Quantizer Design for Turbo-code Channels[1]

J. Bakus and A. K. Khandani

Department of Electrical and Computer Engineering
University of Waterloo
Waterloo, Ontario, Canada, N2L 3G1

Technical Report UW-E&CE#99-04

July 6 1999

[1]A preliminary version of this work has been presented in [1,2]. A more detailed version of this work is reported in [3]. This work is financially supported by Communications and Technology Ontario (CITO).

Abstract

A new method of combined source-channel coding using Turbo-code is presented. The corresponding system transmits a discrete time, continuous amplitude signal over an Additive White Gaussian Noise (AWGN) channel where a Turbo-code is used for error correction purposes. The proposed scheme takes advantage of the available reliability information (generated by the Turbo-code decoder) to improve the quality of the reconstructed signal. The design rules for a quantizer and reconstructor to minimize the overall mean square distortion are derived. Numerical results are presented showing up to 1 dB improvement in the end-to-end distortion with respect to a traditional channel optimized quantizer. This improvement is obtained at the price of a small increase in the system complexity.

1 Introduction

This work studies the problem of reliable transmission of discrete time continuous amplitude signals over an Additive White Gaussian Noise (AWGN) channel. This is achieved by a scalar fixed rate quantizer used in conjunction with a new class of error correction code, called Turbo-code [1, 2]. An improvement in the quantizer performance is achieved by taking advantage of the individual *bit probabilities* that are generated by the Turbo-code decoder. These probabilities are available at no extra cost and the overall increase in complexity of the quantizer is minimal.

Turbo-code is a new class of binary block codes which achieve a performance close to the Shannon limit. The basic structure of a Turbo-code consists of two or more recursive systematic convolutional encoders (which are terminated at the end of each block) concatenated in parallel through a set of pseudo-random interleavers. The decoder consists of a number of elementary MAP decoders with a feedback from the last one to the first one. The decoding operation is performed in an iterative manner that has a linear complexity with respect to the block length. The decoder output is a reliability information expressed as the probability of each bit.

A fixed rate scalar quantizer with a soft reconstruction rule is presented which operates in conjunction with a Turbo-code. The system is designed using an iterative procedure similar to the Lloyd-Max algorithm [3, 4]. The corresponding design procedure iteratively optimizes the quantizer and the reconstructor until no further improvement can be made. This results in a local optimum solution which is not guaranteed to be globally optimal.

The original Lloyd-Max algorithm does not consider the effect of channel noise. Kurtenbach and Wintz were one of the first researchers who investigated the effect of the channel noise [5] in a quantization system. They studied the transmission of pulse coded modulation data over a noisy channel and discovered that the channel noise affects the design of the quantizer. Later, Rydbeck and Sundberg showed that the code word assignment for a quantizer operating over a noisy channel also plays an important role in the system performance [6].

Farvardin and Vaishampayan presented a design algorithm based on Lloyd-Max resulting in the so called *channel optimized scalar quantizer* (COSQ) [7]. The COSQ is designed to operate over a binary symmetric channel with a particular error probability. The channel optimized quantizer algorithm was further extended by the same researchers to encode vector sources [8, 9], and is known as the *channel optimized vector quantizer* (COVQ). One of the first attempts to design a quantizer with soft reconstruction decoding was made by Vaishampayan and Farvardin by extending the COVQ design algorithm to include the modulation signal set [10].

Phamdo and Alajaji also extended the COVQ algorithm to allow for a soft reconstruction [11, 12]. In their work, the soft reconstruction is achieved by quantizing the output of the channel demodulator in conjunction with a channel optimized quantizer. This increases the number of possible reconstruction points, and results in a finer decoding than the classical COVQ.

Skoglund and Hedeling presented a vector soft decision decoding quantizer in which the binary code words of the quantizer are encoded using a Hadamard code [13]. This coding scheme gives the probabilities of the individual code words at the decoder, and the final output is reconstructed as a sum of the reconstruction levels multiplied by the corresponding probabilities. This idea was further adapted by the same researchers for image transmission [14], and multiuser decoding [15].

A similar technique is presented in this work with a scalar quantizer operating in conjunction with Turbo-code (some preliminary results of this work is reported in [16, 17]). The Turbo-code decoder provides bit reliability information expressed as a log likelihood ratio, and a quantizer is designed to take advantage of this information to improve the reconstruction of the samples. Numerical results show an improvement over channel optimized scalar quantizer operating in conjunction with Turbo-code.

Several other works have been devoted to using Turbo-code as part of a combined source channel coding system for image transmission [18, 19]. In these works, the image is source coded using the JPEG standard [20], and then channel coded using Turbo-code. An image model is used to estimate the probability of each pixel from adjacent pixels, and these probabilities are fed back to the Turbo-code decoder to improve the channel decoding.

The Turbo-code decoder is a soft decoding scheme that produces reliability information about each transmitted bit d_k given the channel output sequence R_1^N. This information is expressed as a Log Likelihood Ratio (LLR) Λ as:

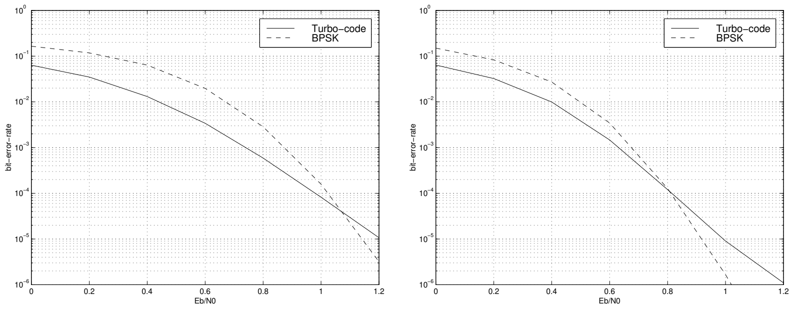
$$\Lambda = \ln \frac{Pr(bit\ 1\ sent \mid channel\ output)}{Pr(bit\ 0\ sent \mid channel\ output)}. \tag{1}$$

In traditional Turbo-code systems, hard decision rule is performed on Λ to determine the most likely transmitted bits. This work presents a method to improve the overall quantizer performance by using the reliability information in a soft decision reconstruction rule. This reliability information is available as part of the decoding process at no extra cost and the extra complexity added to the reconstructor to use this information is also minimal.

