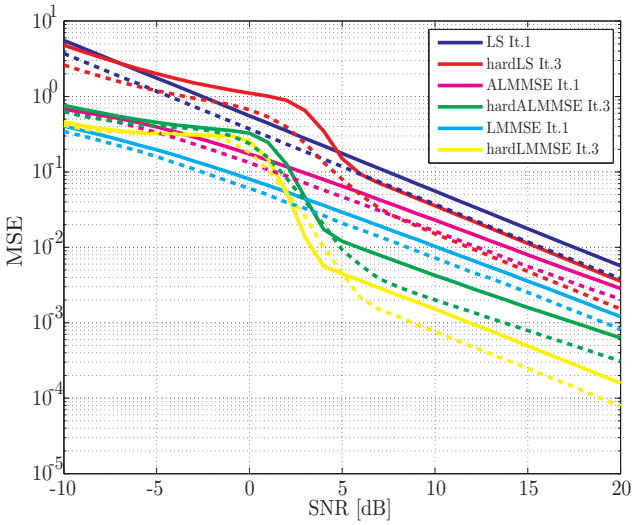

Figure 5.23: MSE comparison for hard information for 4×4 MIMO case (2×2 MIMO case with dashed curves)

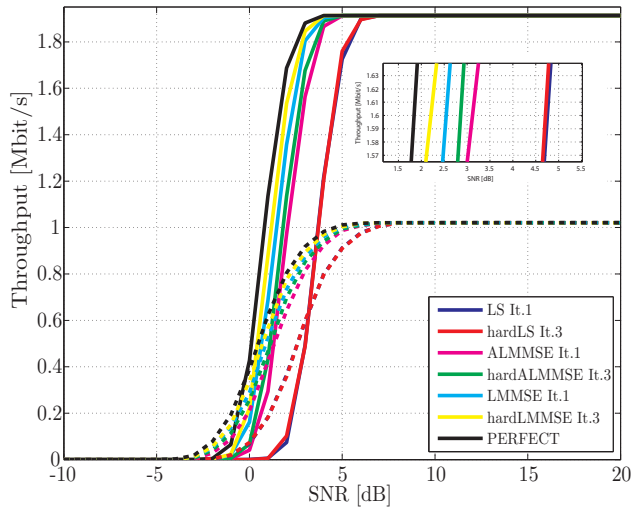

Figure 5.24: Throughput comparison for hard information for 4×4 MIMO case (2×2 MIMO case with dashed curves)

SNR gain values for hard case				
Channel estimator	Init. LS	Init. ALMMSE	Init. LMMSE	Perfect
Iter. LS	0.05 dB	-	-	-
Iter. ALMMSE	1.9 dB	0.25 dB	-	-
Iter. LMMSE	2.5 dB	0.9 dB	0.3 dB	-

Table 5.8: Summary of SNR gain values for hard case (4×4 MIMO case)

5.2.3 Hard versus Soft Feedback Information

Figure 5.25 depicts the throughput for hard and soft case after the 3rd iteration. The improvement of the soft case over the hard case for iterative LS is about 0.3 dB, soft iterative ALMMSE improves about 0.7 dB, and the soft iterative LMMSE improves approximately 0.3 dB.

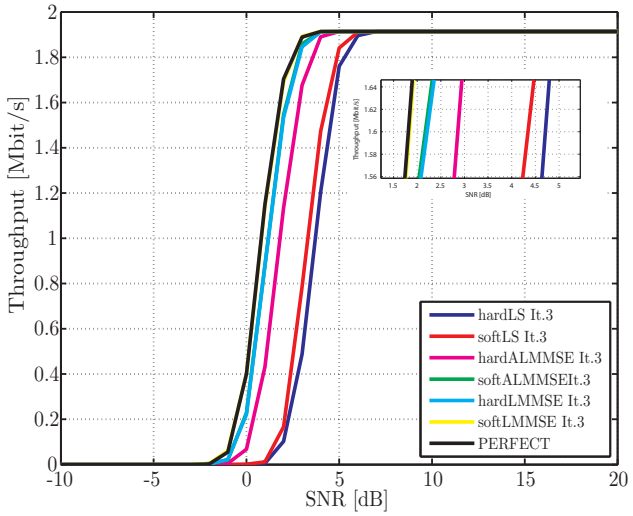

Figure 5.25: Throughput for hard versus soft information for MIMO case

SNR loss values for hard case				
Channel estimator	Init. LS	Init. ALMMSE	Init. LMMSE	Perfect
Iter. LS	-	1.6 dB	2.1 dB	2.8 dB
Iter. ALMMSE	-	-	0.3 dB	1 dB
Iter. LMMSE	-	-	-	0.35 dB

