Ultra Low Latency Implementation of Robust Channel Estimation and Equalization for Industrial Wireless Communication Systems

Ludwig Karsthof, Mingjie Hao, Jochen Rust, Steffen Paul
Institute of Electrodynamics and Microelectronics (ITEM.me)
University of Bremen, Germany
{karsthof,hao,rust,steffen.paul} @item.uni-bremen.de

***Abstract*—In industrial environments, automation companies seek for wireless solutions for their real-time demanding production lines. Complicated wiring, feasibility problems in connecting moving parts and high cost limit the operation of wired communication systems like EtherCAT or POWERLINK. The specified *ultra reliable low latency communication* channel in 5G networks is planned to support robust wireless connection for guaranteed latency of 1 ms. But up to now, it is not yet clear, whether 5G will provide frequently used industry communication interfaces or not. Additionally, to the authors knowledge, there has no 5G transceiver been built up yet, which is proven to fulfill all industry specific demands. For this reason, other standalone solutions for this tasks are developed. To meet the industry requirements, all hardware must be designed for low latency and low hardware complexity at maximum robustness. This paper addresses channel estimation and equalization hardware architectures for use in such systems.**

***Index Terms*—Industrial wireless, channel estimation, low latency, hardware implementation**

I. Introduction and Related Work

Research and development of wireless communication systems for industrial environments is still ongoing, since they have a key role in Industry 4.0 [1]. For example, 5G standardization considers industrial radio channels with the employment of the *ultra reliable low latency communication* channel, called URLLC[2]. For multi-carrier systems, focus is on generalized frequency division multiplexing (GFDM [3]). However, at the moment orthogonal approaches take place in 5G development and standardization. For OFDM, the authors from [4] give a good overview of channel estimation techniques. Least squares (LS) and mean square error (MSE) approaches are presented in their paper. A robust soft-decision-aided channel estimator based on Pareto without a priori channel knowledge is presented in [5]. Most papers related to this topic focus on performance of the algorithm and neglect the feasibility and computational effort to implement the algorithms in hardware. Complex algorithms with high computational effort often run on high performance processors to fulfill the real-time requirement. Hence, in most commercial systems, performing algorithms are supported by hardware accelerators, or are completely implemented in hardware. Doing so saves energy, hardware complexity and cost. In our previous work [6], we proposed our hardware architecture for an industrial wireless solution. We implemented orthogonal frequency division multiplex multicarrier (OFDM) system with 256 subcarriers in 5.8 GHz range. For sufficient robustness, convolutional channel coding was applied with a high code rate of 0.5 and with QPSK modulation. The system fulfills the requirements of 1 ms and provided an SPI interface for sensor and actuator connection. Baseband signal processing can be accelerated even further if re-organized but not re-designed, as single processing blocks in RX path take 1015 clock cycles of the baseband signal processing clock (in the clients) to perform their operation including synchronization, CFO-estimation and compensation, FFT, channel estimation and compensation and channel decoding using Viterbi decoder. Hence, with the hardware operating at 100 MHz, latency of the single blocks is given by 10.15 µs. Latency that small can be achieved either by massive parallelization or by smart simplification of the used hardware blocks.

In this work, we propose an efficient, low complexity and low latency hardware architecture for industrial channel estimation and equalization. The design is built in hardware completely, integrated into our baseband signal processing chain and validated on FPGA-platform.

II. Channel Estimation Techniques

Commonly used pilot based channel estimation is implemented in this work. To keep the overhead as small as possible, equidistant subcarriers of the OFDM-symbol are used for channel estimation. To reduce hardware complexity, linear interpolation between pilot carriers is performed. Since one would need information from the next channel estimation to interpolate between coefficients in time, a delay of more than one OFDM-symbol would have to be accepted. Thus, for low-latency transmission of data, interpolation between channel coefficients in time is neglected. Moreover, in this paper it is assumed that the channel estimation is valid for a specific amount of time (coherence time 85 µs), after which the channel coefficients are estimated again. Adjacent carriers are considered to have similar coefficients, which defines the amount of pilots that are sent over time and frequency. In our system, every fourth subcarrier of every eighth sent OFDM-symbol contains pilots. This pilot design is determined by measurements of the channel and proven suitable in field tests. Because comb type arrangement of the pilots neglect the effect of frequency selective channels in industrial environments, we perform estimation on 56 subcarriers in total. Eight estimations are performed in downlink frame because of time variant industrial radio channels. When implementing a model for the channel estimation and equalization in Matlab, the easiest way of compensation is to divide by the channel coefficient, neglecting noise. In a sequentially processing hardware block, this cannot be implemented easily due to the absence of low-complexity dividers. Commonly used SRT-algorithms for division usually need several clock cycles for operation at higher complexities compared to the hardware that is needed for alternative algorithms. In this work, the commonly used IQ-based approach is implemented and compared to an improved version, which uses phase information only to reduce hardware complexity.

