Implementation Aspects of Channel Estimation for 3GPP LTE Terminals

Article · January 2011

CITATIONS
54

READS
571

5 authors, including:

di wu
Universidad Nacional de Itapúa
27 PUBLICATIONS **332** CITATIONS
SEE PROFILE

Christian Mehlfuehrer
TU Wien
64 PUBLICATIONS **1,567** CITATIONS
SEE PROFILE

Dake Liu
Hainan University
224 PUBLICATIONS **2,166** CITATIONS
SEE PROFILE

Implementation Aspects of Channel Estimation for 3GPP LTE Terminals

Michal Šimko*, Di Wu†, Christian Mehlführer*, Johan Eilert‡ and Dake Liu†
*Institute of Telecommunications, Vienna University of Technology, Vienna, Austria
†Department of Electrical Engineering, Linköping University, Linköping, Sweden
‡ST-Ericsson AT AB, Lund, Sweden
Contact: msimko@nt.tuwien.ac.at

***Abstract*—In this paper, hardware implementation aspects of the channel estimator in 3GPP LTE terminals are investigated. A channel estimation ASIC, which handles the real-time channel estimation, is presented. Compared to traditional correlator-based channel estimators, the channel estimator presented boosts the throughput at feasible silicon cost by adopting a recently proposed estimation method named Approximate Linear Minimum Mean Square Error (ALMMSE). In this paper, both the architecture and VLSI implementation of the estimator are elaborated. Implemented using a 65 nm CMOS process, the channel estimator supports the full 20 MHz bandwidth of 3GPP LTE and consumes only 49 kgates.**

I. Introduction

The 3rd Generation Partnership Project Long-Term Evolution (3GPP LTE) is an emerging mobile broadband communication system that offers high-speed mobile data service. LTE adopts Orthogonal Frequency Division Multiple Access (OFDMA) and Multiple-Input Multiple-Output (MIMO) to achieve a peak bit-rate target of 326 Mbit/s in the downlink (assuming a 4×4 MIMO system with 20 MHz bandwidth, 64 QAM, coding rate 1 and 19% pilot symbol overhead). The performance gain of MIMO heavily depends on the accurate estimation of Channel State Information (CSI), which is crucial for every communications system. As addressed in [1], the complexity of MIMO channel estimation is infeasible for most low-complexity receivers in practice. Hence, the pilots (reference signals) introduced in LTE have been chosen orthogonal to allow low-complexity channel estimation for multi-antenna transmissions. In this work, a channel estimation ASIC is implemented as an accelerator for 3GPP LTE modem platforms to handle real-time channel estimation with different antenna configurations and mobility.

In [9], a practical implementation of a general MIMO OFDM transceiver is presented, including a simple least squares channel estimator. Hardware implementation aspects of a Singular Value Decomposition (SVD) based channel estimator are described in [10]. In [11], authors present an implementation of an LTE terminal, however the presented channel estimator is very simple and its performance is not sufficient. In contrast to the cited works, we utilize a fully standard compliant LTE physical layer link, including the standardized pilot structure. Therefore, our results are representative for:

1) throughput performance,
2) channel estimation performance
3) and implementation complexity

of feasible LTE channel estimators.

The remainder of the paper is organized as follows. In Sec. II, the LTE system model is presented in brief. Sec. III introduces several channel estimation algorithms evaluated in this paper. Sec. IV elaborates the architecture of the channel estimator. In Sec. VI, link-level simulation results with fixed-point datatypes are presented to quantitatively justify the design trade-off between performance and complexity. The ASIC implementation is presented in Sec. VII. Finally, Sec. VIII concludes the paper.

II. System Model

A. Overview

As defined in [4], the duration of an LTE frame is $T_{\text{frame}} = 10\,\text{ms}$. Each frame consists of ten subframes and each subframe of two slots. LTE specifies two different cyclic prefix lengths (normal and extended) to handle different delay spreads. Accordingly, to obtain a constant slot duration of 0.5 ms, each slot contains seven OFDM symbols in case of normal cyclic prefix transmission and six OFDM symbols in case of extended cyclic prefix transmission. A so-called resource block (RB) consists of twelve adjacent subcarriers over seven (six) OFDM symbols.

