Available at www.ComputerScienceWeb.com
POWERED BY SCIENCE DIRECT®

Signal Processing 83 (2003) 1519–1528

SIGNAL PROCESSING

www.elsevier.com/locate/sigpro

Memory efficient adaptation of vector quantizers to time-varying channels☆

Norbert Görtz[a,*], Jörg Kliewer[b]

[a] *Institute for Communications Engineering (LNT), Munich University of Technology (TUM), 80290 Munich, Germany*
[b] *Institute for Circuits and Systems Theory, University of Kiel, Kaiserstr. 2, 24143 Kiel, Germany*

Received 9 September 2002; received in revised form 26 February 2003

Abstract

Channel-optimized vector quantization (COVQ) is approximated by the novel channel-adaptive scaled vector quantization (CASVQ). This new method uses a reference codebook that is optimal for one specific channel condition. However, for a bit-error rate being different from the design assumption for the reference codebook, all codevectors are scaled by a common factor, which depends on the channel condition. It is shown by simulations that a performance close to that of COVQ can be achieved in many practically important situations. Without a significant increase in complexity, the new CASVQ-scheme can be adapted to time-varying channels by adjusting the scaling factor to the current bit-error probability. Another advantage is that only one codebook needs to be stored for all error probabilities, while for COVQ either the performance degrades significantly due to channel mismatch, or a large set of codebooks must be available at the encoder and the decoder.

Keywords: Joint source-channel coding; Channel-optimized vector quantization; Time-varying channels; Complexity and memory requirements

1. Introduction

Channel-optimized vector quantization (COVQ) [4,5] achieves strong quality-improvements over conventional vector quantization (VQ) if the transmission channel is noisy. Variations of COVQ in which simulated and deterministic annealing are used have been proposed, e.g., in [6,10]; the algorithms work superior to "normal" COVQ due to improvements in the codebook design that avoid getting stuck in poor local optima. Although a great deal of the work on COVQ has been done for the binary symmetric channel, the codebook training and the design of optimal encoder-decoder pairs for soft source decoding after the transmission with soft-decision demodulation have also been considered, e.g., in [1,11]. Recently, the adaptation of COVQ to time-varying channels has become a subject of interest [8], where COVQ design approaches are proposed for different types of missing channel information.

In this paper a new memory-efficient COVQ approximation for time-varying channels is presented. More precisely, we address the problem how to

☆ This paper was presented in part at the IEEE International Symposium on Information Theory, Washington, DC, USA, June 2001.

* Corresponding author. Tel.: +49-89-289-23494; fax: +49-89-289-23490.
E-mail address: norbert.goertz@ei.tum.de (N. Görtz).

0165-1684/03/$ - see front matter
doi:10.1016/S0165-1684(03)00071-9

limit the memory and complexity requirements in a COVQ-framework such that these are almost the same as in the conventional VQ case, while keeping good performance for all (time-varying) channel conditions. It is shown that this goal can be achieved by simply scaling a COVQ reference codebook depending on the actual channel bit error rate, which leads to a strong reduction of the overall memory and complexity requirements compared to the COVQ approach. A closely related method has been stated in [2], where scaling is introduced to decrease the distances between the codewords and thus to alleviate the quality decrease at the decoder output due to index permutations caused by bit errors. In our work, however, we employ codevector-scaling to obtain a memory efficient COVQ codebook approximation which is designed for a bit-error probability being different from that used for the reference codebook.

This paper is organized as follows. After a discussion of complexity and memory issues for VQ and COVQ in Section 2, the new approximation of COVQ called channel-adaptive scaled vector quantization (CASVQ) is introduced in Section 3. In Section 4 the performance of the new scheme is compared with COVQ.

2. Memory and complexity issues for VQ and COVQ

Fig. 1 shows the model of the used transmission system. It consists of a channel-optimized vector quantizer with the codebook C, a binary symmetric transmission channel that causes index-permutations by bit errors, and a decoder that essentially performs a table-lookup. The bit-error probability is assumed to be known at both the encoder and the decoder.

The idea of channel-optimized vector quantization (COVQ) is to exploit the knowledge about the channel in the design of the codebook and in the encoding algorithm. The COVQ codebook design algorithm [4] can be regarded as an extension of the classical LBG-algorithm [9] for noiseless channels. In the following, we use the performance of COVQ [4] as a reference for the approximations that are investigated in this paper. Note that the COVQ distance measure, which is used in both the codebook design procedure and the encoding operation, is different from that in

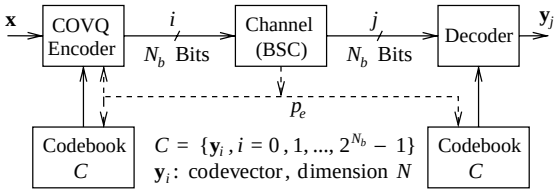

Fig. 1. Transmission system with a channel-optimized vector quantizer.