A. IQ-based calculation of channel coefficients

The most basic way of calculating channel coefficients is the direct way, processing complex data in cartesian form. With known pilot data, reciprocal channel coefficients (which are multiplied with the receive samples later) are defined:

$$\frac{1}{\underline{h}} = \frac{p_I y_I + p_Q y_Q + j \cdot \left(p_Q y_I - p_I y_Q\right)}{y_I^2 + y_Q^2} \tag{1}$$

with $p_{I,Q}$ the in-phase and quadrature component of known pilot symbols, $y_{I,Q}$ the received samples and 1/h the reciprocal channel coefficient. This operation takes 6 multipliers and 3 adders, and real part and imaginary part have to be divided by the divisor of equation (1). Interpolating between complex numbers usually must be performed in phase and amplitude (refer to Fig. 1). Because channel coefficients of direct adjacent subcarriers are commonly expected to be similar, linear interpolation of in-phase and quadrature component is performed as an approximation. For equidistant channels, linear interpolation between two estimated channels $\underline{h}_4$ and $\underline{h}_0$ is implemented as follows:

$$\underline{h}_2 \approx \frac{\underline{h}_4 + \underline{h}_0}{2}. \tag{2}$$

Interpolation between equidistant subcarriers is commonly performed separately in real part and imaginary part and simplifies the interpolation, so that

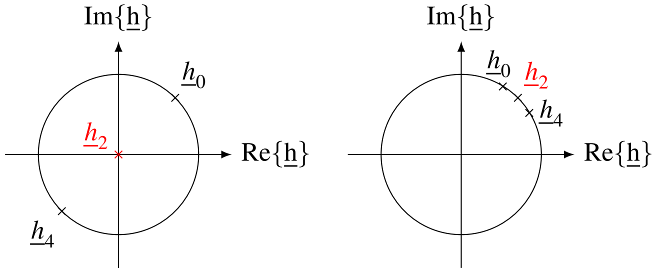
$$\underline{h}_2 \approx \frac{h_{4,RE} + h_{0,RE}}{2} + j \cdot \frac{h_{4,IM} + h_{0,IM}}{2} \tag{3}$$

$\underline{h}_1$ and $\underline{h}_3$ are determined similarly. For shorter critical path, both can be calculated without approximating $\underline{h}_2$ first:

$$\underline{h}_1 \approx \frac{\underline{h}_4 + \underline{h}_0}{4} + \frac{\underline{h}_0}{2} \tag{4}$$

$$\underline{h}_3 \approx \frac{\underline{h}_4}{2} + \frac{\underline{h}_4 + \underline{h}_0}{4} \tag{5}$$

The approximation is only applicable, if the channel coefficients of nearby subcarriers are similar, as illustrated in Fig. 1. Since receive samples are multiplied with the reciprocal channel coefficients, it is necessary that their amplitude value and phase are correct. Analyzing a sample OFDM-symbol from real data transmission experiments in an industrial environment, our channel coefficients prove to be similar to their neigbours, as illustrated in Fig. 2. The more different $\underline{h}_0$ and $\underline{h}_4$ are, the more $\underline{h}_2$ differs from the actual channel coefficient in terms of absolute value. Equalizing samples with a too small reciprocal channel coefficient will lead to equalized receive samples with a too large distance to the actual send value in terms of amplitude. When working with the amplitude information, e.g. when using CDMA-despreading or soft-bit decoding with LLRs (log-likelihood ratio), this problem has to be taken care of. In our design, every fourth subcarrier has a pilot symbol, so linear interpolation takes 3 shift registers and 3 adders. Reciproke channel coefficients are then stored in a register array and updated, as soon as an OFDM-symbol contains new pilots (in our case every 8th symbol). The main advantage of this design is the simplicity of the algorithm due to some major approximations (refer to Sec. III-A). Equalization takes another 3 multipliers for complex multiplication of the receive samples with the complex channel coefficient.



