

RESEARCH

Open Access

Channel-optimized scalar quantizers with erasure correcting codes

Dengyu Qiao^{1,2} and Ye Li^{1*}

Abstract

This paper investigates the design of channel-optimized scalar quantizers with erasure correcting codes over binary symmetric channels (BSC). A new scalar quantizer with uniform decoder and channel-optimized encoder aided by erasure correcting code is proposed. The lower bound for performance of the new quantizer with complemented natural code (CNC) index assignment is given. Then, in order to approach it, the single parity check code and corresponding decoding algorithm are added into the new quantizer encoder and decoder, respectively. Analytical results show that the performance of the new quantizers with CNC is better than that of the original quantizers with CNC and natural binary code (NBC) when crossover probability is at a certain range.

1 Introduction

For a uniform scalar source, it is well known from [1] that if over a noiseless channel, in terms of the mean squared distortion (MSD), the uniform scalar quantizer is optimal among all quantizers. But if across noisy channels, the uniform quantizer is no longer optimal. Therefore, joint source and channel coding has attracted much attention in [2,3], which has been seen as a promising scheme for effective data transmission over wireless channels due to its ability to cope with varying channel conditions [4-6]. Generally, there are two approaches to improving the performance of a quantizer over a noisy channel.

One is that the encoding cells in quantizer's encoders are designed to be affected by the transmission channel characteristics, for example [7], or the final positioning of the reconstruction in decoders depends on channel characteristics, for example [8], which is called as channel optimized quantizers. In [9-12], some necessary optimality conditions are given. Alternatively, an error detecting code can be cascaded with the quantizer at the expense of added transmission rate in [13].

The other one is index assignment, which is a mapping of source code symbols to channel code symbols and was studied in [14,15]. The usual goal to design an index

assignment for noisy channels is to minimize the end-to-end MSD over all possible index assignments. Some famous index assignments, such as natural binary code (NBC), Gray code and randomly chosen index assignments, are studied on a binary symmetric channel (BSC) in [16-21]. In [16], it is proved that NBC is optimal for uniform scalar quantizers and uniform source. In [17], McLaughlin et al. extended it to uniform vector quantizers. Farber and Zeger [8] also proved the optimality of NBC for uniform source and quantizers with uniform encoders and channel-optimized decoders. In [7], they not only studied on NBC code, but also proposed a new affine index assignment, named as complemented natural code (CNC).

Interestingly, it is known from [7] that for uniform decoder and channel-optimized encoder quantizer with CNC index assignment, as crossover probability increases to a certain value, some of the encoding cells appear to be empty, which is also discovered in [22,23]. These empty cells can be looked upon as redundancy, just like a form of implicit channel coding. Besides, it can be discovered in [7] that there is an implicit assumption is that transmitters should know transition error probabilities. If this assumption comes true, then decoders would exactly know the encoding cells in encoders and can judge if the receiving index is one of the indices of empty cells, but [7] does not apply any methods to defeat the receiving index of empty cells.

*Correspondence: ye.li@siat.ac.cn

¹Shenzhen Institutes of Advanced Technology, Chinese Academy of Sciences, Shenzhen, China

Full list of author information is available at the end of the article

In this paper, a genie-aided erasure code is applied into the ideal quantizer decoders, which can correct the receiving index of empty cells. The lower bound for performance of the uniform decoder and channel-optimized encoder quantizer with CNC index assignment is given. Then, the single parity check (SPC) block code is utilized to approach the lower bound. The main scheme is that, in transmitters, several indices are grouped into a block and parity check bits are appended in every block. Afterwards, if a receiving index is detected to be the index of empty cells in receivers, then it will be marked as an erasure. Then, the SPC code is used to correct erasures column by column for every block. In view of the structure, the SPC block code is a little similar to the SPC product code in [24], but their decoders are very different. The decoding scheme of the SPC block code is very simple; thus, only a little complexity will be added.

The rest of this paper is organized as follows. Section 2 gives definitions and notation. In Section 3, the lower bound for distortion of quantizers with uniform decoders and channel-optimized encoders is analyzed. Section 4 introduces the SPC block code and gives the distortion of the quantizer appending it. In Section 5, analytical results are shown. Finally, the conclusion is presented in Section 6.

2 Background

Throughout this paper, a continuous real-valued source random variable X uniformly distributed on $[0, 1]$ is considered. Some of the mathematical notations and definitions follow [7] in this section. Then, a rate n -bit quantizer is a mapping from the source to one of the real-valued codepoints (quantization level) $y_n(i)$,

$$\mathcal{Q} : \{X|X \in [0, 1]\} \longrightarrow \{y_n(i) | i = 0, 1, \dots, 2^n - 1\}.$$

A rate 2 quantizer is plotted in Figure 1 as an example. The quantizer includes a quantizer encoder that is a mapping from source X to a certain index i

$$\mathcal{Q}_e : \{X|X \in [0, 1]\} \longrightarrow \{i|i = 0, 1, \dots, 2^n - 1\},$$

and a quantizer decoder that is a mapping from the index i to codepoint $y_n(i)$,

$$\mathcal{Q}_d : \{i|i = 0, 1, \dots, 2^n - 1\} \longrightarrow \{y_n(i) | i = 0, 1, \dots, 2^n - 1\}.$$

In the encoder, the i th encoding cell can be denoted by the set

$$R_n(i) = \mathcal{Q}_e^{-1}(i).$$

If $R_n(i) = \phi$, we say $R_n(i)$ is an *empty cell*. In general, for most quantizers, there are no empty cells in encoding cells. But a kind of quantizers in [7] is considered, in which the quantizer encoder may contain empty encoding cells. The *centroid* of the i th cell of the quantizer \mathcal{Q} is presented by a conditional expectation

$$c_n(i) = E[X|X \in R_n(i)].$$

For a noisy channel, an index assignment π_n is often used to debate noise, which is a permutation of the set $\{0, 1, \dots, 2^n - 1\}$. Then, if index J is received by a quantizer decoder with index assignment, a random variable $X \in [0, 1]$ will be quantized to the quantization level

$$\mathcal{Q}_d(\pi_n^{-1}(J)) = y_n(\pi_n^{-1}(J)).$$

Then, the end-to-end mean squared error (MSE) can be written as

$$D_{(\pi_n)} = E\left[(X - \mathcal{Q}_d(\pi_n^{-1}(J)))^2\right]. \quad (1)$$

In this paper, we focus on the quantizers with *uniform decoders* and *channel-optimized encoders* [7]. A quantizer

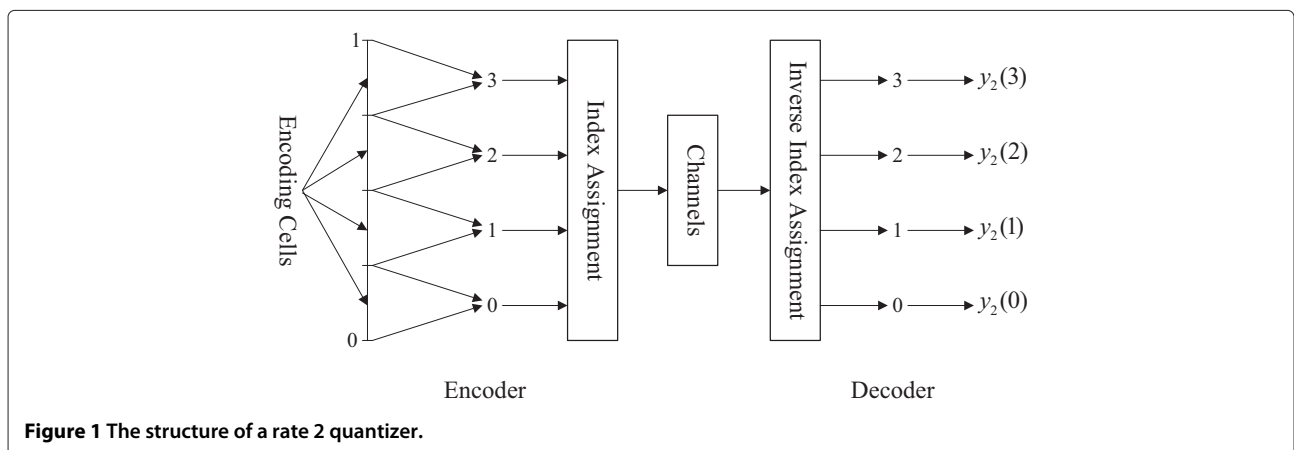


Figure 1 The structure of a rate 2 quantizer.

decoder is said to be uniform if for each i , the i th code-point satisfies

$$y_n(i) = \left(i + \frac{1}{2}\right) 2^{-n}. \quad (2)$$

A quantizer encoder is said to be a *channel-optimized encoder* if it satisfies the *weighted nearest neighbor condition*, that is,

$$W_i \subset R_n(i) \subset \overline{W}_i,$$

where

$$W_i = \left\{ x : \sum_{j=0}^{2^n-1} (x - y_n(j))^2 p_n(\pi_n(j) | \pi_n(i)) < \sum_{j=0}^{2^n-1} (x - y_n(j))^2 p_n(\pi_n(j) | \pi_n(k)), \forall k \neq i \right\}. \quad (3)$$

Here, \overline{W}_i denotes the closure of W_i , and $p_n(j|i)$ denotes the probability that index j was received, given that index i was sent. If assuming a binary symmetric channel with crossover probability ϵ , $p_n(j|i)$ can be defined as

$$p_n(j|i) = \epsilon^{H_n(i,j)} (1 - \epsilon)^{n-H_n(i,j)} \quad (4)$$

for $0 \leq \epsilon \leq 1/2$, where $H_n(i,j)$ is the Hamming distance between n -bit binary words i and j . Then, according to [7], a quantizer with a uniform decoder and a channel-optimized encoder satisfies, for all i ,

$$\overline{R}_n(i) = \{x \in [0, 1] : \alpha_n(i, k) x \geq \beta_n(i, k), \forall k \neq i\}, \quad (5)$$

$$\overline{R}_n(i) = \begin{cases} \left[0, \delta 2^{-n} - \frac{\epsilon^2}{\delta}\right], & \text{for } i = 0 \\ \left[i\delta 2^{-n} - \frac{\epsilon^2}{\delta}, (i+1)\delta 2^{-n} + \frac{\epsilon(2+\epsilon)}{1+2\epsilon}\right], & \text{for } 1 \leq i \leq 2^{n-1} - 3, i \text{ odd} \\ \left[i\delta 2^{-n} + \frac{\epsilon(2+\epsilon)}{1+2\epsilon}, (i+1)\delta 2^{-n} - \frac{\epsilon^2}{\delta}\right], & \text{for } 2 \leq i \leq 2^{n-1} - 2, i \text{ even} \\ \left[(2^{-1} - 2^{-n})\delta - \frac{\epsilon^2}{\delta}, 1/2\right], & \text{for } i = 2^{n-1} - 1 \\ \left[1/2, (2^{-1} + 2^{-n})\delta + \frac{\epsilon(2-3\epsilon)}{\delta}\right], & \text{for } i = 2^{n-1} \\ \left[i\delta 2^{-n} + \frac{\epsilon(2-3\epsilon)}{\delta}, (i+1)\delta 2^{-n} + \frac{3\epsilon^2}{1+2\epsilon}\right], & \text{for } 2^{n-1} + 1 \leq i \leq 2^n - 3, i \text{ odd} \\ \left[i\delta 2^{-n} + \frac{3\epsilon^2}{1+2\epsilon}, (i+1)\delta 2^{-n} + \frac{\epsilon(2-3\epsilon)}{\delta}\right], & \text{for } 2^{n-1} + 2 \leq i \leq 2^n - 2, i \text{ even} \\ \left[(1 - 2^{-n})\delta + \frac{\epsilon(2-3\epsilon)}{\delta}, 1\right], & \text{for } i = 2^n - 1 \end{cases} \quad (9)$$

where

$$\alpha_n(i, k) = \sum_{j=0}^{2^n-1} j [p_n(\pi_n(j) | \pi_n(i)) - p_n(\pi_n(j) | \pi_n(k))] \quad (6)$$

$$\beta_n(i, k) = 2^{-n-1} \left(\alpha_n(i, k) + \sum_{j=0}^{2^n-1} j^2 [p_n(\pi_n(j) | \pi_n(i)) - p_n(\pi_n(j) | \pi_n(k))] \right). \quad (7)$$

Let an encoder-optimized uniform quantizer (EOUQ) denote a rate n quantizer with a uniform decoder and a channel-optimized encoder, along with a uniform source on $[0,1]$, and a binary symmetric channel with crossover probability ϵ . For each n , the CNC index assignment [7] is defined by

$$\pi_n^{\text{CNC}}(i) = \begin{cases} i, & \text{for } 0 \leq i \leq 2^{n-1} - 1 \\ i + 1, & \text{for } 2^{n-1} \leq i \leq 2^n - 2 \text{ and } i \text{ even} \\ i - 1, & \text{for } 2^{n-1} + 1 \leq i \leq 2^n - 1 \text{ and } i \text{ odd.} \end{cases} \quad (8)$$

Denote

$$\epsilon_n^* = \frac{2^n + 4}{12} \cdot \left[\sqrt{3} \sin\left(\frac{\arctan(\tau/\sigma) + \pi}{3}\right) - \cos\left(\frac{\arctan(\tau/\sigma) + \pi}{3}\right) - 1 \right] + \frac{1}{2}$$

$$\sigma = 2^{n-5} - \frac{1}{27} (2^{n-2} + 1)^3$$

$$\tau = \sqrt{2^{n-4} (2^{n-6} - \sigma)}.$$

The encoding cells of an EOUQ with the CNC index assignment are given in [7] as follows.

If $n \geq 3$ and $\epsilon \in [0, \epsilon^*]$, then

where $\delta = 1 - 2\epsilon$.

