Maximum mutual information vector quantization of log-likelihood ratios for memory efficient HARQ implementations

Danieli, Matteo; Forchhammer, Søren; Andersen, Jakob Dahl; Christensen, Lars P.B.; Skovgaard Christensen, Søren

Published in:
Data Compression Conference (DCC), 2010

Link to article, DOI:
10.1109/DCC.2010.98

Publication date:
2010

Document Version
Publisher's PDF, also known as Version of record

Maximum Mutual Information Vector Quantization of Log-Likelihood Ratios for Memory Efficient HARQ Implementations

Matteo Danieli, Søren Forchhammer, Jakob Dahl Andersen,
{matd, sofo, jdan}@fotonik.dtu.dk, *Technical University of Denmark*
Lars P.B. Christensen, Søren Skovgaard Christensen
{lars.christensen, soren.sk.christensen}@nokia.com, *Nokia Denmark*

Abstract

Modern mobile telecommunication systems, such as 3GPP LTE, make use of Hybrid Automatic Repeat reQuest (HARQ) for efficient and reliable communication between base stations and mobile terminals. To this purpose, marginal posterior probabilities of the received bits are stored in the form of log-likelihood ratios (LLR) in order to combine information sent across different transmissions due to requests. To mitigate the effects of ever-increasing data rates that call for larger HARQ memory, vector quantization (VQ) is investigated as a technique for temporary compression of LLRs on the terminal. A capacity analysis leads to using maximum mutual information (MMI) as optimality criterion and in turn Kullback-Leibler (KL) divergence as distortion measure. Simulations run based on an LTE-like system have proven that VQ can be implemented in a computationally simple way at low rates of 2-3 bits per LLR value without compromising the system throughput.

1 Introduction

In modern mobile telecommunication standards such as HSDPA, HSUPA and their evolution LTE [1], Hybrid Automatic Repeat reQuest (HARQ) is used in order to guarantee transmission reliability and increase the channel throughput [2]. HARQ is a powerful combination of forward error correction, error detection and if necessary packet retransmission for high channel throughput when feedback is available. The main idea that underlies HARQ is that even if a message is not decoded correctly upon reception, valuable information can be stored temporarily and combined to additional information to correctly decode it. In order to combine the information about the same message sent across different transmissions, the information available at the receiver must be stored on the HARQ memory. Data compression can help reduce the amount of memory needed on the receiving circuit for HARQ. To maximize the performance of the error correcting code (ECC), posterior probabilities of the bits that were transmitted need to be stored while waiting for the next retransmission, usually in the form of *log-likelihood ratios* (LLRs). Due to the ever increasing data rates in telecommunication networks and the need for supporting multiple HARQ processes in parallel, LLRs storage is bound to require an increasing amount of memory. This is what makes data compression desirable in this context.

In this paper we focus on defining a lossy compression technique for LLRs. This technique should optimize the compression rate under the constraint that the loss introduced is minimal in terms of reduction of system throughput in comparison to the case where no or

1068-0314/10 $26.00 © 2010 IEEE
DOI 10.1109/DCC.2010.98

little lossy compression is performed. Vector quantization (VQ) was selected for compression of LLRs for multi-bit modulation schemes. Choosing maximum mutual information (MMI) between transmitted bits and quantizer output (which corresponds to maximizing the channel capacity) as optimality measure leads to using Kullback-Leibler (KL) divergence as distortion measure for both minimum distance encoding and VQ design in conjunction with a Lloyd clustering algorithm. The use of KL and other kind of Bregman divergences as a measure of the distortion incurred when performing quantization of probabilities distributions is rather old. For instance, it has been used to design optimal quantizers for input variables to a binary decision system [3]. It has also already been used together with Lloyd clustering for the design of optimal VQs for pattern classification applications such as speech recognition [4]. To the knowledge of the authors, though, no previous attempts have been made to examine the efficacy of these techniques when turbo decoding is performed based on the quantized LLRs. The same MMI optimality criterion was chosen in [5], but the analysis was limited to scalar quantization (SQ) in case of additive white Gaussian noise (AWGN) channel. We consider VQ for the more realistic model of a Rayleigh-fading channel, which provides further motivation to the use of VQ due to the dependency it induces between LLRs of bits that are demapped from the same received symbol.

