Channel-Optimized Vector Quantization with Mutual Information as Fidelity Criterion

Andreas Winkelbauer*, Gerald Matz*, and Andreas Burg†

*Institute of Telecommunications, Vienna University of Technology, Gusshausstrasse 25/389, 1040 Vienna, Austria
†Telecommunications Circuits Laboratory, EPF Lausanne, 1015 Lausanne, Switzerland
email: {andreas.winkelbauer, gerald.matz}@tuwien.ac.at, andreas.burg@epfl.ch

***Abstract*—We consider the problem of channel-optimized vector quantization (COVQ) with mutual information as fidelity criterion. This problem is relevant in a communications context, where the goal is to maximize the end-to-end rate. We propose an algorithm which is similar to the information bottleneck method and solves the considered COVQ problem. In contrast to existing COVQ designs, the proposed method implicitly optimizes the labels of the quantizer outputs, thereby avoiding separate label optimization. The usefulness of the proposed method is corroborated by application examples and numerical results.**

I. Introduction

Channel-optimized vector quantization (COVQ) is a well-known approach to lossy joint source-channel coding [1], [2] that minimizes the end-to-end distortion when the quantizer output is transmitted over a noisy channel. The squared-error distortion is the most commonly used distortion measure for lossy source coding problems. While the squared-error distortion often leads to analytically tractable results and may be suitable for continuous-amplitude sources, it is less appropriate for other problems, e.g., in a communications context.

In digital communications, low-rate quantization is of interest due to constraints regarding bandwidth, memory size, power consumption, and chip area. Example scenarios in which quantizer outputs are corrupted by a subsequent channel are distributed systems like relay networks with noisy links, distributed inference in sensor networks, and practical receiver implementations with unreliable memories [3].

In this paper, we consider COVQ with the aim of *maximizing the achievable end-to-end information rate*; hence, we use mutual information as fidelity criterion. This problem is fundamentally different from distortion-based quantizer design in that the reproduction alphabet at the quantizer output, and hence the "distance" between quantizer input and output, is immaterial since mutual information depends only on the statistics. Although quantizer design for communication systems has recently attracted some attention [4]–[8], we appear to be the first to solve the COVQ problem together with label optimization.

The remainder of this paper is organized as follows. In Sec. II we state the COVQ problem mathematically. Sec. III reviews the *information bottleneck* (IB) method [9]. Our IB-based COVQ design algorithm is presented in Sec. IV. Application examples are given in Sec. V and conclusions are provided in Sec. VI.

Funding by WWTF Grant ICT12-054.

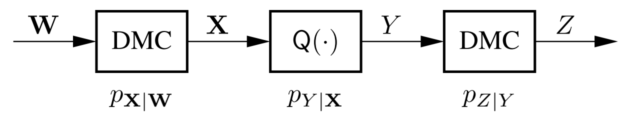

Figure 1: System model for COVQ. The quantizer is designed to maximize the end-to-end mutual information $I(\mathbf{W};Z)$.

Notation: We use boldface letters for vectors and uppercase letters for random variables. Sets are denoted by calligraphic letters and their cardinality is denoted by $|\cdot|$. We write $\mathbb{E}\{\cdot\}$ and $\delta_{i,j}$ for the expectation and the Kronecker delta, respectively. Entropy, mutual information, and relative entropy are denoted by $H(\cdot)$, $I(\cdot\,;\cdot)$, and $D(\cdot\|\cdot)$, respectively. The Hamming distance is denoted by $d_{\mathrm{H}}(\cdot,\cdot)$ and $\mathcal{N}(\mu,\sigma^2)$ is the Gaussian distribution with mean μ and variance σ^2.

II. Problem Statement

We consider the system depicted in Fig. 1. The sequence $\mathbf{W} = (W_1,\dots,W_k) \in \mathcal{W}^k$ (here, $\mathcal{W}$ denotes the input alphabet) with distribution $p_{\mathbf{W}}(\mathbf{w})$ is transmitted over the *discrete memoryless channel* (DMC) $p_{\mathbf{X}|\mathbf{W}}(\mathbf{x}|\mathbf{w}) = \prod_{l=1}^{k} p_{X|W}(x_l|w_l)$ with length-k output sequence $\mathbf{X} \in \mathcal{X}^k$ (here, $\mathcal{X}$ denotes the output alphabet). The sequence $\mathbf{X}$ is then passed through a k-dimensional *vector quantizer* (VQ) $\mathsf{Q} : \mathcal{X}^k \to \mathcal{Y}$, described by the transition probabilities $p_{Y|\mathbf{X}}(y|\mathbf{x})$. The number of quantization levels is denoted by $m \triangleq |\mathcal{Y}| \leq |\mathcal{X}|^k$. The quantizer output $Y = \mathsf{Q}(\mathbf{X})$ is then transmitted over another DMC $p_{Z|Y}(z|y)$, which finally yields Z from the alphabet $\mathcal{Z}$. We will refer to $p_{\mathbf{X}|\mathbf{W}}(\mathbf{x}|\mathbf{w})$ and $p_{Z|Y}(z|y)$ as the "unquantized channel" and the "forward channel", respectively.