If $n \geq 3$ and $\epsilon \in [\epsilon^*, 1/2)$, then

$$\bar{R}_n(i) = \begin{cases} \left[0, \delta 2^{-n} - \frac{\epsilon^2}{\delta}\right], & \text{for } i = 0 \text{ and } \epsilon < 1/(2^{n/2} + 2) \\ \left[\delta 2^{-n} - \frac{\epsilon^2}{\delta}, (4\delta + \delta^2) 2^{-n-1} + \epsilon\right] \cap [0, 1], & \text{for } i = 1 \\ \left[\frac{2\delta(i-1)+\delta^2}{2^{n+1}} + \epsilon, \frac{2\delta(i+1)+\delta^2}{2^{n+1}} + \epsilon\right], & \text{for } 3 \leq i \leq 2^{n-1} - 3, i \text{ odd} \\ \left[\frac{((2^n - 4)\delta + \delta^2 + 2^{n+1}\epsilon) 2^{-n-1}, 1/2\right], & \text{for } i = 2^{n-1} - 1 \\ \left[1/2, ((2^n + 2)\delta + 1 - 4\epsilon^2 + 2^{n+1}\epsilon) 2^{-n-1}\right], & \text{for } i = 2^{n-1} \\ \left[\frac{2\delta(i-1)+1-4\epsilon^2}{2^{n+1}} + \epsilon, \frac{2\delta(i+1)+1-4\epsilon^2}{2^{n+1}} + \epsilon\right], & \text{for } 2^{n-1} + 2 \leq i \leq 2^n - 4, i \text{ even} \\ \left[\frac{(2^{n+1}-6)\delta+1-4\epsilon^2}{2^{n+1}} + \epsilon, (1 - 2^{-n})\delta + \frac{\epsilon(2-3\epsilon)}{\delta}\right] \cap [0, 1], & \text{for } i = 2^n - 2 \\ \left[(1 - 2^{-n})\delta + \frac{\epsilon(2-3\epsilon)}{\delta}, 1\right] \cap [0, 1], & \text{for } i = 2^n - 1 \text{ and } \epsilon < 1/(2^{n/2} + 2) \\ \phi, & \text{else.} \end{cases} \quad (10)$$

As above, when $n \geq 3$ and $\epsilon \in [\epsilon^*, 1/2)$, there exists an empty cell set E , which has $2^{n-1} - 2$ elements and consists of all even numbers from 2 to $2^{n-1} - 2$ and all odd numbers from $2^{n-1} + 1$ to $2^n - 3$, given by

$$E \triangleq \{i : R_n(i) = \phi\} \quad (11)$$

$$= \{i : 2 \leq i \leq 2^{n-1} - 2, i \text{ even}\} \cup \{i : 2^{n-1} + 1 \leq i \leq 2^n - 3, i \text{ odd}\}.$$

Let $D_{(\pi_n)}$ denote the end-to-end MSD of EOUQ with index assignment π_n . The MSE of an EOUQ with the CNC index assignment [7] is

$$D_{(\text{CNC})} = \begin{cases} D_1(n, \epsilon), & \text{for } 0 \leq \epsilon \leq \epsilon^* \\ D_2(n, \epsilon), & \text{for } \epsilon^* \leq \epsilon \leq \frac{1}{2^{n/2}+2} \\ D_3(n, \epsilon), & \text{for } \frac{1}{2^{n/2}+2} \leq \epsilon \leq 1/2 \end{cases} \quad (12)$$

where

$$D_1(n, \epsilon) = \frac{2^{-2n}}{3(1+2\epsilon)} \left(\frac{1}{4} + \left(2^{2n} + \frac{5}{2}\right) \epsilon - (2^{2n+1} - 15 \cdot 2^n + 4) \epsilon^2 + 6(2^{2n} - 2^{n+2} - 4) \epsilon^3 + (2^n - 4)(2^n - 2) \epsilon^4 - 12(2^n - 4) \epsilon^5 \right)$$

$$D_2(n, \epsilon) = \frac{2^{-3n}}{3} \left(2^n - 3 + [(2^n - 3)(2^{2n} + 10) - 2^{n+1} + 48] \epsilon - [(2^n - 6)(2^n - 5)(2^n - 4) - 3 \cdot 2^{2n}] \epsilon^2 + 2(2^n - 6)(2^n - 5)(2^n - 4) \epsilon^3 + 12(2^n - 5)(2^n - 4) \epsilon^4 + 24(2^n - 4) \epsilon^5 \right)$$

$$D_3(n, \epsilon) = \frac{2^{-3n}}{3} \left(2^n + 3 + [(2^n - 3)(2^{2n} + 10) - 2^{n+1}] \epsilon - [(2^n - 6)(2^n - 5)(2^n - 4) - 3 \cdot 2^{2n}] \epsilon^2 + 2(2^n - 6)(2^n - 5)(2^n - 4) \epsilon^3 + 12(2^n - 5)(2^n - 4) \epsilon^4 + 24(2^n - 4) \epsilon^5 \right).$$

3 Channel-optimized quantizers with erasure correcting codes

As to channel-optimized encoders, the implied key assumption in [7] is that channel state information (CSI) is known to transmitters. In a time division duplexing (TDD) mode system, it is easy to know CSI for transmitters, because uplink and downlink share the same channel. But in a frequency division duplexing (FDD) mode system, it is much harder to know CSI for transmitters than the TDD mode. Generally, after CSI is estimated in the receivers, it must be fed back to transmitters through an extra reliable channel. Here, in this paper, for a binary symmetric channel, CSI only includes crossover probability ϵ . In other words, if the above assumption comes true, for a given channel-optimized encoder in [7], receivers absolutely know temporal ϵ and can judge if there exist empty cells in encoding cells and whether the receiving index belongs to empty cells or not. This is because the encoding cells are determined by ϵ . Once the empty cell index appears, receivers should recognize that the received index is erroneously detected. However, in [7], the index that is known to be an error is still sent to quantizer decoders.

An assumption is made in this section that the decoders can correct all the receiving indices of empty cells, using a genie-aided erasure correcting code, which will be discussed in the subsequent section. In other words, all encoding cells belonging to the set E are fixed to empty in the encoder, and the receiving index of empty cells can be thought as an erasure to be marked and then recovered in the decoder. Because of this assumption, the probability that index j was received given that index i was sent will be changed as

$$\forall i \in E^c \quad p'_n(\pi_n(j) | \pi_n(i)) = \begin{cases} p_n(\pi_n(i) | \pi_n(i)) + \sum_{k \in E} p_n(\pi_n(k) | \pi_n(i)) & j = i \\ 0 & j \in E \\ p_n(\pi_n(j) | \pi_n(i)) & j \in E^c \text{ and } j \neq i \end{cases} \quad (13)$$

Then, similarly, if the proposed quantizer encoder is said to satisfy the weighted nearest neighbor condition, then the encoding cells should satisfy

$$W'_i \subset R_n(i) \subset \overline{W'_i} \quad \forall i \in E^c,$$

where

$$W'_i = \left\{ x : \sum_{j \in E^c} (x - y_n(j))^2 p_n(\pi_n(j) | \pi_n(i)) + \sum_{j \in E} (x - y_n(i))^2 p_n(\pi_n(j) | \pi_n(i)) < \sum_{j \in E^c} (x - y_n(j))^2 p_n(\pi_n(j) | \pi_n(k)) + \sum_{j \in E} (x - y_n(k))^2 p_n(\pi_n(j) | \pi_n(k)), \forall k \neq i \right\}. \quad (14)$$

It is worthwhile to note that the second term in each side of the less-than operator in W'_i is different from W_i in (3). This term means that if receiving an empty cell index, our proposed quantizer decoder is able to correct it. In order to be easier to solve (14), it is rewritten as follows.

Lemma 1. For all i , the encoding cells of our proposed EOUQ satisfy,

$$\overline{R'_n(i)} = \{x \in [0, 1] : \alpha'_n(i, k)x \geq \beta'_n(i, k), \forall k \neq i\}, \quad (15)$$

where

$$\alpha'_n(i, k) = \sum_{j \in E^c} j [p_n(\pi_n(j) | \pi_n(i)) - p_n(\pi_n(j) | \pi_n(k))] + \sum_{j \in E} [i \cdot p_n(\pi_n(j) | \pi_n(i)) - k \cdot p_n(\pi_n(j) | \pi_n(k))] \quad (16)$$

$$\beta'_n(i, k) = 2^{-n-1} \left(\alpha'_n(i, k) + \sum_{j \in E^c} j^2 [p_n(\pi_n(j) | \pi_n(i)) - p_n(\pi_n(j) | \pi_n(k))] + \sum_{j \in E} [i^2 \cdot p_n(\pi_n(j) | \pi_n(i)) - k^2 \cdot p_n(\pi_n(j) | \pi_n(k))] \right) \quad (17)$$

After substituting (8), $\alpha'_n(i, k)$ and $\beta'_n(i, k)$ can be simplified as the following two corollaries.

Corollary 1. $\alpha'_n(i, k)$ can be simplified as follows.

1. $i = 0$

a. $1 \leq k \leq 2^{n-1}-1, k \text{ odd}; 2^{n-1} \leq k \leq 2^n-2, k \text{ even}$

$$\begin{aligned} \alpha'_n(0, k) = & -k + (-2^n + 2k + 2)\epsilon + (2^{n+1} - 2k - 4)\epsilon^2 \\ & + (2^n - 1)\epsilon^{n-1} - (2^n - 1)\epsilon^n \\ & - (2^n - k - 1)p_n(2^n - 2 | \pi_n(k)) \\ & + k \cdot p_n(0 | \pi_n(k)) \end{aligned}$$

b. $k = 2^n - 1$

$$\begin{aligned} \alpha'_n(0, 2^n - 1) = & -2^n + 1 + 2\epsilon + (2^{n+1} - 6)\epsilon^2 \\ & + (2^{n+1} - 2)\epsilon^{n-1} - (2^{n+1} - 2)\epsilon^n \end{aligned}$$

2. $1 \leq i \leq 2^n - 1, i \text{ odd}; 2^n - 1 \leq i \leq 2^n - 2, i \text{ even}$

a. $k = 0$

$$\begin{aligned} \alpha'_n(i, 0) = & i + (2^n - 2i - 2)\epsilon + (-2^{n+1} + 2i + 4)\epsilon^2 \\ & + (-2^n + 1)\epsilon^{n-1} + (2^n - 1)\epsilon^n \\ & + (2^n - i - 1)p_n(2^n - 2 | \pi_n(i)) - ip_n(0 | \pi_n(i)) \end{aligned}$$

b. $1 \leq k \leq 2^n - 1, k \text{ odd}; 2^n - 1 \leq k \leq 2^n - 2, k \text{ even}; k \neq i$

$$\begin{aligned} \alpha'_n(i, k) = & (i - k) - 2(i - k)\epsilon + 2(i - k)\epsilon^2 \\ & + (2^n - i - 1)p_n(2^n - 2 | \pi_n(i)) \\ & - i \cdot p_n(0 | \pi_n(i)) - (2^n - k - 1)p_n(2^n - 2 | \pi_n(k)) \\ & + k \cdot p_n(0 | \pi_n(k)) \end{aligned}$$

c. $k = 2^n - 1$

$$\begin{aligned} \alpha'_n(i, 2^n-1) = & - (2^n-i-1) + (2^n-2i) \epsilon + (2i-2) \epsilon^2 \\ & + (2^n-1) \epsilon^{n-1} + (-2^n+1) \epsilon^n \\ & + (2^n-i-1) p_n(2^n-2|\pi_n(i)) \\ & - ip_n(0|\pi_n(i)) \end{aligned}$$

3. $i = 2^n - 1$

a. $k = 0$

$$\begin{aligned} \alpha'_n(2^n-1, 0) = & 2^n - 1 - 2\epsilon - (2^{n+1}-6) \epsilon^2 \\ & - (2^{n+1}-2) \epsilon^{n-1} + (2^{n+1}-2) \epsilon^n \end{aligned}$$

b. $1 \leq k \leq 2^n-1$, k odd; $2^n - 1 \leq k \leq 2^n-2$, k even

$$\begin{aligned} \alpha'_n(2^n-1, k) = & (2^n-k-1) - (2^n-2k) \epsilon - (2k-2) \epsilon^2 \\ & - (2^n-1) \epsilon^{n-1} + (2^n-1) \epsilon^n \\ & - (2^n-k-1) p_n(2^n-2|\pi_n(k)) \\ & + kp_n(0|\pi_n(k)) \end{aligned}$$

Corollary 2. $\beta'_n(i, k)$ can be simplified as seen in the Appendix ('The simplification of $\beta'_n(i, k)$ ').

It is known from the above two corollaries that for a given i and k , $\alpha'_n(i, k)$ and $\beta'_n(i, k)$ have only one variable ϵ , so the symbolic toolbox in *Matlab* can be used to solve

the inequality set. The encoding cells $R_n(i)$ are solved for the 3-bit and 4-bit quantizer, as shown in the following two theorem.