In Section 2 we describe the general architecture of a HARQ scheme, we provide a definition of LLRs, we mention how they are normally computed and we finally elaborate the problem statement. In Section 3 we explain why MMI is a suitable optimality criterion in our case and we touch on the principles of VQ. Section 4 presents a combination of Lloyd clustering with KL divergence to implement MMI VQ of LLRs. In Section 5 we describe the system that we used as a test bench for our findings and we provide simulation results that compare the performance of MMI VQ to MSE VQ and simpler scalar quantizers.

2 System overview and problem definition

2.1 Hybrid ARQ

HARQ works together with ECCs such as turbo or LDPC codes. It starts out by sending only some of the redundancy bits generated by the code in addition to the information bits. The number of bits to be sent, hence the *code rate*, is chosen according to the channel conditions, in order to match the code rate to the estimated noise level. A decoding algorithm is run at the receiver, and if it fails (which is signaled by an error detecting code, such as CRC), the received bits are stored in the HARQ memory and more redundancy is sent (hence the expression *incremental redundancy*) by the transmitter in order to be combined with the previously received bits.

2.2 Marginal posterior probabilities and log-likelihood ratios

Rather than bits x, communication systems usually send symbols $s \in \mathbb{C}$ that are selected from a given alphabet or *constellation*, according to a mapping that associates groups of K bits to points in the constellation, $s = \text{map}(x_1, \ldots, x_K)$. The presence of a channel causes the received symbol r to be generally different from the transmitted symbol s. The distortion introduced by a wireless channel in the symbol is usually modeled as a multiplication by a complex factor and the addition of noise, so that we can write $r = as + n$. Based on r and on the channel model that is assumed, the receiver can associate marginal posterior probabilities $p(x_k|r),\ x_k \in \{0, 1\}$ to each of the K original bits. We talk about *soft decoding* when probabilities are fed into the decoding algorithm instead of the most likely bit, in which case we talk about *hard decoding*. Using probabilities instead of hard bits as

input, the performance of error decoding can be improved significantly. LLRs are compact representations of these marginal posterior probabilities, which for transmitted bit x_k can be computed as:

$$l_k = \log \frac{p(x_k = 1|r)}{p(x_k = 0|r)}. \quad (1)$$

Notice that using the LLR, l_k instead of $p(x_k|r)$ no loss of information occurs, since the operation (1) is invertible.

2.3 Efficient compression of LLRs

The desire to store LLRs instead of hard bits in the HARQ memory poses a storage issue. On one hand, LLRs are real-valued, which implies that enough physical bits should be allocated for each soft bit in order to store it with negligible loss of information. On the other hand, the system needs to keep soft bits in the HARQ memory until a message is correctly decoded, only then may the memory be released. Considering the ever-growing data rates of modern telecommunication systems, and the fact that several HARQ processes are preferably run in parallel in order to take advantage of the idle time between the moment when a message is transmitted and the moment when the transmitter is notified of whether a retransmission is needed, the HARQ memory is bound to require an increasing share of the receiving circuit. This is where data compression techniques can make a difference by reducing the number of physical bits required to represent LLRs and make the HARQ technique more memory efficient.

The problem we address in this paper is to find a lossy compression algorithm for efficient storage of LLRs in the HARQ memory. The algorithm should assure a significant compression ratio and negligible performance degradation in terms of throughput. Since LLRs are used to decode bits at the receiver, an appropriate way to assess the performance of our compression algorithm is to observe how it influences the bit error rate (BER) in comparison to the case where no quantization is performed. A lower BER implies higher throughput or capacity, in that less retransmissions are needed to send a message correctly to destination. Finally, the algorithm should be fast and its complexity should be low, so that the memory savings are not eliminated by the increased complexity.

If we refer to the communication system sketched in Section 2.2, we notice that a dependency appears between LLRs referring to bits that were mapped to the same symbol s (see Fig. 1a). A simple yet efficient way to capture this dependency and convert it to a higher compression ratio (or alternatively to improve performance) is vector quantization (VQ). These considerations led us to focus our investigation on the quantization of vectors of LLRs that are related to bits that have been mapped to the same symbol s, therefore only constellations where more than one bit is carried by each transmitted symbol fall within the scope of this paper, but the principle is applicable in general to vectors whose components are LLRs.