conventional VQ. This is due to the fact that transitions from the transmitted indices i to some indices j at the receiver occur with probabilities $P_{j|i}$ that depend on the channel. For instance, if we transmit over a binary-symmetric channel (BSC) with bit-error probability p_e, the transition probabilities are given by

$$P_{j|i} = p_e^{h(i,j)} \cdot (1 - p_e)^{N_b - h(i,j)}, \tag{1}$$

where N_b is the number of bits required to encode the index i and $h(i,j)$ is the Hamming distance between the transmitted and the received index, i.e., $h(i,j)$ is the number of bit errors inserted by the channel. The probabilities $P_{j|i}$ are used in the COVQ-encoder to minimize the expected distortion at the receiver output due to the quantization decision. The expected distortion, the *COVQ distance measure*, is given by [4]

$$d_{\mathrm{covq}}(\mathbf{x}, \mathbf{y}_i) = \sum_{j=0}^{N_C - 1} P_{j|i} \cdot d_{\mathrm{vq}}(\mathbf{x}, \mathbf{y}_j), \tag{2}$$

where $\mathbf{x}$ is the data-vector to be quantized and $\mathbf{y}_i$ the codevector with the index $i = 0, 1, \ldots, N_C - 1$. The number of codevectors is denoted as $N_C \doteq 2^{N_b}$ (size of the codebook), and $d_{\mathrm{vq}}(\mathbf{x}, \mathbf{y}_j)$ is the distance measure that would be used in conventional VQ. In COVQ-encoding, (2) is computed for each codevector $\mathbf{y}_i$, and the index i_{opt} corresponding to the codevector that produces the minimum value $d_{\mathrm{covq}}(\mathbf{x}, \mathbf{y}_{i_{\mathrm{opt}}})$ is selected for the transmission over the channel. Besides, (2) is also used in the codebook training process. Often (and therefore also in this paper) the mean-squared error

$$d_{\mathrm{vq}}(\mathbf{x}, \mathbf{y}_i) = \frac{1}{N} \sum_{l=0}^{N-1} (x_l - y_{i,l})^2 \tag{3}$$

is used as the *VQ distance measure*. The vector dimension is denoted by N and the vector components of $\mathbf{y}_i$ and $\mathbf{x}$ are indexed by l. The computation of (3) requires $3N$ floating-point operations per codevector of dimension N, where "floating-point operation" will be used as a generic term for addition, subtraction, multiplication, or division in the following. It is a standard technique to simplify the implementation by expanding the sum in (3) as follows:

$$d_{\mathrm{vq}}(\mathbf{x},\mathbf{y}_i) = \frac{1}{N}\sum_{l=0}^{N-1} x_l^2 - \frac{2}{N}\sum_{l=0}^{N-1} x_l y_{i,l} + \frac{1}{N}\sum_{l=0}^{N-1} y_{i,l}^2 = \frac{1}{N}\sum_{l=0}^{N-1} x_l^2 + \frac{2}{N}\, d'_{\mathrm{vq}}(\mathbf{x},\mathbf{y}_i). \tag{4}$$

The first term is non-negative and does not depend on the codevector. Since, furthermore, $2/N$ is a constant factor it is sufficient to minimize

$$d'_{\mathrm{vq}}(\mathbf{x},\mathbf{y}_i) = w(\mathbf{y}_i) - \sum_{l=0}^{N-1} x_l y_{i,l} \tag{5}$$

with

$$w(\mathbf{y}_i) \doteq \frac{1}{2}\sum_{l=0}^{N-1} y_{i,l}^2 \tag{6}$$

by a proper selection of the codevector index i. In what follows, $d'_{\mathrm{vq}}(\mathbf{x},\mathbf{y}_i)$ will be referred to as the *simplified VQ distance measure*. The (halves of the) "energies" $w(\mathbf{y}_i)$ of the codevectors can be precomputed and stored in advance (since they do not depend on the data $\mathbf{x}$), so by use of N_{C} additional scalar memory locations we may reduce the complexity of the codebook search to $2N$ floating-point operations per codevector.

For COVQ, the calculation of the expected distortion (2) for each of the codevectors requires the values of the VQ distance measure (3) for all codevectors and the corresponding index transition probabilities $P_{j|i}$. Simplified implementations, which are algorithmically equivalent to the minimization of (2), have been stated in [5]: while the decoder uses the COVQ-codebook directly, the encoder uses a "transformed" codebook that includes the index transition probabilities. We will briefly describe the method here: if we insert (3) into (2) and expand the sums we obtain

$$\begin{aligned} d_{\mathrm{covq}}(\mathbf{x},\mathbf{y}_i) &= \sum_{j=0}^{N_{\mathrm{C}}-1} P_{j|i}\cdot d_{\mathrm{vq}}(\mathbf{x},\mathbf{y}_j) \\ &= \sum_{j=0}^{N_{\mathrm{C}}-1} P_{j|i}\cdot\frac{1}{N}\sum_{l=0}^{N-1}(x_l - y_{j,l})^2 \\ &= \frac{1}{N}\sum_{l=0}^{N-1} x_l^2 + \frac{2}{N}\left\{\underbrace{\sum_{j=0}^{N_{\mathrm{C}}-1} P_{j|i}\,\frac{1}{2}\sum_{l=0}^{N-1} y_{j,l}^2}_{\overline{w(\mathbf{y}_i)}} - \sum_{l=0}^{N-1} x_l \underbrace{\sum_{j=0}^{N_{\mathrm{C}}-1} P_{j|i}\cdot y_{j,l}}_{\overline{y_{i,l}}}\right\}. \end{aligned} \tag{7}$$