Table 5.9: Summary of SNR loss values for hard case (4×4 MIMO case)

5.3 Analysis and Conclusion

In this chapter, we analysed the iterative channel estimation algorithms based on simulations. We investigated them for different feedback information provided by the decoder. Soft information like a-posteriori and extrinsic or hard information can be chosen as feedback information. For the given simulation settings (Table 5.1), the simulations in all cases and scenarios analysed have shown that the soft case outperforms the hard case. Simulations also

have shown that a-posteriori information delivers the best result for iterative LS channel estimate, and extrinsic case performs slightly better for iterative ALMMSE and LMMSE than the a-posteriori case. As stated in Table 5.1, all simulations so far are obtained using 4-QAM as modulation scheme, with CQI = 4. Under the assumption made at the beginning of Chapter 4 for the iterative approaches, it is obvious that increasing the modulation order and/or the ECR (cf. Section 3.3.1, Equation (3.5) and Figure 3.5) the error propagation becomes more severe, which degrades the channel estimation performance and thus the overall system performance. To show how the iterative approaches behave at higher modulation orders some simulations are also carried out for 16-QAM ($\mathrm{CQI} = 7, \mathrm{ECR} = 378$) and 64-QAM ($\mathrm{CQI} = 10, \mathrm{ECR} = 466$) for 2×2 OLSM and TxD schemes. Figure 5.26 and Figure 5.27 depict the MSE and throughput over SNR for soft 2×2 OLSM MIMO scheme for 4-QAM, 16-QAM and 64-QAM. It is obvious that for higher-order modulation schemes the a-priori information is less reliable and the error propagation impacts the overall system performance. Therefore, the iterative approaches for modulation orders 16-QAM and 64-QAM for OLSM scheme perform worse than the initial approaches. On the other side, it is observed that decreasing the ECR the iterative approaches will improve the overall system performance also for 16-QAM and 64-QAM modulation orders. This is because lower code rates can generally correct more channel errors than higher code rates and thus are more energy efficient. However, higher code rates are more bandwidth efficient than the lower code rates because the amount of overhead (in form of parity bits) is lower.

In TxD case, the same data are sent from all transmit antenna ports. The use of multiple spatially differentiated signals increases the chances that data lost due to poor SNR will be different from each channel. Figure 5.28 and Figure 5.29 depicts the MSE and throughput over SNR for 2×2 TxD MIMO scheme for 4-QAM, 16-QAM and 64-QAM. From the MSE plot, we can see that for low SNR values, increasing the modulation order the MSE for iterative approaches increases, but as the SNR values further increase the MSE decreases considerably for all modulation orders with respect to the initial channel estimation. On the other hand, the overall system performance has been improved for all modulation orders and SNR gains are observed with respect to initial channel estimation.

Based on the above analysis and under the assumption made at the beginning of Chapter 4, we come to the conclusion that the iterative approaches are suitable for 4-QAM modulation order for all transmission schemes. Thus, they improve the overall system performance with respect to initial channel esti-