3 MMI as optimality criterion and VQ

Whenever we deal with lossy compression, we are confronted with a trade-off between rate and distortion [6, p. 301]. While an operational rate is easily measured by the amount of bits that are necessary to store the compressed data, distortion must be chosen carefully based on the application. In this paper we show that when storing LLRs, an information theoretical approach is much more suitable than conventional distortion measures such as MSE. In particular, we can look at the concatenation of symbol mapping, transmission

channel, and LLRs compression as a channel itself. According to Shannon's channel coding theorem, the channel capacity represents an upper bound for the rate at which data can be sent, provided a good channel code is chosen, with arbitrarily low error probability, therefore it makes sense to maximize the capacity of that equivalent channel. The expression for the channel capacity for a single (marginalized) LLR, l becomes:

$$C = I(X_k; Y_k) = H(X_k) - H(X_k|Y_k) = 1 + \sum_{x_k \in \{0,1\}} \sum_{y_k} p(x_k|y_k)p(y_k) \log_2 p(x_k|y_k), \quad (2)$$

where x_k is the original bit and $y_k = Q(l)$ represents the reconstruction value for its LLR, $I(\cdot;\cdot)$ is the mutual information between random variables, $H(\cdot)$ is the entropy and $H(\cdot|\cdot)$ is the conditional entropy of a random variable given that another is observed. If we assume equally likely input bits $H(X_k) = 1$, the channel capacity only depends on the conditional entropy of the original bit given that we observe the reconstructed LLR, $H(X_k|Y_k)$. The problem of maximizing channel capacity is equivalent to that of maximizing the *mutual information* $I(X;Y)$ between original bits and compressed LLRs, which is why we selected MMI as the design criterion for our compression scheme. Another way of putting it is expressing the distortion introduced by the lossy compression as mutual information loss between the case where no compression is performed, and the case where compression is taken into account i.e. $\Delta I = H(X|Y) - H(X|L)$. We shall see that this last alternative point of view will help us find a solution to our problem. We remark that the capacity would be higher if we avoid marginalizing the posterior probabilities, but that would deviate from our set-up since the turbo decoder accepts marginals as inputs.

On top of providing low rate and distortion, our compression scheme should also have low computational complexity, since it must be implemented on devices with limited computational power. For this reason, and for its capability of capturing the dependency between LLRs computed from the same symbol r, we chose to use fixed-rate VQ to perform LLR compression. VQ is based on the same principles as SQ, only it applies to higher-dimensional spaces. When performing VQ in $\mathbb{R}^K$, where $K > 1$, we need to define a set of N K-dimensional reconstruction values (or a *codebook*) and decision boundaries (in this case $(K-1)$-dimensional manifolds) that circumscribe the *regions* that make up the partition [7, p. 310]. We shall see in the next section that by defining a proper distance in $\mathbb{R}^K$ both the codebook design and the vector encoding tasks become much simpler. In particular, nearest neighbor encoding can be performed, by associating each input vector with the closest vector in the codebook.

4 MMI VQ Design for LLRs

A simple yet powerful technique for the design of VQ based on a set of training points representing its typical input is given by the Generalized Lloyd Algorithm [7, p. 362], here (Lloyd algorithm for brevity). In order to be applied, though, it requires a distance to be defined in $\mathbb{R}^K$. The algorithm itself consists of iterations over two steps to enforce the optimality conditions for a quantizer. In the first step, the training points are associated with the closest points in the codebook based on the selected distance measure (nearest neighbor condition), while in the second step the centroids of each set of training points are selected as the new reconstruction values (centroid condition), in order to minimize the average distance between training and reconstruction points. The algorithm is started by selecting an initial codebook, which can be provided by other design algorithms, or simply taken randomly from the training set.

Although the Lloyd algorithm is usually applied in conjunction with Euclidean distance, which results in the minimization of the MSE, it is easily applicable to other distance measures as well. In the following we will show that an MMI approach leads to KL divergence as a distortion measure. KL divergence is not a distance, due to its asymmetry and to the fact that it does not satisfy the triangle inequality. Nevertheless, Csiszàr proved geometric properties that allow it to be used as a distortion measure [8]. We can thereby adapt the Lloyd algorithm to the design of an MMI VQ for LLRs. We will now present the mathematical derivation of our approach. Our purpose is to quantize random vector $\mathbf{l} = \{l_1, \ldots, l_K\}$. To design an optimal codebook, we draw n_T samples of $\mathbf{l}$ as training points, which we can model with a random vector, $\mathbf{t} = \{t_1, \ldots, t_K\}$, with values in an alphabet $\mathcal{T}$. We shall explain in Section 5.1 why we resort to samples rather than to the actual input vector $\mathbf{l}$. Occurrences of $\mathbf{t}$ are equally likely because of the way they are generated, each having probability $p(\mathbf{t}) = 1/n_T$. The VQ splits $\mathbb{R}^K$ in K-dimensional regions R_i. Training points falling in R_i are associated with the quantizer output represented by index i. This index can be modeled as a random variable with alphabet $\mathcal{I} = \{1, \ldots, N\}$, where N represents the number of cells of the VQ.