B. Reference Signal Structure

As in many other OFDM systems (e.g. DVB-T), pilot symbols (called reference signals in LTE) are inserted during subcarrier mapping in both time and frequency directions such that the receiver can estimate time-variant radio channels. One major enhancement in LTE is the introduction of multiple antennas. Being different from the typical "MIMO-OFDM" channel estimation problems [1], [8], [12] in academia, the reference signals transmitted from multiple antennas are orthogonal to each other. As a consequence the channel impulse response between different Tx-Rx antenna pairs can be estimated individually. We therefore consider in our system model only a single transmit and a single receive antenna. The received OFDM symbol $\mathbf{y}$ at one receive antenna can be written as

$$\mathbf{y} = \mathbf{X}\mathbf{h} + \mathbf{w}, \tag{1}$$

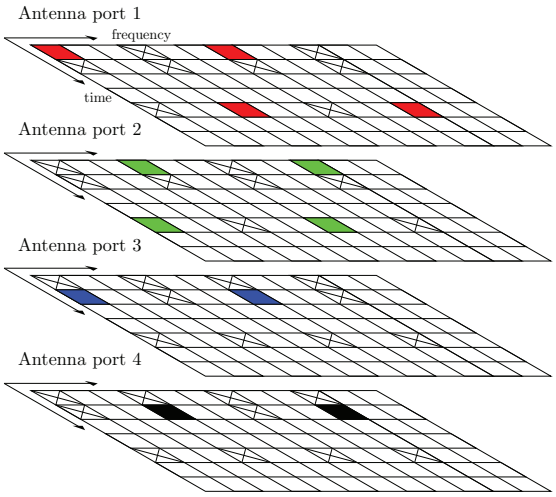

Fig. 1: Reference Signal Mapping in LTE

where the vector $\mathbf{h}$ contains the channel coefficients in the frequency domain and $\mathbf{w}$ is additive zero mean white Gaussian noise with variance σ_w^2 at the receive antenna. The matrix $\mathbf{X}$ comprises permuted data symbols $\mathbf{x}_\mathrm{d}$ and pilot symbols $\mathbf{x}_\mathrm{p}$ on the main diagonal. The permutation is given by the permutation matrix $\mathbf{P}$:

$$\tilde{\mathbf{x}} = \left[\mathbf{x}_\mathrm{p}^\mathrm{T}\, \mathbf{x}_\mathrm{d}^\mathrm{T}\right]^\mathrm{T}, \tag{2}$$

$$\mathbf{x} = \mathbf{P}\,\tilde{\mathbf{x}}, \tag{3}$$

$$\mathbf{X} = \mathrm{diag}\left(\mathbf{x}\right). \tag{4}$$

The vectors $\mathbf{y}$, $\mathbf{h}$ and $\mathbf{w}$ in Eq. (1) can be divided according to Eq. (2) into two parts:

1) a part corresponding to the pilot symbol positions,
2) a part corresponding to the remaining data symbol positions.

III. Channel Estimation in LTE

In this section, the different types of channel estimation techniques considered in this paper are explained.

A. LS

The Least-Squares (LS) channel estimator for subcarriers on which pilot symbols are located, is given by

$$\hat{\mathbf{h}}_\mathrm{p}^\mathrm{LS} = \mathbf{X}_\mathrm{p}^\mathrm{H}\,\mathbf{y}_\mathrm{p}. \tag{5}$$

The remaining channel coefficients have to be obtained by interpolation. In this work, we apply linear interpolation.