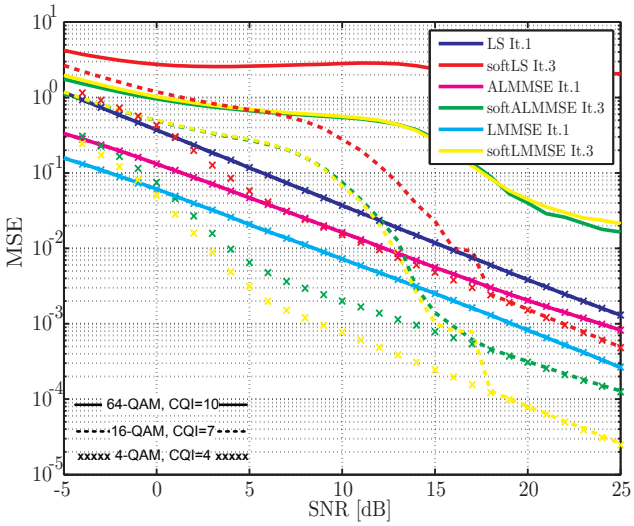

Figure 5.26: MSE for soft 2×2 OLSM scheme for 4-QAM, 16-QAM and 64-QAM

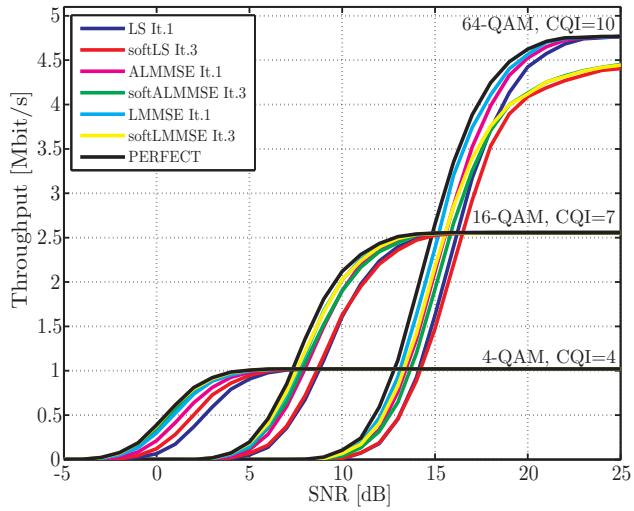

Figure 5.27: Throughput for soft 2×2 OLSM scheme for 4-QAM, 16-QAM and 64-QAM

mation. For 16-QAM and 64-QAM modulation orders, iterative approaches show to improve the system performance only for TxD transmission scheme and show worse performance for OLSM scheme with respect to the initial channel estimation.