In our derivation we use a Monte Carlo approach to compute probabilities. In this perspective, we chose to compute $p(i)$, namely the probability that an input point falls in region R_i, as n_i/n_T, where n_i is defined as the number of points associated with the output value indexed by i. K LLRs are grouped together in a vector and quantized simultaneously. As previously said, LLRs related to bits mapped to the same symbol are grouped together to exploit their coupling. With x_k we indicate the original bit to which the k^{th} LLR in a vector refers.

As stated in Section 3, an optimal VQ ensures minimum $\Delta I = H(X_k|Y_k) - H(X_k|L_k)$. To express ΔI in our case, we remark that entropies depend on the probability distributions and not on the actual values random variables or vectors take. From the point of view of probabilities, whether we refer to y_k, which is the reconstruction value for the k^{th} LLR, or to i, which is used to index it, makes no difference. At the same time, for $n_T \to +\infty$, $\mathbf{t}$ approximates $\mathbf{l}$, which enables us to write $\Delta I \approx H(X_k|I_{\mathcal{I}}) - H(X_k|\mathbf{T})$.

A fundamental ingredient to the computation of the conditional entropy $H(X_k|\mathbf{T})$ is the conditional probability $p(x_k|t_k)$, where t_k is the k^{th} component of vector $\mathbf{t}$, but this value can be directly inferred from the very values we need to quantize, simply by inverting the definition of LLR. Conditioning on t_k rather than on r does not modify the marginals, since LLRs encompass all our probabilistic knowledge after marginalization. Hence, we can equivalently refer to $p(x_k|r)$ or $p(x_k|t_k)$. If we now define $p = p(x_k = 0|r)$ and we reverse the definition of LLR:

$$t_k = \text{LLR}_k(r) = \log \frac{p(x=1|r)}{p(x=0|r)} = \log \frac{1-p}{p} \;\Rightarrow\; p = \frac{1}{1+e^{t_k}}, t_k \in \mathbb{R}. \tag{3}$$

Let us compute the conditional entropy of the original bit given the quantized version of the LLRs, which can be represented by the index associated with the vector quantizer output, i:

$$H(X_k|I_{\mathcal{I}}) = -\sum_{x_k \in \{0,1\}} \sum_{i=1}^{N} p(i)p(x_k|i) \log_2 p(x_k|i). \tag{4}$$

As mentioned earlier, we set $p(i) = n_i/n_T$, $n_i = |\{\mathbf{t} \in R_i\}|$ that is the probability of a given quantizer output is approximated by the relative number of training points that fall in the region associated with the output value i. The posterior $p(x_k|i)$ can be obtained by means

of marginalization:

$$p(x_k|i) = \sum_{\mathbf{t}\in R_i} p(x_k|i,\mathbf{t})p(\mathbf{t}) = \frac{n_i}{n_T}\sum_{\mathbf{t}\in R_i} p(x_k|t_k), \tag{5}$$

which corresponds to computing the average posterior probability for the original bit conditioned on the training points belonging to R_i. For the sake of our mathematical derivation, we can in general indicate this average as a function of the training point, $q(x_k|\mathbf{t}) = p(x_k|i),\ \mathbf{t} \in R_i$, where the value for q depends on the region $\mathbf{t}$ belongs to. The final expression for ΔI becomes:

$$\begin{aligned}\Delta I =& -\sum_{x_k\in\{0,1\}} \frac{1}{n_T}\sum_{\mathbf{t}\in\mathcal{T}} p(x_k|\mathbf{t})\log_2 q(x_k|\mathbf{t}) + \sum_{x_k\in\{0,1\}}\sum_{\mathbf{t}\in\mathcal{T}} p(\mathbf{t})p(x_k|\mathbf{t})\log_2 p(x_k|\mathbf{t}) \\ =& \frac{1}{n_T}\sum_{\mathbf{t}\in\mathcal{T}}\sum_{x_k\in\{0,1\}} p(x_k|\mathbf{t})\log_2\frac{p(x_k|\mathbf{t})}{q(x_k|\mathbf{t})} = \frac{1}{n_T}\sum_{\mathbf{t}\in\mathcal{T}} D_{\mathrm{KL}}(p_{x_k|\mathbf{t}}||q_{x_k|\mathbf{t}}),\end{aligned} \tag{6}$$