Theorem 1. For $0 \leq \epsilon \leq 0.5$, the encoding cells for the 3-bit quantizer are

$$\overline{R'_3}(i) = \begin{cases} \left[0, \frac{1+42\epsilon^3-48\epsilon^2+12\epsilon}{8(1+12\epsilon^3-15\epsilon^2+3\epsilon)} \right], & \text{for } i = 0 \\ \left[\frac{1+42\epsilon^3-48\epsilon^2+12\epsilon}{8(1+12\epsilon^3-15\epsilon^2+3\epsilon)}, \frac{5-30\epsilon^3+6\epsilon^2-3\epsilon}{8(2-6\epsilon^3+3\epsilon^2-3\epsilon)} \right], & \text{for } i = 1 \\ \phi, & \text{for } i = 2 \\ \left[\frac{5-30\epsilon^3+6\epsilon^2-3\epsilon}{8(2-6\epsilon^3+3\epsilon^2-3\epsilon)}, \frac{1+2\epsilon^3-2\epsilon}{2(2\epsilon^3-2\epsilon+1)} \right], & \text{for } i = 3 \\ \left[\frac{1+2\epsilon^3-2\epsilon}{2(2\epsilon^3-2\epsilon+1)}, \frac{11-18\epsilon^3+18\epsilon^2-21\epsilon}{8(2-6\epsilon^3+3\epsilon^2-3\epsilon)} \right], & \text{for } i = 4 \\ \phi, & \text{for } i = 5 \\ \left[\frac{11-18\epsilon^3+18\epsilon^2-21\epsilon}{8(2-6\epsilon^3+3\epsilon^2-3\epsilon)}, \frac{7+54\epsilon^3-72\epsilon^2+12\epsilon}{8(1+12\epsilon^3-15\epsilon^2+3\epsilon)} \right], & \text{for } i = 6 \\ \left[\frac{7+54\epsilon^3-72\epsilon^2+12\epsilon}{8(1+12\epsilon^3-15\epsilon^2+3\epsilon)}, 1 \right], & \text{for } i = 7 \end{cases} \quad (18)$$

Theorem 2. For $0 \leq \epsilon \leq 0.5$, the encoding cells for the 4-bit quantizer are

$$\overline{R'_4}(i) = \begin{cases} \left[0, \frac{1+240\epsilon^4-223\epsilon^3-55\epsilon^2+52\epsilon}{16(1+30\epsilon^4-18\epsilon^3-23\epsilon^2+11\epsilon)} \right], & \text{for } i = 0 \\ \left[\frac{1+240\epsilon^4-223\epsilon^3-55\epsilon^2+52\epsilon}{16(1+30\epsilon^4-18\epsilon^3-23\epsilon^2+11\epsilon)}, \frac{-5+240\epsilon^4-173\epsilon^3+65\epsilon^2-13\epsilon}{16(-2+30\epsilon^4-21\epsilon^3+2\epsilon^2+3\epsilon)} \right], & \text{for } i = 1 \\ \phi, & \text{for } i = 2 \\ \left[\frac{-5+240\epsilon^4-173\epsilon^3+65\epsilon^2-13\epsilon}{16(-2+30\epsilon^4-21\epsilon^3+2\epsilon^2+3\epsilon)}, \frac{9+37\epsilon^3-33\epsilon^2-4\epsilon}{32(1+\epsilon^3+\epsilon^2-2\epsilon)} \right], & \text{for } i = 3 \\ \phi, & \text{for } i = 4 \\ \left[\frac{9+37\epsilon^3-33\epsilon^2-4\epsilon}{32(1+\epsilon^3+\epsilon^2-2\epsilon)}, \frac{13+240\epsilon^4-335\epsilon^3+115\epsilon^2-20\epsilon}{16(2+30\epsilon^4-43\epsilon^3+17\epsilon^2-4\epsilon)} \right], & \text{for } i = 5 \\ \phi, & \text{for } i = 6 \\ \left[\frac{13+240\epsilon^4-335\epsilon^3+115\epsilon^2-20\epsilon}{16(2+30\epsilon^4-43\epsilon^3+17\epsilon^2-4\epsilon)}, \frac{1}{2} \right], & \text{for } i = 7 \\ \left[\frac{1}{2}, \frac{19+240\epsilon^4-353\epsilon^3+157\epsilon^2-44\epsilon}{16(2+30\epsilon^4-43\epsilon^3+17\epsilon^2-4\epsilon)} \right], & \text{for } i = 8 \\ \phi, & \text{for } i = 9 \\ \left[\frac{19+240\epsilon^4-353\epsilon^3+157\epsilon^2-44\epsilon}{16(2+30\epsilon^4-43\epsilon^3+17\epsilon^2-4\epsilon)}, \frac{23-5\epsilon^3+65\epsilon^2-60\epsilon}{32(1+\epsilon^3+\epsilon^2-2\epsilon)} \right], & \text{for } i = 10 \\ \phi, & \text{for } i = 11 \\ \left[\frac{23-5\epsilon^3+65\epsilon^2-60\epsilon}{32(1+\epsilon^3+\epsilon^2-2\epsilon)}, \frac{-27+240\epsilon^4-163\epsilon^3-33\epsilon^2+61\epsilon}{16(-2+30\epsilon^4-21\epsilon^3+2\epsilon^2+3\epsilon)} \right], & \text{for } i = 12 \\ \phi, & \text{for } i = 13 \\ \left[\frac{-27+240\epsilon^4-163\epsilon^3-33\epsilon^2+61\epsilon}{16(-2+30\epsilon^4-21\epsilon^3+2\epsilon^2+3\epsilon)}, \frac{15+240\epsilon^4-65\epsilon^3-313\epsilon^2+124\epsilon}{16(1+30\epsilon^4-18\epsilon^3-23\epsilon^2+11\epsilon)} \right], & \text{for } i = 14 \\ \left[\frac{15+240\epsilon^4-65\epsilon^3-313\epsilon^2+124\epsilon}{16(1+30\epsilon^4-18\epsilon^3-23\epsilon^2+11\epsilon)}, 1 \right], & \text{for } i = 15 \end{cases} \quad (19)$$

For $i \in E$, define the quantities

$$\begin{aligned} I_r(i) &= \underset{j \in E^c}{\operatorname{argmin}} c_n(j) \\ I_l(i) &= \underset{j \in E^c}{\operatorname{argmax}} c_n(j) \\ z_n(i) &= \sup_{R_n(i)} \\ V &= \{i : 1 \notin \bar{R}_n(i)\} \cap E^c \\ I_1 &= V^c \cap E^c \end{aligned}$$

Let $D_{(\pi_n)}^{\text{lower bound}}$ denote the end-to-end MSE of modified EOUQ with index assignment π_n appending a genie-aided erasure correcting code, which is the lower bound for the MSE of modified EOUQ.

Theorem 3. *The lower bound for MSE of a modified EOUQ with the CNC index assignment is*

$$\begin{aligned} D_{(\text{CNC})}^{\text{lower bound}} &= \frac{1}{3} - 2^{-n-1} + 2^{-2n-2} + 2^{-n} \left[\sum_{i \in V} z_n^2(i) \cdot \alpha_n(i, I_r(i)) \right. \\ &\quad - \left(\sum_{i \in E^c} j p_n(\pi_n^{(\text{CNC})}(j) | \pi_n^{(\text{CNC})}(I_1)) \right. \\ &\quad \left. \left. + \sum_{i \in E} i p_n(\pi_n^{(\text{CNC})}(j) | \pi_n^{(\text{CNC})}(I_1)) \right) \right] \\ &\quad + 2^{-2n} \left(\sum_{i \in E^c} (j + j^2) p_n(\pi_n^{(\text{CNC})}(j) | \pi_n^{(\text{CNC})}(I_1)) \right. \\ &\quad \left. + \sum_{i \in E} (i + i^2) p_n(\pi_n^{(\text{CNC})}(j) | \pi_n^{(\text{CNC})}(I_1)) \right). \end{aligned} \quad (20)$$

Proof. According to (1) and the assumption we make,

$$\begin{aligned} D_{(\text{CNC})}^{\text{lower bound}} &= \sum_{i \in E^c} \sum_{j \in E^c} p_n(\pi_n^{(\text{CNC})}(j) | \pi_n^{(\text{CNC})}(i)) \\ &\quad \times \int_{R_n(i)} (x - y_n(j))^2 dx \\ &\quad + \sum_{i \in E^c} \sum_{j \in E} p_n(\pi_n^{(\text{CNC})}(j) | \pi_n^{(\text{CNC})}(i)) \\ &\quad \times \int_{R_n(i)} (x - y_n(i))^2 dx \end{aligned}$$

$$D_{(\text{CNC})}^{\text{lower bound}} = \frac{5 + 243\epsilon + 135\epsilon^2 + 579\epsilon^3 - 66,276\epsilon^6 + 39,330\epsilon^5 - 13,932\epsilon^4 + 13,176\epsilon^9 - 59,184\epsilon^8 + 84,888\epsilon^7}{768(1 + 12\epsilon^3 - 15\epsilon^2 + 3\epsilon)(2 - 6\epsilon^3 + 3\epsilon^2 - 3\epsilon)} \quad (21)$$

$$\begin{aligned} &= \sum_{i \in E^c} \sum_{j \in E^c} p_n(\pi_n^{(\text{CNC})}(j) | \pi_n^{(\text{CNC})}(i)) \\ &\quad \times \int_{R_n(i)} (x^2 - 2xy_n(j) + y_n^2(j)) dx \\ &\quad + \sum_{i \in E^c} \sum_{j \in E} p_n(\pi_n^{(\text{CNC})}(j) | \pi_n^{(\text{CNC})}(i)) \\ &\quad \times \int_{R_n(i)} (x^2 - 2xy_n(i) + y_n^2(i)) dx \\ &= \sum_{i \in E^c} \sum_{j \in E^c} p_n(\pi_n^{(\text{CNC})}(j) | \pi_n^{(\text{CNC})}(i)) \\ &\quad \times \int_{R_n(i)} \left(x^2 - 2x \frac{j+0.5}{2^n} + \left(\frac{j+0.5}{2^n} \right)^2 \right) dx \\ &\quad + \sum_{i \in E^c} \sum_{j \in E} p_n(\pi_n^{(\text{CNC})}(j) | \pi_n^{(\text{CNC})}(i)) \\ &\quad \times \int_{R_n(i)} \left(x^2 - 2x \frac{i+0.5}{2^n} + \left(\frac{i+0.5}{2^n} \right)^2 \right) dx \\ &= \frac{1}{3} - 2^{-n-1} + 2^{-2n-2} \\ &\quad - 2^{-n} \sum_{i \in E^c} [z_n^2(i) - z_n^2(I_l(i))] \\ &\quad \times \left[\sum_{i \in E^c} j p_n(\pi_n^{(\text{CNC})}(j) | \pi_n^{(\text{CNC})}(i)) \right. \\ &\quad \left. + \sum_{i \in E} i p_n(\pi_n^{(\text{CNC})}(j) | \pi_n^{(\text{CNC})}(i)) \right] \\ &\quad + 2^{-2n} \sum_{i \in E^c} [z_n(i) - z_n(I_l(i))] \\ &\quad \times \left[\sum_{i \in E^c} (j + j^2) p_n(\pi_n^{(\text{CNC})}(j) | \pi_n^{(\text{CNC})}(i)) \right. \\ &\quad \left. + \sum_{i \in E} (i + i^2) p_n(\pi_n^{(\text{CNC})}(j) | \pi_n^{(\text{CNC})}(i)) \right]. \end{aligned}$$

After re-expressing $I_l(i)$ the above formula with respect to $I_r(i)$, and merging like terms according to the definitions of $\alpha'_n(i, k)$ and $\beta'_n(i, k)$,

$$\begin{aligned} D_{(\text{CNC})}^{\text{lower bound}} &= \frac{1}{3} - 2^{-n-1} + 2^{-2n-2} + 2^{-n} \left[\sum_{i \in V} z_n^2 \cdot \alpha_n(i, I_r(i)) \right. \\ &\quad - \left(\sum_{j \in E^c} j p_n(\pi_n^{(\text{CNC})}(j) | \pi_n^{(\text{CNC})}(I_1)) \right. \\ &\quad \left. \left. + \sum_{j \in E} i p_n(\pi_n^{(\text{CNC})}(j) | \pi_n^{(\text{CNC})}(I_1)) \right) \right] \\ &\quad + 2^{-2n} \left(\sum_{j \in E^c} (j + j^2) p_n(\pi_n^{(\text{CNC})}(j) | \pi_n^{(\text{CNC})}(I_1)) \right. \\ &\quad \left. + \sum_{j \in E} (i + i^2) p_n(\pi_n^{(\text{CNC})}(j) | \pi_n^{(\text{CNC})}(I_1)) \right). \end{aligned}$$

□

After substituting (18) and (19), respectively, Theorem 3 gives the following results.

Theorem 4. *The lower bound for MSE of a 3-bit modified EOUQ with the CNC index assignment is*

Theorem 5. *The lower bound for MSE of a 4-bit modified EOUQ with the CNC index assignment is*

$$\begin{aligned}
 D_{(CNC)}^{lower\ bound} = & (11,740\epsilon + 5,884\epsilon^2 - 476,023\epsilon^3 \\
 & + 3,328,344\epsilon^4 - 1,195,090,122\epsilon^{12} \\
 & - 5,134,292,766\epsilon^{15} + 4,009,702,089\epsilon^{14} \\
 & - 801,350,784\epsilon^{13} + 156,129,623\epsilon^8 \\
 & - 107,373,749\epsilon^7 + 47,124,939\epsilon^6 \\
 & - 14,855,907\epsilon^5 + 1,183,613,619\epsilon^{11} \\
 & - 441,562,375\epsilon^{10} - 43,375,642\epsilon^9 \\
 & + 2,844,849,870\epsilon^{16} + 155,520,000\epsilon^{19} \\
 & - 518,967,000\epsilon^{18} - 143,067,600\epsilon^{17} + 52) \\
 & /12,288 [(1 + 30\epsilon^4 - 18\epsilon^3 - 23\epsilon^2 + 11\epsilon) \\
 & \times (2 - 30\epsilon^4 + 21\epsilon^3 - 2\epsilon^2 - 3\epsilon) \\
 & \times (1 + \epsilon^3 + \epsilon^2 - 2\epsilon) \\
 & (2 + 30\epsilon^4 - 43\epsilon^3 + 17\epsilon^2 - 4\epsilon)]
 \end{aligned}
 \tag{22}$$

4 EOUQ with CNC aided by single parity check block code

A good erasure code for our proposed quantizers should have excellent ability to correct erasures but no ability to correct errors. This is because the codes that have error correcting ability can improve the performance of not only our proposed quantizers but also the quantizers in [7], and the benefits are equal. Thus, in this section, we focus on SPC code to correct the empty cell index that appears in receivers, in order to approach the lower bound for EOUQ with CNC index assignment.