Berrou *et al.* [1] made the observation that the LLR values follow a Gaussian distribution with variance σ_L^2 and mean $\{+M_l, -M_l\}$, where the mean depends on the value of the bit transmitted. This observation is used in the quantizer-reconstructor design algorithm. The Turbo-code encoder, noisy channel, and the decoder are grouped together to form an equivalent channel, which is modeled as an equivalent Binary Phase Shift Keying (BPSK) system with modulation points $\{+M_l, -M_l\}$ operating over an AWGN channel with noise variance σ_l^2.

The values of M_l and σ_l^2 depend on the channel noise, as well as the structure of the Turbo-code and are measured by passing a test bit stream through the Turbo-code scheme and observing the LLR values at the output. Figure 1 shows the bit-error-rate for the actual Turbo-code (obtained through simulation) compared with the bit-error-rate of the equivalent BPSK system (computed using the standard methods based on error function), where the Turbo-code used has $M = 2$ memory elements and block lengths of 2048 and 10 000 bits. It is observed that the difference between the two curves is generally small, with some differences which we conjecture to be due to the following reasons.

The validity of the Gaussian model for the distribution of LLR values can be explained by



(a) N=2048 (b) N=10000

Figure 1: Bit-error-rate for the equivalent BPSK system and the actual rate 1/3 Turbo-code system with $M = 2$ memory elements and block length of 10 000.

noting that in the water fall region the significant error events have a large weight, and consequently, the LLR value of each decoded bit is composed of the sum of many small components (reflecting the indepenent contributions from different bit positions involved in a large weight error event). Due to the central limit, the corresponding distribution tends to become Gaussian. On the other hand, in the model presented, one averages the argument of the error function for different bit positions over the entire block, and uses the result to calculate an estimate of the overall bit-error probability. We note that in general different bit positions in a Turbo-code block have different error probabilities (due to non-homogeneous structure of the interleaver). In this case, as the error-function is convex, the averaging of its argument results in an overestimate of the true bit-error probability in the water-fall region as noticed in Figure 1. This effect is less pronounced as the block length increases. We note that the relative positioning of the two curves changes as we move towards the error floor region. This can be explained by noting that in the error floor region, the significant error events are of a small weight, and consequently, the assumption of Gaussian distribution is not valid any more.

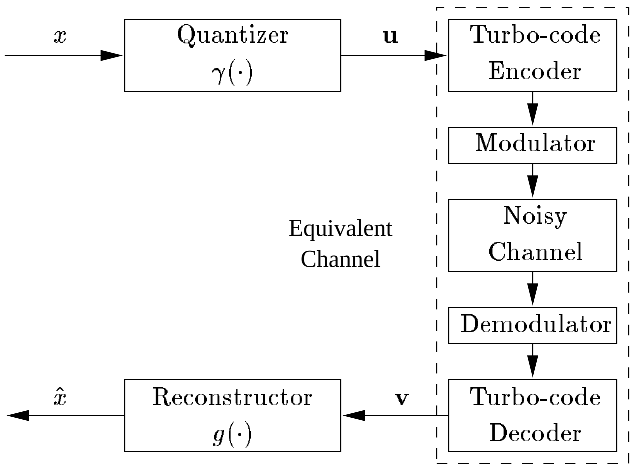


Figure 2: System block diagram.

2 System Overview

The block diagram of the system is shown in Figure 2. The Turbo-code decoder produces a soft output reliability information about each transmitted bit which is used by the reconstructor to improve the quality of the reconstruction. The quantizer and the reconstructor are jointly optimized to give the minimum end-to-end distortion.

The scalar quantizer $\gamma(\cdot)$ maps the input sample x to a partition S which is represented by a code word $\mathbf{u}$. There are N disjoint and exhaustive partitions available which are grouped together in a partition set $\{S_1, \ldots, S_N\}$. Each partition S_i has a corresponding binary code word $\mathbf{u}_i$ in a code word set $\{\mathbf{u}_1, \ldots, \mathbf{u}_N\}$. Each code word is a vector composed of r bits, where $2^r \leq N$. The code word $\mathbf{u}$ representing the sample x is passed to the Turbo-code encoder for channel encoding and transmission.

At the receiver side, the first step in the reconstruction of the source sample is a Turbo-code decoder. For each transmitted code word $\mathbf{u}$, the Turbo-code decoder produces a reliability information vector $\mathbf{v}$ composed of r log likelihood ratio values. The reconstruction rule maps the reliability information vector to an output sample $\hat{x}$. To design the quantizer and the reconstructor, the relationship between the transmitted code word $\mathbf{u}$ and the received reliability vector $\mathbf{v}$ is required. The Turbo-code encoder, noisy channel, and the decoder are grouped into an equivalent channel, as shown in Figure 2, and a model is used to approximate the relationship between $\mathbf{u}$ and $\mathbf{v}$.