(a) Approximation fails (b) Approximation applicable

Fig. 1: Approximation of linear interpolation between channel coefficients in cartesian coordinates can be done if the neighboured samples are similar (b).

B. The role of amplitude in QPSK-systems

Since QPSK demodulation is done by determining the quadrant in which a sample is located, only phase information is needed for hard decision of the QPSK-demapper. A simplification can be introduced: neglecting the amplitude of the signal (IQ-samples) yields lower hardware complexity. In Fig. 4, a block diagram of the proposed hardware is shown. The use of amplitude information for QPSK modulated signals is given, e.g. when soft bit decoding (like Viterbi) or despreading code divided payload (code division multiple access, CDMA).

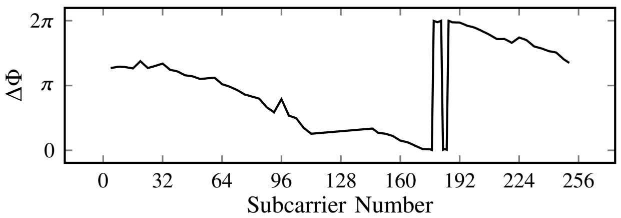


Fig. 2: Real industrial environment channel coefficients measured by our channel estimation hardware. ΔΦ describes the needed correction angle for the corresponding subcarrier's payload bits.

C. Phase- and amplitude based calculation of the channel coefficients

In Sec. II-B, estimation of the sample phase is treated. To estimate the amplitude information as well, additional logic is needed, which is shown in figure 5. For CDMA or appropriate LLR-input for soft bit decoding, the amplitude of the signal cannot be neglected and is equalized as

$$|\tilde{x}| = \frac{\sqrt{y_I^2 + y_Q^2}}{|h|} \tag{6}$$

with $\tilde{x}$ the equalized sent symbol, y the receive sample and h the absolute of the channel coefficient (its amplitude). The inverse channel coefficient is calculated according to

$$h^{-1} = h_r = h_{r,approx} + \frac{h_{r,approx} - (y_I^2 + y_Q^2)h_{r,approx}^3}{2} \tag{7}$$

in e.g. 3 iterations. $h_{r,approx}$ is approximated in a LUT in the first iteration. Calculating angle and amplitude of the IQ data can be advantageous due to the exact calculating of linear interpolation.

III. Architecture

Two different approaches are implemented here and compared to each other in terms of hardware complexity and latency. For all channel estimation techniques, the same frame design is used, where every fourth subcarrier is used for pilot symbols.

A. Achitecture of the IQ-based solution

When implementing the IQ-based channel estimation and equalization hardware, we designed sub-blocks as shown in Fig. 3. The data input in our channel estimation block are still bit-inversed from the FFT-operation. After estimating their reciprocal channel coefficients, $\frac{1}{h}$-values are stored in a memory for later equalization of carriers by multiplication. Since payload bits are located between the pilot carriers (in our case 3 of them), every OFDM-symbol is stored for later equalization. When equalizing an OFDM-symbol, channel estimation of the previous symbol must be completed. For this purpose, a second memory block has been integrated. The multiplication block shown in Fig. 3 is implemented as complex multiplication, so that it actually contains 4 multiplications, according to Eq. (8).

$$(I_0 + jQ_0) \cdot (I_1 + jQ_1) = I_0 \cdot Q_0 - I_1 \cdot Q_1 + j(I_0 \cdot Q_1 + I_1 \cdot Q_0) \tag{8}$$

In the $\frac{1}{h}$-block, operation is performed as defined in Eq. (1). For this, a division is needed which was implemented as SRT-algorithm which takes 12 clock cycles to perform division. The division is pipelined, so that we don't have to wait for the hardware to finish one coefficient estimation before starting calculation of the next one.