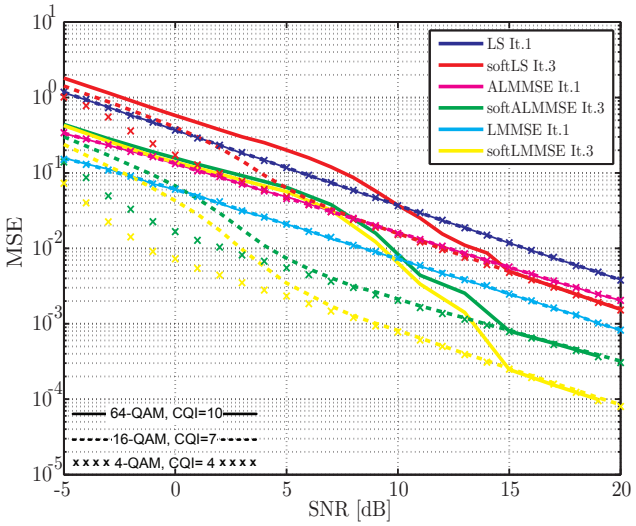

Figure 5.28: MSE for soft 2×2 TxD scheme for 4-QAM, 16-QAM and 64-QAM

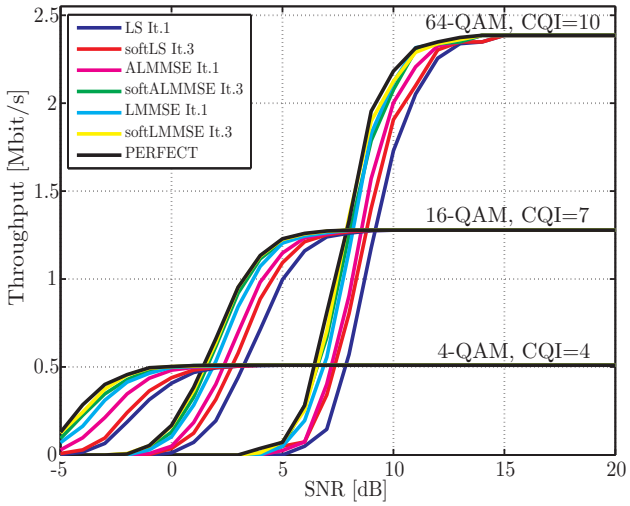

Figure 5.29: Throughput for soft 2×2 TxD scheme for 4-QAM, 16-QAM and 64-QAM

Chapter 6

Conclusions

In this thesis, different iterative channel estimation algorithms for Long Term Evolution (LTE) downlink are investigated. The channel estimators are derived for slowly changing channels following the assumption that channel impulse response remains constant during the transmission of one subframe. This thesis focuses on answering the question of whether channel estimate based on pilot symbols are sufficiently accurate to achieve high data rate transmission. Channel estimate can be further enhanced, if besides pilot symbols, additional information about data symbols can be exploited by the channel estimator. In order to take advantage of such information the conventional receiver has to be modified. The decoding unit, as the most important component of the receiver, is also modified and new component such as the soft symbol mapper is added to the receiver. The modified decoding unit, besides hard decoded bits, outputs also soft estimated coded bits. This feedback information is utilized by the soft symbol mapper to calculate the estimated data symbols, which in turn, together with the pilot symbols are applied by the channel estimators to re-estimate the channel. In the first iteration the channel is estimated only at the pilot symbols, then the detector exploits this channel state information to detect the symbols and provide the decoding unit with soft information for each coded bit. From the second iteration on, the channel estimator avails both the pilot symbols and hard or soft estimated data symbols and re-estimates the channel. The impact of processing either hard or soft feedback information on channel estimation is investigated, and based on simulations the soft feedback information outperforms the hard feedback information for all iterative channel estimation algorithms.

The channel estimators come with different complexities and performance behaviours. First, we derive the iterative Least Squares (LS) channel estima-

tor that does not make any assumption about statistical channel properties. Initial LS channel estimate is calculated only at the pilot symbols. From the second iteration on, the channel is re-estimated and enhanced with respect to the initial channel estimation. For SISO (4×4 MIMO) case and soft feedback information, iterative LS estimator after the third iteration improves approximately 0.8 dB (0.45 dB) with respect to initial LS estimate and loses about 1.2 dB (2.5 dB) with respect to the system with perfect channel knowledge.

Considering the statistical channel properties, iterative Linear Minimum Mean Square Error (LMMSE) channel estimator is derived which improves the performance of the iterative LS estimator. Perfect knowledge of channel autocorrelation is assumed. For SISO (4×4 MIMO) case and soft feedback information, iterative LMMSE improves approximately 0.45 dB (0.7 dB) with respect to the initial channel estimation. An SNR loss of approximately 0.05 dB with respect to the system with perfect channel knowledge is observed. Thus, the performance of iterative LMMSE channel estimator approximately achieves the performance of the system with perfect channel knowledge.

Despite the fact that the performance of the iterative LMMSE estimator is excellent, its complexity is too high for a real-time implementation. Therefore, the goal of iterative Approximate Linear Minimum Mean Square Error (ALMMSE) estimator is to reduce the complexity of the iterative LMMSE estimator and in turn preserve its performance. The iterative ALMMSE estimator exploits only the correlation between the L closest subcarriers. Thus, there is a trade-off between complexity and performance by varying the parameter L. For SISO (4×4 MIMO) case and for a chosen $L = 12$, iterative ALMMSE channel estimator gains about 0.7 dB (0.9 dB) with respect to initial channel estimation, and loses 0.2 dB (0.35 dB) compared to iterative LMMSE channel estimator.

