Field Trial Evaluation of Compression Algorithms for Distributed Antenna Systems

Michael Grieger, Peter Helbing, Gerhard Fettweis
Technische Universität Dresden,
Vodafone Chair Mobile Communications Systems,
Email: {michael.grieger, peter.helbing, fettweis}@ifn.et.tu-dresden.de

Patrick Marsch
Nokia Siemens Networks - Research
Wroclaw, Poland
patrick.marsch@nsn.com

Abstract—Coordinated multi-point (CoMP) in the cellular uplink, offering large improvements in spectral efficiency and fairness, appears to be an effective option to combat inter-cell interference. Current approaches range from coordinated scheduling to coherent joint detection. A major drawback of coherent joint detection is the large extent of additional backhaul infrastructure required for the exchange of received signals among base stations. Theoretical research on this topic emphasizes the benefits of source coding as a method to reduce the backhaul that is required. This paper complements previous publications through field trial results obtained in a representative urban setup. Different scalar and vector compression algorithms are compared in terms of their complexity as well as the average distortion of the compressed signal. System performance, evaluated in terms of the SINR of the equalized transmit signals, was determined for measurement data to investigate performance of compression algorithms under real-world conditions.

I. INTRODUCTION

The progress in capability and availability of cellular mobile communication systems has led to a steadily increasing offer and demand for services based on these technologies, subsequently requiring higher network capacities. As available spectral resources are scarce, research focus lies on increasing spectral efficiency. For today’s cellular networks, the main spectral efficiency limiting factor is inter-cell interference. In conventional systems, frequency resources are distributed among neighboring cells to limit this interference. As a result, every cell utilizes only a fraction of the bandwidth available to the system. Future systems are likely to use some form of CoMP, where interference is either avoided by coordinated scheduling or signals received across cell borders are jointly exploited by coherent joint detection. As shown in [1], [2] the latter results in increased spectral efficiency due to the removal of limitations on resource reuse between cells.

To enable this cooperation, a means of information exchange is required. In practice, the necessary information would be forwarded via a backhaul channel between base stations (BSs), resulting in additional traffic which must be supported by the infrastructure of the network operator. For this reason, backhaul-efficient cooperation is an important goal for putting CoMP into practice.

Various schemes for coordinated detection have been proposed, such as distributed decoding in [3] and distributed compression in [4]. In the scope of this paper, a group of cooperating BSs will be conceptualized as a distributed antenna system (DAS). Uplink signals sent by mobile terminals (MTs) are received at several BSs. Compressed representations of the receive signals are forwarded to a single decoding BS, where they are exploited as additional information available for joint detection. For this scheme, the achievable uplink sum rates of multiple MTs, which are significant even for strongly limited backhaul rates, has been presented in [4].

The problem of minimizing signal distortion introduced by quantization, with optimization techniques depending on the statistics of the original signal, has been researched for several decades. Several of these techniques are presented in Section II. The subject of Section III is the efficient implementation of the quantization process with regards to computational complexity and the representation of the input signal.

The influence of quantization on joint decoding performance was assessed in field trials. Section IV details the field trial setup and Section V explains the signal processing architecture. Quantization can be included at various processing stages of an OFDM system. In particular, we will compare the quantization in time and frequency domain.

Finally, practicality of the schemes implemented was assessed. This evaluation is presented in Section VI, where the computational complexity of the algorithms was considered, along with comparative results obtained by testing the extended signal processing chain with various quantization schemes.

II. CODEBOOK GENERATION

While quantization can be used for compliance with rate limitations of a channel, the trade-off is an introduction of a distortion which limits the throughput of the entire system. A quantization function maps input values from a partitioning of the input range to a set of output or reproduction values (or codebook). To minimize distortion, this mapping is optimized according to the probability distribution of the input signal which changes constantly due to time varying channels. There are two possible ways for this adaptation: Either statistical properties of the signal are estimated and used in appropriate algorithms or the quantizer is found by using training.

The key difference in quantization algorithms is whether the input signal is interpreted as a set of real-valued samples (or scalars) or as a set of points in a multi-dimensional vector
space. These interpretations lead to the concepts of scalar quantization (SQ) and vector quantization (VQ) respectively. Furthermore, the generation of optimal quantizers depends on whether a continuous distribution with variable parameters is assumed (e.g., the Gaussian distribution), or whether an estimate of the underlying distribution is generated from a sampled test sequence. An overview of all quantizers that are considered in this work is given in Figure 1.