Thus, similar as in the VQ-case, it is sufficient to minimize the *simplified COVQ distance measure*

$$d'_{\mathrm{covq}}(\mathbf{x},\mathbf{y}_i) = \overline{w(\mathbf{y}_i)} - \sum_{l=0}^{N-1} x_l \overline{y_{i,l}}, \tag{8}$$

with

$$\overline{w(\mathbf{y}_i)} \doteq \sum_{j=0}^{N_{\mathrm{C}}-1} P_{j|i}\,\frac{1}{2}\sum_{l=0}^{N-1} y_{j,l}^2 = \sum_{j=0}^{N_{\mathrm{C}}-1} P_{j|i} w(\mathbf{y}_j) \tag{9}$$

and

$$\overline{y_{i,l}} \doteq \sum_{j=0}^{N_{\mathrm{C}}-1} P_{j|i}\cdot y_{j,l}, \tag{10}$$

instead of the minimization of (2). The "transformed" codevectors $\overline{\mathbf{y}_i} = \{\overline{y_{i,l}},\ l = 0,\ldots,N-1\}$ from (10), which can be interpreted as the expectation of the codevector at the receiver conditioned on the possibly transmitted index i, and the "energies" $\overline{w(\mathbf{y}_i)}$ from (9) may be stored at the encoder in place of the actual codebook, which is only required at the decoder. Thus, the memory requirements and the complexity of COVQ encoding by minimization of (8) over i

are not larger[1] than in the conventional VQ-case (where (5) is minimized)—as long as the channel is time-invariant.

If the channel is time-varying the channel-optimized codebook might not be matched to the current channel statistics (channel mismatch), i.e., the performance degrades, compared to the optimal case. If the assumption of the bit-error probability p_e for COVQ-codebook training differs only slightly from its true value on the channel, the performance of unmatched COVQ is not significantly worse compared to optimally matched COVQ [4]. But if, for instance, a COVQ-codebook designed for a bit-error probability of $p_e = 0.05$ is used on an uncorrupted channel ($p_e = 0$), a significant loss in the "clean-channel performance" can be observed.

One way to improve the performance in such a situation is to switch between a finite number of codebooks at the encoder and the decoder depending on the current channel state [7]; the codebooks can cover the range of possible channel variations (although no perfect match is possible since, in general, p_e is a real number). The robustness of COVQ against small variations of p_e preserves close-to-optimum performance if the number of codebooks is high enough. However, this strategy requires the storage of several codebooks at both the encoder and the decoder, which may be quite memory consuming. In the next section a new algorithm is stated that does not show this disadvantage by reducing the number of required codebooks to just one for all channel conditions.

3. Channel-adaptive scaled vector quantization (CASVQ)

3.1. Basic principle

In Fig. 2 codebooks with $N_C = 32$ two-dimensional codevectors are depicted that were trained for a Gauss–Markov source with a correlation coefficient of $\rho = 0.9$. Such a source model is representative for the long-term statistics of many source signals such as speech, audio, and images but also for the parameters that are extracted from a block of input samples by advanced source coding schemes. The expected mean-squared error (2) was used as a distance measure for COVQ-codebook training. The plots show the codevectors (marked by "×") that result from the training procedure for the COVQ-codebooks [5] (with "splitting"[2] for the initialization) for several assumptions of the bit-error probability p_e on a binary symmetric channel (BSC). For $p_e = 0$ the codebook is equal to a conventional VQ-codebook (Fig. 2(a)).

Fig. 2 shows that the codevectors are placed closer to the "all-zero" vector (i.e. the average of all training-data vectors) when the channel quality gets worse. This is because a permutation of a transmitted index i (which, for instance, may quantize the data point Z_1) to some other index j (caused by channel errors) leads to a large distortion, if the associated codevectors have a large distance in the signal space; this is the case for the codevectors marked by i and j in Fig. 2(a). The distance between the vectors indexed by i and j in Fig. 2(b) is smaller, i.e., the distortion caused by channel errors that permute both vectors is also smaller. On the other hand, the quantization of data-points with large distance from the origin (e.g., Z_2) is less accurate in Fig. 2(b); thus, we have a loss in "clean-channel" performance ($p_e = 0$) using COVQ-codebooks optimized for $p_e > 0$. In other words: a quality decrease due to higher quantizer overload-distortion is traded for the quality improvement due to a higher robustness against index permutations.

A precise analysis of the positions of the codevectors reveals (Fig. 2(c) and (d)) that the codevectors form clusters, i.e., the codevectors are not only shrunken if p_e increases but they also have new relative locations. This clustering corresponds to empty coding regions that cause an implicit error control coding (redundant coding levels) as reported in [5]. However, a COVQ-codebook for $p_e = p_1 > 0$ may be viewed as a shrunken version of COVQ-codebook

[1] For COVQ on very noisy channels (e.g., bit-error probabilities $p_e > 0.05$) empty encoding regions (corresponding to implicit error control coding) can be observed: certain indexes are never selected by the encoder for transmission, i.e., the corresponding transformed codevectors do not need to be considered in the distance computation (8). If many empty regions occur, this can lead to a significantly lower complexity [5] compared to the "standard" VQ-case.