Appendix A

64-QAM Soft Symbols

Symbol constellation for 64-Quadrature Amplitude Modulation (QAM) according to [7] is depicted in Figure A.1. There are M=64 symbols and six bits $(f_1, f_2, f_3, f_4, f_5, f_6)$ are mapped to a constellation symbol. As in the 4-QAM and 16-QAM case, any Gray-mapped QAM constellation symbol can be divided into real and imaginary part [16]. Therefore, the 1st, 3rd and 5th bit (f_1, f_3, f_5) are mapped to the real part and the 2nd, 4th and 6th bit (f_2, f_4, f_6) to the imaginary part of a 64-QAM constellation symbol. This implies that the real and imaginary components of each soft symbol $\tilde{s}_{d,n_\mathrm{t},n_\mathrm{s},n_k} = \mathrm{Re}(\tilde{s}_{d,n_\mathrm{t},n_\mathrm{s},n_k}) + j\mathrm{Im}(\tilde{s}_{d,n_\mathrm{t},n_\mathrm{s},n_k})$ are independent. Therefore, for the real $\mathrm{Re}(\tilde{s}_{d,n_\mathrm{t},n_\mathrm{s},n_k})$ and the imaginary part $\mathrm{Im}(\tilde{s}_{d,n_\mathrm{t},n_\mathrm{s},n_k})$ we have

$$\begin{aligned}
\mathrm{Re}(\tilde{s}_{d,n_\mathrm{t},n_\mathrm{s},n_k}) = {} & 1 \cdot \mathrm{P}(f_1 = 0)P(f_3 = 0)P(f_5 = 1) \\
& + 3 \cdot \mathrm{P}(f_1 = 0)P(f_3 = 0)P(f_5 = 0) \\
& + 5 \cdot \mathrm{P}(f_1 = 0)P(f_3 = 1)P(f_5 = 0) \\
& + 7 \cdot \mathrm{P}(f_1 = 0)P(f_3 = 1)P(f_5 = 1) \\
& - 1 \cdot \mathrm{P}(f_1 = 1)P(f_3 = 0)P(f_5 = 1) \\
& - 3 \cdot \mathrm{P}(f_1 = 1)P(f_3 = 0)P(f_5 = 0) \\
& - 5 \cdot \mathrm{P}(f_1 = 1)P(f_3 = 1)P(f_5 = 0) \\
& - 7 \cdot \mathrm{P}(f_1 = 1)P(f_3 = 1)P(f_5 = 1),
\end{aligned} \tag{A.1}$$

and

$$\begin{aligned}\mathrm{Im}(\tilde{s}_{d,n_\mathrm{t},n_\mathrm{s},n_k}) = {} & 1 \cdot \mathrm{P}(f_1 = 0)P(f_3 = 0)P(f_5 = 1) \\ & + 3 \cdot \mathrm{P}(f_1 = 0)P(f_3 = 0)P(f_5 = 0) \\ & + 5 \cdot \mathrm{P}(f_1 = 0)P(f_3 = 1)P(f_5 = 0) \\ & + 7 \cdot \mathrm{P}(f_1 = 0)P(f_3 = 1)P(f_5 = 1) \\ & - 1 \cdot \mathrm{P}(f_1 = 1)P(f_3 = 0)P(f_5 = 1) \\ & - 3 \cdot \mathrm{P}(f_1 = 1)P(f_3 = 0)P(f_5 = 0) \\ & - 5 \cdot \mathrm{P}(f_1 = 1)P(f_3 = 1)P(f_5 = 0) \\ & - 7 \cdot \mathrm{P}(f_1 = 1)P(f_3 = 1)P(f_5 = 1).\end{aligned} \tag{A.2}$$

Inserting the expressions for bit probabilities from Equation (3.16) and Equation (3.17) into Equation (A.1) and Equation (A.2), and after some calculations the real and imaginary parts can be written in terms of tangent hyperbolic functions

$$\begin{aligned}\mathrm{Re}(\tilde{s}_{d,n_\mathrm{t},n_\mathrm{s},n_k}) = {} & \\ & - 4\tanh\left(\frac{\mathrm{L}(f_1)}{2}\right) - 2\tanh\left(\frac{\mathrm{L}(f_1)}{2}\right)\tanh\left(\frac{\mathrm{L}(f_3)}{2}\right) \\ & - \tanh\left(\frac{\mathrm{L}(f_1)}{2}\right)\tanh\left(\frac{\mathrm{L}(f_3)}{2}\right)\tanh\left(\frac{\mathrm{L}(f_5)}{2}\right),\end{aligned} \tag{A.3}$$

