Soft-Decoding Vector Quantizer Using Reliability Information from Turbo-Codes

Keang-Po Ho, *Member, IEEE*

***Abstract*—The optimum soft-decoding vector quantizer using the reliability information from Turbo-codes is derived for combined source-channel coding. The encoder and decoder of the quantizer are optimized iteratively. For a four-dimensional vector quantizer having a rate of 1 bit/sample transmitted through a noisy channel, the soft-decoding channel-optimized quantizer can achieve about 3–3.7 dB performance improvement over conventional source-optimized quantizer.**

***Index Terms*—Combined source-channel coding, Turbo-codes, vector quantization.**

I. Introduction

VECTOR quantization is an important method for data compression and source encoding. It is for great importance that vector quantizers are robust against channel noise. When Turbo-codes are decoded, reliability information is provided as the likelihood of the transmitted binary data [1]–[3]. Taking advantage of the reliability information (or, in general, any decoder with soft output), the performance of vector quantizer transmitted through noisy channel can be greatly improved. Combined with the Turbo-code structure, a minimum mean-squared error (MMSE) estimator can be used to provide the best estimation of the original transmitted source. Instead of using the hard-decoded binary data, the MMSE estimator uses the log–likelihood ratio (LLR) from Turbo-codes directly. Similar to channel-optimized vector quantizers (COVQ) [4]–[8] for binary symmetry channel, the encoder and decoder of the soft-decoding COVQ for Turbo-codes can be iteratively optimized using the generalized Lloyd algorithm [5]–[9].

The remaining parts of this paper will derive the MMSE estimator using reliability information, describe the COVQ for Turbo-codes, and provide numerical results for Gauss–Markov source.

II. Vector Quantizer for Turbo-Codes

Fig. 1 shows a schematic diagram of a communication system to transmit vector quantized discrete-time analog signals. In the system, the encoder inputs a K-dimensional analog vector $\boldsymbol{x}$, quantizes it using a quantizer, and then maps the result to a binary codeword selected from $\{b_0, \cdots, b_{Q-1}\}$ with a length of B, where Q is the number of quantization levels. Usually, $Q = 2^B$ for low-noise channel but $Q < 2^B$ for very noisy channel [8]. The combined quantizer and mapping is a function, $b_i = \gamma(x)$, $x \in \Omega_i$ in which Ω_i, $i = 0, \cdots, Q-1$ are the partitions of the input space.

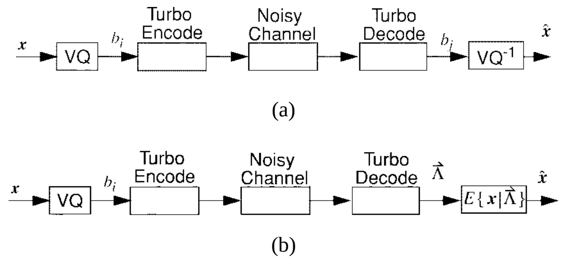

Fig. 1. Schematic diagram of a system for vector quantizer using Turbo-codes with (a) conventional hard-decision mapping based decoder and (b) MMSE estimator based soft-decision decoder.

The binary codewords are then concatenated, encoded by the Turbo-encoder, and sent through the noisy channel. The system in Fig. 1(a) uses conventional mapping based hard-decision decoder for a source-optimized vector quantizer (SOVQ) [9] designed for a noiseless channel. The Turbo-decoder outputs the binary codeword, which is subsequently mapping to a codeword of the vector quantizer. Using the LLR from the Turbo-decoder, the system in Fig. 1(b) uses an MMSE estimator to provide an estimation of the original transmitted analog signal to minimize the mean-squared distortion.