Our aim is to design the quantizer such that the achievable rate over the end-to-end channel $p_{Z|\mathbf{W}}(z|\mathbf{w})$, is maximized. Quantifying the achievable rate by the mutual information $I(\mathbf{W};Z)$ [10], the COVQ $\mathsf{Q}^\star$ is defined by

$$p^\star_{Y|\mathbf{X}}(y|\mathbf{x}) = \underset{p_{Y|\mathbf{X}}(y|\mathbf{x})}{\arg\max}\, I(\mathbf{W};Z). \qquad (1)$$

Note that the quantizer is optimally matched to both the unquantized channel and the forward channel. We have formulated the COVQ problem in terms of DMCs. If the unquantized channel is memoryless with continuous output, our framework can still be applied by discretizing $\mathbf{X}$ to the required precision. The extension to continuous-output forward channels is more difficult, since $p_{Z|Y}$ may depend on p_Y, (e.g., through an average power constraint), but p_Y is not known *a priori*.

Since $\mathbf{W} - \mathbf{X} - Y - Z$ forms a Markov chain, we have

$$p_{Z|\mathbf{W}}(z|\mathbf{w}) = \sum_{y\in\mathcal{Y}} p_{Z|Y}(z|y) \sum_{\mathbf{x}\in\mathcal{X}^k} p_{Y|\mathbf{X}}(y|\mathbf{x}) p_{\mathbf{X}|\mathbf{W}}(\mathbf{x}|\mathbf{w}). \tag{2}$$

Hence, (1) can be rewritten as in (3) at the bottom of the page. For fixed $p_{\mathbf{W}}$, $I(\mathbf{W};Z)$ is convex in $p_{Z|\mathbf{W}}$ [10]; furthermore, it can be shown that (2) implies that $I(\mathbf{W};Z)$ is convex in $p_{Y|\mathbf{X}}$, too, provided $p_{\mathbf{X}|\mathbf{W}}$ and $p_{Z|Y}$ are also fixed. Therefore, (3) is a convex *maximization* problem which makes it hard to find a global optimum $p^\star_{Y|\mathbf{X}}$. Hence, we restrict our attention to finding a locally optimal solution of (3) in the following. In general we have $p_{Y|\mathbf{X}} \in \mathcal{P}$, where $\mathcal{P}$ is a probability simplex. However, due to the structure of the problem in (3), the *optimal quantizer is in fact deterministic*, i.e., we have $p^\star_{Y|\mathbf{X}}(y|\mathbf{x}) \in \{0,1\}$. This follows from the fact that deterministic distributions are extreme points of the (convex and closed) probability simplex and from the following result.

Lemma 1 (cf. [11]). *Let $\mathcal{C}$ be a closed and convex set that has at least one extreme point. A convex function $f : \mathcal{C} \to \mathbb{R}$ that attains a maximum over $\mathcal{C}$ attains the global maximum at some extreme point of $\mathcal{C}$.*

With the optimal quantizer, some of the m quantizer outputs may be unused, i.e., $p_Y(y) = 0$ for some $y \in \mathcal{Y}$. This essentially depends on the forward channel and is in contrast to conventional (non-channel-optimized) VQ where in general $p_Y(y) > 0$ for all $y \in \mathcal{Y}$.

For an error-free forward channel, (1) is equivalent to

$$p^\star_{Y|\mathbf{X}}(y|\mathbf{x}) = \operatorname*{arg\,max}_{p_{Y|\mathbf{X}}(y|\mathbf{x})} I(\mathbf{W};Y), \tag{4}$$

i.e., to conventional VQ design with mutual information as fidelity criterion. In particular, the method we propose in Section IV can also be used to solve (4) by setting $p_{Z|Y}(z|y) = \delta_{z,y}$. We note that even for the simpler problem (4), there exists in general no efficient algorithm for finding the globally optimal solution. Only for the special case of scalar quantization (SQ) and binary channel inputs ($k = 1$ and $|\mathcal{W}| = 2$) the global optimum of (4) can be found via dynamic programming [8]. However, dynamic programming cannot be used to solve the COVQ design problem (3).