Fig. 1: Overview of implemented quantization variants

A. Scalar Quantization

For scalar quantization, the signal is interpreted as a sequence of real-numbered samples which are realizations of a random variable. Due to the linear ordering of real numbers when compared with vector quantization, both scalar quantizer generation and the actual quantization are much less demanding in terms of computational complexity and show better scaling behavior with increasing rates. Correlation between samples can be exploited using transform coding (Section III-C), without losing the complexity advantage.

An algorithm for optimal quantization was first documented by Lloyd in 1957 and published in [5]. Since a similar algorithm was independently described by Max [6], it is also known as the Lloyd-Max algorithm. The global convergence to minimal average MSE distortion of the scalar Lloyd algorithm has been formally proven in [7] for any smooth density function.

B. Vector Quantization

The concepts for quantization of scalars established in the previous section can be generalized for vectors, in this context any ordered set of real-valued components. The advantage of vector quantization lies in the availability of multiple dimensions and thus additional structural degrees of freedom for the definition of partition cells which allows for a larger flexibility in quantizer design, referred to as shaping gain.

The working principle of the Lloyd algorithm is preserved for multidimensional distributions (generalized Lloyd algorithm (GLA)). However, for reasons of computational simplicity, we limit ourselves to quantizers generated from sampled test sequences. This approach leads to two significant computational simplifications:

1) The algorithm does not require a geometrical representation of the partition cells. The assignment of an test vector to a partition cell reduces to finding the minimum-distortion reproduction value.

2) The computation of the centroids reduces to a sequence of vector additions and a scalar multiplication instead of multidimensional integration.

The convergence behavior of the GLA on a sufficiently large training sequence has been shown in [8].

III. QUANTIZER IMPLEMENTATION

After the generation of optimal quantizers for known or estimated probability distributions has been discussed, efficient implementation of the quantization process for a given codebook is the focus of this section. In the quantization process, input symbols are mapped onto reproduction values. In the following, we start out addressing the full search quantizer. We then present an algorithm which scales linearly with the binary resolution of the quantizer. The computational complexity of all algorithms considered is summarized in Table I.

A. Full Search Quantizer

The general problem of optimal quantization is finding the minimum distortion reproduction value (or vector) from the codebook, e.g., for the scalar case:

\[
Q(x) = \hat{x} = \min_{\hat{x} \in C} d(x, \hat{x}).
\]

A brute force solution to this problem is to calculate the distortion between the input value \(x\) and each reproduction value \(\hat{x}_i\) and then determine the smallest distortion measure. This may be called a “full search” quantizer. For scalar quantizers, only a comparison (CMP) between the input value \(x\) and a threshold value \(\hat{x}_i \in C\) is required to add an additional bit of quantizer accuracy.

A major disadvantage of the regular “full search” algorithm for vector quantizer generation arises from the fact that there is no sequential ordering of partition cells. Thus, the distortion of the input vector for each codebook vector must be computed. Thus, the distortion metric calculation for all \(2^r\) reproduction vectors is needed. Then, the minimum distortion value has to be found by an additional \(2^r\) comparisons. This exponential scaling applies not only to the actual quantization process itself, but to quantizer generation, where the main complexity lies in the assignment of \(M\) training vectors to partition cells. When the required multiplications (MULs) are considered, the cost for a Lloyd iteration of a \(k\)-dimensional FSFQ with the training sequence length \(M\) and rate \(r = kr_q\), \(k = 2^j, j = 1, 2, \ldots\) is:

\[
\text{MUL}_{\text{all}}(k, r, M) = M2^{kr_q+2},
\]

where \(r_q\) is the quantization rate per real dimension.

![Table I: Required operations for quantization of one input symbol with 4 real components](image)
B. Tree-Structured Quantizer

A major complexity reduction is achieved by the tree search algorithm which was first introduced in [9]. The key idea is to recursively split the training set as well as the reproduction vectors to generate a sequence of codebooks each with resolution \( r = 1 \), consisting of two codewords which are “test vectors” to form a decision at each tree node during quantization, by finding the one which produces minimal distortion. Compared to the full search algorithm, two significant advantages emerge for the tree search quantizers:

1) The complexity scales linearly with the quantization resolution \( r \) unlike the exponential scale for the full search algorithm. In the scalar case, the tree search algorithm does not even require multiplication.