The reconstruction function $g(\cdot)$ is defined in terms of the reconstruction level set $\{R_1, \ldots, R_N\}$. This function inputs a reliability vector $\mathbf{v}$ from the Turbo-code decoder, and calculates a vector of code word probabilities $\{P(\mathbf{u} = \mathbf{u}_1|\mathbf{v}), \ldots, P(\mathbf{u} = \mathbf{u}_N|\mathbf{v})\}$, where $P(\mathbf{u} = \mathbf{u}_i|\mathbf{v})$ is the probability that code word $\mathbf{u}_i$ was transmitted, conditioned on the received probability vector $\mathbf{v}$. The code word reliability vector is used to reconstruct the output sample $\hat{x}$ from the received reliability vector $\mathbf{v}$ using one of two possible reconstruction rules: *hard reconstruction rule*, and *soft reconstruction rule.*

The hard reconstruction rule selects the code word which is most likely to be transmitted and outputs the corresponding reconstruction level, i.e.,

$$\hat{x} = R_j : P(\mathbf{u} = \mathbf{u}_j|\mathbf{v}) > P(\mathbf{u} = \mathbf{u}_i|\mathbf{v}), \quad \forall\, j \neq i. \tag{2}$$

The soft reconstruction rule generates the reconstructed sample as a sum of the code word probabilities multiplied with their corresponding reconstruction levels, i.e.,

$$\hat{x} = \sum_{j=1}^{N} R_j P(\mathbf{u} = \mathbf{u}_j|\mathbf{v}). \tag{3}$$

The hard decision rule is essentially a threshold operation on the received vector $\mathbf{v}$, and systems with this reconstruction rule are well established in the literature [7, 3, 4, 5, 8, 9]. The soft decision rule is a relatively new idea, where some earlier relevant results are described in [10, 11, 13, 16].

3 Quantizer and Reconstructor Design

The quantizer-reconstructor pair is designed using an iterative algorithm similar to [7, 3, 4]. Consider a sample x which is transmitted to the reconstructor, as shown in Figure 2. The mean square error distortion, averaged over all possible values of $\mathbf{v}$, results in the sample distortion:

$$D_i(x) = \int_{\mathbf{v}} P(\mathbf{v}|\mathbf{u} = \mathbf{u}_i)\,[x - g(\mathbf{v})]^2\, d\mathbf{v}, \tag{4}$$

where $P(\mathbf{v}|\mathbf{u} = \mathbf{u}_i)$ is the probability of receiving a reliability vector $\mathbf{v}$ conditioned on transmitting the code word $\mathbf{u}_i$. Averaging the sample distortion $D_i(x)$ results in the overall average distortion for the quantizer, i.e.,

$$D = \sum_{i=1}^{N} \int_{S_i} P(x) \int_{\mathbf{v}} P(\mathbf{v}|\mathbf{u} = \mathbf{u}_i) \left[x - g(\mathbf{v})\right]^2 d\mathbf{v}dx \tag{5}$$

where $P(x)$ is the source probability distribution in partition S_i. The sample distortion in (5) can be written as,

$$D_i(x) = x^2 - 2xB_i + C_i. \tag{6}$$

resulting in,

$$D = \sum_{i=1}^{N} \int_{S_i} P(x) \left(x^2 - 2xB_i + C_i\right) dx, \tag{7}$$

where,

$$\begin{aligned} B_i &= \int_{\mathbf{v}} g(\mathbf{v})P(\mathbf{v}|\mathbf{u} = \mathbf{u}_i)d\mathbf{v}, \\ C_i &= \int_{\mathbf{v}} g(\mathbf{v})^2 P(\mathbf{v}|\mathbf{u} = \mathbf{u}_i)d\mathbf{v}. \end{aligned} \tag{8}$$

The quantities B_i and C_i are defined in terms of the equivalent channel transition probabilities, $P(\mathbf{v}|\mathbf{u} = \mathbf{u}_i)$, which will be later computed using a proper model for the equivalent channel.

The quantizer is designed by fixing the reconstruction levels, and finding the best set of partitions $\{S_1, \ldots, S_N\}$ corresponding to the selected set of code words $\{\mathbf{u}_1, \ldots, \mathbf{u}_N\}$. Partition S_i is defined as,

$$S_i = \{x : D_i(x) \leq D_j(x), \ \forall\ j \neq i\} \tag{9}$$

where $D_i(x)$ is the sample distortion given in (6).

The reconstruction function $g(\cdot)$ uses one of two reconstruction rules described by (2) and (3), both of which are defined in terms of the reconstruction level set $\{R_1, \ldots, R_N\}$. The design algorithm keeps the quantizer partition set and the binary code word set fixed, and finds the reconstruction level set $\{R_1, \ldots, R_N\}$ that minimizes the average distortion in (7) by setting its derivative with respect to each reconstruction level R_m to zero. This results in N equations in terms of the N reconstruction levels:

$$2\sum_{i=1}^{N}\int_{S_i} xP(x)dx\frac{dB_i}{dR_m} = \sum_{i=1}^{N}\int_{S_i} P(x)dx\frac{dC_i}{dR_m}, \quad m = 1, \ldots, N. \tag{10}$$

The design of the system iterates between the quantizer and reconstructor until the total average distortion given in (7) reaches to a local minimum. The algorithm is summarized as:

1. Initialize the reconstruction level set $\{R_1, \ldots, R_N\}$.
2. Using (8) calculate the expressions B_i and C_i.
3. Using (9) calculate the order of partitions, thresholds between partitions, and code word set.
4. Using (10) calculate the reconstruction level set $\{R_1, \ldots, R_N\}$.
5. Using (7) calculate the overall distortion. If the overall distortion has not reached to a local minimum, go to step 2.

When the closed form expression of the source sample distribution is not known, the training sequence version of the algorithm is used. In this case, for each training sample x, the partition distortion $D_i(x)$ which gives the lowest distortion is found. As each sample is quantized, statistics are collected for the partitions to estimate $P(x \in S_i)$, $E(x|x \in S_i)$ and $E(x^2|x \in S_i)$, namely the probability that a sample x is in S_i, the expectance of all samples and the expectance of the square of all samples in S_i, respectively. These values will be used in our subsequent calculations for the design or performance evaluation of the proposed scheme.