Finally, we note that the labeling of the quantizer outputs enters in the objective function in (3) through $p_{Z|Y}$. Hence, the solution of (3) consists of an optimal partition of $\mathcal{X}^k$ together with the corresponding optimal labels of the m quantizer outputs.

III. The Information Bottleneck Method

Before we proceed with the discussion of our IB-based COVQ design algorithm, we first review the IB method. In [9], Tishby et al. introduced the notion of *relevance through another variable*. Relevant information in a random variable is defined as the mutual information between this random variable and a relevance variable W. The IB method finds a compressed representation Y of the data X such that Y preserves as much information about W as possible. Figuratively speaking, the compression variable Y constitutes a "bottleneck" through which the information that X provides about W is squeezed. The IB method requires the joint statistics of W and X to be known and amounts to solving

$$p^\star_{Y|X}(y|x) = \operatorname*{arg\,min}_{p_{Y|X}(y|x)} \big[I(X;Y) - \beta I(W;Y)\big], \tag{5}$$

where β is a Lagrange multiplier for the relevant information $I(W;Y)$. It was shown in [9] that the optimal assignment $p_{Y|X}(y|x)$ is characterized by

$$p_{Y|X}(y|x) = \frac{p_Y(y)}{\psi(x,\beta)} \exp\big[-\beta D\big(p_{W|X}(w|x) \| p_{W|Y}(w|y)\big)\big], \tag{6}$$

where $\psi(x,\beta)$ is a normalization (partition) function. Eq. (6) describes the solution of (5) only implicitly since $p_Y(y)$ and $p_{W|Y}(w|y)$ themselves depend on $p_{Y|X}(y|x)$. We have

$$p_Y(y) = \sum_x p_{Y|X}(y|x) p_X(x), \tag{7}$$

$$p_{W|Y}(w|y) = \frac{1}{p_Y(y)} \sum_x p_{W,X}(w,x) p_{Y|X}(y|x), \tag{8}$$

where (8) follows from the fact that $W - X - Y$ forms a Markov chain.

The IB equations (6)-(8) are satisfied simultaneously at the minima of the *free energy* $\mathbb{E}\{-\log\psi(X,\beta)\}$. The minimization of the free energy can be carried out by performing the following alternating iterations (ℓ is the iteration index) [9]:

$$p_Y^{(\ell+1)}(y) = \sum_x p_X(x) p^{(\ell)}_{Y|X}(y|x), \tag{9a}$$

$$p^{(\ell+1)}_{W|Y}(w|y) = \frac{1}{p_Y^{(\ell+1)}(y)} \sum_x p_{W,X}(w,x) p^{(\ell)}_{Y|X}(y|x), \tag{9b}$$

$$p^{(\ell+1)}_{Y|X}(y|x) = \frac{p_Y^{(\ell+1)}(y)\exp\Big[-\beta D\Big(p_{W,X}(w|x)\Big\|p^{(\ell+1)}_{W|Y}(w|y)\Big)\Big]}{\psi^{(\ell+1)}(x,\beta)}. \tag{9c}$$

$$\begin{aligned} p^\star_{Y|\mathbf{X}}(y|\mathbf{x}) = \operatorname*{arg\,max}_{p_{Y|\mathbf{X}}(y|\mathbf{x})} \sum_{\mathbf{w}\in\mathcal{W}^k} p_{\mathbf{W}}(\mathbf{w}) \sum_{z\in\mathcal{Z}} \sum_{y\in\mathcal{Y}} p_{Z|Y}(z|y) \sum_{\mathbf{x}\in\mathcal{X}^k} p_{Y|\mathbf{X}}(y|\mathbf{x}) p_{\mathbf{X}|\mathbf{W}}(\mathbf{x}|\mathbf{w}) \\ \cdot \log \frac{\sum_{y\in\mathcal{Y}} p_{Z|Y}(z|y) \sum_{\mathbf{x}\in\mathcal{X}^k} p_{Y|\mathbf{X}}(y|\mathbf{x}) p_{\mathbf{X}|\mathbf{W}}(\mathbf{x}|\mathbf{w})}{\sum_{\mathbf{w}'\in\mathcal{W}^k} p_{\mathbf{W}}(\mathbf{w}') \sum_{y\in\mathcal{Y}} p_{Z|Y}(z|y) \sum_{\mathbf{x}\in\mathcal{X}^k} p_{Y|\mathbf{X}}(y|\mathbf{x}) p_{\mathbf{X}|\mathbf{W}}(\mathbf{x}|\mathbf{w}')}. \end{aligned} \tag{3}$$

The initialization $p^{(0)}_{Y|X}(y|x)$ is chosen randomly. The equations (9a)-(9c) constitute the basic IB algorithm which yields a locally optimal solution $p^{\star}_{Y|X}(y|x)$.