B. LMMSE

The Linear Minimum-Mean-Square-Error (LMMSE) channel estimator requires the knowledge of the second order statistics of the channel and the noise. It performs better than the LS estimator, but it requires higher computational complexity. The LMMSE channel estimate can be obtained by filtering the LS estimate

$$\hat{\mathbf{h}}^\mathrm{LMMSE} = \mathbf{R}_{\mathbf{h},\mathbf{h}_\mathrm{p}}\left(\mathbf{R}_{\mathbf{h}_\mathrm{p},\mathbf{h}_\mathrm{p}} + \sigma_w^2\mathbf{I}\right)^{-1}\hat{\mathbf{h}}_\mathrm{p}^\mathrm{LS}, \tag{6}$$

where the $\mathbf{R}_{\mathbf{h}_\mathrm{p},\mathbf{h}_\mathrm{p}}$ is the autocorrelation matrix of the channel at the pilot symbols position and $\mathbf{R}_{\mathbf{h},\mathbf{h}_\mathrm{p}}$ is the crosscorrelation matrix between the channel at the data symbol positions and the channel at the pilot symbol position.

C. ALMMSE

The performance of the LMMSE estimator is in general superior to that of the LS estimator [1] at the cost of higher computational complexity because of the matrix inversion in Eq. (6). In a real-time implementation, a reduction of complexity is desired while preserving the performance of the LMMSE estimator. In this section, we discuss a low complexity estimator originally proposed in [2], where the authors applied this estimator in WiMAX. The main difference between LTE and WiMAX from the channel estimation point of view is that LTE utilizes distributed pilot symbols for the channel estimation instead of a preamble utilized in WiMAX [4], [5]. Consequently, the ALMMSE estimator presented in [2] has to be adopted for the application in LTE.

The two main ideas of the ALMMSE estimator in [2] are:

1) Calculate the LMMSE filtering matrix by using only the correlation between L neighboring subcarriers instead of the full correlation between all subcarriers as in the case of LMMSE estimation.
2) Assume that the correlation is frequency independent and estimate a full rank $L \times L$ autocorrelation matrix utilizing the LS channel estimate.

The ALMMSE algorithm adapted for LTE consists of the following steps:

1) Choose the correlation length L that defines the dimension of $\hat{\mathbf{R}}_\mathbf{h}^{(L)}$ to be $L \times L$. Due to the pilot structure in LTE, L is bounded by $3 \leq L \leq K_\mathrm{sub}$ (K_sub is the number of subcarriers). A small L is generally desirable from the complexity point of view. However, with increasing L also the performance of the estimator will improve. If $L = K_\mathrm{sub}$ is chosen, the ALMMSE estimator is equal to the LMMSE estimator.
2) Choose the interval $\mathcal{I}^k$ of L consecutive subcarrier indices according to the following rule (k is the subcarrier index of the channel coefficient to be estimated):

$$\mathcal{I}^k = \begin{cases} [1,\ldots,L] & ;k \leq \frac{L+1}{2} \\ \left[k - \lfloor\frac{L-1}{2}\rfloor,\ldots,k+\lceil\frac{L-1}{2}\rceil\right] & ;\text{otherwise} \\ [K_\mathrm{sub}-L+1,\cdots,K_\mathrm{sub}] & ;k \geq K_\mathrm{sub} - \frac{L-1}{2} \end{cases} \tag{7}$$

Let $\mathbf{h}^{(\mathcal{I}^k)}$ be the channel vector for the subcarriers from the chosen interval $\mathcal{I}^k$

$$\mathbf{h}^{(\mathcal{I}^k)} = \left[h_{\mathcal{I}^k(1)},\ldots,h_{\mathcal{I}^k(L)}\right]^\mathrm{T}. \tag{8}$$

3) Find the $K_\mathrm{p}^{(L)} = \lfloor\frac{L}{3}\rfloor$ subcarriers on which the pilot symbols are located within the chosen interval $\mathcal{I}^k$. Let

$\mathbf{h}_{\mathrm{p}}^{(\mathcal{I}^k)}$ be the vector of channel coefficients on the pilot symbol positions.