4.1 SPC Code

The SPC code is one of the most popular error detection codes for it is easy to be implemented, which is also able to correct a single erasure. But, if there exist multiple erasures, the typical decoding method for the single-erasure

case will fail to recover them. In this paper, we present a modified decoding method to deal with the multiple-erasure case for the SPC code. The main idea is that when one of the multiple erasures is being recovered, the other erasures are restored to the original value before being marked. Then, the multiple-erasure case can be transferred to several cases for a single erasure. Thus, the typical decoding method for the single-erasure case is still to be effective. Figure 2 gives an illustration of the modified decoding algorithm for the SPC code. According to Figure 2, as multiple erasures are recovered independently if multiple erasures exist, the erasure recovery probability \tilde{P}_c with modified decoding method for the single- or multiple-erasure case is equal to the erasure recovery probability P_c with typical decoding method for the single-erasure case, i.e.,

$$\tilde{P}_c = P_c.
 \tag{23}$$

Then, the probability P_c that a single erasure can be recovered by the SPC typical decoder is given in the following theorem.

Theorem 6. *If a single erasure is detected, the probability that the erasure can be recovered by SPC code is*

$$P_c = \sum_{\substack{j=0 \\ j \text{ is even}}}^{\bar{k}} C_{k-1}^j (1 - \epsilon)^{k-j-1} \epsilon^j,
 \tag{24}$$

where C_n^k denotes an n choose k function, $\bar{k} = 2\lfloor(k-1)/2\rfloor$, and $\lfloor x \rfloor$ means to round x to the nearest integers less than or equal to x .

Proof. If a single erasure appears in the receiving SPC code, all error events can be classified into the following several cases.

Case 1. No error: when there exist no errors in the column, the erasure can be definitely recovered. Then, in this case, the recovery probability is $(1 - \epsilon)^k$.

Case 2. One error: when there exists one error in the column, only if the single error happens to be located at the

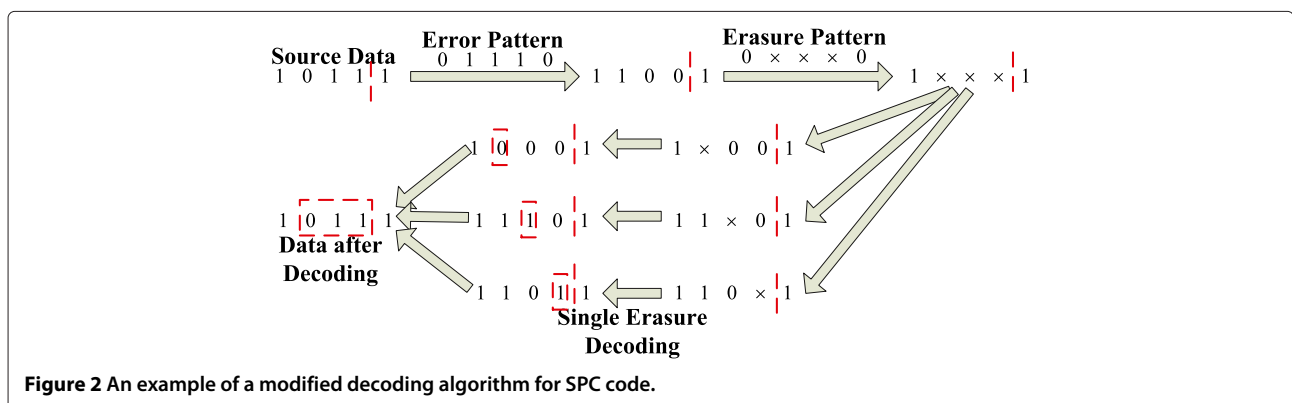


Figure 2 An example of a modified decoding algorithm for SPC code.

erasure, the erasure can be recovered. Then, the recovery probability, in this case, is $(1 - \epsilon)^{k-1} \epsilon$.

Case 3. Two errors: when there exist two errors in the column, only if the two errors are both not located at the erasure, the erasure can be recovered. Then, the recovery probability, in this case, is $C_{k-1}^2 (1 - \epsilon)^{k-2} \epsilon^2$.

Case 4. Odd number of errors: when there exist i (odd number) errors in the column, similar to 'one error case', only if one of the errors happens to be located at the erasure, the erasure can be recovered. Then, the recovery probability, in this case, is $C_{k-1}^{i-1} (1 - \epsilon)^{k-i} \epsilon^i$.

Case 5. Even number of errors: when there exist j (even number) errors in the column, similar to 'two errors case', only if all errors are all not located at the erasure, the erasure can be recovered. Then, the recovery probability, in this case, is $C_{k-1}^j (1 - \epsilon)^{k-j} \epsilon^j$.

In sum, according to all cases, the probability that the single erasure can be recovered by SPC code can be written as

$$P_c = (1 - \epsilon)^k + C_{k-1}^2 (1 - \epsilon)^{k-2} \epsilon^2 + \dots + C_{k-1}^j (1 - \epsilon)^{k-j} \epsilon^j + \dots + C_{k-1}^{1-1} (1 - \epsilon)^{k-1} \epsilon + C_{k-1}^{3-1} (1 - \epsilon)^{k-3} \epsilon^3 + \dots + C_{k-1}^{i-1} (1 - \epsilon)^{k-i} \epsilon^i + \dots,$$

where i is an odd number and j is an even number. Assuming k is odd, without of generalization, let $i = j + 1$, so that

$$P_c = (1 - \epsilon)^k + C_{k-1}^2 (1 - \epsilon)^{k-2} \epsilon^2 + \dots + C_{k-1}^j (1 - \epsilon)^{k-j} \epsilon^j + \dots + C_{k-1}^{1-1} (1 - \epsilon)^{k-1} \epsilon + C_{k-1}^{3-1} (1 - \epsilon)^{k-3} \epsilon^3 + \dots + C_{k-1}^{j+1-1} (1 - \epsilon)^{k-j-1} \epsilon^{j+1} + \dots = C_{k-1}^0 \left[(1 - \epsilon)^k + (1 - \epsilon)^{k-1} \epsilon \right] + C_{k-1}^2 \left[(1 - \epsilon)^{k-2} \epsilon^2 + (1 - \epsilon)^{k-3} \epsilon^3 \right] + \dots + C_{k-1}^j \left[(1 - \epsilon)^{k-j} \epsilon^j + (1 - \epsilon)^{k-j-1} \epsilon^{j+1} \right] + \dots + C_{k-1}^{k-1} \left[(1 - \epsilon)^{k-(k-1)} \epsilon^{k-1} + (1 - \epsilon)^{k-(k-1)-1} \epsilon^{k-1+1} \right] = C_{k-1}^0 (1 - \epsilon)^{k-1} + C_{k-1}^2 (1 - \epsilon)^{k-3} \epsilon^2 + \dots + C_{k-1}^j (1 - \epsilon)^{k-j-1} \epsilon^j + C_{k-1}^{k-1} \epsilon^{k-1} = \sum_{\substack{j=0 \\ j \text{ is even}}}^{k-1} C_{k-1}^j (1 - \epsilon)^{k-j-1} \epsilon^j.$$

Supposing k is even, also let $i = j + 1$, so that

$$P_c = (1 - \epsilon)^k + C_{k-1}^2 (1 - \epsilon)^{k-2} \epsilon^2 + \dots + C_{k-1}^j (1 - \epsilon)^{k-j} \epsilon^j + \dots + C_{k-1}^{1-1} (1 - \epsilon)^{k-1} \epsilon + C_{k-1}^{3-1} (1 - \epsilon)^{k-3} \epsilon^3 + \dots + C_{k-1}^{j+1-1} (1 - \epsilon)^{k-j-1} \epsilon^{j+1} + \dots = C_{k-1}^0 \left[(1 - \epsilon)^k + (1 - \epsilon)^{k-1} \epsilon \right] + C_{k-1}^2 \left[(1 - \epsilon)^{k-2} \epsilon^2 + (1 - \epsilon)^{k-3} \epsilon^3 \right] + \dots + C_{k-1}^j \left[(1 - \epsilon)^{k-j} \epsilon^j + (1 - \epsilon)^{k-j-1} \epsilon^{j+1} \right] + \dots + C_{k-1}^{k-2} \left[(1 - \epsilon)^{k-(k-2)} \epsilon^{k-2} + (1 - \epsilon)^{k-(k-2)-1} \epsilon^{k-2+1} \right] = C_{k-1}^0 (1 - \epsilon)^{k-1} + C_{k-1}^2 (1 - \epsilon)^{k-3} \epsilon^2 + \dots + C_{k-1}^j (1 - \epsilon)^{k-j-1} \epsilon^j + C_{k-1}^{k-2} (1 - \epsilon) \epsilon^{k-2} = \sum_{\substack{j=0 \\ j \text{ is even}}}^{k-2} C_{k-1}^j (1 - \epsilon)^{k-j-1} \epsilon^j.$$

Therefore, for any $k \geq 2$, the probability that the single erasure can be recovered by SPC code is

$$P_c = \sum_{\substack{j=0 \\ j \text{ is even}}}^{2\lfloor (k-1)/2 \rfloor} C_{k-1}^j (1 - \epsilon)^{k-j-1} \epsilon^j. \quad (25)$$

□

4.2 SPC block code

In this paper, $k - 1$ transmitting indices with parity bits are grouped into a $k \times n$ SPC block code, as shown in Figure 3. In a SPC block code, every index is converted to a binary word and then placed row by row, and bits in every column are grouped to a SPC code, respectively. If an index in a row is detected to belong to the empty cell set, all entries in the row are marked as an erasure word. In this paper, erasure word denotes the whole erased bits in one row, and then, a bit in the erasure word is

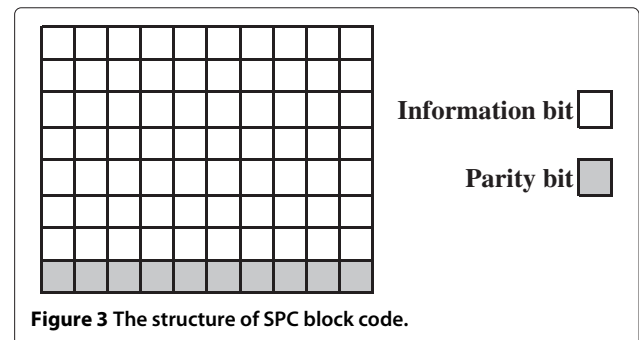


Figure 3 The structure of SPC block code.

called as erasure bit. Then, the modified SPC decoding method shown in Figure 2 is used to recover every erasure bit column by column. Figure 4 gives an illustration of the decoding algorithm for a 3-bit quantizer aided by 5×3 SPC block code. As shown in Figure 4, if multiple rows are marked as erasure word, when the erasure words in one row are being recovered, the erasure words in other rows are restored to the original value before being marked.

In order to avoid confusion, we define that I denotes the transmitted index, K denotes the input of the SPC block decoder, and J denotes the output of the SPC block decoder. Then, the SPC block code-aided transition probability $\tilde{p}_n(j|i)$ can be defined as follows.

Theorem 7. *If aided by SPC block code, the transition probability $\tilde{p}_n(j|i)$ that index j is output from the SPC*

block decoder given that index i is transmitted can be written as

$$\tilde{p}_n(j|i) = \begin{cases} p_n(j|i) + (\sum_{k \in E} p_n(k|i)) \cdot (1 - P_c)^{H_n(i,j)} P_c^{n-H_n(i,j)} & j \notin E \\ (\sum_{k \in E} p_n(k|i)) \cdot (1 - P_c)^{H_n(i,i)} P_c^{n-H_n(i,i)} & j \in E \end{cases} \quad (26)$$

Proof.

$$\begin{aligned} \tilde{p}_n(j|i) &\triangleq \tilde{p}_n(J = j|I = i) \\ &= \frac{\tilde{p}_n(J = j, I = i)}{\tilde{p}_n(I = i)} \\ &= \frac{\sum_k \tilde{p}_n(J = j, K = k, I = i)}{\tilde{p}_n(I = i)} \\ &= \frac{\sum_k \tilde{p}_n(J = j|K = k, I = i) \cdot \tilde{p}_n(K = k, I = i)}{\tilde{p}_n(I = i)} \\ &= \sum_k \tilde{p}_n(J = j|K = k, I = i) \cdot \tilde{p}_n(K = k|I = i) \\ &= \sum_k \tilde{p}_n(J = j|K = k, I = i) \cdot p_n(k|i), \end{aligned} \quad (27)$$

where $p_n(k|i)$ is defined in (4). Obviously, $\tilde{p}_n(K = k|I = i) = p_n(k|i)$.

Case 1: $k \notin E$ (index input into decoders does *not* belong to empty cells)

$$\tilde{p}_n(J = j|K = k, I = i) = \begin{cases} 1 & k = j \text{ and } j \notin E \\ 0 & k \neq j. \end{cases} \quad (28)$$

This is because if the input index does not belong to the empty cell, it will not be marked as erasure and not be changed by the SPC block decoder.