[2] It is known from literature [3] that the splitting method in VQ-codebook training leads to a relatively good index assignment. In case of COVQ-codebook training with $p_e > 0$ the index assignment is automatically optimized.

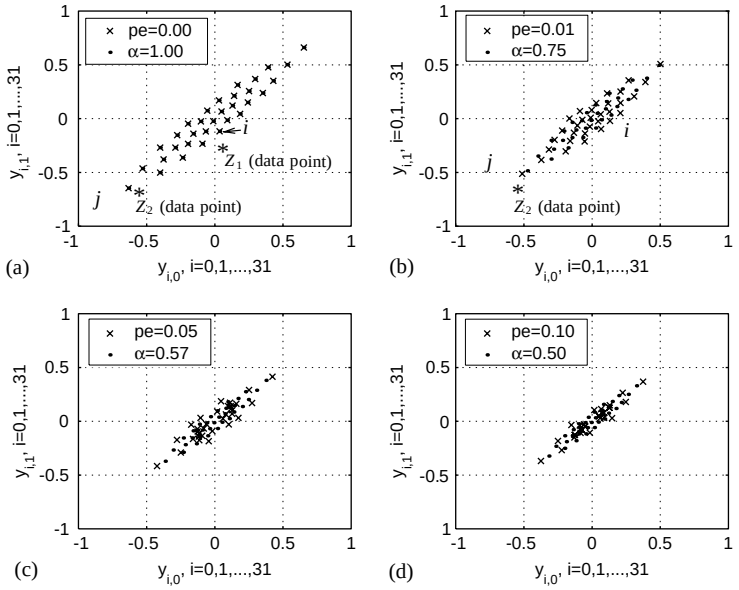

Fig. 2. Comparison of COVQ-codebooks ("×") and CASVQ-codebooks ("·"), both with $N_C = 32$ codevectors. The codebooks in plot (a) are identical to a conventional VQ-codebook designed by the LBG algorithm [9] (with "splitting").

optimized for $p_e < p_1$, if only the "rough shape" is considered.

This observation leads to the basic idea of *channel-adaptive scaled vector quantization* (*CASVQ*): the codevectors $\mathbf{y}_i^{(r)}$ from a reference COVQ-codebook are scaled by a channel dependent factor $\alpha(p_e) \geq 0$, where $\alpha(p_e) \neq 1$ if the channel does *not* match the training assumption of the reference codebook. Note that $\alpha(p_e) < 1$ if the reference codebook is obtained for a smaller BER compared to the actual one (as in the examples of Fig. 2) and $\alpha(p_e) > 1$ if a larger BER is used for the design of the COVQ reference codebook. Thus, the channel-matched COVQ-codevectors $\mathbf{y}_i(p_e)$ are approximated by the CASVQ-codevectors

$$\mathbf{y}_i'(p_e) = \alpha(p_e) \cdot \mathbf{y}_i^{(r)}. \tag{11}$$

For the sake of brevity we omit the dependence on p_e for $\mathbf{y}_i'(p_e)$ and $\alpha(p_e)$ in the following.

As an example, CASVQ-codevectors (marked by "·") are included in Fig. 2. They have been derived from the VQ reference codebook (COVQ-codebook with $p_e = 0$ as the design assumption) in Fig. 2(a) by the scaling factors α stated in the legends. It is obvious that the rough shape of the COVQ-codebooks (marked by "×") can be approximated by the CASVQ-codebooks quite well.

3.2. Optimization of CASVQ

The question arises, how to find appropriate scaling factors for each channel condition, i.e., the function $\alpha = f(p_e)$ is required. Unfortunately, there is no way to find the optimal factor analytically by variational techniques, because the average distortion for a training-set of source vectors depends on the individual quantization decisions that again depend on the scaling factor. Hence, the quantization decisions change, as the scaling factor α is varied to minimize the average distortion. However, since the function $\alpha = f(p_e)$ is individual for a codebook and its source signal, it may be determined in advance before the system is used. Thus, we can accept some computational complexity in the off-line optimization of the CASVQ-codebook for a reasonable number of samples from α and p_e.

As an example, let us think of a correlated Gaussian source signal, which is vector-quantized by a codebook with $N_c = 32$ two-dimensional codevectors; (2) is used as a distance measure for the quantization

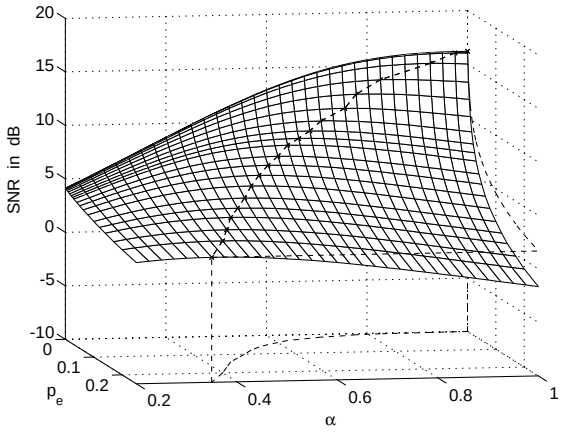

Fig. 3. Performance of CASVQ for codebooks with $N_C = 32$ two-dimensional codevectors which are used to encode a strongly correlated Gauss–Markov source (correlation coefficient $\rho = 0.9$). The SNR-value is plotted versus the bit-error probability p_e and the scaling factor α. A COVQ-codebook for $p_e = 0$ (i.e., a conventional VQ-codebook) is used as a reference codebook.