where $D_{\mathrm{KL}}(p||q)$ is the KL divergence between probability distributions p and q, defined on random variables which share the same alphabet. Equation 6 demonstrates that it is possible to compute ΔI as the average KL divergence between the posterior probabilities associated to each LLR value and the average posterior probabilities associated with the training points that belong to the same quantization region. Notice that the expression also holds for continuous random vectors by substituting integrals with summations.

Let us introduce some further notation with the purpose of presenting the modified Lloyd algorithm. $\mathbf{y}_i$ is the reconstruction value related to region R_i, and $\mathcal{C}$ represents our codebook. Quantization is done over vectors of LLRs, but so far we have only defined divergence between single LLRs. The overall divergence between LLR vectors is easily computed by summing the divergences between components. Making use of an abuse of notation (KL divergence is defined between probability distributions, not between LLRs), we let $D(t_k||y_k)$ denote the KL divergence between couples of LLRs and $D(\mathbf{t}||\mathbf{y}) = \sum_{k=1}^{K} D(t_k||y_k)$ the divergence between LLR vectors. Here we sum the vector components because the joint posterior distribution has been factorized in marginals, and this product translates into a summation in the KL divergence expression [4]. This point is explained more in detail in Section 5.1. In our algorithm, the closest reconstruction point to each training point is computed by means of an exhaustive search.

The approach to MMI LLR quantization presented in [5] was only applied to the case of BPSK transmission over AWGN channel. In [9] it is argued that an extension of that approach to different modulations and channel models is analytically infeasible, hence the paper deviates from MMI and proposes a simpler implementation that degrades the information rate. In this paper we show that by using a training set-based design process for SQ and VQ, MMI quantization can be performed in combination with any modulation scheme and channel model, provided that a set of samples drawn from the distribution of the LLRs we need to quantize is available. Consequently, our technique makes it possible to take advantage of the optimality of MMI quantization even in very complex scenarios. Ease of implementation and flexibility are among the main advantages of our method.

5 Simulations setup and results

In this section we present the settings we have chosen for our simulations and the results we obtained. In the first part we show that the performance of MMI VQ is significantly

Algorithm 1 Lloyd algorithm for the design of a Maximum Mutual Information Vector Quantizer

$\mathcal{C} \leftarrow N$ randomly picked vectors $\mathbf{t} \in \mathcal{T}$ {Initialize codebook}
$\Delta I_{\text{old}} \leftarrow +\infty$ {Initialize distortion to infinity}
repeat
  for all $\mathbf{t} \in \mathcal{T}$ **do** {assignment step}
    assign $\mathbf{t}$ to R_i so that $\mathbf{c}_i = \arg\min_{\mathbf{c}\in\mathcal{C}} D(\mathbf{t}||\mathbf{c})$
  end for
  for all $i \in \mathcal{I}$ **do** {update step}
    for $k = 1$ **to** K **do**
      $p_k \leftarrow \frac{1}{n_i} \sum_{\mathbf{t}\in R_i} \frac{1}{1+e^{t_k}}$ {compute average posterior probability}
    end for
    $\mathbf{y}_i \leftarrow \{\log \frac{1-p_1}{p_1}, \ldots, \log \frac{1-p_K}{p_K}\}$ {compute reconstruction LLR vector}
  end for
  use Eq. 6 to compute $\Delta I_{\text{new}} = \sum_{k=1}^{K} H(X_k|I_{\mathcal{I}}) - H(X_k|\mathbf{T})$
until $\Delta I_{\text{old}} - \Delta I_{\text{new}} < \epsilon$

higher than MMI SQ in terms of channel capacity. In the second part we show that this channel capacity gain corresponds to improved bit error rates when channel coding is taken into account.