The choice of β in the IB algorithm allows us to trade off compression rate $I(X;Y)$ against relevant information $I(W;Y)$. For large β, we obtain weak compression and preserve much relevant information; for small β, compression is strong and little relevant information is preserved. When we design deterministic quantizers using the IB method, we fix the size of the output alphabet $\mathcal{Y}$. Thus the quantization rate is upper bounded as $I(X;Y) = H(Y) \leq \log|\mathcal{Y}|$ and we let $\beta \to \infty$ to preserve as much relevant information as possible. The IB method has first been used for quantizer design in a communications context in [12].

IV. Channel-Optimized Vector Quantizer Design

In this section we develop an algorithm that is based on the IB method and yields a locally optimal solution of (3). To this end, we will make the following major modifications compared to the basic IB algorithm (9):

- Since we know that the optimal COVQ is deterministic, we ensure that the output of the algorithm corresponds to a deterministic quantizer.
- We let $\beta \to \infty$ since we are interested in preserving as much relevant information as possible at a given quantization rate.
- We include the forward channel $p_{Z|Y}(z|y)$ into the quantizer optimization and we adapt the iterative algorithm accordingly.

We first rewrite the objective function as

$$I(\mathbf{W};Z) = I(\mathbf{W};Y) + \underbrace{I(\mathbf{W};Z|Y)}_{=0} - I(\mathbf{W};Y|Z) \tag{10}$$

$$= I(\mathbf{W};\mathbf{X}) - I(\mathbf{W};\mathbf{X}|Y) \tag{11}$$

$$- I(\mathbf{W};\mathbf{X}|Z) + I(\mathbf{W};\mathbf{X}|Y,Z)$$

$$= I(\mathbf{W};\mathbf{X}) - I(\mathbf{W};\mathbf{X}|Z). \tag{12}$$

Since the first term in (12) does not depend on $p_{Y|\mathbf{X}}$, we can rewrite (3) as follows:

$$p^{\star}_{Y|\mathbf{X}}(y|\mathbf{x}) = \underset{p_{Y|\mathbf{X}}(y|\mathbf{x})}{\arg\min}\ I(\mathbf{W};\mathbf{X}|Z) \tag{13}$$

$$= \underset{p_{Y|\mathbf{X}}(y|\mathbf{x})}{\arg\min}\ \mathbb{E}\big\{\mathbb{E}\left\{C(\mathbf{X},Z)|\mathbf{X}=\mathbf{x}\right\}\big\}, \tag{14}$$

where we have defined

$$C(\mathbf{x},z) \triangleq D\big(p_{\mathbf{W}|\mathbf{X}}(\mathbf{w}|\mathbf{x})\big\|p_{\mathbf{W}|Z}(\mathbf{w}|z)\big). \tag{15}$$

We can further rewrite the conditional expectation in (14) as

$$\mathbb{E}\left\{C(\mathbf{X},Z)|\mathbf{X}=\mathbf{x}\right\} = \sum_{y\in\mathcal{Y}} p_{Y|\mathbf{X}}(y|\mathbf{x}) \sum_{z\in\mathcal{Z}} p_{Z|Y}(z|y)C(\mathbf{x},z). \tag{16}$$

We next choose $p_{Y|\mathbf{X}}(y|\mathbf{x})$ such that $\mathbb{E}\left\{C(\mathbf{X},Z)|\mathbf{X}=\mathbf{x}\right\}$ is minimized for each $\mathbf{x} \in \mathcal{X}^k$. To this end, we note that for fixed y the second sum in (16) is a scalar. Hence, we let

$$p_{Y|\mathbf{X}}(y|\mathbf{x}) = \delta_{y,y^{\star}(\mathbf{x})}, \tag{17}$$

where $y^{\star}(\mathbf{x})$ is the particular $y \in \mathcal{Y}$ which minimizes the second sum in (16):

$$y^{\star}(\mathbf{x}) = \underset{y\in\mathcal{Y}}{\arg\min} \sum_{z\in\mathcal{Z}} p_{Z|Y}(z|y)C(\mathbf{x},z). \tag{18}$$

By minimizing (16) for each $\mathbf{x} \in \mathcal{X}^k$ separately, we also minimize (14). Due to (17) we furthermore have

$$p_{Z|\mathbf{X}}(z|\mathbf{x}) = \sum_{y\in\mathcal{Y}} p_{Z|Y}(z|y)p_{Y|\mathbf{X}}(y|\mathbf{x}) = p_{Z|Y}\big(z|y^{\star}(\mathbf{x})\big). \tag{19}$$

Based on the above relations, we now formulate our algorithm for COVQ design, see Algorithm 1.