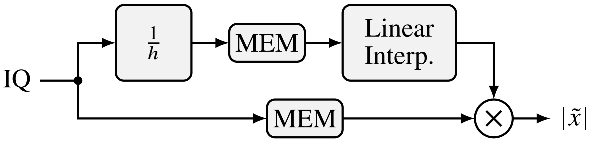


Fig. 3: Block diagram for IQ-based solution of channel estimation and equalization

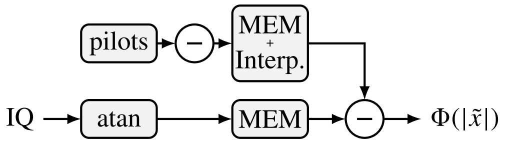


Fig. 4: Block diagram for angle based solution, estimating and equalizing the angle of sent symbols.

B. Architecture of the angle-based solution

In the angle based-solution, amplitude and phase of the IQ-data must be determined separately directly from the beginning of the algorithm. The FFT-output drives the input of the block diagrams in Fig. 4 and Fig. 5. In Fig. 4, the first stage is calculating the phase information of the received samples. For low complexity implementation of the arctangent function, a piecewise linear approximation function was used. The algorithm outputs results for our trigonometric function in one clock cycle. The arctangent function is calculated with all input samples and the result is forwarded to the channel estimation unit. Once pilots have been input, a correction angle is calculated for estimating the channel. Preamble data, DC carrier and guard band are ignored by the channel estimation FSM (finite state machine). While calculating correction angles (representing channel coefficients) and saving them in a RAM, all payload data is stored in a separate memory for later equalization. Linear interpolation between the correction angles is performed in parallel to receiving the OFDM-symbol that contains the pilot data. When receiving the second OFDM-symbol, channel estimation is complete and equalization of the first received OFDM-symbol's payload is done. Calculating the amplitude is described in Sec. II-C. In hardware it is implemented as follows: In the first clock cycle,

$$|h|^2 = Re\{y\}^2 + Im\{y\}^2 \tag{9}$$

is calculated. The value is buffered in a register, and send to the next block, which performs a coarse estimation of the reciprocal absolute value using a lookup-table of the size 256. All algorithms in the square root estimation provide 8 Bit wide datapaths. Further iterations of approximating the square root of $\frac{1}{h}$ are performed as defined in Eq.(7).

IV. Evaluation

A. Measurements

After equalizing the input samples with the estimated channel coefficients in phase, the equalized samples look similar to the ones in Fig. 6. Most of the equalized samples are located on or near the expected QPSK-symbols in constellation diagram now. Since in our design, amplitude information is neglected, and therefore not calculated, a histogram for

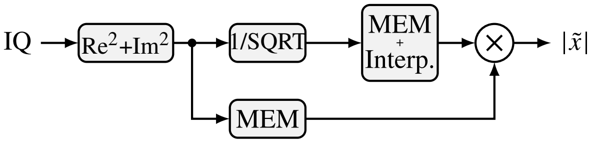


Fig. 5: Block diagram for amplitude extension of the channel estimation/equalization hardware

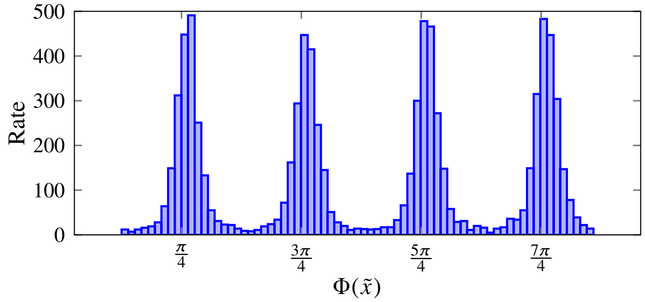


Fig. 6: Histogram for the equalized sample angles from one example frame. The data was recorded in an industrial production hall.

the received symbols is rather suitable than a constellation diagram. Based on the information on the data from Fig. 6, decoding is performed. Decoding data from the example frame was successful, even though some sample angles were located on or near the axes. For this, every communication system includes a channel coding technique, in our case, a convolutional code was used (with Viterbi decoding in the RX path). To gain the data, we transmitted packages in the industry hall, received them and transmitted the data for the histogram to our host-PC for analysis.