2) Variable quantization rate: as the most significant bit is determined first and subsequent bits are concatenated after further decisions, a lower quantization rate can easily be achieved by an incomplete passage through the tree without changing the underlying codebook.

For scalar quantizers, the computational complexity for tree-structured vector quantizers (TSVQ) scales linearly with the rate \( r \) and the number of vector components, requiring \( k^2 \) multiplications and \( k \) additions/subtractions (ADD).

As is the case for the actual quantization, the complexity for codebook generation using the Lloyd algorithm scales linearly with \( r_q \) for TSVQ. The total of \( 2^r - 1 \) tree nodes exist, for each of which a codebook with \( r = 1 \) must be generated:

\[
\text{MUL}_{\text{TSVQ}} = 2^{kr_q - 1} \cdot \text{MUL}_{\text{LI}}(k, 1, M^*),
\]

where \( M^* \) is the average number of elements in the training sequence of a single quantizer. At each level of the tree, every training vector is assigned to a unique node (corresponding to a single quantizer), thus

\[
M^* = \frac{Mr_qk}{2kr_q - 1}.
\]

Thus the number of multiplications per Lloyd iteration for a TSVQ is

\[
\text{MUL}_{\text{TSVQ}} = M2^{k+2}r_qk.
\]

C. Transform Coding

Another approach is transform coding, where correlation between symbols is exploited for a reduction in distortion by an orthogonal transformation that is fully reversible at the decoding BS. The optimality of the Karhunen-Loeve (KL) transform which eliminates correlation between vector components for Gaussian sources has been shown in [10]. It has been experimentally observed that the elimination of this correlation leads to more efficient scalar quantization. In a mobile environment the transformation applied has to be adapted according to the current channel state. Note that transform coding can also be applied for VQ. In this case it will not achieve a better performance, but will allow using a standard codebook without correlation and , thus, may reduce complexity.

IV. Field Trial Setup

The performance of the proposed algorithms for backhaul signal quantization was evaluated by processing measurement data collected by LTE-Advanced equipment at TU Dresden. Radio link signals accumulated in field trials were digitized and recorded, while signal processing (including synchronization, channel estimation, and joint equalization) was done offline. As shown in Figure 2, the measurement setup consists of two BSs deployed on the rooftop of a building in Dresden about 150 m apart. The BSs are synchronized through GPS fed reference normals and connected via a microwave link. Each BS is equipped with a cross-polarized directional antenna, hence with \( N_{bs} = 2 \) individual antenna elements per BS. The UEs - employing one omnidirectional antenna each - transmit OFDM symbols using 4QAM or 16QAM. Both UEs are scheduled to transmit on the same resource in time and frequency. Various transmission parameters are listed in Table II.

V. Signal Processing Architecture

In the next few subsections, we will briefly explain the general signal processing steps performed in our architecture. For a more thorough discussion, we refer the reader to [11].

Neglecting residual synchronization errors and assuming the channel has a coherence bandwidth significantly larger than the sub-carrier spacing of \( \Delta F = 15 \text{kHz} \), the transmission of each symbol on a single sub-carrier of the OFDM system in frequency domain can be stated as

\[
y_1 = h_{1,1}x_1 + h_{1,2}x_2 + n_1,
\]

\[
y_2 = h_{2,1}x_1 + h_{2,2}x_2 + n_2,
\]

where \( y_m \in \mathbb{C}^{[N_{bs} \times 1]} \) denotes a vector of signals received by \( N_{bs} \) antennas of BS \( m \), \( h_{m,n} \in \mathbb{C}^{[N_{bs} \times 1]} \) denotes the channel matrix from UE \( n \) to BS \( m \), \( x_n \in \mathbb{C} \) is a symbol transmitted by UE \( n \), and \( n_m \in \mathbb{C}^{[N_{bs} \times 1]} \) is additive, uncorrelated noise.
Frame and symbol synchronization exploit the autocorrelation properties of the time domain signals. The cyclic prefix of each OFDM symbol is removed and a fast Fourier transform (FFT) recovers the frequency domain symbols where each subcarrier is associated with an FFT point.