Algorithm 1 *COVQ design algorithm*

Input: $\varepsilon > 0$, $n > 0$, $k > 0$, $p_{\mathbf{W},\mathbf{X}}(\mathbf{w},\mathbf{x})$, $p_{Z|Y}(z|y)$, $\mathcal{X}$, $\mathcal{Y}$, $\mathcal{Z}$
Initialization: $\overline{C}_{-1} \leftarrow \infty$, $\eta \leftarrow \infty$, $\ell \leftarrow 0$, randomly initialize $C_{-1}(\mathbf{x},z) \in \mathbb{R}_{+}, \forall \mathbf{x} \in \mathcal{X}^k$ and $\forall z \in \mathcal{Z}$

1: **while** $\eta \geq \varepsilon$ **and** $\ell < n$ **do**
2: **for all** $\mathbf{x}$ **do**
3: $y^{\star} \leftarrow \arg\min_{y\in\mathcal{Y}} \sum_{z\in\mathcal{Z}} p_{Z|Y}(z|y)C_{\ell-1}(\mathbf{x},z)$
4: $p^{(\ell)}_{Y|\mathbf{X}}(y|\mathbf{x}) \leftarrow \delta_{y,y^{\star}},\ y \in \mathcal{Y}$
5: $p^{(\ell)}_{Z|\mathbf{X}}(z|\mathbf{x}) \leftarrow p_{Z|Y}(z|y^{\star}),\ z \in \mathcal{Z}$
6: **end for**
7: $p^{(\ell)}_{Z}(z) \leftarrow \sum_{\mathbf{x}\in\mathcal{X}^k} p^{(\ell)}_{Z|\mathbf{X}}(z|\mathbf{x})p_{\mathbf{X}}(\mathbf{x})$
8: $p^{(\ell)}_{\mathbf{W}|Z}(\mathbf{w}|z) \leftarrow \frac{1}{p^{(\ell)}_{Z}(z)} \sum_{\mathbf{x}\in\mathcal{X}^k} p_{\mathbf{W},\mathbf{X}}(\mathbf{w},\mathbf{x})p^{(\ell)}_{Z|\mathbf{X}}(z|\mathbf{x})$
9: $C_{\ell}(\mathbf{x},z) \leftarrow D\Big(p_{\mathbf{W}|\mathbf{X}}(\mathbf{w}|\mathbf{x})\big\|p^{(\ell)}_{\mathbf{W}|Z}(\mathbf{w}|z)\Big)$
10: $\overline{C}_{\ell} \leftarrow \sum_{\mathbf{x}\in\mathcal{X}^k} p(\mathbf{x}) \sum_{z\in\mathcal{Z}} p^{(\ell)}_{Z|\mathbf{X}}(z|\mathbf{x})C_{\ell}(\mathbf{x},z)$
11: $\eta \leftarrow (\overline{C}_{\ell-1} - \overline{C}_{\ell})/\overline{C}_{\ell}$
12: $\ell \leftarrow \ell + 1$
13: **end while**

Unless it terminates after n iterations, our algorithm converges within an accuracy of ε to a local optimum of (3) (depending on the initialization $C_{-1}(\mathbf{x},z)$) and returns a deterministic quantizer. Algorithm 1 can be run repeatedly and the best solution retained; this helps to avoid getting stuck at a bad local optimum.

We note that our algorithm finds the optimal quantizer *jointly* with corresponding labels for the quantizer output (cf. lines 3 and 4 in Algorithm 1). This is in contrast to distortion-based COVQ design algorithms (e.g., the LBG algorithm [13]), which usually require that the labeling be fixed in advance and therefore the NP-hard labeling problem has to be considered separately. Furthermore, the *a posteriori* probabilities $p_{\mathbf{W}|Z}(\mathbf{w}|z)$ are obtained as a by-product of the COVQ design which simplifies subsequent receiver processing.

V. Application Examples

We next discuss apply our COVQ design to quantization for relaying and to receivers with unreliable memory. For simplicity we focus on SQ in the following.

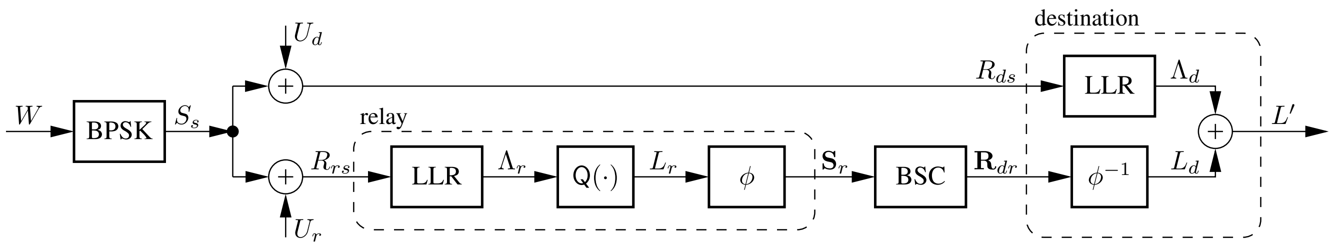

Figure 2: Quantize-and-forward relay channel. The channel-optimized quantizer $\mathsf{Q}(\cdot)$ maximizes $I(W; L_d)$.