Case 2: $k \in E$ (index input into decoders belongs to empty cells)

In this case, $\tilde{p}_n(J = j|K = k, I = i)$ denotes the probability that index j is output from the SPC block code, given that index i is sent and index k , which belongs to empty cells, is received and input into the SPC block decoder. The whole bits in the row where index k lies in will be marked as erasure word. According to the proposed SPC block decoding algorithm shown in Figure 4, every erasure bit will try to be recovered by the modified SPC decoder (mentioned in Figure 2) column by column, respectively. If multiple rows are marked as erasure word, when erasure words in one row are being recovered, the erasure words in other row are restored to original value before marked. Recalled from (23), the recovery probability for one erasure bit, \tilde{P}_c , can be obtained from Theorem 6. Assuming N denotes the number of bits that fail to be recovered for an erasure word, then the number of bits that succeed

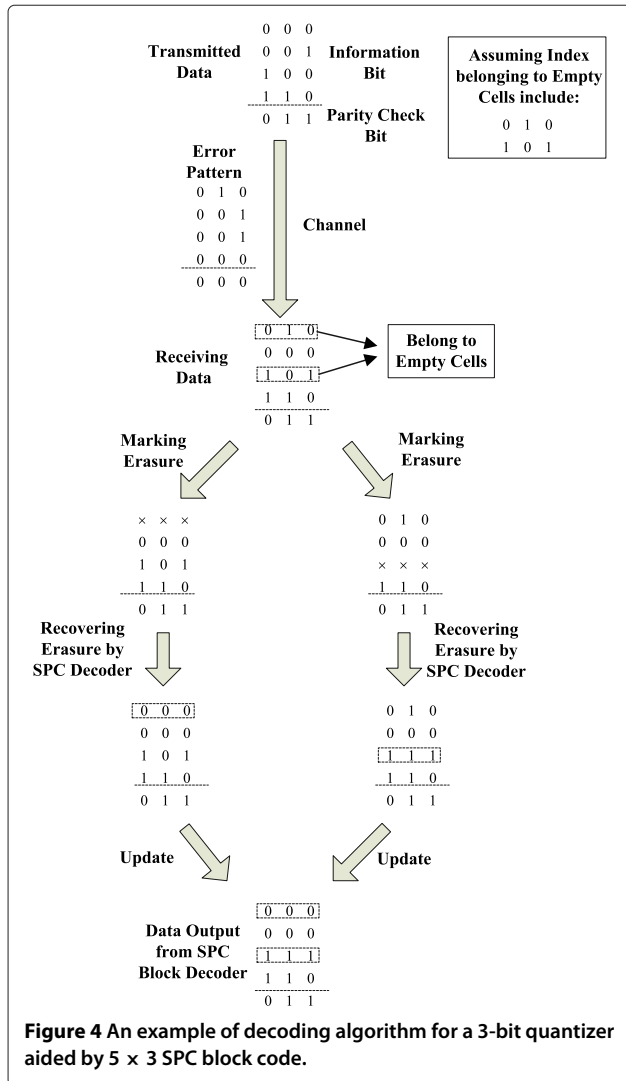


Figure 4 An example of decoding algorithm for a 3-bit quantizer aided by 5×3 SPC block code.

to be recovered for an erasure word is equal to $n - N$. Then,

$$\begin{aligned} \tilde{P}_n (J = j|K = k, I = i) &= (1 - \tilde{P}_c)^N \cdot \tilde{P}_c^{n-N} \\ &= (1 - P_c)^N \cdot P_c^{n-N} \end{aligned} \quad (29)$$

Obviously, $N = H_n(i, j)$, where $H_n(i, j)$ is the Hamming distance between n -bit binary words i and j .

Therefore,

$$\tilde{P}_n (J = j|K = k, I = i) = (1 - P_c)^{H_n(i,j)} P_c^{n-H_n(i,j)}. \quad (30)$$

After substituting (28) and (30), (27) can be rewritten as

$$\tilde{p}_n(j|i) = \begin{cases} p_n(j|i) + (\sum_{k \in E} p_n(k|i)) \cdot (1 - P_c)^{H_n(i,j)} P_c^{n-H_n(i,j)} & j \notin E \\ (\sum_{k \in E} p_n(k|i)) \cdot (1 - P_c)^{H_n(i,j)} P_c^{n-H_n(i,j)} & j \in E \end{cases} \quad (31)$$

□

Then, MSD can be written as

$$\begin{aligned} D_{(\text{CNC})}^{\text{SPC}} &= \sum_i \sum_j \tilde{P}_n(\pi_n^{(\text{CNC})}(j) | \pi_n^{(\text{CNC})}(i)) \\ &\quad \times \int_{R_n(i)} (x - y_n(j))^2 dx \\ &= \sum_i \sum_j \tilde{P}_n(\pi_n^{(\text{CNC})}(j) | \pi_n^{(\text{CNC})}(i)) \\ &\quad \times \left(\frac{x^3}{3} - y_n(j) x^2 + y_n^2(j) x \right) \Big|_{R_n(i)}. \end{aligned}$$

Now, $D_{(\text{CNC})}^{\text{SPC}}$ is a function of ϵ and k . Thus, we can use the Symbolic toolbox in Matlab to obtain the exact expression for every special case as follows.

For 3-bit quantizer, $K = 3$

$$\begin{aligned} D_{(\text{CNC})}^{K=3} &= (7,962,624\epsilon^{21} - 66,686,976\epsilon^{20} \\ &\quad + 264,508,416\epsilon^{19} - 669,171,456\epsilon^{18} \\ &\quad + 1,226,192,256\epsilon^{17} - 1,745,986,752\epsilon^{16} \\ &\quad + 2,013,793,056\epsilon^{15} - 1,922,961,168\epsilon^{14} \\ &\quad + 1,526,564,664\epsilon^{13} - 992,715,912\epsilon^{12} \\ &\quad + 509,652,360\epsilon^{11} - 192,400,668\epsilon^{10} \\ &\quad + 44,669,958\epsilon^9 - 1,219,602\epsilon^8 - 2,647,899\epsilon^7 \\ &\quad + 71,307\epsilon^6 + 553,449\epsilon^5 - 180,846\epsilon^4 \\ &\quad + 4,059\epsilon^3 + 2,763\epsilon^2 + 501\epsilon + 10) \\ &\quad / [768(-2 + 6\epsilon^3 - 3\epsilon^2 + 3\epsilon)^2 \\ &\quad \times (1 + 12\epsilon^3 - 15\epsilon^2 + 3\epsilon)^2] \end{aligned} \quad (32)$$

For 3-bit quantizer, $K = 4$

$$\begin{aligned} D_{(\text{CNC})}^{K=4} &= (63,700,992\epsilon^{24} - 629,047,296\epsilon^{23} \\ &\quad + 2,944,180,224\epsilon^{22} - 8,785,760,256\epsilon^{21} \\ &\quad + 18,977,504,256\epsilon^{20} - 31,903,206,912\epsilon^{19} \\ &\quad + 43,683,065,856\epsilon^{18} - 50,100,968,448\epsilon^{17} \\ &\quad + 48,911,316,480\epsilon^{16} - 40,974,579,648\epsilon^{15} \\ &\quad + 29,499,005,376\epsilon^{14} - 18,128,322,720\epsilon^{13} \\ &\quad + 9,340,103,424\epsilon^{12} - 3,904,109,052\epsilon^{11} \\ &\quad + 1,250,869,332\epsilon^{10} - 273,128,574\epsilon^9 \\ &\quad + 26,813,814\epsilon^8 + 3,148,215\epsilon^7 + 92,601\epsilon^6 \\ &\quad - 947,925\epsilon^5 + 277,830\epsilon^4 \\ &\quad - 7,539\epsilon^3 - 3,735\epsilon^2 - 501\epsilon - 10) \\ &\quad / [-768(1 + 12\epsilon^3 - 15\epsilon^2 + 3\epsilon)^2 \\ &\quad \times (-2 + 6\epsilon^3 - 3\epsilon^2 + 3\epsilon)^2] \end{aligned} \quad (33)$$

For 3-bit quantizer, $K = 5$

$$\begin{aligned} D_{(\text{CNC})}^{K=5} &= (509,607,936\epsilon^{27} - 5,796,790,272\epsilon^{26} \\ &\quad + 31,484,215,296\epsilon^{25} - 109,374,603,264\epsilon^{24} \\ &\quad + 274,864,472,064\epsilon^{23} - 535,778,758,656\epsilon^{22} \\ &\quad + 847,229,552,640\epsilon^{21} - 1,120,016,581,632\epsilon^{20} \\ &\quad + 1,262,654,369,280\epsilon^{19} - 1,229,166,254,592\epsilon^{18} \\ &\quad + 1,040,713,894,656\epsilon^{17} - 768,801,597,696\epsilon^{16} \\ &\quad + 495,151,866,816\epsilon^{15} - 276,513,260,880\epsilon^{14} \\ &\quad + 132,315,403,416\epsilon^{13} - 53,156,640,288\epsilon^{12} \\ &\quad + 17,336,334,384\epsilon^{11} - 4,320,019,992\epsilon^{10} \\ &\quad + 713,651,970\epsilon^9 - 39,095,634\epsilon^8 \\ &\quad - 11,051,139\epsilon^7 + 193,347\epsilon^6 + 1,467,897\epsilon^5 \\ &\quad - 386,442\epsilon^4 + 9,711\epsilon^3 \\ &\quad + 4,707\epsilon^2 + 501\epsilon + 10) / [768(-2 + 6\epsilon^3 \\ &\quad - 3\epsilon^2 + 3\epsilon)^2 (1 + 12\epsilon^3 - 15\epsilon^2 + 3\epsilon)^2] \end{aligned} \quad (34)$$

For 4-bit quantizer, $K = 3$

$$\begin{aligned} D_{(\text{CNC})}^{K=3} &= (14,276,736,000,000\epsilon^{39} 107,794,022,400,000\epsilon^{38} \\ &\quad + 290,417,114,880,000\epsilon^{37} 33,167,086,848,000\epsilon^{36} \\ &\quad - 2,096,081,266,694,400\epsilon^{35} \\ &\quad + 7,146,138,215,312,640\epsilon^{34} 11,787,493,179,759,552\epsilon^{33} \\ &\quad + 4,294,650,642,726,528\epsilon^{32} \\ &\quad + 30,745,837,163,362,608\epsilon^{31} 93,407,604,210,871,728\epsilon^{30} \\ &\quad + 152,012,048,832,668,196\epsilon^{29} 160,041,493,676,832,384\epsilon^{28} \\ &\quad + 101,864,889,812,630,040\epsilon^{27} \\ &\quad - 14,665,606,004,624,670\epsilon^{26} 45,599,458,954,787,901\epsilon^{25} \\ &\quad + 54,961,029,779,720,829\epsilon^{24} \\ &\quad - 31,880,510,555,625,807\epsilon^{23} \\ &\quad + 7,071,644,674,463,064\epsilon^{22} \\ &\quad + 4,425,365,815,868,583\epsilon^{21} \\ &\quad - 5,028,119,456,259,998\epsilon^{20} \\ &\quad + 2,223,302,804,647,541\epsilon^{19} 234,761,920,605,312\epsilon^{18} \\ &\quad - 379,032,669,202,821\epsilon^{17} \\ &\quad + 329,357,542,510,982\epsilon^{16} 163,086,464,543,222\epsilon^{15} \\ &\quad + 57,945,094,589,773\epsilon^{14} 15,146,844,338,710\epsilon^{13} \end{aligned}$$

$$\begin{aligned}
 &+2,665,837,646,460\epsilon^{12}154,116,255,422\epsilon^{11} \\
 &-97,607,466,956\epsilon^{10} + 50,967,593,269\epsilon^9 16,469,653,107\epsilon^8 \\
 &+4,300,904,858\epsilon^7 - 1,004,660,075\epsilon^6 \\
 &+202,215,844\epsilon^5 22,094,182\epsilon^4 1,646,856\epsilon^3 + 400,968\epsilon^2 \\
 &+48,104\epsilon + 208) / \left[12,288(2 + 30\epsilon^4 43\epsilon^3 + 17\epsilon^2 4\epsilon)^2 \right. \\
 &(1 + 30\epsilon^4 18\epsilon^3 23\epsilon^2 + 11\epsilon)^2 (-2 + 30\epsilon^4 21\epsilon^3 + 2\epsilon^2 + 3\epsilon)^2 \\
 &\left. (1 + \epsilon^3 + \epsilon^2 - 2\epsilon)^2 \right]
 \end{aligned} \tag{35}$$

$$\begin{aligned}
 &-18,259,673,092,737,874,944\epsilon^{36} + 3,150,966,289,987,906,560\epsilon^{37} \\
 &+2,903,688,409,553,842,176\epsilon^{38} - 2,899,835,033,082,052,608\epsilon^{39} \\
 &+1,258,694,025,775,349,760\epsilon^{40} - 228,653,524,883,865,600\epsilon^{41} \\
 &-47,498,124,460,032,000\epsilon^{42} + 39,377,206,149,120,000\epsilon^{43} \\
 &-9,639,950,745,600,000\epsilon^{44} + 913,711,104,000,000\epsilon^{45}) \\
 &/ \left[12,288(-2 + 30\epsilon^4 - 21\epsilon^3 + 2\epsilon^2 + 3\epsilon)^2 (1 + \epsilon^3 + \epsilon^2 - 2\epsilon)^2 \right. \\
 &\left. (2 + 30\epsilon^4 - 43\epsilon^3 + 17\epsilon^2 - 4\epsilon)^2 (1 + 30\epsilon^4 - 18\epsilon^3 - 23\epsilon^2 + 11\epsilon)^2 \right]
 \end{aligned} \tag{37}$$