5.1 Optimal vector quantizer design

The first step consisted in evaluating whether quantizers designed by means of our modified Lloyd algorithm could approach the theoretical channel capacity with a limited number of quantization cells. Since LTE provided us with the motivation for this work, we referred to the LTE standard when designing our test bench. LTE employs QPSK, 16-QAM and 64-QAM as higher-order modulations. Under the assumption of isotropic AWGN, these two-dimensional constellations can be split in two independent one-dimensional ones which can be treated separately, namely BPSK, 4-PAM and 8-PAM respectively. LLRs show a dependency that can be exploited by VQ when they are computed from the same received symbol and given that BPSK only carries one bit per symbol, we focus on 4-PAM and 8-PAM.

Pairs or triplets of bits x_k are mapped to s—a 4-PAM or 8-PAM transmission symbol, respectively—using Gray mapping. The type of dependency between LLRs referring to bits mapped to the same symbol t depends on the channel model. We used independent Rayleigh distributed gains $a \sim \mathcal{R}(1/\sqrt{2})$ and AWGN samples $n \sim \mathcal{N}(0, \sigma^2)$ to simulate the mobile radio channel. We assume perfect equalization at the receiver, so that the received symbol can be expressed as $r = s + n/a$. Under this assumption the above mentioned independence of the two dimensions still holds. The next step is to compute the LLRs relative to the original bits based on r. This requires computing $p(x_k = 0|r)$, and since we can look at a single realization of this channel as a Gaussian channel with variance $\sigma^2_{\text{eq}} = \sigma^2/a^2$, the computation comes down to applying Bayes' theorem to Gaussian distributions.

The reason why we need to resort to samples instead of using the actual probability density function for the LLRs l_k is that our channel model makes it hard to keep track of it. Samples are easy to generate and enable us to apply the framework developed so far to different scenarios as well, as long as samples of the VQ input are available. Once we obtain an optimal quantizer by means of the method in Section 4, the capacity of the channel

which connects original bits to their vector quantized LLRs can be computed as:

$$C_{\mathrm{VQ}} = \sum_{k=1}^{K} I(X_k; I_{\mathcal{I}}) = \sum_{k=1}^{K} (H(X_k) - H(X_k|I_{\mathcal{I}})), \tag{7}$$

where the expression for $H(X_k|I_{\mathcal{I}})$ is shown in (4) and $H(X_k)$ is trivially 1 if we assume that the original bits are evenly distributed. The summation is due to the fact that LLR decomposition implies marginalization. C_{VQ} has to be compared to the capacity of 4-PAM and 8-PAM over a Rayleigh fading channel, which can be evaluated using numerical integration. It is worth pointing out that the overall capacity of the channel is here computed as summation of the individual capacities of the channels that connect one bit with its corresponding LLR. When comparing this capacity with the theoretical channel capacity without quantization, the latter has to be computed in a *bitwise* rather than in a *symbol-wise* fashion, meaning that instead of using the joint posterior probabilities of the bits mapped to the same symbol, we use their marginals. Due to the mutual dependency between bits if conditioned on the received symbol, the joint distribution is not equal to the products of the marginals, and this implies a capacity loss which is inherent in marginalization.

The calculation of C_{Rayleigh} involves two steps. The first is to compute the capacity of 2^K-PAM over a Gaussian channel as a function of the variance σ^2:

$$C_{2^K\text{-PAM}} = I(X;R) = \sum_{k=1}^{K} I(X_k;R) = \sum_{k=1}^{K} (h_k(R) - h(N)) \tag{8}$$

where $h_k(R)$ is the differential entropy of the distribution for the received symbol triggered by the k^{th} bit, which is with probability 1/2 a mixture of Gaussians with equal variance centered in the constellation points where the k^{th} bit is 0 and with probability 1/2 centered where it is 1, and $h(N)$ is the differential entropy of a Gaussian distribution [6, p. 263]. The second step is to compute the average of $C_{2^K\text{-PAM}}(\sigma^2/a^2)$ over the Rayleigh distribution.

Figure 1b shows the channel capacity for 4-PAM with and without quantization, for different values of the *rate* $b = (\log_2 N)/K$, denoting the number of bits needed to store an LLR. With as few as 2.5 bits per LLR, the noise introduced by MMI VQ causes only a capacity loss equivalent to 0.2 dB in E_S/N_0 compared to the theoretical bound. VQ yields an approximate 0.1 dB capacity gain over SQ. Optimal SQs are separately designed for l_0 and l_1. Optimal allocation of the $\log_2 N$ bits for l_0 and l_1 has been performed by means of an exhaustive search.