The reconstructor is designed using the expression in (10). The integral terms in the expression are evaluated with the collected statistics using following substitutions:

$$\begin{aligned} \int_{S_i} xP(x)dx &= P(x \in S_i)E(x|x \in S_i) \\ \int_{S_i} P(x)dx &= P(x \in S_i) \end{aligned} \tag{11}$$

3.1 Equivalent Channel Models

The iterative algorithm presented requires the channel dependent parameters B_i and C_i defined in (8). These quantities depend on the relationship between the codeword $\mathbf{u}$ produced by the quantizer and the reliability vector $\mathbf{v}$ received by the reconstructor as shown in Figure 2. This relationship is very complex and includes the effects of the channel noise and the Turbo-code encoder/decoder pair. Instead, the Turbo-code channel encoder/decoder pair and the noisy channel are replaced with an equivalent channel, which inputs a code word $\mathbf{u}$ and outputs a reliability vector $\mathbf{v}$.

Two classical models are discussed that result in the classical Lloyd-Max quantizer [3, 4] and the channel optimized quantizer [7], respectively. These two are compared with two alternative models for a soft decision quantizer. Both of the soft decision quantizers perform better than Lloyd-Max or channel optimized quantizer.

The first model is the *Noiseless Channel Model*, which ignores the effect of the channel noise, i.e., $\mathbf{u}_i = \hat{\mathbf{u}}_j$, resulting in,

$$\begin{aligned} B_i &= R_i \\ C_i &= R_i^2. \end{aligned} \tag{12}$$

This is equivalent to the classical Lloyd-Max design algorithm [3, 4], where the reconstruction level is given by:

$$R_m = \frac{\int_{S_m} xP(x)dx}{\int_{S_m} P(x)dx}. \tag{13}$$

The second model for a hard decision quantizer is the the *Binary Symmetric Channel Model.* It models the channel as a binary symmetric channel with a probability of error ϵ, and the relationship between the transmitted code word $\mathbf{u}_i$ and the received code word $\hat{\mathbf{u}}_j$ is given by the transition probability:

$$\begin{aligned} P(j|i) &= Pr(\hat{\mathbf{u}}_j|\mathbf{u}_i) \\ &= (\epsilon)^h(1-\epsilon)^{(r-h)} \end{aligned} \tag{14}$$

where h is the Hamming distance between code words $\mathbf{u}_i$ and $\hat{\mathbf{u}}_j$ and the r is the total number of bits per code word. With this model, the channel dependent quantities B_i and C_i in (8) become:

$$B_i = \sum_{j=1}^{N} P(j|i) R_j$$
$$C_i = \sum_{j=1}^{N} P(j|i) R_j^2. \tag{15}$$

This results in the channel optimized quantizer of [7], with the reconstruction levels given by,

$$R_m = \frac{\sum_{i=1}^{N} P(m|i) \int_{S_i} x P(x) dx}{\sum_{i=1}^{N} P(m|i) \int_{S_i} P(x) dx} \tag{16}$$

Systems operating with soft decision reconstruction rule use the reliability vector $\mathbf{v}$ to calculate the probability $P(\mathbf{u} = \mathbf{u}_j|\mathbf{v})$ of the transmitted code word. The resulting vector of code word probabilities is used in a weighted average reconstruction rule (3) to generate the output sample $\hat{x}$. To calculate the probability $P(\mathbf{u} = \mathbf{u}_j|\mathbf{v})$, consider the l^{th} bit u of the code word $\mathbf{u}$. This bit is transmitted over the equivalent channel, resulting in a single LLR value v in the l^{th} position in the reliability vector $\mathbf{v}$. Using the LLR definition (1) and given that $P(u = 0|v) + P(u = 1|v) = 1$, the probability of the transmitted bit conditioned on the received LLR value v is:

$$P(u = 0|v) = \frac{1}{1 + \exp(v)}$$
$$P(u = 1|v) = \frac{\exp(v)}{1 + \exp(v)}. \tag{17}$$

All source bits in the code word $\mathbf{u}$ are considered independent, and therefore the probability $P(\mathbf{u} = \mathbf{u}_i|\mathbf{v})$ for an arbitrary code word is calculated by multiplying the individual bit probabilities given in (17).

Consider the channel dependent parameters B_i and C_i defined in (8). Substituting these in (3) results in:

$$B_i = \sum_{m=1}^{N} R_j \Theta_i(k)$$
$$C_i = \sum_{j=1}^{N} \sum_{k=1}^{N} R_j R_k \Psi_i(j, k) \tag{18}$$

where

$$\begin{aligned} \Theta_i(k) &= \int_{\mathbf{v}} P(\mathbf{v}|\mathbf{u}=\mathbf{u}_i)P(\mathbf{u}=\mathbf{u}_k|\mathbf{v})d\mathbf{v} \\ \Psi_i(j,k) &= \int_{\mathbf{v}} P(\mathbf{v}|\mathbf{u}=\mathbf{u}_i)P(\mathbf{u}=\mathbf{u}_j|\mathbf{v})P(\mathbf{u}=\mathbf{u}_k|\mathbf{v})d\mathbf{v}. \end{aligned} \tag{19}$$

The two sets of parameters, $\Theta_i(k)$ and $\Psi_i(j,k)$, capture the characteristics of the channel with soft reconstruction rule. The indices i, j and k denote the individual code words, and the two parameters are computed for all such possible code word combinations. This results in N^2 different values for $\Theta_i(k)$ and N^3 different values for $\Psi_i(j,k)$.