4) Create a permutation matrix $\mathbf{P}$ of dimension $L \times L$ with

$$\tilde{\mathbf{h}}^{(\mathcal{I}^k)} = \left[{\mathbf{h}_{\mathrm{p}}^{(\mathcal{I}^k)}}^{\mathrm{T}} {\mathbf{h}_{\mathrm{d}}^{(\mathcal{I}^k)}}^{\mathrm{T}} \right]^{\mathrm{T}} = \mathbf{P}^{\mathrm{T}} \mathbf{h}^{(\mathcal{I}^k)}, \tag{9}$$

where $\mathbf{h}_{\mathrm{d}}^{(\mathcal{I}^k)}$ is the channel vector on the data positions within the chosen interval $\mathcal{I}^k$.

5) Permute $\hat{\mathbf{R}}_{\mathbf{h}}^{(L)}$ with $\mathbf{P}$

$$\tilde{\hat{\mathbf{R}}}_{\mathbf{h}}^{(L)} = \mathbf{P}^{\mathrm{T}} \hat{\mathbf{R}}_{\mathbf{h}}^{(L)} \mathbf{P}. \tag{10}$$

6) Extract $\tilde{\hat{\mathbf{R}}}_{\mathbf{h}_{\mathrm{LS}}}^{(L)}$ and $\tilde{\hat{\mathbf{R}}}_{\mathbf{h},\mathbf{h}_{\mathrm{LS}}}^{(L)}$ from $\tilde{\hat{\mathbf{R}}}_{\mathbf{h}}^{(L)}$ as

$$\tilde{\hat{\mathbf{R}}}_{\mathbf{h}_{\mathrm{LS}}}^{(L)} = \left(\tilde{\hat{\mathbf{R}}}_{\mathbf{h}}^{(L)} \right)_{K_{\mathrm{p}}^{(L)}, K_{\mathrm{p}}^{(L)}}, \tag{11}$$

$$\tilde{\hat{\mathbf{R}}}_{\mathbf{h},\mathbf{h}_{\mathrm{LS}}}^{(L)} = \left(\tilde{\hat{\mathbf{R}}}_{\mathbf{h}}^{(L)} \right)_{L, K_{\mathrm{p}}^{(L)}}. \tag{12}$$

(The operator $(\mathbf{A})_{M,N}$ creates a submatrix of matrix $\mathbf{A}$ which is given by the first M rows and first N columns).

7) Calculate the filtering matrix $\tilde{\mathbf{F}}^{(L)}$

$$\tilde{\mathbf{F}}^{(L)} = \tilde{\hat{\mathbf{R}}}_{\mathbf{h},\mathbf{h}_{\mathrm{LS}}}^{(L)} \left(\tilde{\hat{\mathbf{R}}}_{\mathbf{h}_{\mathrm{LS}}}^{(L)} + \sigma_w^2 \mathbf{I} \right)^{-1}. \tag{13}$$

8) Obtain an estimate of the channel coefficients by multiplying (filtering) the LS estimate on the pilot positions from the chosen interval $\mathcal{I}^k$ with $\tilde{\mathbf{F}}^{(L)}$ and permuting (multiplying by $\mathbf{P}^{\mathrm{T}}$). Finally, the k-th element has to be selected

$$\mathbf{q} = \underbrace{\mathbf{P}^{\mathrm{T}} \tilde{\mathbf{F}}^{(L)}}_{\mathbf{F}_l} \tilde{\mathbf{h}}_{\mathrm{LS}}^{(\mathcal{I}^k)} \tag{14}$$

$$\hat{h}_{\mathrm{ALMMSE},k} = \begin{cases} [\mathbf{q}]_k & ; k \leq \frac{L+1}{2} \\ [\mathbf{q}]_{\lceil \frac{L+1}{2} \rceil} & ; \text{otherwise} \\ [\mathbf{q}]_{L+k-K_{\mathrm{sub}}} & ; k \geq K_{\mathrm{sub}} - \frac{L-1}{2} \end{cases} \tag{15}$$

$[\mathbf{q}]_k$ means that the k-th element of vector $\mathbf{q}$ is selected. Due to the fact, that in LTE at least one pilot symbol is transmitted every third subcarrier, just three different permutations of the matrix $\tilde{\mathbf{F}}^{(L)}$ have to be calculated, if the channel is quasi-static within one OFDM symbol.