For 4-bit quantizer, $K = 4$

$$\begin{aligned}
 D_{(\text{CNC})}^{K=4} = & (114,213,888,000,000\epsilon^{42} - 1,033,673,011,200,000\epsilon^{41} \\
 & +3,285,980,835,840,000\epsilon^{40} - 427,035,193,344,000\epsilon^{39} \\
 & -29,615,491,894,579,200\epsilon^{38} + 103,980,817,090,897,920\epsilon^{37} \\
 & -164,725,822,832,887,296\epsilon^{36} + 34,515,810,866,297,088\epsilon^{35} \\
 & +472,361,930,923,112,064\epsilon^{34} - 1,315,712,396,208,910,656\epsilon^{33} \\
 & +2,125,947,487,263,270,816\epsilon^{32} - 2,411,219,122,612,440,912\epsilon^{31} \\
 & +1,921,993,430,825,213,592\epsilon^{30} - 851,450,424,470,925,900\epsilon^{29} \\
 & -285,553,775,320,155,900\epsilon^{28} + 988,348,010,466,804,588\epsilon^{27} \\
 & -1,071,182,372,882,329,488\epsilon^{26} + 716,211,751,000,262,427\epsilon^{25} \\
 & -268,380,717,446,800,437\epsilon^{24} - 24,693,601,109,397,063\epsilon^{23} \\
 & +117,058,813,839,308,184\epsilon^{22} - 91,679,254,830,762,285\epsilon^{21} \\
 & +40,953,822,541,585,406\epsilon^{20} - 8,513,469,892,437,467\epsilon^{19} \\
 & -2,817,431,401,552,608\epsilon^{18} + 3,612,067,778,674,335\epsilon^{17} \\
 & -1,903,032,054,185,672\epsilon^{16} + 647,855,202,371,522\epsilon^{15} \\
 & -132,028,998,320,989\epsilon^{14} - 1,029,960,719,786\epsilon^{13} \\
 & +13,574,222,503,848\epsilon^{12} - 6,451,548,073,534\epsilon^{11} \\
 & +1,919,139,169,130\epsilon^{10} - 412,300,519,087\epsilon^9 \\
 & +64,933,542,813\epsilon^8 - 7,683,962,162\epsilon^7 + 1,043,876,171\epsilon^6 \\
 & -249,828,052\epsilon^5 + 40,140,886\epsilon^4 + 201,576\epsilon^3 \\
 & -467,304\epsilon^2 - 48,104\epsilon - 208) \\
 & / \left[-12,288(1 + \epsilon^3 + \epsilon^2 - 2\epsilon)^2 (-2 + 30\epsilon^4 - 21\epsilon^3 + 2\epsilon^2 + 3\epsilon)^2 \right. \\
 & \left. (2 + 30\epsilon^4 - 43\epsilon^3 + 17\epsilon^2 - 4\epsilon)^2 (1 + 30\epsilon^4 - 18\epsilon^3 - 23\epsilon^2 + 11\epsilon)^2 \right]
 \end{aligned} \tag{36}$$

For 4-bit quantizer, $K = 5$

$$\begin{aligned}
 D_{(\text{CNC})}^{K=5} = & (208 + 48,104\epsilon + 533,640\epsilon^2 + 1,165,656\epsilon^3 \\
 & -58,458,982\epsilon^4 + 281,191,396\epsilon^5 \\
 & -1,006,697,867\epsilon^6 + 15,450,699,362\epsilon^7 \\
 & -19,1847,469,599\epsilon^8 + 1,488,428,462,173\epsilon^9 \\
 & -8,189,718,621,524\epsilon^{10} + 33,774,417,209,110\epsilon^{11} \\
 & -103,453,615,005,204\epsilon^{12} + 207,719,199,032,498\epsilon^{13} \\
 & -66,906,562,649,267\epsilon^{14} - 1,513,988,103,500,822\epsilon^{15} \\
 & +7,587,789,874,788,170\epsilon^{16} - 22,537,124,796,414,813\epsilon^{17} \\
 & +44,795,080,053,420,036\epsilon^{18} - 45,411,147,473,877,727\epsilon^{19} \\
 & -64,724,051,540,787,962\epsilon^{20} + 441,781,898,032,166,091\epsilon^{21} \\
 & -1,209,986,072,300,463,732\epsilon^{22} + 2,194,513,539,257,525,709\epsilon^{23} \\
 & -2,581,081,879,055,831,871\epsilon^{24} + 852,406,151,910,028,791\epsilon^{25} \\
 & +4,532,585,756,743,770,546\epsilon^{26} - 13,587,290,869,775,967,948\epsilon^{27} \\
 & +23,129,883,155,269,680,864\epsilon^{28} - 26,732,149,422,180,934,236\epsilon^{29} \\
 & +17,882,765,153,323,206,432\epsilon^{30} + 4,450,563,097,100,665,344\epsilon^{31} \\
 & -32,275,487,629,836,383,808\epsilon^{32} + 51,938,010,511,486,921,344\epsilon^{33} \\
 & -53,591,598,103,908,331,776\epsilon^{34} + 38,728,483,984,293,107,712\epsilon^{35}
 \end{aligned}$$

5 Distortion analysis

In this section, several figures are plotted to analyze performance of the standard [7] and proposed modified EOUQ. Distortion of the standard EOUQ and the lower bound for distortion of the modified EOUQ with CNC index assignment for 3-bit and 4-bit are shown in Figures 5 and 6, respectively, which are plotted in imitation of Figure six in [7] in order to clearly display the difference between those. According to Figures 5 and 6, when the crossover probability ϵ is greater than 0.01, the performance of our proposed EOUQ is better than that of the standard EOUQ with CNC index assignment, which shows the benefit obtained from a good erasure correcting code. Additionally, when $\epsilon < 0.01$, the proposed one is worse than the standard one. This is because the modified EOUQ would increase the quantization error due to the fixed empty cells for all ϵ in (18) and (19). But this problem can be solved by switching work mode in encoders and decoders. As mentioned before, the difference between the standard and proposed EOUQ is that an erasure correcting code into encoders and the corresponding decode scheme into decoders are appended. So, in a practical system, we can initially give a threshold value $\hat{\epsilon}_n$ for ϵ , for example, $\hat{\epsilon}_n = 0.01$ for 3-bit and 4-bit quantizers. When $\epsilon < \hat{\epsilon}_n$, the standard quantizer is adopted. Then, when encoders and decoders realize $\epsilon \geq \hat{\epsilon}_n$, an erasure correcting code is appended.

The X axis (which represents crossover probability) of Figure 5 is numbered from 0 to 0.5. However, in communication systems nowadays, the crossover probability seldom happens to reach 0.5. So, it is a preferable way to clearly display the performance that the logarithmic axis numbered from 10^{-3} to 10^{-1} should be chosen as X axis. Then, the proposed modified EOUQ appending a SPC block with $K = 3, 4, 5$ in MSE achieved by CNC index assignment for rate $n = 3$ and $n = 4$ is shown in Figures 7 and 8, respectively. As shown in Figures 7 and 8, when crossover probability is less than 10^{-2} , all lines are almost overlapped, that is to say that most of empty cell indices are recovered by SPC block decoders. However, when crossover probability is greater than 10^{-2} , the curve of our proposed quantizer using SPC block code gradually deviates from the lower bound, because

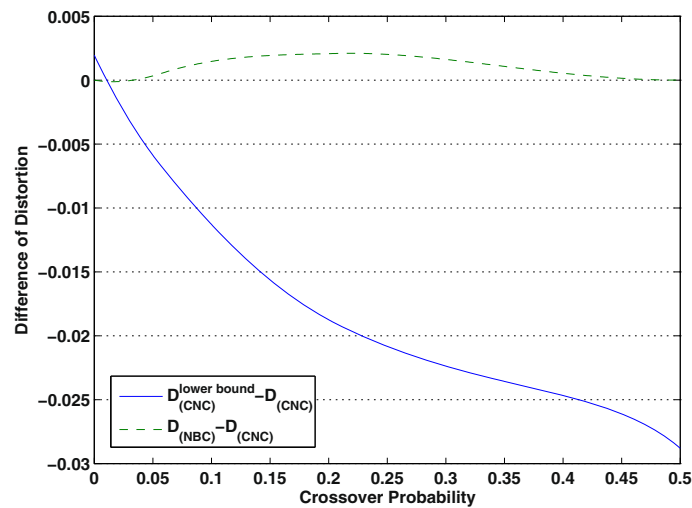


Figure 5 The difference between EOQ and proposed modified EOQ in MSE achieved by CNC index assignment for rate $n = 3$.

the SPC block decoder does not have enough capability of correcting all erasures so that a few of empty cell indices cannot be recovered according to Theorem 6. This is a performance penalty incurred by the sub-optimal erasure correcting code used in this paper. If another well-designed SPC code that can increase recovery probability in Theorem 6, for example SPC product code, or better erasure correcting code is adopted, the performance will approach the lower bound. But it is important that when crossover probability is numbered from 10^{-2} to 10^{-1} , the performance of proposed quantizers with CNC index assignment using SPC block code is still

better than that of standard quantizers with CNC index assignment.

6 Conclusion

In this paper, the lower bound for a modified uniform decoder and channel-optimized encoder quantizer with CNC index assignment is proposed. After appending the SPC block code and its decoding algorithm into encoders and decoders, respectively, the performance of this modified quantizer approaches the lower bound and is better than that of the standard quantizer when crossover error probability ϵ is greater than a threshold $\hat{\epsilon}_n$.

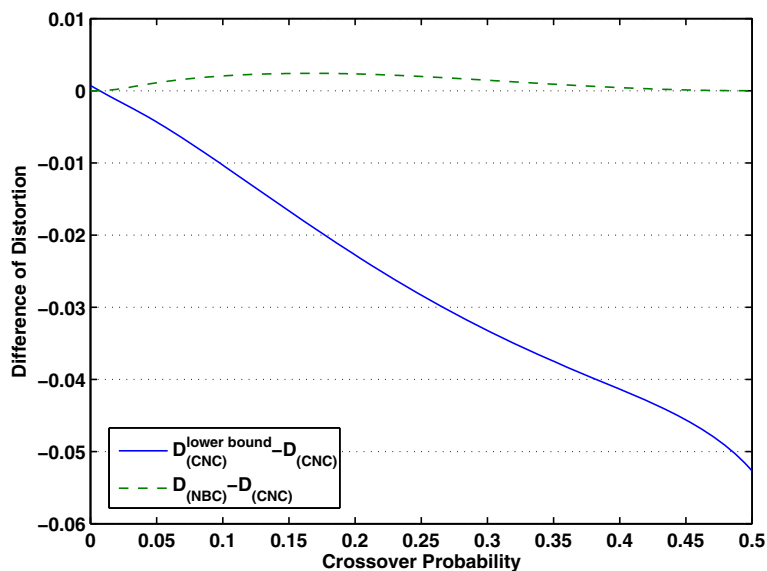


Figure 6 The difference between EOQ and proposed modified EOQ in MSE achieved by CNC index assignment for rate $n = 4$.

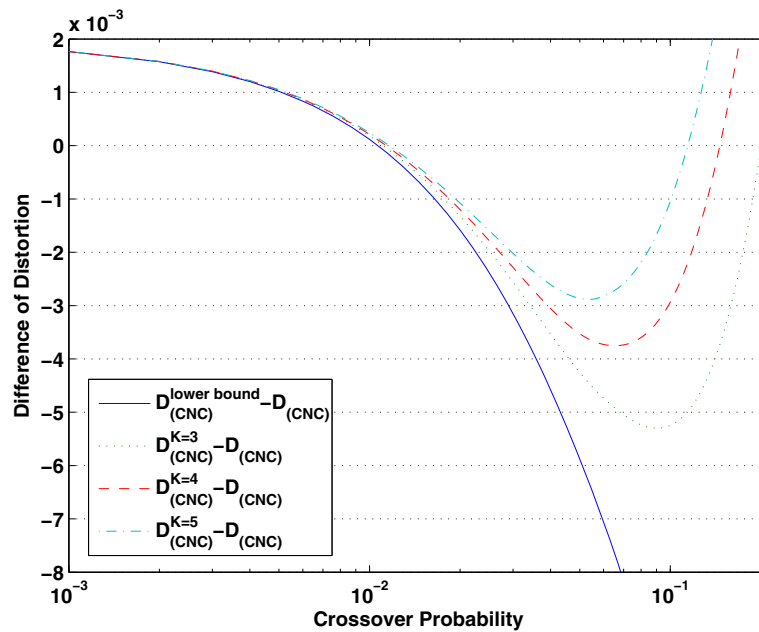


Figure 7 The MSE difference between original and modified EOUC appending a SPC block with $K = 3, 4, 5$ for rate $n = 3$.

Appendix

Proof of Lemma 1

Proof. For any integer $i, k \in [0, 2^n - 1]$ but $i \neq k$, the inequality in (14) can be rewritten as

$$\begin{aligned} & \sum_{j \in E^c} (x^2 - 2xy_n(j) + y_n^2(j)) p_n(\pi_n(j) | \pi_n(i)) \\ & + \sum_{j \in E} (x^2 - 2xy_n(i) + y_n^2(i)) p_n(\pi_n(j) | \pi_n(i)) \\ \leq & \sum_{j \in E^c} (x^2 - 2xy_n(j) + y_n^2(j)) p_n(\pi_n(j) | \pi_n(k)) \\ & + \sum_{j \in E} (x^2 - 2xy_n(k) + y_n^2(k)) p_n(\pi_n(j) | \pi_n(k)). \end{aligned}$$

Merging like terms gives

$$\begin{aligned} & x^2 \sum_j p_n(\pi_n(j) | \pi_n(i)) + \sum_{j \in E^c} [-2xy_n(j) + y_n^2(j)] p_n(\pi_n(j) | \pi_n(i)) \\ & + \sum_{j \in E} [-2xy_n(i) + y_n^2(i)] p_n(\pi_n(j) | \pi_n(i)) \\ \leq & x^2 \sum_j p_n(\pi_n(j) | \pi_n(k)) + \sum_{j \in E^c} [-2xy_n(j) + y_n^2(j)] p_n(\pi_n(j) | \pi_n(k)) \\ & + \sum_{j \in E} [-2xy_n(k) + y_n^2(k)] p_n(\pi_n(j) | \pi_n(k)). \end{aligned}$$

Since π_n is bijective and $\sum_j p_n(j|i) = 1, \forall j$, cancellation of terms gives

$$\begin{aligned} & \sum_{j \in E^c} 2xy_n(j) [p_n(\pi_n(j) | \pi_n(i)) - p_n(\pi_n(j) | \pi_n(k))] \\ & + \sum_{j \in E} 2x [y_n(i) p_n(\pi_n(j) | \pi_n(i)) - y_n(k) p_n(\pi_n(j) | \pi_n(k))] \\ \geq & \sum_{j \in E^c} y_n^2(j) [p_n(\pi_n(j) | \pi_n(i)) - p_n(\pi_n(j) | \pi_n(k))] \\ & + \sum_{j \in E} [y_n^2(i) p_n(\pi_n(j) | \pi_n(i)) - y_n^2(k) p_n(\pi_n(j) | \pi_n(k))]. \end{aligned}$$