A. Quantize-and-Forward Relaying

We consider the relay channel depicted in Fig. 2. The source transmits BPSK modulated symbols $S_s \in \{\pm 1\}$ and the signals received at the destination and relay are given by

$$R_{ds} = S_s + U_d, \qquad R_{rs} = S_s + U_r. \tag{20}$$

Here, $U_d \sim \mathcal{N}(0, \sigma_d^2)$ and $U_r \sim \mathcal{N}(0, \sigma_r^2)$ are additive Gaussian noise terms. The destination and the relay then compute the *log-likelihood ratios* (LLRs)

$$\Lambda_d = \frac{2}{\sigma_d^2} R_{ds}, \qquad \Lambda_r = \frac{2}{\sigma_r^2} R_{rs}. \tag{21}$$

Next, the relay performs quantization with a resolution of $n = \log_2(m)$ bits, yielding $L_r = \mathsf{Q}(\Lambda_r) \in \mathcal{L}$ with $|\mathcal{L}| = m$. The index of the quantizer output is then mapped to a length-n binary label $\mathbf{S}_r = \phi(L_r)$ which is transmitted over the relay-destination channel, modeled by a binary symmetric channel (BSC) with crossover probability ε. The destination receives the noisy label $\mathbf{R}_{dr}$ according to

$$p(\mathbf{R}_{dr} = \mathbf{r} | \mathbf{S}_r = \mathbf{s}) = \varepsilon^{d_{\mathrm{H}}(\mathbf{r},\mathbf{s})} (1-\varepsilon)^{n - d_{\mathrm{H}}(\mathbf{r},\mathbf{s})}. \tag{22}$$

The noisy label $\mathbf{R}_{dr}$ is converted back to the corresponding quantized LLR $L_d = \phi^{-1}(\mathbf{R}_{dr}) \in \mathcal{L}$, where $L_d = L_r$ with probability $(1-\varepsilon)^n$. Finally, the destination sums up the LLRs received via the direct link and via the relay. The resulting LLR $L' = \Lambda_d + L_d$ can be passed on to a decoder.

In our example we choose $\sigma_d^2 = 2\sigma_r^2$, i.e., the SNR on the source-relay link is 3 dB higher than on the source-destination link. The capacity on the relay-destination link is chosen as 0.5 bit per channel use (bpcu), i.e., $\varepsilon = 0.11003$. The relay uses an SQ with $n = 2$ bits, i.e., 4 quantization levels.

In Fig. 3 we plot the mutual information $I(W; L')$ versus the source-destination SNR σ_d^{-2} in dB. The red curve ('$\circ$' markers) shows the rates achievable with our channel-optimized quantizer design. The blue curve ('+' markers) constitutes a simple upper bound given by 2 bit quantization with error-free transmission of the quantizer output. By using capacity-achieving coding on the relay-destination link, we can achieve the black curve ('□' markers) by 1 bit quantization with error-free transmission (rate-1/2 code, 2 channel uses; ignoring complexity and delay due to coding). We can see that for low to moderate rates channel-optimized SQ (blocklength equal to 1) performs within 1 dB of separate channel coding (indefinite blocklength) and SQ. The solid (dashed) green

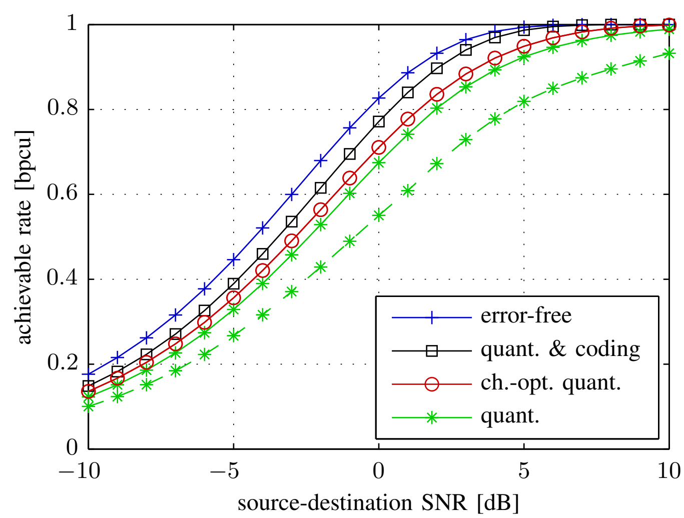

Figure 3: Comparison of achievable rates for quantize-and-forward relaying.

curve ('$*$' markers) shows the rates achievable by non-channel-optimized quantization maximizing $I(W; L_r)$ using the best (worst) labeling for the quantizer output.