IV. Processing Flow

The algorithm presented in Sec. III-C needs further optimizations for real-time implementation. As illustrated in Fig. 2, the processing flow of the LTE channel estimator includes both LS and ALMMSE channel estimation. The first stage is the scaling in Eq. (5), which computes the channel response on pilot symbol subcarriers. ALMMSE is only used when the UE is in low-speed mode (the mode in which MIMO is more likely to be used). When the UE velocity is higher than the threshold velocity K_v, LS channel estimation is used. When the UE velocity is lower than K_v, ALMMSE estimation is used. Meanwhile, to estimate the correlation matrix $\tilde{\hat{\mathbf{R}}}_{\mathbf{h}_{\mathrm{LS}}}^{(L)}$ and $\tilde{\hat{\mathbf{R}}}_{\mathbf{h},\mathbf{h}_{\mathrm{LS}}}^{(L)}$, a number of subframes (those with subframe number $< N$ after the UE enters Connected Mode) will be used as training. During the training phase, the channel estimator works in LS mode and calculates the correlation matrices using the result of LS estimation. The updated correlation matrix is stored in a buffer and updated every subframe. The computation of $\tilde{\hat{\mathbf{R}}}_{\mathbf{h}_{\mathrm{LS}}}^{(L)}$ and $\tilde{\hat{\mathbf{R}}}_{\mathbf{h},\mathbf{h}_{\mathrm{LS}}}^{(L)}$ is only needed when the training phase is finished and the estimator enters ALMMSE mode.

In ALMMSE mode, three coefficient matrices $\mathbf{F}_1, \mathbf{F}_2, \mathbf{F}_3$ need to be computed from the correlation matrices. This involves the matrix inversion of 4×4 matrices (for 4×2 and 2×2 MIMO) assuming $L = 12$. However, such an operation is only needed when the SNR changes significantly. If the SNR estimation is subframe based, it will not change within one subframe. Thus, the major computational cost of ALMMSE estimation is the matrix multiplication in Eq. (14).

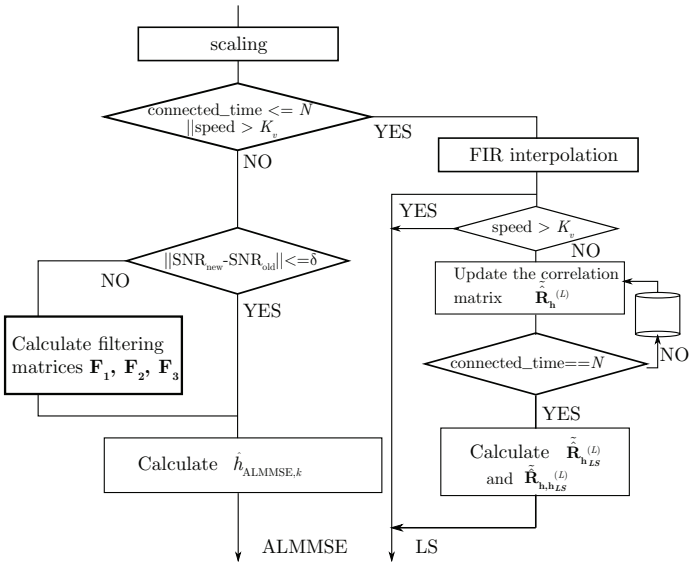

Fig. 2: Processing Flow of the LTE Channel Estimator

V. Channel Estimator Architecture

The channel estimator architecture is shown in Fig. 3. It contains two major parts, the controller unit and the Interpolation Unit (IU). Our implementation allows real-time channel estimation for a Category 4 LTE modem (with up to 4×2 spatial multiplexing and 20 MHz maximum bandwidth).