The Turbo-decoder provides the LLR Λ of each bit, according to [1], [2], [10]

$$\Lambda = \ln\left[\frac{\Pr(u_k = 1|\text{channel output})}{\Pr(u_k = -1|\text{channel output})}\right] \tag{1}$$

where u_k is the transmitted bit. Conventionally, the decoder of Fig. 1(a) makes a hard decision on the LLR to detect the corresponding bit which is the sign of the LLR, i.e., 1 if the LLR larger than 0 and -1 otherwise. The LLR can be used to calculate the bit *a posteriori* probability according to [10]

$$P(u_k = \pm 1|\Lambda) = \frac{\exp(\pm\Lambda)}{1 + \exp(\pm\Lambda)}. \tag{2}$$

The LLR *a posteriori* probability is $P(\Lambda|u_k) = P(u_k|\lambda)p(\Lambda)/P(u_k)$ from the Bayes rule. Usually, $p(u_k = \pm 1) = 1/2$ and $p(\Lambda)$ are the same for both $u_k = \pm 1$. Ignoring all common factors, we may write

Manuscript received June 24, 1998. The associate editor coordinating the review of this letter and approving it for publication was Prof. E. Biglieri. This work was supported by the Research Grant Council of HKSAR, Earmarked Grant CUHK 4370/99E.

The author is with the Department of Information Engineering, The Chinese University of Hong Kong, Shatin, NT, Hong Kong (e-mail: kpho@ie.cuhk.edu.hk).

Publisher Item Identifier S 1089-7798(99)06558-8.

$P(\Lambda|u_k) = P(u_k|\Lambda)$ for simplicity. The LLR *a posteriori* probabilities of the bits within a binary codeword can be used to calculate a set of LLR *a posteriori* probabilities $\{P_0, \cdots, P_{Q-1}\}$, where $P_i = P(\overrightarrow{\Lambda}|b_i)$, $i = 0, \cdots, Q-1$, is the LLR *a posteriori* probability given the binary codewords b_i, and $\overrightarrow{\Lambda} = (\Lambda_1, \cdots, \Lambda_B)$ is the LLR for the binary codeword. Using those *a posteriori* probabilities, the optimal estimation for the original input signal is the MMSE estimator $\hat{x}(\overrightarrow{\Lambda}) = E\{x|\overrightarrow{\Lambda}\}$ [4]–[8]

$$\hat{x}(\overrightarrow{\Lambda}) = \sum_{i=0}^{Q-1} E\{\boldsymbol{x}|b_i\}p(b_i|\overrightarrow{\Lambda}) = \sum_{i=0}^{Q-1} \boldsymbol{c}_i p(b_i|\overrightarrow{\Lambda}) \tag{3}$$

where $\boldsymbol{c}_i = E\{\boldsymbol{x}|b_i\}$, $i = 0, \cdots, Q-1$, are the codeword or centroid of each vector quantization partition. From the Bayes rule, $p(b_i|\overrightarrow{\Lambda}) = p(\overrightarrow{\Lambda}|b_i)p_{b_i}/p(\overrightarrow{\Lambda})$, the soft-decision decoder is

$$\hat{x}(\overrightarrow{\Lambda}) = \frac{\sum_{i=0}^{Q-1} P_i p_{b_i} \boldsymbol{c}_i}{\sum_{i=0}^{Q-1} P_i p_{b_i}} \tag{4}$$

where P_{b_i} is the *a priori* probability of the binary codewords b_i. In Bakus and Khandani [11], the output vector is estimated by, using our symbol, $\hat{x}(\overrightarrow{\Lambda}) = \sum_{i=0}^{Q-1} P_i \boldsymbol{c}_i$ for a scalar quantizer with the assumption that all binary codewords have an identical *a priori* probability. In general, the system distortion using the approximation of [11] is very close to the optimal decoder (4). However, in some special cases with a huge difference in *a priori* probabilities, the approximation of [11] is much worse than the optimum decoder (4).