Our channel-optimized approach uniformly outperforms conventional (non-channel-optimized) quantization. Although the performance gain is small, our algorithm is attractive because it avoids separate label optimization and allows us to choose the labeling ϕ arbitrarily.

B. Unreliable Memories

We next consider transmission of BPSK modulated symbols (as depicted in Fig. 4) over the AWGN channel $R = S + U$, where $S \in \{\pm 1\}$ and $U \sim \mathcal{N}(0, \sigma^2)$. The receiver calculates the LLRs $\Lambda = 2R/\sigma^2$ which are then quantized, yielding $L = \mathsf{Q}(\Lambda)$. Next, the quantizer output is mapped to the binary label $\mathbf{B}$ which is stored in unreliable memory. Reading the noisy label $\mathbf{B}'$ from the memory yields the quantized LLR L'. We use the stuck-at channel (SAC) to model the failure of bit cells in unreliable memory [14]. For the SAC with error probability $0 < \varepsilon < 1$ and equally likely stuck-at errors, each bit cell is in one of the following three states:

- $\Xi = \xi_0$: error-free bit cell (with probability $1 - \varepsilon$)
- $\Xi = \xi_+$: bit cell is stuck at "1" (with probability $\varepsilon/2$)
- $\Xi = \xi_-$: bit cell is stuck at "0" (with probability $\varepsilon/2$)

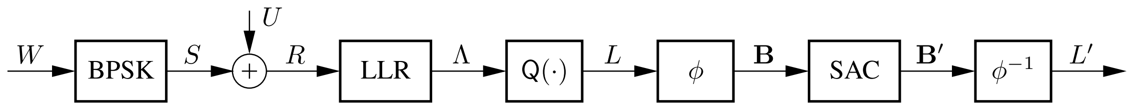

Figure 4: Receiver with unreliable LLR memory. The channel-optimized quantizer $\mathsf{Q}(\cdot)$ maximizes $I(W; L')$.

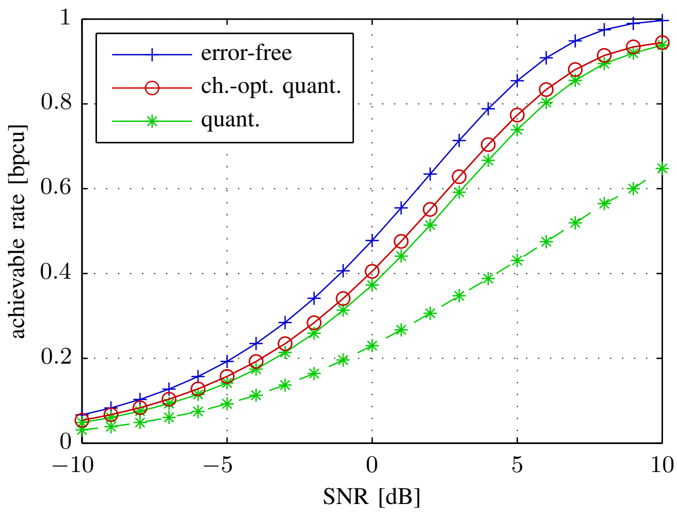

Figure 5: Comparison of achievable rates for receiver processing with unreliable LLR memory.

The capacity of a single bit cell in the SAC model is zero since the content of the bit cell is independent of the input with positive probability. However, we use the SAC to model unreliable memory in the following way: we assume that each bit cell fails independently and data is stored without knowledge about the state of the individual bit cells. This is equivalent to multiple channel uses of a single bit cell where the state of the bit cell is chosen at random before each channel use. In this case, the SAC with error probability ε and equally likely stuck-at errors is equivalent to a BSC with crossover probability $\varepsilon/2$. Indeed, we have

$$p_{B'|B}(b'|b) = \sum_{\xi \in \{\xi_0, \xi_+, \xi_-\}} p_{B'|B,\Xi}(b'|b,\xi) p_{\Xi}(\xi) \tag{23}$$

$$= (1-\varepsilon)\delta_{b',b} + \frac{\varepsilon}{2}\delta_{b',0} + \frac{\varepsilon}{2}\delta_{b',1} \tag{24}$$

$$= \begin{cases} 1 - \frac{\varepsilon}{2}, & b = b', \\ \frac{\varepsilon}{2}, & b \neq b', \end{cases} \tag{25}$$

where $b, b' \in \{0, 1\}$.