The quantities B_i and C_i are used directly in the quantizer design algorithm (9). The reconstructor is designed by substituting B_i and C_i in (10) resulting in a set of linear equations:

$$\sum_{i=1}^{N} \Theta_i(k) \int_{S_i} xP(x)dx = \sum_{j=1}^{N} R_j \sum_{i=1}^{N} \Psi_i(j,k) \int_{S_i} P(x)dx \tag{20}$$

where $k = 1, \ldots, N$. This set of linear equations can be expressed in matrix form as:

$$\begin{bmatrix} E(1) \\ \vdots \\ E(N) \end{bmatrix} = \begin{bmatrix} F(1,1) & \ldots & F(1,N) \\ \vdots & \ddots & \vdots \\ F(N,1) & \ldots & F(N,N) \end{bmatrix} \begin{bmatrix} R_1 \\ \vdots \\ R_N \end{bmatrix} \tag{21}$$

where

$$\begin{aligned} E(k) &= \sum_{i=1}^{N} \Theta_i(k) \int_{S_i} xP(x)dx \\ F(j,k) &= \sum_{i=1}^{N} \Psi_i(j,k) \int_{S_i} P(x)dx. \end{aligned} \tag{22}$$

The solution $\{R_1, \ldots, R_N\}$ to this set of equations results in the optimal reconstruction levels.

Two different models are presented to estimate the parameters $\Theta_i(k)$ and $\Psi_i(j,k)$: *Gaussian Channel Model* and *Reliability Information Channel Model*. Both of these develop the parameters as one bit versions $\theta_i(k)$ and $\psi_i(j,k)$, and extend the results to composite code words by multiplying the corresponding bit values together. This is based on the assumption that the individual bits of the code words are independent of each other which is a common assumption in Turbo-codes. Given the transmitted code word as a single bit u, and the received reliability vector as a single LLR value v, the one bit parameters are given as:

$$\begin{aligned} \theta_i(k) &= \int_v P(v|u=i)P(u=k|v)dv \\ \psi_i(j,k) &= \int_v P(v|u=i)P(u=j|v)P(u=k|v)dv. \end{aligned} \tag{23}$$

The indices i, j and k denote the individual bit values, which can take the values 0 or 1. The two quantities are evaluated for all bit combinations, resulting in 4 different values for $\theta_i(k)$ and 8 different values for $\psi_i(j,k)$. The individual bits of the code word are assumed to be independent, and the parameters for all the individual bits of the code word are multiplied together to calculate parameters for an arbitrary code word.

The first model presented is the *Gaussian Channel Model*, which takes advantage of the fact that the probability distribution of the LLR values is very close to a Gaussian distribution [1]. The channel is modeled as a BPSK system with modulation signal points at $\{-M_l, +M_l\}$ and a noise variance of σ_l^2. Both of the parameters, M_l and σ_l^2, are measured by passing a test bit stream through the channel, and observing the mean and the variance of the received LLR values.

In a one bit system the probability distribution of the received LLR value v conditioned on the transmitted bit u is given by:

$$\begin{aligned} Pr(v \mid u=0) &= \frac{1}{\sqrt{2\pi\sigma_l^2}} \exp\left[-\frac{(v+M_l)^2}{2\sigma_l^2}\right], \\ Pr(v \mid u=1) &= \frac{1}{\sqrt{2\pi\sigma_l^2}} \exp\left[-\frac{(v-M_l)^2}{2\sigma_l^2}\right]. \end{aligned} \tag{24}$$

The relationships (17), (24) are substituted into the definition of $\theta_i(k)$ and $\psi_i(j,k)$ in (23).

The second model is called the *Reliability Information Channel Model*, which uses the definition of the LLR values in (1) to calculate the two channel parameters. For this purpose, we define a random variable X_k as the probability that the bit u takes the value $k=0,1$, conditioned on the received LLR value v:

$$X_k = P(u=k|v), \quad k=0,1. \tag{25}$$

The equations in (17) establish a one-to-one relationship between the random variable v, namely the LLR value, and the random variable X_k. The value of $k = 0, 1$ determines which of the two equations are used for this relationship. With this one-to-one mapping, the probability distribution of X_k is expressed in terms of the probability distribution of v by the transformation:

$$P(X_k)dX_k = P(v)dv \tag{26}$$

Conditioning both sides of the equation on $u = i$ results in:

$$P(X_k|u = i)dX_k = P(v|u = i)dv \tag{27}$$

Substituting the equations (25) and (27) in the definition of one bit parameter $\theta_i(k)$ in (23) results in:

$$\begin{aligned}\theta_i(k) &= \int_v P(v|u = i)P(u = k|v)dv \\ &= \int_0^1 X_k P(X_k|u = i)dX_k \\ &= E(X_k|u = i)\end{aligned} \tag{28}$$

The quantity $E(X_k|u = i)$ can be measured by passing a test bit stream through the channel and observing the channel output.

The quantity $\psi_i(j, k)$ is calculated in a similar way to $\theta_i(k)$. For this purpose, we define a random variable $Y_{j,k}$ as,

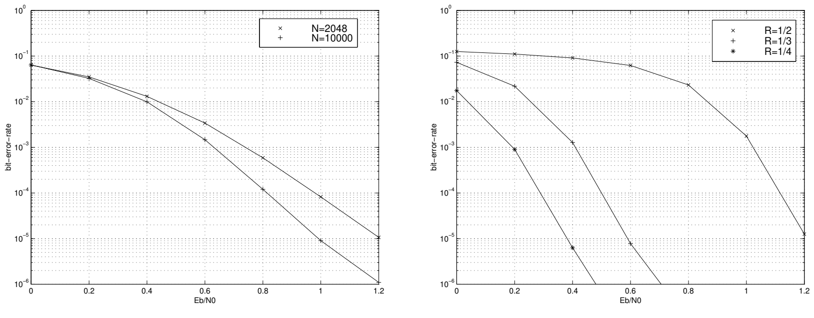
$$Y_{j,k} = P(u = j|v)P(u = k|v), \quad j, k = 0, 1. \tag{29}$$

Again, using (17), the random variables $P(u = \{0, 1\}|v)$ are uniquely related to the random variable v, resulting in,

$$P(Y_{j,k})dY_{j,k} = P(v)dv \tag{30}$$

Conditioning both sides on $u = i$ results in:

$$P(Y_{j,k}|u = i)dY_{j,k} = P(v|u = i)dv. \tag{31}$$



(a) $M = 2$, Rate $1/3$ (b) $M = 3$, $N = 10\,000$

Figure 3: Bit-error-rate curves for Turbo-code channel coding.