The overall distortion from the input of source encoder to the output of the source decoder is

$$D = \frac{1}{K} \sum_{i=0}^{Q-1} \int_{\Omega_i} E\{p(\overrightarrow{\Lambda}|b_i)\|\boldsymbol{x} - \hat{\boldsymbol{x}}(\overrightarrow{\Lambda})\|^2\} p(\boldsymbol{x})\, d\boldsymbol{x}. \tag{5}$$

Given the MMSE source decoder, the optimal partitions are

$$\begin{aligned} x \in \Omega_i \quad &\text{if } E\{p(\overrightarrow{\Lambda}|b_i)\|\boldsymbol{x} - \hat{\boldsymbol{x}}(\overrightarrow{\Lambda})\|^2\} \\ &< E\{p(\overrightarrow{\Lambda}|b_k)\|\boldsymbol{x} - \hat{\boldsymbol{x}}(\overrightarrow{\Lambda})\|^2\}, \quad \text{for all } k \neq i \end{aligned} \tag{6}$$

or

$$x \in \Omega_i, \quad \text{if } \Gamma_i - 2\boldsymbol{x} \cdot \boldsymbol{v}_i < \Gamma_k - 2\boldsymbol{x} \cdot \boldsymbol{v}_k, \quad \text{for all } k \neq i \tag{7}$$

where

$$\Gamma_i = E\{p(\overrightarrow{\Lambda}|b_i)\|\hat{\boldsymbol{x}}(\overrightarrow{\Lambda})\|^2\} \tag{8}$$

$$\boldsymbol{v}_i = E\{p(\overrightarrow{\Lambda}|b_i)\hat{\boldsymbol{x}}(\overrightarrow{\Lambda})\} \tag{9}$$

and the expectation $E\{\cdot\}$ is with respect to all possible additive channel noise and input sequence. The relation of (7) is the generalized centroid condition for the optimal encoder.

A COVQ for Turbo-codes can be designed iteratively using the soft-decision decoder. Using a test sequence of input vectors $\boldsymbol{x}$ and transmitting the codewords using the schematic diagram of Fig. 1(b), the design algorithm can be functioned as the following.

1) Choose initial codewords $\boldsymbol{c}_i$, assign $\boldsymbol{v}_i = \boldsymbol{c}_i$ and $\Gamma_i = \|\boldsymbol{c}_i\|^2$.
2) Encode all vectors using (7) to find the partitions Ω_i.
3) Find the centroids of each partition.
4) Transmit binary codewords according to Fig. 1(b).
5) Decode the channel outputs using (4).
6) Update $\boldsymbol{v}_i$ and Γ_i using (8) and (9), also find the overall distortion using (5).
7) Repeat steps 2–6 until the convergence of the overall distortion.

In the above algorithm, if both the $\boldsymbol{v}_i$ and Γ_i of (8) and (9) are evaluated in step 6 without uncertainty, the overall distortion decreases in each iteration and leads to convergence. This algorithm is similar to the generalized Lloyd algorithm for vector quantizer design [8]. In practice, the expectation of (8) and (9) may have small uncertainty in the numerical calculation and the overall distortion does not decrease monotonically. Inferior to the optimal convergence condition, the convergence condition of this paper is to ensure the standard deviation of the last ten overall distortion values for less than 0.05 dB.

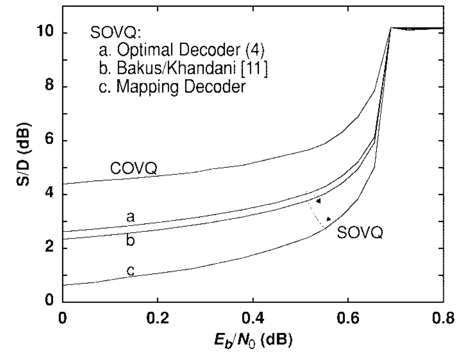

Fig. 2. The signal-to-distortion ratio (S/D) as a function of channel signal-to-noise ratio, E_b/N_0, for a four-dimensional vector quantizer decoded by COVQ for Turbo-codes, MMSE decoder (4) for SOVQ, Bakus/Khandani decoder [10] for SOVQ, and mapping based decoder for SOVQ [8].

III. Numerical Results

The LLR based soft-decoder and the COVQ for Turbo-codes is tested for vector quantization. The analog source is a first-order Gauss–Markov source having a correlation coefficient of 0.9. Four-dimensional vector quantizer is used with a rate of 1 bit/sample. The initial codewords are SOVQ designed using the LBG algorithm [9]. The 16-state rate-1/2 Turbo-codes [1], [2] are used with a block length of 256×256 and generator of $G_1 = 37$ and $G_2 = 21$. To limit the computational complexity, the number of decoding iterations is ten. The error probability as a function of the channel signal-to-noise ratio (S/N) in our simulation is about the same as that in [1], [2].