For our example we assume $\varepsilon = 0.11003$ and we perform SQ with a resolution of $n = 3$ bits. In Fig. 5 we plot the mutual information $I(W; L')$ versus the SNR σ^{-2} in dB. The red curve ('$\circ$' markers) shows the rates achievable by our channel-optimized quantizer design. The blue curve ('$+$' markers) constitutes a simple upper bound given by 3 bit quantization with error-free storage of the quantizer output. The solid (dashed) green curve ('$*$' markers) shows the rates achievable by non-channel-optimized quantization maximizing $I(W; L)$ using the best (worst) labeling for the quantizer output.

We note that in this case there are $(2^3)! = 40320$ different bit mappings and the performance penalty may be very large if the optimal mapping is not found. This is in contrast to our approach which outperforms non-channel-optimized quantization without the need to perform separate optimization of the bit labels.

VI. Conclusions

We have studied COVQ design in a communications context with the aim to maximize the end-to-end rate. We have formulated the problem as a convex maximization of a mutual information. After reviewing the IB method, we have provided an IB-like algorithm which provides a locally optimal solution to the considered COVQ design problem. The proposed algorithm can also be used for non-channel-optimized VQ design and allows us to circumvent the NP-hard label optimization problem. This makes the proposed algorithm attractive in terms of performance and computational complexity. We have provided two application examples which confirm that our algorithm provides superior performance.

References

[1] N. Farvardin, "A study of vector quantization for noisy channels," *IEEE Trans. Inf. Theory*, vol. 36, no. 4, pp. 799–809, July 1990.

[2] N. Farvardin and V. Vaishampayan, "On the performance and complexity of channel-optimized vector quantizers," *IEEE Trans. Inf. Theory*, vol. 37, no. 1, pp. 155–160, Jan. 1991.

[3] S. Ghosh and K. Roy, "Parameter variation tolerance and error resiliency: New design paradigm for the nanoscale era," *Proc. IEEE*, vol. 98, no. 10, pp. 1718–1751, Oct. 2010.

[4] J. Singh, O. Dabeer, and U. Madhow, "On the limits of communication with low-precision analog-to-digital conversion at the receiver," *IEEE Trans. Comm.*, vol. 57, no. 12, pp. 3629–3639, Dec. 2009.

[5] G. Zeitler, R. Kötter, G. Bauch, and J. Widmer, "On quantizer design for soft values in the multiple-access relay channel," in *Proc. IEEE ICC 2009*, June 2009.

[6] C. Novak, P. Fertl, and G. Matz, "Quantization for soft-output demodulators in bit-interleaved coded modulation systems," in *Proc. IEEE Int. Symposium on Inf. Theory*, July 2009, pp. 1070–1074.

[7] M. Danieli, S. Forchhammer, J. Andersen, L. Christensen, and S. Christensen, "Maximum mutual information vector quantization of log-likelihood ratios for memory efficient HARQ implementations," in *Proc. Data Compression Conference (DCC 2010)*, March 2010, pp. 30–39.

[8] B. Kurkoski and H. Yagi, "Concatenation of a discrete memoryless channel and a quantizer," in *Proc. IEEE Information Theory Workshop (ITW 2010)*, Jan. 2010.

[9] N. Tishby, F. Pereira, and W. Bialek, "The information bottleneck method," in *Proc. 37th Allerton Conf. on Communication, Control, and Computing*, Sept. 1999, pp. 368–377.

[10] T. M. Cover and J. A. Thomas, *Elements of Information Theory*. New York: Wiley, 1991.

[11] D. P. Bertsekas, A. Nedić, and A. E. Ozdaglar, *Convex Analysis and Optimization*. Athena Scientific, 2003.

[12] G. Zeitler, R. Kötter, G. Bauch, and J. Widmer, "Design of network coding functions in multihop relay networks," in *Proc. 5th Int. Symposium on Turbo Codes and Related Topics*, Sept. 2008, pp. 249–254.

[13] Y. Linde, A. Buzo, and R. Gray, "An algorithm for vector quantizer design," *IEEE Trans. Comm.*, vol. 28, no. 1, pp. 84–95, Jan. 1980.

[14] C. Roth, C. Benkeser, C. Studer, G. Karakonstantis, and A. Burg, "Data mapping for unreliable memories," in *Proc. 50th Annual Allerton Conference on Communication, Control, and Computing (Allerton 2012)*, Oct. 2012, pp. 679–685.