and a conventional VQ codebook is used as a reference codebook for CASVQ. Thus, the range of the scaling factors is limited to $0 < \alpha \leqslant 1$ in this case. The SNR-values, measured between the input of the quantizer and the output of a table-lookup decoder, are depicted in Fig. 3 in a three-dimensional plot, for a simulation of several (α, p_e)-pairs. Fig. 3 shows that the SNR-surface is quite smooth, i.e., a limited number of (α, p_e)-pairs is sufficient to achieve a good approximation of the best possible performance. The dashed curve on the SNR-surface indicates the best SNR (and the associated α-value) for each simulated value of p_e; the required function $\alpha = f(p_e)$ is given by the projection of the dashed curve into the p_e–α-plane. The projection of the dashed curve into the SNR-p_e-plane indicates the best SNR-values achievable by CASVQ for each value of p_e. This curve appears again in the simulation results in Fig. 5 (labeled "CASVQ, covq-dist.").

Note that the smooth behavior of the SNR surface in Fig. 3 and the function $\alpha = f(p_e)$, resp., holds for arbitrary input source processes. This is due to the fact that the COVQ codevectors may be interpreted as pre-computed mean-square estimates of the vector-quantized transmitted source symbols. If the channel is strongly distorted the mean-square estimate at the output of the decoder tends to zero for any zero-mean input source, since the COVQ code vector locations approach the zero vector. The CASVQ approach approximates this behavior for large p_e by $\alpha(p_e) \to 0$. Since the mean-square error between the source signal points and the CASVQ codevector locations is a continuous function in α the SNR surface and thus also $\alpha = f(p_e)$ are smooth functions.

By means of a simulation as described above, a close approximation of the function $\alpha = f(p_e)$ can be found for any practically relevant codebook in reasonable time.

3.3. Complexity and memory requirements of CASVQ on time-varying channels

If the CASVQ-codebook is used, the simplified COVQ distance measure (8) corresponds to

$$d'_{\mathrm{covq}}(\mathbf{x}, \alpha\mathbf{y}_i^{(r)}) = \overline{w(\alpha\mathbf{y}_i^{(r)})} - \sum_{l=0}^{N-1} x_l \cdot \overline{\alpha y_{i,l}^{(r)}}, \tag{12}$$

with

$$\overline{w(\alpha\mathbf{y}_i^{(r)})} = \sum_{j=0}^{N_C-1} P_{j|i} \frac{1}{2} \sum_{l=0}^{N-1} \alpha^2 (y_{j,l}^{(r)})^2$$

$$= \alpha^2 \cdot \sum_{j=0}^{N_C-1} P_{j|i} w(\mathbf{y}_j^{(r)}) = \alpha^2 \cdot \overline{w(\mathbf{y}_i^{(r)})} \tag{13}$$

and

$$\overline{\alpha y_{i,l}^{(r)}} = \sum_{j=0}^{N_C-1} P_{j|i} \cdot \alpha y_{j,l}^{(r)}$$

$$= \alpha \cdot \sum_{j=0}^{N_C-1} P_{j|i} \cdot y_{j,l}^{(r)} = \alpha \cdot \overline{y_{i,l}^{(r)}}, \tag{14}$$

where (9) and (10) have been used. Note that both α and $P_{j|i}$ are functions of the *current* bit-error probability p_e, which is distinct from the reference bit-error probability. Thus, (12) may be written as

$$d'_{\mathrm{covq}}(\mathbf{x}, \alpha\mathbf{y}_i^{(r)}) = \alpha^2 \cdot \overline{w(\mathbf{y}_i^{(r)})} - \alpha \cdot \sum_{l=0}^{N-1} x_l \cdot \overline{y_{i,l}^{(r)}}. \tag{15}$$

As we can observe from (15) it is possible to adapt CASVQ to any value of p_e by use of an appropriate

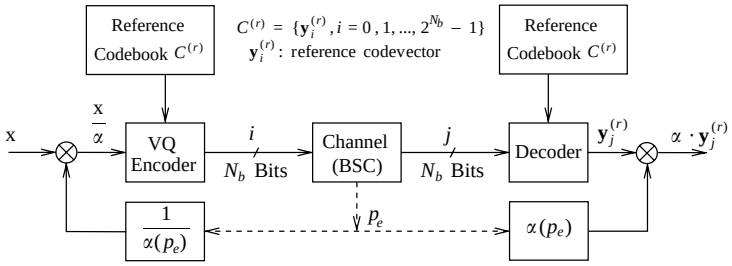

Fig. 4. CASVQ system with VQ distance measure for encoding.