After canceling terms and substituting (2), the left side of the inequality gives

$$\begin{aligned} & 2^{-n} x \sum_{j \in E^c} (2j + 1) [p_n(\pi_n(j) | \pi_n(i)) - p_n(\pi_n(j) | \pi_n(k))] \\ & + 2^{-n} x \sum_{j \in E} [(2i + 1) p_n(\pi_n(j) | \pi_n(i)) \\ & - (2k + 1) p_n(\pi_n(j) | \pi_n(k))] \\ = & 2^{-n+1} x \left(\sum_{j \in E^c} j [p_n(\pi_n(j) | \pi_n(i)) - p_n(\pi_n(j) | \pi_n(k))] \right. \\ & \left. + \sum_{j \in E} [i \cdot p_n(\pi_n(j) | \pi_n(i)) - k \cdot p_n(\pi_n(j) | \pi_n(k))] \right) \\ & + 2^{-n} x \sum_j [p_n(\pi_n(j) | \pi_n(i)) - p_n(\pi_n(j) | \pi_n(k))] \\ = & 2^{-n+1} \alpha'_n(i, k) x \end{aligned}$$

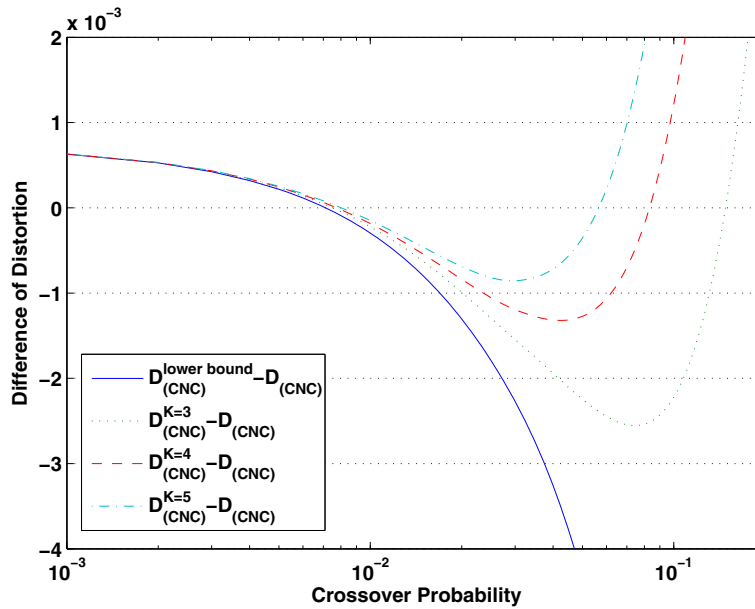


Figure 8 The MSE difference between original and modified EOUQ appending a SPC block with $K = 3, 4, 5$ for rate $n = 4$.

and similarly, the right side of the inequality gives

$$\begin{aligned}
 & 2^{-2n} x \sum_{j \in E^c} \left(j^2 + j + \frac{1}{4} \right) [p_n(\pi_n(j) | \pi_n(i)) - p_n(\pi_n(j) | \pi_n(k))] \\
 & + 2^{-n} x \sum_{j \in E} \left[\left(i^2 + i + \frac{1}{4} \right) p_n(\pi_n(j) | \pi_n(i)) \right. \\
 & \quad \left. - \left(k^2 + k + \frac{1}{4} \right) p_n(\pi_n(j) | \pi_n(k)) \right] \\
 & = 2^{-n+1} \beta'_n(i, k) + 2^{-2n-2} \sum_j [p_n(\pi_n(j) | \pi_n(i)) \\
 & \quad - p_n(\pi_n(j) | \pi_n(k))] \\
 & = 2^{-n+1} \beta'_n(i, k).
 \end{aligned}$$

Therefore, the inequality in (14) can be rewritten as

$$\alpha'_n(i, k) x \geq \beta'_n(i, k).$$

□

The simplification of $\beta'_n(i, k)$

$$\begin{aligned}
 \beta'_n(i, k) = 2^{-n-1} & \left(\alpha'_n(i, k) + \sum_{j \in E^c} j^2 [p_n(\pi_n(j) | \pi_n(i)) \right. \\
 & \quad \left. - p_n(\pi_n(j) | \pi_n(k))] + \sum_{j \in E} [i^2 \cdot p_n(\pi_n(j) | \pi_n(i)) \right. \\
 & \quad \left. - k^2 \cdot p_n(\pi_n(j) | \pi_n(k))] \right).
 \end{aligned}$$

Case 1. $i = 0$

a. $1 \leq k \leq 2^n - 1, k$ odd; $2^n - 1 \leq k \leq 2^n - 2, k$ even

$$\begin{aligned}
 \beta'_n(0, k) = 2^{-n-1} & \left\{ \alpha'_n(0, k) + \left[\frac{2^{2n+2}}{3} - 6 \cdot 2^n + \frac{32}{3} \right. \right. \\
 & \quad \left. \left. + 4k^2 - (2^{n+2} - 4)k \right] \epsilon^3 + [2^n - 8 \right. \\
 & \quad \left. - 8k^2 + (3 \cdot 2^{n+1} - 6)k \right] \epsilon^2 \\
 & \quad \left. + \left[\frac{-2^{2n} + 4}{3} + 2^n + 4k^2 \right. \right. \\
 & \quad \left. \left. + (-2^{n+1} + 2)k \right] \epsilon - k^2 \right. \\
 & \quad \left. + (2^n - 1)^2 \epsilon^{n-1} - (2^n - 1)^2 \epsilon^n \right. \\
 & \quad \left. + k^2 p_n(0 | \pi_n(k)) - [(2^n - 1)^2 - k^2] \right. \\
 & \quad \left. \times p_n(2^n - 2 | \pi_n(k)) \right\}
 \end{aligned}$$

b. $k = 2^n - 1$

$$\begin{aligned}
 \beta'_n(0, 2^n - 1) = 2^{-n-1} & \left\{ \alpha'_n(0, 2^n - 1) - 2(2^n - 1)^2 \epsilon^n \right. \\
 & \quad \left. + 2(2^n - 1)^2 \epsilon^{n-1} + (2 \cdot 2^{2n} - 8 \cdot 2^n + 6) \epsilon^2 \right. \\
 & \quad \left. + (2^{n+1} - 2) \epsilon - (2^n - 1)^2 \right\}
 \end{aligned}$$

Case 2. $1 \leq i \leq 2^n - 1, i$ odd; $2^n - 1 \leq i \leq 2^n - 2, i$ even

a. $k = 0$

$$\beta'_n(i, 0) = 2^{-n-1} \left\{ \alpha'_n(i, 0) - \left[\frac{2^{2n+2}}{3} - 6 \cdot 2^n + \frac{32}{3} + 4i^2 - (2^{n+2} - 4) i \right] \epsilon^3 - [2^n - 8 - 8i^2 + (3 \cdot 2^{n+1} - 6) i] \epsilon^2 - \left[\frac{-2^{2n} + 4}{3} + 2^n + 4i^2 + (-2^{n+1} + 2) i \right] \epsilon + i^2 - (2^n - 1)^2 \times \epsilon^{n-1} + (2^n - 1)^2 \epsilon^n - i^2 p_n(0|\pi_n(i)) + [(2^n - 1)^2 - i^2] p_n(2^n - 2|\pi_n(i)) \right\}$$

b. $1 \leq k \leq 2^n - 1, k$ odd; $2^n - 1 \leq k \leq 2^n - 2, k$ even; $k \neq i$

$$\beta'_n(i, k) = 2^{-n-1} \left\{ \alpha'_n(i, k) + [-4i^2 + 4k^2 + (2^{n+2} - 4) \times (i - k)] \epsilon^3 + [8i^2 - 8k^2 + (-3 \cdot 2^{n+1} + 6) (i - k)] \epsilon^2 + [-4i^2 + 4k^2 + (2^{n+1} - 2) (i - k)] \times \epsilon + i^2 - k^2 - i^2 p_n(0|\pi_n(i)) + [(2^n - 1)^2 - i^2] p_n(2^n - 2|\pi_n(i)) + k^2 p_n(0|\pi_n(k)) - [(2^n - 1)^2 - k^2] \times p_n(2^n - 2|\pi_n(k)) \right\}$$

c. $k = 2^n - 1$

$$\beta'_n(i, 2^n - 1) = 2^{-n-1} \left\{ \alpha'_n(i, 2^n - 1) - \left[\frac{2^{2n+2}}{3} - 6 \cdot 2^n + \frac{32}{3} + 4i^2 - (2^{n+2} - 4) i \right] \epsilon^3 + [2^{2n+1} - 9 \cdot 2^n + 14 + 8i^2 + (-3 \cdot 2^{n+1} + 6) i] \epsilon^2 + \left[\frac{2^{2n} - 10}{3} + 2^n - 4i^2 + (2^{n+1} - 2) i \right] \times \epsilon + i^2 - (2^n - 1)^2 + (2^n - 1)^2 \epsilon^{n-1} - (2^n - 1)^2 \times \epsilon^n - i^2 p_n(0|\pi_n(i)) + [(2^n - 1)^2 - i^2] p_n(2^n - 2|\pi_n(i)) \right\}$$

$$\sum_{\substack{j=2 \\ j \text{ even}}}^{2^n-4} j p_n(j|i) = \begin{cases} (2^n - 2) \epsilon^n - (2^n - 2) \epsilon^{n-1} - (2^n - 2) \epsilon^2 + (2^n - 2) \epsilon & i = 0 \\ (2^n - 2) \epsilon^2 + (i - 1) (1 - 2\epsilon) \epsilon - (2^n - 2) p_n(2^n - 2|i) & 1 \leq i \leq 2^n - 1 \text{ and } i \text{ is odd} \\ (2^n - 2) (1 - \epsilon)^2 - (2^n - 2) (1 - \epsilon)^n & i = 2^n - 2 \end{cases} \quad (40)$$

Case 3. $i = 2^n - 1$

a. $k = 0$

$$\beta'_n(2^n - 1, 0) = 2^{-n-1} \left\{ \alpha'_n(2^n - 1, 0) + 2(2^n - 1)^2 \epsilon^n - 2(2^n - 1)^2 \epsilon^{n-1} - (2 \cdot 2^{2n} - 8 \cdot 2^n + 6) \epsilon^2 - (2^{n+1} - 2) \epsilon + (2^n - 1)^2 \right\}$$

b. $1 \leq k \leq 2^n - 1, k$ odd; $2^n - 1 \leq k \leq 2^n - 2, k$ even

$$\beta'_n(2^n - 1, k) = 2^{-n-1} \left\{ \alpha'_n(2^n - 1, k) + \left[\frac{2^{2n+2}}{3} - 6 \cdot 2^n + \frac{32}{3} + 4k^2 - (2^{n+2} - 4) k \right] \epsilon^3 - [2^{2n+1} - 9 \cdot 2^n + 14 + 8k^2 + (-3 \cdot 2^{n+1} + 6) k] \epsilon^2 - \left[\frac{2^{2n} - 10}{3} + 2^n - 4k^2 + (2^{n+1} - 2) k \right] \times \epsilon - k^2 + (2^n - 1)^2 - (2^n - 1)^2 \epsilon^{n-1} + (2^n - 1)^2 \epsilon^n + k^2 p_n(0|\pi_n(k)) - [(2^n - 1)^2 - k^2] p_n(2^n - 2|\pi_n(k)) \right\}$$

Proof of Corollary 1

In order to prove Corollary 1, the following corollaries will be firstly evidenced.

Corollary 3. *If $0 \leq i \leq 2^n - 1$, then*

$$\sum_{j=0}^{2^n-1} j p_n(j|i) = (2^n - 1) \epsilon + i (1 - 2\epsilon), \quad (38)$$

and

$$\sum_{j=0}^{2^n-1} j p_n(\pi_n^{(CNC)}(j) | \pi_n^{(CNC)}(i)) = \begin{cases} (2^n - 1) \epsilon + (1 - 2\epsilon) (i + \epsilon), & \text{for } i \text{ even} \\ (2^n - 1) \epsilon + (1 - 2\epsilon) (i - \epsilon), & \text{for } i \text{ odd.} \end{cases} \quad (39)$$

Corollary 3 was proved in [7].

Corollary 4.

Proof.

$$\begin{aligned} \sum_{\substack{j=2 \\ j \text{ even}}}^{2^n-4} j p_n(j|0) &= \sum_{\substack{j=2 \\ j \text{ even}}}^{2^n-2} j p_n(j|0) - (2^n - 2) p_n(2^n - 2|0) \\ &= 2(1 - \epsilon) \sum_{j=0}^{2^{n-1}-1} j p_{n-1}(j|0) - (2^n - 2)(1 - \epsilon) \epsilon^{n-1} \\ &= (2^n - 2) \epsilon^n - (2^n - 2) \epsilon^{n-1} - (2^n - 2) \epsilon^2 + (2^n - 2) \epsilon \end{aligned}$$

$$\begin{aligned} \sum_{\substack{j=2 \\ j \text{ even}}}^{2^n-4} j p_n(j|2^n - 2) &= \sum_{\substack{j=2 \\ j \text{ even}}}^{2^n-2} j p_n(j|2^n - 2) \\ &\quad - (2^n - 2) p_n(2^n - 2|2^n - 2) \\ &= 2(1 - \epsilon) \sum_{j=0}^{2^{n-1}-1} j p_{n-1}(j|2^{n-1} - 1) \\ &\quad - (2^n - 2)(1 - \epsilon)^n \\ &= (2^n - 2)(1 - \epsilon)^2 - (2^n - 2)(1 - \epsilon)^n \end{aligned}$$

If $1 \leq i \leq 2^n - 1$ and i is odd, then

$$\begin{aligned} \sum_{\substack{j=2 \\ j \text{ even}}}^{2^n-4} j p_n(j|i) &= \sum_{\substack{j=2 \\ j \text{ even}}}^{2^n-2} j p_n(j|i) - (2^n - 2) p_n(2^n - 2|i) \\ &= 2\epsilon \sum_{j=0}^{2^{n-1}-1} j p_{n-1}\left(j \left| \frac{i-1}{2} \right.\right) - (2^n - 2) p_n(2^n - 2|i) \\ &= (2^n - 2) \epsilon^2 + (i - 1)(1 - 2\epsilon) \epsilon \\ &\quad - (2^n - 2) p_n(2^n - 2|i). \end{aligned}$$

□

Denote $a_n(i) \triangleq \sum_{j \in E^c} j p_n(\pi_n^{(\text{CNC})}(j) | \pi_n^{(\text{CNC})}(i)) + i \sum_{j \in E} p_n(\pi_n^{(\text{CNC})}(j) | \pi_n^{(\text{CNC})}(i))$.