B. RTL-synthesis and FPGA-platform

Synthesis results for the RTL-code can be seen in table I. Due to the example systems's OFDM-symbol size, the frame size for our design was set to 256. Since the number of channel coefficients define the memory size needed, this must be taken into account when evaluating the total numbers of slices listed in table I. The number of channel coefficients, which are stored and interpolated, are equal in both cases. It is easy to see, that (even if the amplitude information is needed) calculating the channel coefficients in angle and amplitude results in lower hardware demand compared to calculating it the usual way in cartesian coordinates. For LUT and register slices, this is due to more complexity in interpolating 2x16 bit wide coefficients compared to interpolating on 10 bit wide phase information. Since estimation and equalization using phase information only need addition/subtraction with correction angles and pilot phase information, it is easy to see, that DSP resources are not needed in this case. Like table II shows, latency of both algorithms is low.

Algorithm	Module	LUTs	Regs	BRAM	DSPs
IQ	Estimation	4299	5435	0	6
IQ	EQ	84	303	0.5	4
IQ	**Total**	**4384**	**5738**	**0.5**	**10**
Φ	Estimation	1086	1288	0	0
Φ	EQ	43	290	0.5	0
Φ	atan	247	10	0	0
Φ	**Total**	**1376**	**1588**	**0.5**	**0**
$\|\|IQ\|\|$	**Total**	**878**	**872**	**0**	**10**

TABLE I: FPGA resource utilization

Algorithm	Module	Latency (Cycles)
IQ	Estimation	12
IQ	EQ	1
IQ	**Total**	**257***
Φ	Estimation	1
Φ	EQ	1
Φ	atan	1
Φ	**Total**	**257***
$\|\|IQ\|\|$	**Total**	**257***

TABLE II: Latency of the hardware implementation. Channel estimation of our IQ-Design is pipelined, the SRT-divider uses 12 clock cycles. *Note that channel coefficients are used one OFDM-symbol later (256 clock cycles).

V. Conclusion

In this paper, a fast channel estimation and equalization hardware was presented, which meets industry requirements regarding latency with robust QPSK modulation and linear channel coefficient interpolation. Real receive samples from industry environments are used for validation and prove the design an effective, low complexity and low latency implementation for wireless receiver channel estimation and equalization. For QPSK modulation, the design could be simplified neglecting the amplitude information. This yields a lower hardware utilization of the FPGA's resources.

VI. Acknowledgements

This work has been partially funded by the Federal Ministry of Education and Research, Germany in the project “HiFlecs” (Ref. No.: 16KIS0271) and “Estimate” (Ref. No. 16KIS0637).

References

[1] Varghese, A. and Tandur, D.: Wireless requirements and challenges in industry 4.0, International Conference on in Contemporary Computing and Informatics (IC3I) 2014.

[2] 3GPP consortium,Technical Report on: “Study on enhancement of Ultra-Reliable Low-Latency Communication (URLLC) support in the 5G Core network”, Release 16, Reference 23725.

[3] N. Michailow et al., “Generalized Frequency Division Multiplexing for 5th Generation Cellular Networks,” in IEEE Transactions on Communications, vol. 62, no. 9, pp. 3045-3061, Sept. 2014.

[4] Pallaviram Sure, Chandra Mohan Bhuma: “A survey on OFDM channel estimation techniques based on denoising strategies”, International Journal Engineering Science and Technology, Vol 20, Issue 2, 2017, ISSN 2214-0986

[5] H. Yuan and P. Kam, “Soft-Decision-Aided, Smoothness-Constrained Channel Estimation over Time-Varying Fading Channels With No Channel Model Information,” in IEEE Transactions on Wireless Communications, vol. 16, no. 1, pp. 73-86, Jan. 2017

[6] Hao, M., Karsthof, L., Rust, J., Demel, J., Bockelmann, C., Dekorsy, Al Houry, A, Mackenthun, F. und Paul, St.: “FPGA-based Baseband Solution for High Performance Industrial Wireless Communication.” IEEE International Conference on Digital Signal Processing (DSP). 2018

[7] Karsthof, L., Hao, M., Rust, J., Block, D., Meier, U. and Paul, St., “Dynamically reconfigurable real-time hardware architecture for channel utilisation analysis in industrial wireless communication,” 2016 IEEE Nordic Circuits and Systems Conference (NORCAS), Copenhagen, 2016, pp. 1-6.