factor α and only *one* reference codebook for calculating the "transformed" reference codevectors $\overline{\mathbf{y}_i^{(r)}}$ and the "energies" $\overline{w(\mathbf{y}_i^{(r)})}$ in (15). Whenever p_e changes, $\overline{\mathbf{y}_i^{(r)}}$ and $\overline{w(\mathbf{y}_i^{(r)})}$ have to be recomputed from the reference codebook by applying (9) and (10), respectively. Thus, the proposed CASVQ method with COVQ distance measure for encoding, which will be denoted with "CASVQ, covq-dist." in the following, is more complex than conventional VQ. Furthermore, memory for two codebooks, the reference codebook $\mathbf{y}_i^{(r)}$ and the current transformed codebook $\overline{\mathbf{y}_i^{(r)}}$ used for encoding, is required at the encoder. Note that the "energies" $\overline{w(\mathbf{y}_i^{(r)})}$ only need to be stored for the current bit-error probability, so here, no additional memory is required for time-variant channels. In contrast, for perfectly channel-matched COVQ a new codebook has to be trained and stored in both the encoder and the decoder for each value of p_e.

In order to save the complexity for the recomputations (due to a change of p_e) and the memory for the second codebook in the CASVQ approach, one may use the conventional VQ distance measure instead of the COVQ distance measure for encoding. Certainly, this is another approximation that will decrease the performance compared to the optimal COVQ; however, the degradation is only moderate but the simplification of the encoding is significant as we will see in the following. From (5) and (6) we obtain

$$d'_{\mathrm{vq}}(\mathbf{x}, \alpha\mathbf{y}_i^{(r)}) = \alpha^2 \cdot \underbrace{\frac{1}{2}\sum_{l=0}^{N-1}(y_{i,l}^{(r)})^2}_{w(\mathbf{y}_i^{(r)})} - \alpha \cdot \sum_{l=0}^{N-1} x_l y_{i,l}^{(r)}$$

$$= \alpha^2 \left(w(\mathbf{y}_i^{(r)}) - \sum_{l=0}^{N-1} \frac{x_l}{\alpha}\, y_{i,l}^{(r)} \right). \qquad (16)$$

Since α^2 in (16) is equal for each codevector, it is sufficient to minimize

$$d''_{\mathrm{vq}}(\mathbf{x}, \alpha\mathbf{y}_i^{(r)}) \doteq w(\mathbf{y}_i^{(r)}) - \sum_{l=0}^{N-1} \left(\frac{x_l}{\alpha}\right) y_{i,l}^{(r)} \qquad (17)$$

over i in place of (16). Now, all we have to do additionally (compared with conventional VQ) is to scale the components of the *input* data vector $\mathbf{x}$ by $1/\alpha$ and to rescale the output codevector $\mathbf{y}_j^{(r)}$ by α; a small table to represent the function $\alpha = f(p_e)$ is also required. The rest is the same as in conventional VQ including the memory and complexity requirements.

The CASVQ-system with the VQ-distance measure for encoding is depicted in Fig. 4. If this system is used, the corresponding simulation results are labeled "CASVQ, vq-dist.". However, the simulation results in Section 4 show that the quality decrease obtained by replacing the COVQ distance measure with the simple VQ distance measure is only small.

Note that a change of α due to varying channel statistics affects both the encoder and decoder codebook. Therefore, we assume that a backward channel is available (which holds true for many wireless and wireline communication systems) such that the estimated bit-error rate at the decoder can be communicated to the encoder.

4. Simulation results

In Fig. 5 the performances of COVQ and CASVQ for the transmission of a strongly autocorrelated

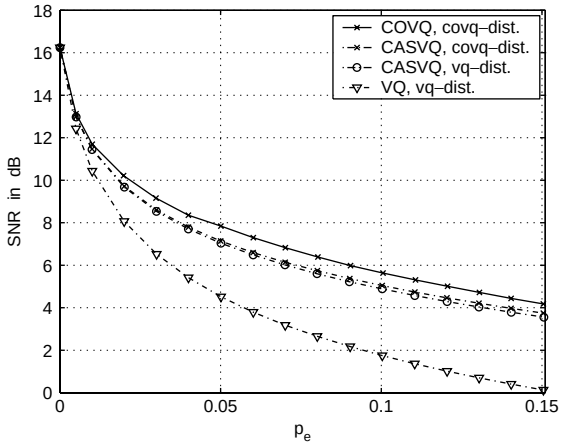

Fig. 5. Performance of COVQ, CASVQ, and VQ for the quantization of a strongly correlated Gauss–Markov source; codebooks with $N_C = 32$ two-dimensional codevectors are used.

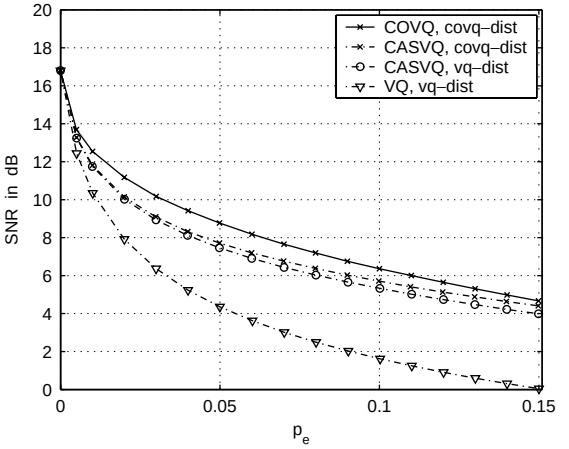

Fig. 6. Performance of COVQ, CASVQ, and VQ for the quantization of a strongly correlated Gauss–Markov source; codebooks with $N_C = 128$ three-dimensional codevectors are used.