Channel estimation evaluates the pilot symbols transmitted per TTI at the standard positions (4th and 11th) defined by LTE Release 8. An orthogonal code design for the pilot pilots. In a cellular setup, we also loose the ability to form different cooperation clusters on different sub-carriers and the required backhaul bandwidth does not scale with the number of sub-carriers used, which would be desired in low loaded cells. For both $N_{bs} = 2$ local antennas, each received sample is initially digitized with a resolution of $r_q = 12$ bit per real dimension. Without any further compression, the number of bits required for the transmission of one TTI (1 ms) is

$$N_b^{TTI} = N_{bs} \times 2 \times \frac{f_s}{1000} \times r_q = 737280 \text{ bits}. \quad (8)$$

### B. Quantization in Frequency Domain

By performing synchronization and FFT at the remote base station before quantization, there is an additional increase in flexibility because symbols on different subcarriers can be exchanged to different BSs in a cellular network. Furthermore, the rate requirements are significantly lowered when compared to the quantization of time domain samples, because of the exchange via the backhaul is limited to data and pilot symbols.

Due to the assignment of frequency domain symbols to individual sub-carriers, quantizer generation and quantization can be performed separately for groups of adjacent subcarriers. Under the assumption of similar channel realizations for adjacent subcarriers, the distribution of the unequalized data symbols would also be tighter for the whole range of subcarriers, leading to a possible reduction in average quantization distortion. As in section V-A, channel estimation is performed on the quantized data after backhaul transmission.

### C. Equalization

The MMSE filter for joint detection is

$$G_{\text{biased}} = \hat{H}^H \left( \hat{H} \hat{H}^H + \Phi \right)^{-1} , \quad (9)$$

where $\hat{H} = \begin{bmatrix} \hat{h}_{1,1} & \hat{h}_{1,2} \\ \hat{h}_{2,1} & \hat{h}_{2,2} \end{bmatrix}$ and $\Phi = \begin{bmatrix} \sigma_q^2 I & 0 \\ 0 & \sigma_q^2 I \end{bmatrix}$.

Note that the MMSE filter has to be computed for each symbol due to frequency and time variance of the channel. In order to compare the performance of the quantizers investigated, we use the SINR after equalization which is calculated according to a standard error vector magnitude approach. Thus, we compute the root-mean-square of the difference between the

---

**TABLE II: Transmission parameters**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Carrier frequency</td>
<td>2.53 GHz</td>
</tr>
<tr>
<td>System bandwidth</td>
<td>10 MHz</td>
</tr>
<tr>
<td>Sample rate $f_s$</td>
<td>15.36 MHz</td>
</tr>
<tr>
<td>Resource blocks (PRBs)</td>
<td>$N_{PRB} = 10$</td>
</tr>
<tr>
<td>No. of sub-carriers per PRB</td>
<td>$N_{SC} = 12$</td>
</tr>
<tr>
<td>Transmit Power</td>
<td>$-5$ dBm</td>
</tr>
<tr>
<td>AD-converter resolution</td>
<td>$r_q = 12$ bit per real dim.</td>
</tr>
</tbody>
</table>

---

Fig. 3: Comparison of quantization algorithms in time domain, $M = 10000$ (1 TTI)
measured symbols and ideal symbols, which is accurate even in the low SINR regime because the transmitted symbols are known under field trial conditions.

VI. FIELD TRIAL RESULTS

In this section, individual algorithms are assessed in terms of computational complexity requirements and performance.

A. Quantization in Time Domain

Figure 3 depicts performance results of scalar quantization on the received time-domain signals. For each real component, an optimized codebook of rate \( r_q = 3 \ldots 8 \) bit per real dim. was used. The Lloyd-generated quantizer based on a training sequence (train) and the one based on a parameterized Gaussian distribution (gauss) achieve comparable SINRs for all rates observed. This shows that the Gaussian codebook is close to the best codebook achievable by the optimality criteria, and thereby suggests that the approximation of Gaussian receive signals is valid. The linear quantizer clearly performs worse than the other two; for low quantization rates, the gain achieved by joint equalization barely surpasses the SINR for non-cooperative decoding. For higher quantization rates, any quantizer shows convergence towards the upper bound (exchange of receive signals without compression loss), although the steepness of convergence for the optimized quantizers is much greater: For a rate of \( r_q = 6 \) bit per real dim. (being half the initial rate of \( r_{\text{init}} = 12 \) bit per real dim.), there is a negligible degradation when compared to the optimum.