A. Controller Unit

The Controller Unit is in general finite state machine that performs matrix inversion and handles the address calculations and configuration of the IU for all tasks including the extraction of RS from the resource blocks and permutation of channel estimates. In principle, it contains an address generation unit that generates parallel addresses for multiple data that are fetched to compute the $\hat{\mathbf{H}}_{\mathrm{p}}^{\mathrm{LS}}$ in Eq. (5). As

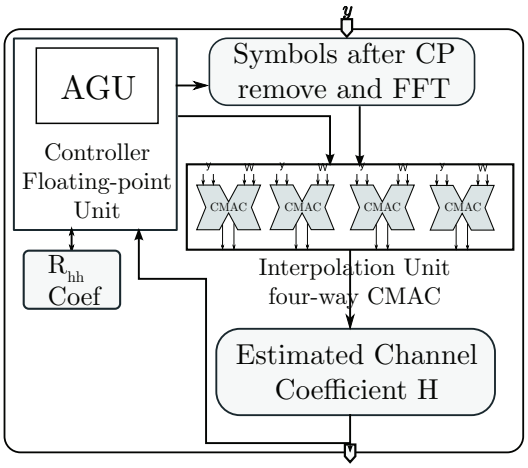

Fig. 3: Block Diagram of the Channel Estimator Implementation

presented in Sec. III-C, in order to exploit the channel statistics, ALMMSE channel estimation involves operations such as matrix inversion defined in Eq. (13). Hence, the Controller also contains a 16-bit floating point datapath (with a multiplier, an accumulator and a reciprocal unit to compute scalar inverses). The matrix inversion method presented in [6] is used to compute the updated channel autocorrelation $\mathbf{R_{hh}}$. Since the inverse is needed only when the SNR changes significantly (which is at a much lower rate compared to the symbol rate), it can easily be handled by the controller software.

B. Interpolation Unit (IU)

The IU is responsible for computing the actual $\hat{\mathbf{h}}_{\mathrm{p}}^{\mathrm{LS}}$ and the channel response at the data resource elements by interpolating $\hat{\mathbf{h}}_{\mathrm{p}}^{\mathrm{LS}}$. The main operation involved is matrix-vector multiplication. The IU mainly consists of a 16-bit fixed point four-way Single Instruction Multiple Data (SIMD) Complex Multiply-ACcumulate (CMAC) unit. Thanks to the regularity of the LTE RS locations, the interpolation coefficients can be efficiently stored in a small look-up table to reduce the hardware cost. The implementation of the four-way SIMD CMAC unit is similar to the one presented in [6].

Note that in this paper, the channel estimator is designed as a hardware accelerator to focus on the signal processing of channel estimation and make it easier to quantize the silicon cost. For more flexibility and hardware reuse, the channel estimator can also be mapped to a baseband DSP processor such as [16] in pure software.

VI. Link-level Performance

In order to evaluate the performance of the channel estimation algorithms, simulation is carried out using a standard-compliant 3GPP LTE simulator [3]. The simulator is implemented partly in Matlab and partly in C. It includes the complete physical layer signal processing such as timing/frequency synchronization [15], channel estimation, subcarrier demapping, rate-matching [14] and turbo decoding. H-ARQ [13] based on the CRC of coded blocks is also enabled to support up to three retransmissions. Ped-B [7] is selected as channel model. It is assumed that the channel is quasi-static within one OFDM symbol duration. The bandwidth is set to 5 MHz in the simulation, the velocity of user equipment is 3 km/h. The parameter L is set to 12. Perfect synchronization and ML detection is assumed to focus the simulation on channel estimation performance.

In Fig. 4, the throughput result are shown in the case a 16-bit floating point datatype is used, the performance severely degrades at high SNR compared to IEEE double precision. This is mainly due to the fact the matrices involved in the ALMMSE processing are nearly singular and require a sufficiently high numerical precision. When 64-bit floating point is used, no degradation is observed. In order to minimize the number of bits and the hardware cost, SNR under-estimation (regularization) is used, which sets a fixed σ^2 value when the SNR is higher than a threshold (12 dB in this paper). This also reduces the amount of processing needed in ALMMSE. The result shows that a 16-bit datatype with SNR under-estimation incur a negligible degradation compared to 64-bit processing without SNR under-estimation.