Gauss–Markov source signal (correlation coefficient $\rho = 0.9$) over a binary symmetric channel are depicted; the quantizations are carried out by codebooks with 32 two-dimensional codevectors. The COVQ-codebooks are optimally matched to the true value of p_e on the channel; the scaling factor $\alpha(p_e)$ of the CASVQ-codebooks is adapted to p_e as described in Section 3.

As expected, the COVQ codebook with COVQ-distance measure for encoding (curve labeled "COVQ, covq-dist.") works best of all, but there is only a moderate loss for "CASVQ, covq-dist.". Moreover, the performance of CASVQ with the conventional VQ distance measure for encoding ("CASVQ, vq-dist.") is only slightly inferior to "CASVQ, covq-dist.", i.e., the "better" distance measure does not significantly improve the performance of CASVQ.

The results for conventional VQ with VQ distance measure for encoding ("VQ, vq-dist.") have also been included in Fig. 5 to evaluate the difference between COVQ and CASVQ: when a moderate loss in performance compared to optimally channel-matched COVQ can be accepted, it is possible to apply a CASVQ-codebook with the simple VQ distance measure for encoding. This allows a very simple and memory-efficient adaptation of the coding scheme to time-varying channels while keeping most of the performance-gain of optimally matched COVQ.

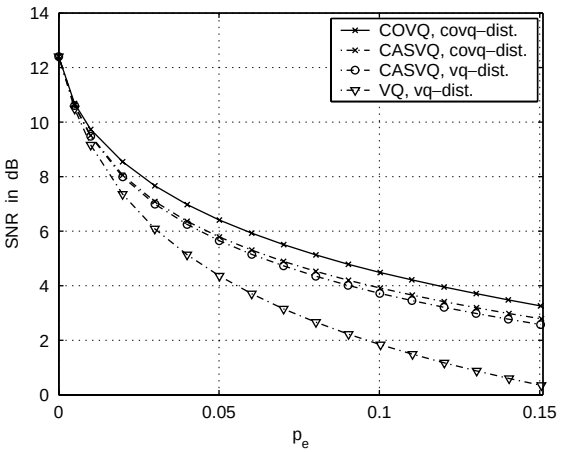

Fig. 7. Performance of COVQ, CASVQ, and VQ for the quantization of an uncorrelated Gaussian source; codebooks with $N_C = 32$ two-dimensional codevectors are used.

Qualitatively, the same implications hold for the second simulation, which again was carried out for the correlated Gaussian source signal, but with codebooks containing 128 three-dimensional codevectors. The results are depicted in Fig. 6.

The performances for an uncorrelated Gaussian source signal quantized with 32 two-dimensional codevectors are shown in Fig. 7. As above, CASVQ with the conventional VQ distance measure performs close to COVQ.

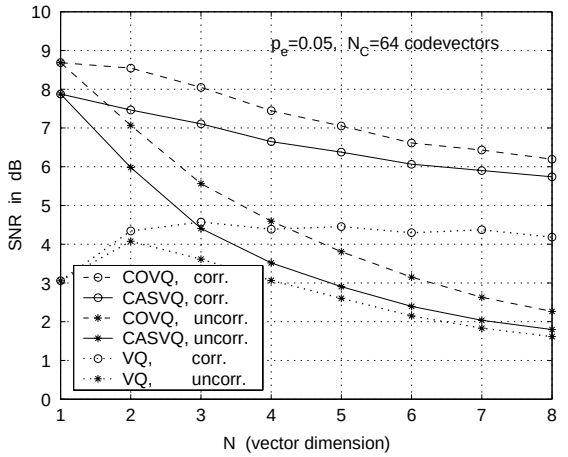

Fig. 8. Performance of COVQ, CASVQ (with VQ distance measure), and conventional VQ for the quantization of strongly correlated and uncorrelated Gaussian sources for various codevector dimensions N; codebooks with $N_C = 64$ codevectors are used; the bit-error probability is $p_e = 0.05$.

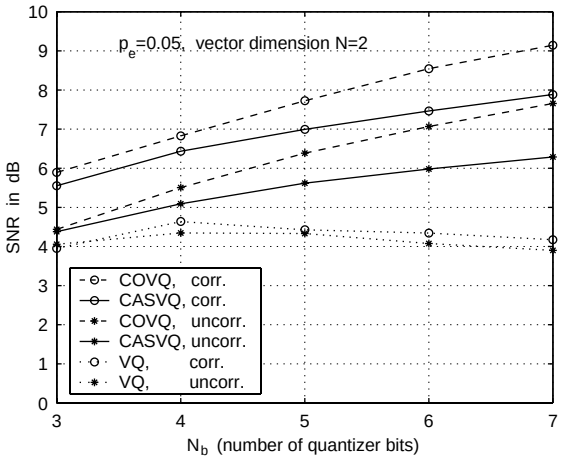

Fig. 9. Performance of COVQ, CASVQ (with VQ distance measure), and conventional VQ for the quantization of strongly correlated and uncorrelated Gaussian sources for various codebook sizes $N_C = 2^{N_b}$; codevectors with dimension $N = 2$ are used; the bit-error probability is $p_e = 0.05$.