Fig. 2 shows the signal-to-distortion ratio (S/D) as a function of the channel S/N per bit, E_b/N_0. The performance provided by the iteratively optimized COVQ, the MMSE decoder of (4) for SOVQ, the Bakus/Khandani decoder [11] for SOVQ, and hard-decision mapping based decoder for

SOVQ is shown for comparison. The COVQ provides the smallest distortion and the largest S/D. The improvement of COVQ over conventional hard-decision mapping based decoder increases with the decrease of channel S/N and approaches 3–3.7 dB for channel S/N lower than 0.6 dB.

The MMSE estimator (4) can be applied to the SOVQ. The improvement over conventional mapping based decoder also increases with the decrease of channel S/N and approaches about 1.4–2 dB for channel S/N smaller than 0.6 dB. Applied for SOVQ, the MMSE estimator performs about 0.20–0.28 dB better than the suboptimal Bakus–Khandani decoder [11] which is confirmed to be suboptimal.

IV. Conclusion

The optimal soft-decoding vector quantizer using the reliability information from a Turbo-decoder is derived for combined source-channel coding. The decoder is an MMSE estimator to provide the optimal estimation of the source for minimum distortion. The encoder and decoder of the COVQ for Turbo-codes are optimized iteratively by fixing one of them using the generalized Lloyd algorithm. For a four-dimensional vector quantizer with a rate of 1 bit/sample, the COVQ for Turbo-codes can achieve about 3–3.7 dB performance improvement SOVQ having mapping based decoder.

References

[1] C. Berrou, A. Glavieux, and P. Thitimajshima, "Near Shannon limit error-correcting coding and decoding: Turbo-code (1)," in *Proc. IEEE Int'l Conf. on Commun.*, Geneva, Switzerland, May 1993, pp. 1064–1070.

[2] C. Berrou and A. Glavieux, "Near optimum error correcting coding and decoding: Turbo-codes," *IEEE Trans. Commun.*, vol. 44, pp. 1261–1271, 1996.

[3] Special issues on "Concatenated coding techniques and iterative decoding: Sailing toward channel capacity," *IEEE J. Selected Areas in Commun.*, vol. 16, no. 2, Feb. 1998.

[4] A. Kurtenbach and P. Wintz, "Quantizing for noisy channels," *IEEE Trans. Commun. Technol.*, vol. COM-17, pp. 291–302, 1969.

[5] H. Kumazawa, M. Kasahara, and T. Namekawa, "A construction of vector quantizers for noisy channels," *Electron. & Eng. in Japan*, vol. 67-B, pp. 39–47, 1984.

[6] N. Farvardin and V. Vaishampayan, "Optimal quantizer design for noisy channels: An approach to combined source-channel coding," *IEEE Trans. Inform. Theory*, vol. IT-33, pp. 827–838, 1987.

[7] N. Farvardin, "A study of vector quantization for noisy channels," *IEEE Trans. Inform. Theory*, vol. 36, pp. 799–809, 1990.

[8] N. Farvardin and V. Vaishampayan, "On the performance and complexity of channel-optimized vector quantizers," *IEEE Trans. Inform. Theory*, vol. 37, pp. 155–160, 1991.

[9] Y. Linde, A. Buzo, and R. M. Gray, "An algorithm for vector quantizer design," *IEEE Trans. Commun.*, vol. COM-28, pp. 84–95, 1980.

[10] J. Hagenauer, E. Offer, and L. Papke, "Iterative decoding of binary block and conventional codes," *IEEE Trans. Info. Theory*, vol. 42, pp. 429–445, 1996.

[11] J. Bakus and A. K. Khandani, "Combined source-channel coding using turbo-codes," *Electron. Lett.*, vol. 33, pp. 1613–1614, 1997.