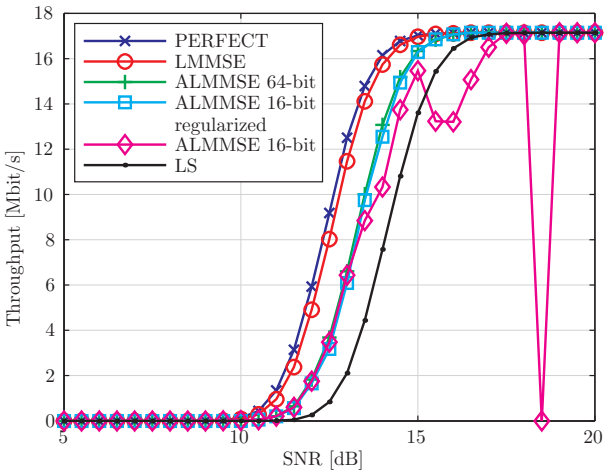

Fig. 4: Coded Throughput (rate 0.602, 16-QAM)

Fig. 5 shows the MSE for different LTE channel estimators. The LMMSE channel estimator outperforms the remaining channel estimators. However, its hardware implementation requires the most computational power. The ALMMSE channel estimator using IEEE double precision shows 4 dB SNR improvement when compared to the LS channel estimator. Unfortunately, implementation using IEEE double precision is too costly. The ALMMSE channel estimator using 16 bit implementation is at high SNR unstable. However, using SNR underestimation the performance is close the the ALMMSE with IEEE double precision. This channel estimator offer good performance-complexity tradeoff.

VII. Implementation

The channel estimator is implemented using the ST CMOS 65 nm process libraries and Synopsys low-power design flow.

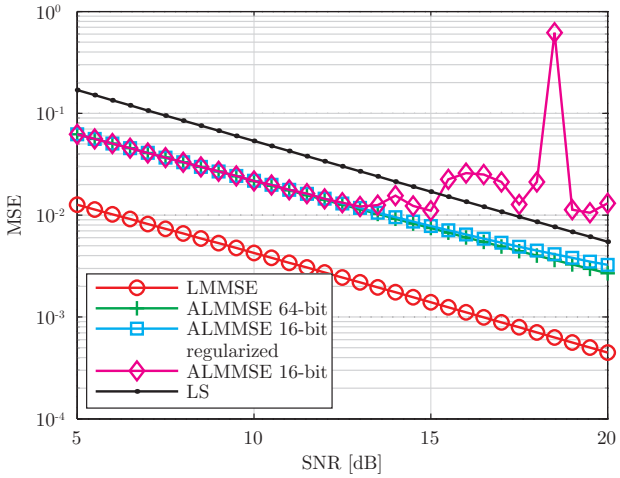

Fig. 5: MSE of the different channel estimator

Table I depicts the synthesized gate count, and working frequency.

Area of Controller (kgate)	17
Area of IU (kgate)	32
Total Area (kgate)	49
Working Frequency (MHz)	200

TABLE I: Implementation Cost Estimate

In LTE, assuming the channel estimation has to be performed every subframe (1 ms), up to 1200 channel coefficients have to be estimated per transmit-receive antenna pair to support the full 20 MHz downlink bandwidth. The proposed architecture running at 200 MHz can handle ALMMSE estimation for up to 4×2 MIMO systems in real-time.

VIII. Conclusions

The result shows that algorithm-architecture cooptimization can further simplify the ALMMSE channel estimation algorithm. A short wordlength can be used to allow a low cost ASIC implementation. With SNR under-estimation, a 16-bit floating-point datatype provides sufficient precision to support the 4×4 matrix inversion involved in ALMMSE with negligible degradation of performance. A trade-off between performance and complexity has been reached at a feasible silicon cost with a 1 dB throughput gain compared to the LS channel estimation.