and

$$\begin{aligned}\mathrm{Im}(\tilde{s}_{d,n_\mathrm{t},n_\mathrm{s},n_k}) = {} & \\ & - 4\tanh\left(\frac{\mathrm{L}(f_2)}{2}\right) - 2\tanh\left(\frac{\mathrm{L}(f_2)}{2}\right)\tanh\left(\frac{\mathrm{L}(f_4)}{2}\right) \\ & - \tanh\left(\frac{\mathrm{L}(f_2)}{2}\right)\tanh\left(\frac{\mathrm{L}(f_4)}{2}\right)\tanh\left(\frac{\mathrm{L}(f_6)}{2}\right).\end{aligned} \tag{A.4}$$

Therefore, the soft estimated symbols for 64-QAM are obtained as

$$\tilde{s}_{d,n_\mathrm{t},n_\mathrm{s},n_k} = \frac{\mathrm{Re}(\tilde{s}_{d,n_\mathrm{t},n_\mathrm{s},n_k}) + j\mathrm{Im}(\tilde{s}_{d,n_\mathrm{t},n_\mathrm{s},n_k})}{\sqrt{42}}\,, \tag{A.5}$$

where, after inserting the Equation (A.3) and the Equation (A.4) into the Equation (A.5) we obtain the soft estimated symbols for 64-QAM.