5.2 Interaction with error correcting coding

The capacity gain caused by MMI implies lower bit error rate (BER), when channel coding is performed. For the choice of an ECC we referred to the LTE standard [1] and chose a rate 1/3 turbo code with block length $l_{\mathrm{bl}} = 6144$ bits, which uses two identical 8-state recursive convolutional codes as constituent encoders [1].

Fig. 2a shows that there is a strong correlation between the results in terms of channel capacity and the corresponding results in terms of BER when using VQ or SQ. In particular, we verified that MMI VQ with $b = 2.5$ bits causes a 0.2 dB loss in terms of BER w.r.t. the case were no quantization is performed, which is consistent with the 0.2 dB capacity loss. Moreover, simulations show that MMI SQ with $b = 2.5$ performs slightly worse than MMI VQ with $b = 2$, which means that VQ allows to save more than 0.5 bit per LLR over SQ.

[1] Transfer function $G(D) = d_1(D)/d_0(D)$, $d_0(D) = 1 + D^2 + D^3$, $d_1(D) = 1 + D + D^3$

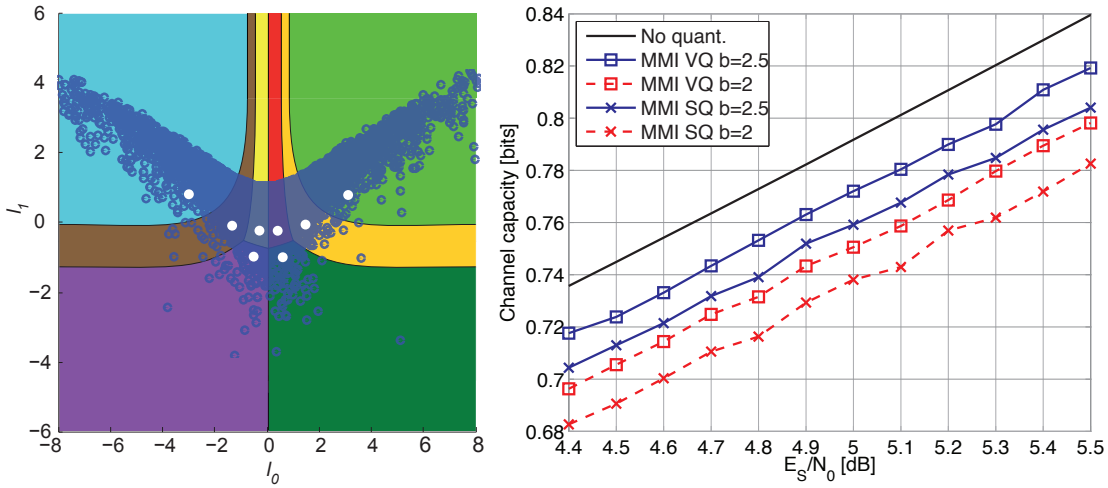


Figure 1: On the left, samples $t \in \mathcal{T}$ (scatter points), regions R_i and reconstruction points $\mathbf{y}$ (white circles) in case of 4-PAM with Rayleigh-fading channel and on the right capacity comparison between VQ and SQ

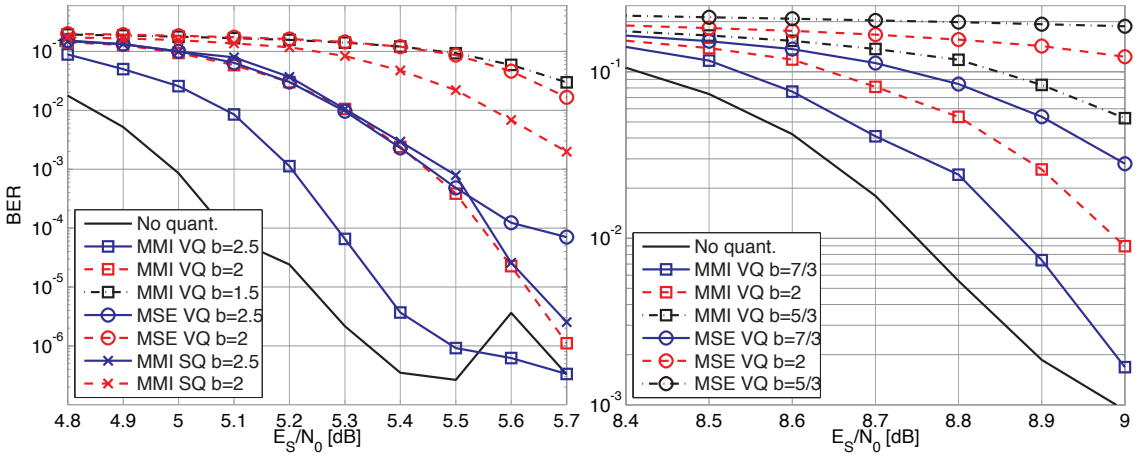
Lastly, Using MSE instead of MMI quantization causes an approximate performance loss of 0.1 dB with $b = 2.5$ and an even larger loss when $b = 2$. In general, while MMI VQ still performs decently, MSE VQ and MMI SQ perform much worse when $b = 2$ bits. Fig. 2b shows that when moving to higher-order modulations, VQ using 2 bits per LLR suffice to limit the performance loss to about 0.2 dB, which goes well with the fact that the more dimensions involved, the more dependency can be exploited and converted into performance gains.