The inherent gain postulated for vector quantization in section VI-B can be observed in figure 3b: wherein an additional gain of 0.2-0.3 dB in post-detection SINR is observable for all vector quantizers.

While the SINR values measured for all quantizer variants are very close, further differentiation is possible: higher-dimensional quantizers result in higher SINRs, and the codebooks of the FSVQs (fsvq,k) are slightly better as there is no subdivision of the training sequence for less significant bits as with TSVQ (tsvq,k), where \( k \) is the number of dimensions. Another notable fact is that there is no additional observable gain by transform coding (kltrans) in comparison with scalar quantization without further decorrelation (train).

B. Quantization in Frequency Domain

For scalar quantizers in frequency domain (see figure 4a), only variants with optimized codebooks were considered. The Gaussian assumption does not hold for the frequency domain signal as each symbol is a superposition of \( K = 2 \) QAM symbols and noise, a small loss in SINR (\( \approx 0.1 - 0.2 \) dB) is observed at lower quantization rates (\( r_q \leq 4 \) bit per real dim.) for quantization in frequency domain. Nevertheless, for increasing \( r_q \), the same convergent behavior occurs. For this representation, channel estimation is performed on quantized data as well.

As observed for scalar quantization in the time domain, the improved codebook realization by Lloyd optimization, while not reflected in the average distortion, leads to further SINR gain for lower quantization rates when compared to the assumption of a Gaussian distribution.

The inherent structural advantages of vector quantization result in an additional SINR gain of 0.2-0.4 dB compared to the scalar algorithms, depending on \( r_q \) (see figure 4b). For fsvq4 and tsvq4, there is a clearly visible fall-off for the highest investigated rate \( r_q = 3 \) bit per real dim.. At this rate, a quantizer codebook size of \( N = 2^{4 r_q} = 2^{12} = 4096 \) vectors is required for quantizers with dimension \( k = 4 \), resulting in a training ratio \( M/N \) only slightly larger than 1.

C. Length of Training Sequence

For quantizers based on a training sequence, the average distortion depends on the training length \( M \), or more specifically, the training ration \( M/N \). This is especially relevant for the quantization of frequency-domain symbols, as less sample data is available per dump file when using this representation. Figure 5a shows test results for varying lengths of \( M \) using the kltrans and fsvq4 quantizers after FFT, each with a fixed rate of \( r_q = 3 \) bit per real dim.. Both schemes profit from an increased length of the training sequence. Note that there is little degradation in average distortion and SINR.

Fig. 4: Post-detection SINR of quantization algorithms in frequency domain, \( M = 10000 \)
for both algorithms between $M = 2000$ real symbols and $M = 10000$ real symbols. This suggests that even for low training sequence lengths, a good performance trade-off is attainable, which also has the positive side-effect of lower computational requirements.

D. Quantization of Channel Coefficients

When quantization is performed in the frequency domain, pilot and data symbols can be quantized with different accuracy. An improved accuracy of channel equalization by using channel coefficients estimated from uncompressed symbols is visible for low rates ($r_q = 2$ bit per real dim.). There is a gain of about 0.4 dB for all quantizer variants compared with the SINR values shown in Figure 5b. For higher $r_q$, the channel estimation based on quantized symbols also increases in accuracy, so this gain is steadily reduced.

VII. Conclusions

In this paper, several quantization methods have been investigated and compared using field trials for distributed antenna systems. Several practical schemes can be recommended based on the measurement data evaluated, when quantizing OFDM symbols:

- for low processing capabilities, use scalar quantizers with Gaussian codebooks which are scaled based on periodical estimations of mean and variance of the input signal.
- for higher capabilities, use two-dimensional vector quantizers (per antenna) with a tree-structured codebook which is periodically updated based on training data.
- a reasonable compromise between detection performance gain and computational complexity would be the use of scalar quantizers based on periodically updated training data.

ACKNOWLEDGEMENT

The research leading to these results has received funding from the European Commission’s seventh framework programme FP7-ICT-2009 under grant agreement no 247223 also referred to as ARTIST4G. Further, this work would not have been possible without the support from Ainoa Navarro Caldevilla, Sven-Einar Breuer, Vincent Kotzsch, Fabian Diehm, André Zoch, Zhijun Rong, and Eckhard Ohlmer.

REFERENCES