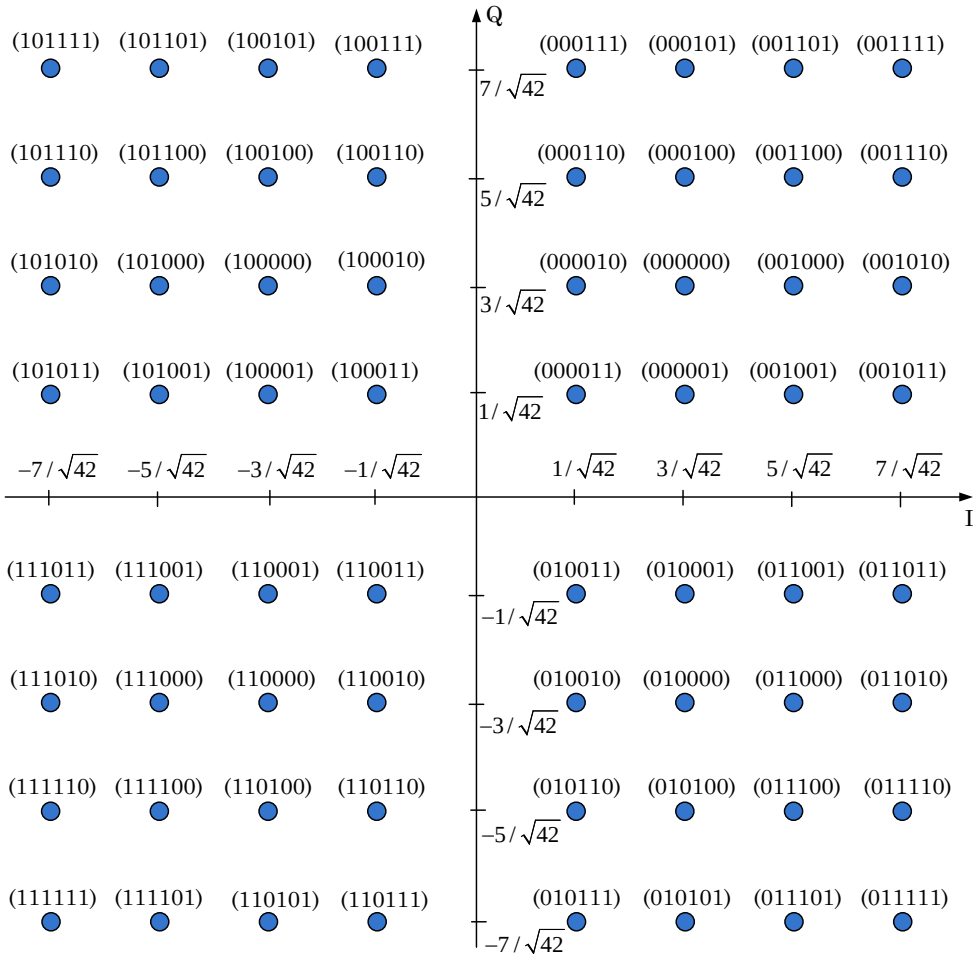

Figure A.1: 64-QAM signal constellation

Appendix B

Acronyms

3GPP	3rd Generation Partnership Project
ALMMSE	Approximate Linear Minimum Mean Square Error
CB	Code Block
CLSM	Closed Loop Spatial Multiplexing
CP	Cyclic Prefix
CQI	Channel Quality Indicator
CRC	Cyclic Redundancy Check
EDGE	Enhanced Data rates for GSM Evolution
ECR	Effective Code Rate
FFT	Fast Fourier Transform
GPRS	General Packet Radio Service
GSM	Global System for Mobile communications
HSPA	High Speed Packet Access
IFFT	Inverse Fast Fourier Transform
LLRs	Log-Likelihood Ratios
LS	Least Squares
LTE	Long Term Evolution
LMMSE	Linear Minimum Mean Square Error
MIMO	Multiple Input Multiple Output
MSE	Mean Square Error
OFDM	Orthogonal Frequency Division Multiplexing
OLSM	Open Loop Spatial Multiplexing
PCCC	Parallel Concatenated Convolutional Code
QAM	Quadrature Amplitude Modulation
SISO	Single Input Single Output
SM	Spatial Multiplexing

SNR Signal to Noise Ratio
TB Transport Block
TxD Transmit Diversity
W-CDMA Wideband Code Division Multiple Access

Bibliography

[1] S. Sesia, I. Toufik, and M. Baker, **LTE - The UMTS Long Term Evolution: From Theory to Practice**. John Wiley Sons Ltd., 2009.

[2] "**Austrian Regulatory Authority for Broadcasting and Telecommunications - Frequency Allocation**." [Online]. Available: http://www.rtr.at/en/tk/Frequenzen

[3] M. Simko, D. Wu, C. Mehlführer, J. Eilert, and D. Liu, "**Implementation aspects of channel estimation for 3GPP LTE terminals**," in *Proc. European Wireless 2011*, Vienna, April 2011.

[4] C. Studer, M. Wenk, A. Burg, and H. Bölcskei, "**Soft-output sphere decoding: Performance and implementation aspects**," in *Proc. of Asilomar Conf. on Signals, Systems, and Computers*, pp. 2071–2076, Nov. 2006. [Online]. Available: http://www.nari.ee.ethz.ch/commth/pubs/p/asilomar06

[5] L. Á. M. R. de Temiño, C. N. i Manchon, C. Rom, T. B. Sørensen, and P. E. Mogensen, "**Iterative channel estimation with robust wiener filtering in LTE downlink**," in *VTC Fall*, pp. 1–5, 2008.

[6] L. Boher, R. Legouable, and R. Rabineau, "**Performance analysis of iterative receiver in 3GPP/LTE DL MIMO OFDMA system**," in *Proc. IEEE 10th International Symposium on Spread Spectrum Techniques and Applications 2008 (ISSSTA 2008)*, pp. 103–108, Aug 2008.

[7] 3GPP, "**Evolved Universal Terrestrial Radio Access (E-UTRA); Physical channels and modulation**," 3rd Generation Partnership Project (3GPP), TS 36.211, Sept. 2008. [Online]. Available: http://www.3gpp.org/ftp/Specs/html-info/36211.htm

[8] C. Mehlführer, M. Wrulich, J. C. Ikuno, D. Bosanska, and M. Rupp, "**Simulating the long term evolution physical layer**," in *Proc. 17th European Signal Processing Conference (EUSIPCO 2009)*, pp. 1471–1478, Glasgow, Scotland, August 2009.