Substituting (29) and (31) in the definition of the single bit parameter $\psi_i(j,k)$ in (23) results in:

$$\begin{aligned} \psi_i(j,k) &= \int_v P(v|u=i)P(u=j|v)P(u=k|v)dv \\ &= \int_0^1 Y_{j,k} P(v|u=i) dY_{j,k} \\ &= E(Y_{j,k} \mid u=i) \end{aligned} \tag{32}$$

The quantity $E(Y_{j,k}|u=i)$ can also be measured by passing a test bit stream through the channel and observing the output.

4 Numerical Results

4.1 Simulation Setup

The presented quantization system has been simulated using samples of a Gaussian distribution with zero mean and unit variance. Four different quantizers were designed using the channel models discussed. Since the source sample distribution is known, the quantizers are designed by a direct substitution in the design formulation without the use of a training sequence.

The main simulation is performed using a rate 1/3 Turbo-code, composed of two *RSC* encoders with $M = 2$ memory elements, and the generator polynomials: $backward = (111)$ and

$forward = (101)$. To investigate the effect of the channel code block length, two Turbo-code systems are tested with block length of 2048 bits and 10 000 bits. The bit-error-rate curves for these two codes are shown in Figure 3(a).

To show the trade-off between the channel coding and the source coding, and to investigate the spectral efficiency of the system, some simulations are also performed with rates 1/2, 1/3, and 1/4 Turbo-code. The Turbo-code used for these simulations has $M = 3$ memory elements, and block length of 10 000 bits. The bit-error-rate curves for these codes using a block length of 10 000 are shown in Figure 3(b). To generate the different rates, the basic rate 1/3 Turbo-code has been modified to form a larger encoder by replacing each of the RSC encoders with a pair of RSC encoders. The two encoders in each pair have the same backward polynomial (1101), and each of the encoders has a unique forward polynomial: (1011) and (1111). Different coding rates are achieved by puncturing the encoder outputs.

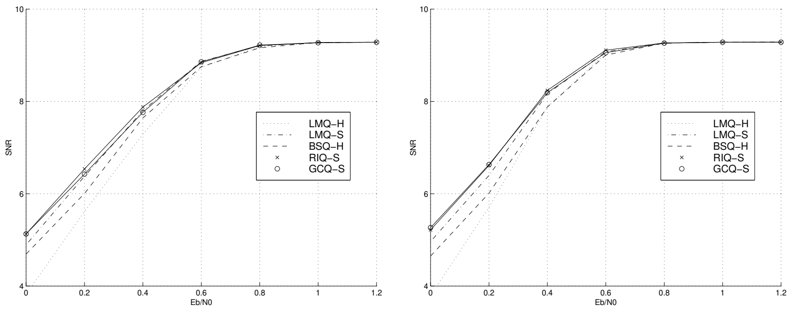
All Turbo-codes presented use an S-random interleaver. The interleaver for codes with block length of 2048 has $S = 15$, and the interleaver for codes with block length of 10 000 has $S = 20$. All the systems are simulated over an AWGN channel with channel signal-to-noise ratio E_b/N_0 in the range of 0.0 to 1.2 dB, and results are obtained for systems with 2, 3, 4, and 5 bit quantizers.

Total of five different quantizers are designed and tested, as listed in Table 1. The first quantizer is the classical Lloyd-Max quantizer, which is designed with the noiseless channel model and a hard decision rule. To investigate the effect of the soft decision, this quantizer is simulated with both hard and soft reconstruction rules, and the two systems are designated as LMQ-H and LMQ-S, respectively. The next quantizer is designed using the binary symmetric channel model and hard decision rule. This is designated as BSQ-H and is identical to the channel optimized quantizer in [7].

The above three quantizers are compared to the two soft decision quantizers, which are designed to operate in conjunction with Turbo-code. The first soft decision algorithm uses the definition of the reliability information to model the channel noise and is referred to as RIQ-S. The second soft decision algorithm approximates the Turbo-code and the noisy channel as a BPSK channel with additive white Gaussian noise, and is referred to as GCQ-S.

Name	Channel Model	Reconstruction
Name	Channel Model	Rule
LMQ-H	Noiseless Channel	Hard
LMQ-S	Noiseless Channel	Soft
BSQ-H	Binary Symmetric Channel	Hard
RIQ-S	Reliability Information Channel	Soft
GCQ-S	Gaussian BPSK Channel	Soft

Table 1: Designed quantization systems



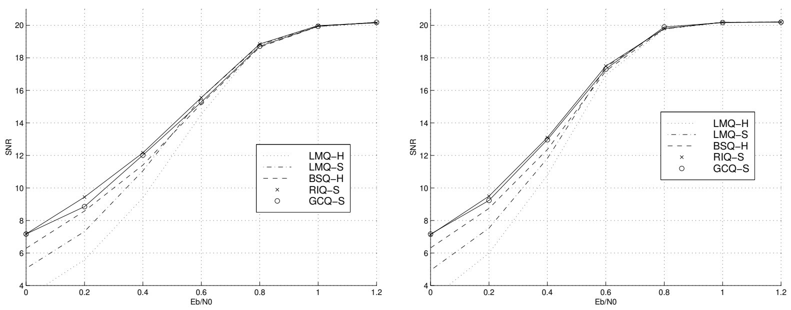
(a) $N = 2048$ (b) $N = 10\,000$

Figure 4: Distortion for 2 bit quantizers with rate 1/3 Turbo-code with $M = 2$ memory elements.