Fig. 8 shows the performance dependencies of COVQ and CASVQ on the codevector dimension N for a fixed number $N_C = 64$ of codevectors and a bit-error probability of $p_e = 0.05$. Since N is reciprocally proportional to the source coding rate it is clear that the SNR is essentially decreasing for increasing N. However, the largest gain of COVQ and CASVQ over conventional VQ is obtained for small vector dimensions. For strongly correlated ($\rho = 0.9$) source signals (where the use of vector quantization really pays off) the gain of CASVQ over VQ remains significant also for moderate vector dimensions. For uncorrelated source signals and high vector dimensions there is no significant gain by CASVQ over VQ, but in this case even COVQ works only moderately better than conventional VQ.

In Fig. 9, in contrast to the previous simulation, the vector dimension is fixed ($N = 2$) but now the number N_b of quantizer index bits (and thus the codebook size $N_C = 2^{N_b}$) is variable, again for a bit-error probability of $p_e = 0.05$. For both strongly correlated ($\rho = 0.9$) and uncorrelated sources the gains of COVQ and CASVQ over VQ increase with the codebook size.

In summary, CASVQ works significantly better than conventional VQ, especially if the source signal is correlated and the bit-rate is high or, equivalently, when the vector dimension is small and the number of codevectors is large. The loss of CASVQ compared to COVQ gets larger with an increasing number of codevectors, but at the same time the gain of CASVQ over conventional VQ increases.

5. Conclusions

As a new result we have proposed channel-adaptive scaled vector quantization (CASVQ) as a substitute for channel-optimized vector quantization. The advantage of CASVQ is that on time-varying channels the memory- and complexity requirements are practically the same as for conventional VQ, but, especially for correlated source signals, the performance is close to that of optimally channel-matched COVQ, which would require to store several codebooks for the adaptation to time-varying channels.

The CASVQ codebook is generated by scaling all the codevectors in a reference codebook with a channel-dependent factor, which approximates the "shape" of the COVQ codebook for the current bit-error probability. If, additionally, the conventional VQ distance measure is used for encoding, the scaling may be moved from the codevectors to the input source signal, yielding a further complexity reduction as the normal VQ encoding algorithm can be used

afterwards. The overhead induced by CASVQ with the VQ distance measure (compared with conventional VQ) is negligible, because CASVQ additionally requires just the scaling of the input source vector, rescaling after the transmission, and a small table to store appropriate scaling factors for the bit-error probabilities. Thus, CASVQ is very attractive for the transmission of multimedia signals over time-variant channels, where limitations for complexity and memory exist, which do not allow the application of optimally channel-matched COVQ.

References

[1] F. Alajaji, N. Phamdo, Soft-decision COVQ for Rayleigh-fading channels, IEEE Commun. Lett. 2 (6) (June 1998) 162–164.

[2] G. Ben-David, D. Malah, Simple adaptation of vector-quantizers to combat channel errors, in: Proceedings of the 6th DSP Workshop, Yosemite, California, USA, October 1994, pp. 41–44.

[3] N.-T. Cheng, N. Kingsbury, Robust zero-redundancy vector quantization for noisy channels, in: Proceedings of the IEEE International Conference on Communications (ICC), 1989, pp. 1338–1342.

[4] N. Farvardin, A study of vector quantization for noisy channels, IEEE Trans. Inform. Theory 36 (4) (July 1990) 799–809.

[5] N. Farvardin, V. Vaishampayan, On the performance and complexity of channel-optimized vector quantizers, IEEE Trans. Inform. Theory 37 (1) (January 1991) 155–160.

[6] S. Gadkari, K. Rose, Robust vector quantizer design by noisy channel relaxation, IEEE Trans. Commun. 47 (8) (August 1999) 1113–1116.

[7] H. Jafarkhani, N. Farvardin, Channel-matched hierarchical table-lookup vector quantization for transmission of video over wireless channels, in: Proceedings of the IEEE International Conference on Image Processing (ICIP), September 1996, pp. 755–758.

[8] H. Jafarkhani, N. Farvardin, Design of channel-optimized vector quantizers in the presence of channel mismatch, IEEE Trans. Commun. 48 (1) (January 2000) 118–124.

[9] Y. Linde, A. Buzo, R.M. Gray, An algorithm for vector quantizer design, IEEE Trans. Commun. COM-28 (1) (January 1980) 84–95.

[10] D. Miller, K. Rose, Combined source-channel vector quantization using deterministic annealing, IEEE Trans. Commun. 42 (2/3/4) (February–April 1994) 347–356.

[11] M. Skoglund, Soft decoding for vector quantization over noisy channels with memory, IEEE Trans. Inform. Theory 45 (4) (May 1999) 1293–1307.