[9] 3GPP, "**Evolved Universal Terrestrial Radio Access (E-UTRA); Multiplexing and channel coding**," 3rd Generation Partnership Project (3GPP), TS 36.212, Mar. 2008. [Online]. Available: http://www.3gpp.org/ftp/Specs/html-info/36212.htm

[10] L. Hanzo, T. Liew, B. Yeap, R. Tee, and S. Ng, **Turbo Coding, Turbo Equalisation and Space-Time Coding: EXIT-Chart-Aided Near-Capacity Designs for Wireless Channels**, ser. Wiley - IEEE. John Wiley & Sons, 2011. [Online]. Available: http://books.google.at/books?id=i0gRC0wGVFgC

[11] J. C. Ikuno, S. Schwarz, and M. Simko, "**LTE rate matching performance with code block balancing**," in *Proc. European Wireless 2011*, Vienna, April 2011.

[12] J. C. Ikuno, M. Wrulich, and M. Rupp, "**Performance and modeling of LTE h-ARQ**," in *International ITG Workshop on Smart Antennas (WSA 2009)*, pp. 1–6, Berlin, Germany, February 2009.

[13] "**Iterative Solutions Coded Modulation Library ISCML**." [Online]. Available: http://www.iterativesolutions.com/

[14] "**LTE Link Level Simulator**." [Online]. Available: http://www.nt.tuwien.ac.at/ltesimulator/

[15] A. Papoulis, **Probability, Random Variables, and Stochastic Processes**. Mc-Graw Hill, 1984.

[16] J. Liu, H. Vanhaute, M. Moonen, A. Bourdoux, and H. De Man, "**Efficient computation of symbol statistics from bit a priori information in turbo receivers**," *Trans. Comm.*, vol. 57, no. 7, pp. 1889–1891, July 2009, doi: 10.1109/TCOMM.2009.07.050241. [Online]. Available: http://dx.doi.org/10.1109/TCOMM.2009.07.050241

[17] A. Wilzeck and T. Kaiser, "**Antenna subset selection for cyclic prefix assisted MIMO wireless communications over frequency selective channels**," *EURASIP J. Adv. Signal Process*, vol. 2008, pp. 1–14, 2008, doi: http://dx.doi.org/10.1155/2008/716826.

[18] X. Hou, X. Zhao, C. Yin, and G. Yue, "**Unified view of channel estimation in mimo-ofdm systems**," in *Wireless Communications, Networking and Mobile Computing, 2005. Proceedings. 2005 International Conference on*, vol. 1, pp. 54 – 58, sept. 2005, doi: 10.1109/WCNM.2005.1543977.

[19] S. M. Kay, **Fundamentals of statistical signal processing: estimation theory**. Upper Saddle River, NJ, USA: Prentice-Hall, Inc., 1993.

[20] N. Sun, T. Ayabe, and T. Nishizaki, "**Efficient spline interpolation curve modeling**," in *Proc. Third International Conference on Intelligent Information Hiding and Multimedia Signal Processing, 2007 (IIHMSP 2007)*, vol. 2, pp. 59–62, Nov. 2007, doi: 10.1109/IIHMSP.2007.4457653.

[21] J. Lei, C. Dazhong, and H. Chunping, "**Two novel transform domain estimation methods for OFDM system and their application environment**," in *Proc. Canadian Conference on Electrical and Computer Engineering (CCECE 2004)*, vol. 1, pp. 377–380 Vol.1, May 2004.

[22] J. J. van de Beek, O. Edfors, M. Sandell, S. K. Wilson, and P. O. Borjesson, "**On channel estimation in OFDM systems**," in *Proc. IEEE 45th Vehicular Technology Conference (VTC 1995)*, vol. 2, pp. 815–819, 1995, doi: 10.1109/VETEC.1995.504981. [Online]. Available: http://dx.doi.org/10.1109/VETEC.1995.504981

[23] C. Mehlführer, S. Caban, and M. Rupp, "**An accurate and low complex channel estimator for OFDM WiMAX**," in *Proc. Third International Symposium on Communications, Control, and Signal Processing (ISCCSP 2008)*, pp. 922–926, St. Julians, Malta, mar 2008. [Online]. Available: http://publik.tuwien.ac.at/files/pub-et_13650.pdf

[24] ITU, "**Recommendation ITU-R M.1225: Guidelines for evaluation of radio transmission technologies for IMT- 2000 systems**," International Telecommunication Union, Recommendation ITU-R M.1225, 1998.