4.2 Quantizer Results

The results for the 2 bit quantizer the 4 bit quantizer, both operating with a rate 1/3 Turbo-code are shown in Figures 4 and 5, respectively. The classical Lloyd-Max quantizer operating with hard reconstruction rule (LMQ-H) results in the worst performance of all quantizers. The channel optimized quantizer (BSQ-H) also uses a hard reconstruction rule, however, the design takes the effect of the channel noise into account. The inclusion of the noise effect resulted in a substantial improvement over the classical Lloyd-Max. The Lloyd-Max quantizer is also tested with soft decision reconstruction (LMQ-S). The soft decision improves the performance over the classical Lloyd-Max, resulting in a performance similar to the channel optimized quantizer.

The two quantizers that are designed specifically for Turbo-code channel coding and soft decision rule offers the best performance. Relative to each other the RIQ-S performs slightly better than GCQ-S at shorter block length, however, at longer block length the two show very



(a) $N = 2048$ (b) $N = 10\,000$

Figure 5: Distortion for 4 bit quantizers with rate 1/3 Turbo-code with $M = 2$ memory elements.

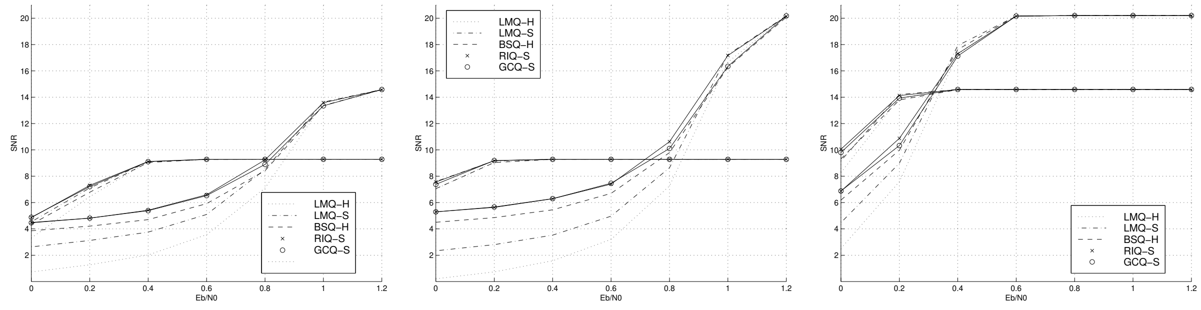
close performance. This means that as the block length increases, the BPSK channel model used to design the GCQ-S becomes more accurate.

4.3 Spectral Efficiency Comparison

To study the effect of the spectral efficiency, both the Turbo-code rate and the quantization rate has been varied to achieve systems that operate with the same sample rate. The Turbo-code used for this comparison is the encoder with $M = 3$ described earlier, with a block length of 10 000 bits.

Figure 6(a) shows the 2 bit quantizers with a rate 1/3 Turbo-code, compared to the 3 bit quantizers used with rate 1/2 Turbo-code. Both of these systems result in a total of 6 channel bits per source sample. At low E_b/N_0, the channel noise has a significant effect on the quantizer performance, and the quantizers that allocate more bits toward the channel coding have better performance. At high E_b/N_0, the channel noise has little impact on the quantizer performance, and the systems that allocate more bits toward the source coding have better performance.

Similar results are shown in Figure 6(b), where Turbo-code rate 1/2 with 4 bit quantizer is compared with Turbo-code rate 1/4 with 2 bit quantizer. Figure 6(c) shows another comparison of Turbo-code rate 1/3 with 4 bit quantizer compared with Turbo-code rate 1/4 with 3 bit quantizer.



(a) Rate 1/2, 3 bit and rate 1/3, 2 bit quantizer

(b) Rate 1/2, 4 bit and rate 1/4, 2 bit quantizer

(c) Rate 1/3, 4 bit and rate 1/4, 3 bit quantizer

Figure 6: Spectral efficiency comparison.

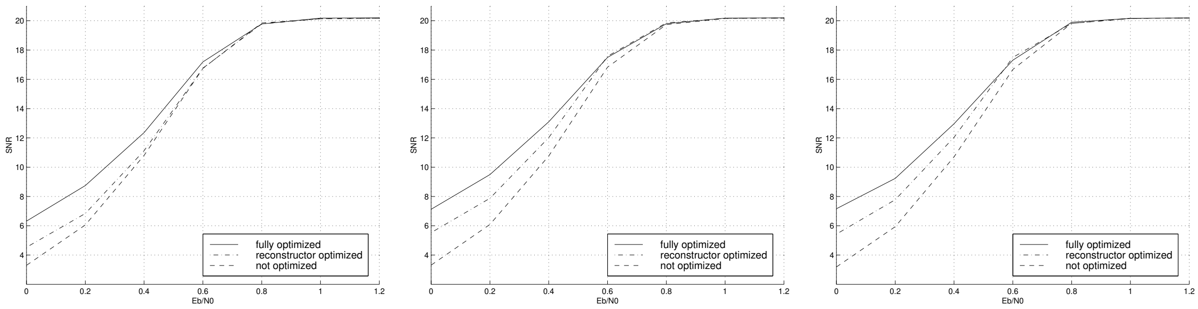
4.4 Channel Mismatch

All the quantizers presented in this work operate under the assumption that the channel noise level is known and does not change during the system operation. In other words, the quantizer and the reconstructor are designed to operate at a particular level of channel noise. In practice, this is not a realistic model, because the channel noise level fluctuates over time, and the system often operates in a channel mismatch state.

Figure 7 shows the effect of the channel mismatch for the 4 bit quantizers. The curve labeled *fully optimized* shows the signal-to-noise ratio for a quantizer and reconstructor designed to match the operating channel noise level. The curve labeled *not optimized* shows the sample signal-to-noise ratio for a quantizer that has been designed for $E_b/N_0 = 0.8$, and operated at different channel noise levels.