In addition, some exploratory tests showed that the quantizers are robust to mismatch (which means that a lower number of codebooks need to be stored and makes it possible to employ predesigned VQs rather than designing them on the fly). Due to the brute force algorithm we use when performing encoding, the complexity is linear w.r.t. the number of levels (it would be logarithmic in case of SQ), but the small codebook size in use limits the complexity increase. Moreover efficient encoding techniques can be used to reduce the complexity even further.

6 Conclusion and future work

A method to design and perform VQ on LLRs according to an MMI optimality criterion has been presented with the aim of reducing memory requirements in modern HARQ implementations. The method is based on the use of KL divergence as a distortion measure for LLRs and proves effective when applied to multi-bit modulations in the case of channel models that create a coupling between LLRs. It was shown that the capacity gain assured by the MMI VQ approach over usual MSE VQ or MMI scalar quantization corresponds to lower bit error rate when channel coding is taken into account. Using as little as 2 bits per soft bit when 4-PAM modulation is taken into account, MMI VQ performs fairly well and outperforms by far MSE VQ and MMI SQ. Further gains are achieved with higher-order modulations.

Our channel model was quite simple, therefore future work should focus on testing



(a) 4-PAM **(b)** 8-PAM

Figure 2: BER comparison between VQ and SQ, Rayleigh-fading channel

whether the advantages of MMI VQ in terms of negligible channel capacity loss and BER increase hold up when more realistic models are used. The complexity of the algorithm and the overhead for codebook storage should also be evaluated to determine in more detail to what extent it is viable and advantageous.

References

[1] 3GPP Technical Specification Group Radio Access Network, *Multiplexing and Channel Coding*, 3GPP TS 36.212 v.8.6.0 (March 2009)

[2] E. Soljanin, R. Liu and P. Spasojevic, Hybrid ARQ with Random Transmission Assignments, *DIMACS Series in Disc. Math. and Theoretical Comp. Sc.* Vol. 66 (2004), 321-327

[3] H.V. Poor and J.B. Thomas, *Applications of Ali-Silvey Distance Measures in the Design of Generalized Quantizers for Binary Decision Systems*, IEEE Trans. on Comm., Vol. 25, No. 9 (Sept. 2007), 893-900

[4] J.E. Shore and R.M. Gray, *Minimum Cross-Entropy Pattern Classification and Cluster Analysis*, IEEE Trans. on Pattern Analysis and Machine Intelligence, Vol. 4, No. 1 (Jan. 1982), 11-17

[5] W. Rave, *Quantization of Log-Likelihood Ratios to Maximize Mutual Information*, IEEE Signal Processing Letters, Vol. 16, No. 4 (2009), pp. 283-286

[6] T.M. Cover and J.A. Thomas, *Elements of Information Theory, 2^{nd} edition*, Wiley (2006)

[7] A. Gersho and R.M. Gray, *Vector Quantization and Signal Compression*, KAP (1992)

[8] I. Csiszár, *I-Divergence Geometry of Probability Distributions and Minimization Problems*, The Annals of Probability, Vol. 3, No. 1 (1975)

[9] C. Novak, P. Fertl and G. Matz, *Quantization for Soft-Output Demodulators in Bit-Interleaved Coded Modulation Systems*, 2009 IEEE International Symposium on Information Theory, pp. 1070-1074