IX. Acknowledgement

This work has been funded by the Christian Doppler Laboratory for Wireless Technologies for Sustainable Mobility under the supervision of Christoph Mecklenbräuker. The authors thank their industrial partners A1 Telekom Austria AG and KATHREIN-Werke KG. Furthermore, the financial support by the Federal Ministry of Economy, Family and Youth and the National Foundation for Research, Technology and Development is gratefully acknowledged. The work of J. Eilert, D. Wu and D. Liu was funded by EU FP7 MultiBase Project in partnership with Ericsson AB et al.

References

[1] Q. Wang, D. Wu, J. Eilert and D. Liu, "Cost Analysis of Channel Estimation in MIMO-OFDM for Software Defined Radio", in *Proc. IEEE Wireless Communications & Networking Conference*, April 2008.

[2] C. Mehlführer, S. Caban, M. Rupp, "An Accurate and Low Complex Channel Estimator for OFDM WiMAX", in *Proc. IEEE ISCCSP*, March 2008.

[3] C. Mehlführer, M. Wrulich, J. C. Ikuno, D. Bosanska and M. Rupp, "Simulating the Long Term Evolution Physical Layer," in *Proc. of the 17th European Signal Processing Conference (EUSIPCO 2009)*, Aug. 2009, Glasgow, Scotland

[4] 3GPP, "Technical Specification Group Radio Access Network; Evolved Universal Terrestrial Radio Access (E-UTRA); Physical Channels and Modulation (Tech. Spec. 36.211 V8.4.0)", Sept 2008

[5] IEEE, "IEEE Standard for Local and Metropolitan Area Networks Part 16: Air Interface for Fixed Broadband Wireless Access Systems", 2004

[6] J. Eilert, D. Wu, D. Liu, "Implementation of a Programmable Linear MMSE Detector for MIMO-OFDM", in *Proc. IEEE ICASSP*, 2008

[7] ITU, "Recommendation ITU-R M.1225: Guidelines for Evaluation of Radio Transmission Technologies for IMT- 2000 Systems", 1998

[8] M. Šimko, C. Mehlführer, M. Wrulich and M. Rupp, "Doubly Dispersive Channel Estimation with Scalable Complexity", in *Proc. IEEE WSA*, 2010

[9] S. Haene, D. Perels, A. Burg, "A Real-Time 4-Stream MIMO-OFDM Transceiver: System Design, FPGA Implementation, and Characterization," IEEE Journal on Selected Areas in Communications, vol.26, no.6, pp.877-889, August 2008

[10] J. Lofgren, S. Mehmood, N. Khan, B. Masood, M. Awan, I. Khan, N.A. Chisty, P. Nilsson, "Hardware implementation of an SVD based MIMO OFDM channel estimator", in *Proc. NORCHIP*, 2009

[11] J. Berkmann,C. Carbonelli, F.Dietrich, C.Drewes, W. Xu, "On 3G LTE Terminal Implementation - Standard, Algorithms, Complexities and Challenges", in *Proc. IWCMC*, 2008

[12] M. Šimko, C. Mehlführer, T. Zemen and M. Rupp, "Inter-Carrier Interference Estimation in MIMO OFDM Systems with Arbitrary Pilot Structure", in *Proc. IEEE VTC Spring*, 2011

[13] J. C. Ikuno, C. Mehlführer and M. Rupp, "A Novel LEP Model for OFDM Systems with HARQ", in *Proc. IEEE ICC*, 2011

[14] J. C. Ikuno, S. Schwarz and M. Šimko, "LTE Rate Matching Performance with Code Block Balancing", in *Proc. European Wireless Conference (EW)*, 2011

[15] Q. Wang, C. Mehlführer and M. Rupp, "Carrier Frequency Synchronization in the Downlink of 3GPP LTE", in *Proc. IEEE PIMRC*, 2010

[16] A. Nilsson, E. Tell, and D. Liu, "An 11 mm^2 70 mW Fully-Programmable Baseband Processor for Mobile WiMAX and DVB-T/H in 0.12μm CMOS", in *IEEE Journal of Solid-State Circuits*, vol. 44, no. 1, pp. 90-97, 2009