In another variation, the encoder is held fixed and the reconstructor is adapted to the actual channel noise level. This situation is shown with the curve labeled *reconstructor optimized*, where the quantizer has been designed for $E_b/N_0 = 0.8$ and the reconstructor is optimized for the true E_b/N_0 value. This situation is realistic because Turbo-code decoder assumes knowledge of the channel noise variance, and consequently, one can assume that the receiver can adjust the reconstructor structure accordingly.

Figure 7(a) shows the channel mismatch for the 4 bit BSQ-H quantizer with a rate 1/3 Turbo-



(a) BSQ-H quantizer (b) RIQ-S quantizer (c) GCQ-S quantizer

Figure 7: Channel Mismatch for the BSQ-H, RIQ-S, and GCQ-S quantizers.

code. At $E_b/N_0 = 0.2$ the mismatch results in 2.6 dB degradation in the sample signal-to-noise ratio from the fully optimized case, and the adaptive reconstructor reduces this degradation to only 1.8 dB. Figure 7(b) shows the channel mismatch for the 4 bit RIQ-S quantizer. The sensitivity to the mismatch is greater than the BSQ-H quantizer resulting in 3.5 dB degradation at the same E_b/N_0, however, the adaptive reconstructor reduces this degradation to only 1.7 dB. Figure 7(c) shows the mismatch for the GCQ-S quantizer. At $E_b/N_0 = 0.2$ the mismatch degradation is 3.2 dB, and the adaptive reconstructor is able to reduce this to only 1.4 dB.

5 Conclusions

This work presents a new method of combined source-channel coding using Turbo-codes. A system is presented that transmits a discrete time, continuous amplitude signal over an additive white Gaussian noise channel. The transmitted samples are encoded using a fixed rate scalar quantizer, and Turbo-code channel coding scheme is used for error correction purposes. The system performance is improved by using the soft reconstruction algorithm and the reliability information available at the Turbo-code decoder.

The performance of the presented quantizers has been shown to be up to 1.0 dB better than the Lloyd-Max or the channel optimized quantizer. Some trade-off between source coding and channel coding is also shown. Furthermore, the channel mismatch distortion is shown to be reduced from approximately 3.5 dB to approximately 1.5 dB with an adaptive receiver.

References

[1] C. Berrou, A. Glavieux, and P. Thitimajshima. Near shannon limit error-correcting coding and decoding: Turbo codes(1). In *Proceedings of ICC'93*, pp. 1064–1070, May 1993.

[2] P. Robertson. Illuminating the structure of code and decoder of parallel concatenated recursive systematic (turbo) codes. In *Globcom Conference*, pp. 1298–1303, 1994.

[3] J. Max, "Quantizing for minimum distortion," *IEEE Transactions On Information Theory*, March 1960, pp. 7–12.

[4] S. P. Lloyd, "Least squares quantization in PCM," *IEEE Transactions On Information Theory*, vol. IT-28, no. 2, March 1982, pp. 129–137.

[5] A. J. Kurntenbach and P. A. Wintz, "Quantizing for noisy channels," *IEEE Transactions on Communciation Technology*, vol. COM-17, no. 2, April 1969, pp. 291–302.

[6] N. Rydbeck and C. W. Sundberg, "Analysis of digital errors in nonlinear PCM systems," *IEEE Transactions on Communications*, vol. COM-24, January 1976, pp. 59–65.

[7] N. Farvardin and V. A. Vaishampayan, "Optimal quantizer design for noisy channels: An approach to combined source-channel coding," *IEEE Transactions On Information Theory*, vol. 33, no. 6, November 1987, pp. 827–838.

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

[9] N. Farvardin and V. A. Vaishampayan, "On the performance and complexity of channel-optimized vector quantizers," *IEEE Transactions On Information Theory*, vol. 37, no. 1, January 1991, pp. 155–160.

[10] V. A. Vaishampayan and N. Farvardin, "Joint design of block source codes and modulation signal sets," *IEEE Transactions On Information Theory*, vol. 38, no. 4, July 1992, pp. 1230–1248.

[11] N. Phamdo and F. Alajaji. Performance of COVQ over AWGN/Rayleigh channels with soft-decision BPSK modulation. In *The 1996 Conference on Information Sciences and Systems*, pp. 137–142, 1997.

[12] F. I. Alajaji and N. C. Phamdo, "Soft-decision COVQ for rayleigh-fading channels," *IEEE Communications Letters*, vol. 2, no. 6, June 1998, pp. 162–164.

[13] M. Skoglund and P. Hedelin. Vector quantization over a noisy channel using soft decision decoding. In *Proceedings of ICASSP 94*, pp. V605–V608, 1994.

[14] M. Skoglund. A soft decoder vector quantizer for a rayleigh fading channel - application to image transmission. In *Proceedings of ICASSP 95*, pp. 2507–2510, 1994.

[15] T. Ottosson and M. Skoglund. Soft multiuser decoding for vector quantization over rayleigh fading CDMA channels. In *Proceedings of ISSTA 96*, pp. 22–25, Mainz, Germany, 1996.

[16] J. Bakus and A. K. Khandani. Combined source-channel coding using turbo codes. In *The 1997 Conference on Information Sciences and Systems*, pp. 309–313, 1997.

[17] J. Bakus and A. K. Khandani, "Combined source-channel coding using turbo codes," *Electronics Letters*, vol. 33, no. 13, September 1997, pp. 1613–1614.

[18] J. He, D. J. Costello, Y. Huang, and R. L. Stevenson. On the application of turbo codes to the robust transmission of compressed images. In *ICIP'97*, 1997.

[19] Z. Peng, Y. Huang, D. J. Costello, and R. L. Stevenson. On the application of turbo codes to the robust transmission of compressed images. In *submitted to the The 1998 Conference on Information Sciences and Systems*, 1998.

[20] W. B. Pennerbaker and J. L. Mithchell, *JPEG: Still Image Compression Standard.* Van Nostrand Rheinhold, New York, NY, 1993.