Corollary 5.

$$a_n(i) = \begin{cases} \epsilon + (2^n - 3) \epsilon^2 + (2^n - 1) \epsilon^{n-1} - (2^n - 1) \epsilon^n & i = 0 \\ (-2^n + 2i + 1) \epsilon^2 + (2^n - 2i - 1) \epsilon + i \\ \quad + (2^n - i - 1) p_n(2^n - 2|i) - i p_n(0|i) & 1 \leq i \leq 2^{n-1} - 1 \text{ and } i \text{ is odd} \\ (-2^n + 2i + 1) \epsilon^2 + (2^n - 2i - 1) \epsilon + i \\ \quad + (2^n - i - 1) p_n(2^n - 2|i + 1) - i p_n(0|i + 1) & 2^{n-1} \leq i \leq 2^n - 2 \text{ and } i \text{ is even} \\ (2^n - 1) \epsilon^n - (2^n - 1) \epsilon^{n-1} + (-2^n + 3) \epsilon^2 - \epsilon + 2^n - 1 & i = 2^n - 1 \end{cases} \quad (41)$$

Proof.

$$\begin{aligned} &\sum_{j \in E^c} j p_n(\pi_n^{(\text{CNC})}(j) | \pi_n^{(\text{CNC})}(i)) + i \sum_{j \in E} p_n(\pi_n^{(\text{CNC})}(j) | \pi_n^{(\text{CNC})}(i)) \\ &= \sum_{j=0}^{2^{n-1}-1} j p_n(\pi_n^{(\text{CNC})}(j) | \pi_n^{(\text{CNC})}(i)) - \sum_{\substack{j=2 \\ j \text{ even}}}^{2^{n-1}-2} j p_n(j | \pi_n^{(\text{CNC})}(i)) \\ &\quad - \sum_{\substack{j=2^{n-1}+1 \\ j \text{ odd}}}^{2^n-3} j p_n(j - 1 | \pi_n^{(\text{CNC})}(i)) \\ &\quad + i \left[\sum_{\substack{j=2 \\ j \text{ even}}}^{2^{n-1}-2} p_n(j | \pi_n^{(\text{CNC})}(i)) + \sum_{\substack{j=2^{n-1}+1 \\ j \text{ odd}}}^{2^n-3} p_n(j - 1 | \pi_n^{(\text{CNC})}(i)) \right] \\ &= \sum_{j=0}^{2^{n-1}-1} j p_n(\pi_n^{(\text{CNC})}(j) | \pi_n^{(\text{CNC})}(i)) - \sum_{\substack{j=2 \\ j \text{ even}}}^{2^{n-1}-2} j p_n(j | \pi_n^{(\text{CNC})}(i)) \\ &\quad - \sum_{\substack{j=2^{n-1} \\ j \text{ even}}}^{2^n-4} (j + 1) p_n(j | \pi_n^{(\text{CNC})}(i)) \\ &\quad + i \left[\sum_{\substack{j=2 \\ j \text{ even}}}^{2^{n-1}-2} p_n(j | \pi_n^{(\text{CNC})}(i)) + \sum_{\substack{j=2^{n-1} \\ j \text{ even}}}^{2^n-4} p_n(j | \pi_n^{(\text{CNC})}(i)) \right] \\ &= \sum_{j=0}^{2^{n-1}-1} j p_n(\pi_n^{(\text{CNC})}(j) | \pi_n^{(\text{CNC})}(i)) - \sum_{\substack{j=2 \\ j \text{ even}}}^{2^n-4} j p_n(j | \pi_n^{(\text{CNC})}(i)) \\ &\quad - \sum_{\substack{j=2^{n-1} \\ j \text{ even}}}^{2^n-4} p_n(j | \pi_n^{(\text{CNC})}(i)) + i \sum_{\substack{j=2 \\ j \text{ even}}}^{2^n-4} p_n(j | \pi_n^{(\text{CNC})}(i)) \end{aligned}$$

If $i = 0$, then

$$\begin{aligned}
 & \sum_{j \in E^c} j p_n(\pi_n^{(\text{CNC})}(j) | \pi_n^{(\text{CNC})}(i)) \\
 & + i \sum_{j \in E} p_n(\pi_n^{(\text{CNC})}(j) | \pi_n^{(\text{CNC})}(i)) \\
 & = \sum_{j=0}^{2^n-1} j p_n(\pi_n^{(\text{CNC})}(j) | 0) - \sum_{\substack{j=2 \\ j \text{ even}}}^{2^n-4} j p_n(j|0) - \sum_{\substack{j=2^{n-1} \\ j \text{ even}}}^{2^n-4} p_n(j|0) \\
 & = \sum_{j=0}^{2^n-1} j p_n(\pi_n^{(\text{CNC})}(j) | 0) - \sum_{\substack{j=2 \\ j \text{ even}}}^{2^n-4} j p_n(j|0) \\
 & \quad - \left[\epsilon (1 - \epsilon) \sum_{j=0}^{2^{n-2}-1} p_{n-2}(j|0) - p_n(2^n - 2|0) \right] \\
 & = \epsilon + (2^n - 3) \epsilon^2 + (2^n - 1) \epsilon^{n-1} - (2^n - 1) \epsilon^n.
 \end{aligned}$$

The proof of the case that $i = 2^n - 1$ is very similar to the above.

If $1 \leq i \leq 2^{n-1} - 1$ and i is odd, then

$$\begin{aligned}
 & \sum_{j \in E^c} j p_n(\pi_n^{(\text{CNC})}(j) | \pi_n^{(\text{CNC})}(i)) \\
 & + i \sum_{j \in E} p_n(\pi_n^{(\text{CNC})}(j) | \pi_n^{(\text{CNC})}(i)) \\
 & = \sum_{j=0}^{2^n-1} j p_n(\pi_n^{(\text{CNC})}(j) | \pi_n^{(\text{CNC})}(i)) - \sum_{\substack{j=2 \\ j \text{ even}}}^{2^n-4} j p_n(j|i) \\
 & \quad - \sum_{\substack{j=2^{n-1} \\ j \text{ even}}}^{2^n-4} p_n(j|i) + i \sum_{\substack{j=2 \\ j \text{ even}}}^{2^n-4} p_n(j|i) \\
 & = \sum_{j=0}^{2^n-1} j p_n(\pi_n^{(\text{CNC})}(j) | \pi_n^{(\text{CNC})}(i)) - \sum_{\substack{j=2 \\ j \text{ even}}}^{2^n-4} j p_n(j|i) - \sum_{\substack{j=2^{n-1} \\ j \text{ even}}}^{2^n-4} p_n(j|i)
 \end{aligned}$$

$$\sum_{\substack{j=2 \\ j \text{ even}}}^{2^n-4} j^2 p_n(j | \pi_n^{(\text{CNC})}(i)) = \begin{cases} (2^n - 2)^2 \epsilon^n - (2^n - 2)^2 \epsilon^{n-1} - \frac{8\epsilon^3}{3} (2^{n-1} - 1) (2^{n-1} - 2) \\ \quad + \frac{\epsilon^2}{3} (2^{2n+1} - 2^{2n} - 3 \cdot 2^{n+2} + 20) + \frac{4}{3} (4^{n-1} - 1) \epsilon & i = 0 \\ \left[\frac{8}{3} (2^{n-1} - 1) (2^{n-1} - 2) - (i - 1) (2^{n+2} - 8) + 4 (i - 1)^2 \right] \epsilon^3 \\ \quad + \left[\frac{4}{3} (4^{n-1} - 1) + (i - 1) (2^{n+1} - 4) - 4 (i - 1)^2 \right] \epsilon^2 \\ \quad + (i - 1)^2 \epsilon - (2^n - 2)^2 p_n(2^n - 2|i) & 1 \leq i \leq 2^n - 1 \text{ and } i \text{ odd} \\ 4(1 - \epsilon) \left[\frac{\epsilon}{3} (4^{n-1} - 1) + \frac{2\epsilon^2}{3} (2^{n-1} - 1) (2^{n-1} - 2) \right. \\ \quad \left. + (2^{n-1} - 1)^2 (1 - 2\epsilon) \right] - (2^n - 2)^2 (1 - \epsilon)^n & i = 2^n - 2 \end{cases} \quad (44)$$

$$\begin{aligned}
 & + i \left[\sum_{\substack{j=0 \\ j \text{ even}}}^{2^n-2} p_n(j|i) - p_n(0|i) - p_n(2^n - 2|i) \right] \\
 & = \sum_{j=0}^{2^n-1} j p_n(\pi_n^{(\text{CNC})}(j) | \pi_n^{(\text{CNC})}(i)) - \sum_{\substack{j=2 \\ j \text{ even}}}^{2^n-4} j p_n(j|i) - \sum_{\substack{j=2^{n-1} \\ j \text{ even}}}^{2^n-4} p_n(j|i) \\
 & + i \left[\epsilon \sum_{j=0}^{2^{n-1}-1} p_{n-1} \left(j \left| \frac{i-1}{2} \right. \right) - p_n(0|i) - p_n(2^n - 2|i) \right] \\
 & = (-2^n + 2i + 1) \epsilon^2 + (2^n - 2i - 1) \epsilon + i \\
 & \quad + (2^n - i - 1) p_n(2^n - 2|i) - i p_n(0|i).
 \end{aligned}$$

The proof of the case that $2^{n-1} \leq i \leq 2^n - 2$ and i is even is very similar to the above. \square

Clearly,

$$\alpha'_n(i, k) = a_n(i) - a_n(k). \quad (42)$$

After substituting (41) for $a_n(i)$ and $a_n(k)$ in (42) and merging like terms, $\alpha'_n(i, k)$ is simplified.

Proof of Corollary 2

In order to prove Corollary 2, the following corollaries will be firstly evidenced.

Corollary 6. *If $0 \leq i \leq 2^n - 1$, then*

$$\begin{aligned}
 \sum_{j=0}^{2^n-1} j^2 p_n(j|i) & = \frac{\epsilon}{3} (4^n - 1) + \frac{2\epsilon^2}{3} (2^n - 1) (2^n - 2) \\
 & \quad + 2i\epsilon (1 - 2\epsilon) (2^n - 1) + i^2 (1 - 2\epsilon)^2.
 \end{aligned} \quad (43)$$

Corollary 6 was proved in [7].

Corollary 7.

Proof. If $i = 0$, then

$$\begin{aligned} & \sum_{\substack{j=2 \\ j \text{ even}}} j^2 p_n(j|0) \\ &= 4(1-\epsilon) \left[\sum_{j=0}^{2^{n-1}-1} j^2 p_{n-1}(j|0) - (2^{n-1}-1)^2 p_{n-1}(2^{n-1}-1|0) \right] \\ &= (2^n-2)^2 \epsilon^n - (2^n-2)^2 \epsilon^{n-1} - \frac{8\epsilon^3}{3} (2^{n-1}-1)(2^{n-1}-2) \\ & \quad + \frac{\epsilon^2}{3} (2^{2n+1} - 2^{2n} - 3 \cdot 2^{n+2} + 20) + \frac{4}{3} (4^{n-1} - 1) \epsilon. \end{aligned}$$

The proof of $i = 2^n - 2$ case is similar to the above.

$$\begin{aligned} & \sum_{\substack{j=2 \\ j \text{ even}}}^{2^n-4} j^2 p_n(j|i) \\ &= 4\epsilon \left[\sum_{j=0}^{2^{n-1}-1} j^2 p_{n-1}\left(j \middle| \frac{i-1}{2}\right) \right] - (2^n-2)^2 p_n(2^n-2|i) \\ &= \left[\frac{8}{3} (2^{n-1}-1)(2^{n-1}-2) - (i-1)(2^{n+2}-8) + 4(i-1)^2 \right] \epsilon^3 \\ & \quad + \left[\frac{4}{3} (4^{n-1}-1) + (i-1)(2^{n+1}-4) - 4(i-1)^2 \right] \epsilon^2 \\ & \quad + (i-1)^2 \epsilon - (2^n-2)^2 p_n(2^n-2|i) \end{aligned}$$

Corollary 8.

$$\sum_{\substack{j=2^{n-1} \\ j \text{ even}}}^{2^n-4} j p_n(j|\pi_n^{(\text{CNC})}(i)) = \begin{cases} (2^n-2)\epsilon^n - (2^n-2)\epsilon^{n-1} + (-2^{n-1}+2)\epsilon^3 - 2\epsilon^2 + 2^{n-1}\epsilon & i = 0 \\ (2^{n-1}-2i)\epsilon^3 + (2^{n-1}+i-1)\epsilon^2 - (2^n-2)p_n(2^n-2|i) & 1 \leq i \leq 2^{n-1}-1 \text{ and } i \text{ odd} \\ (2i+2-3 \cdot 2^{n-1})\epsilon^3 + (-3i-2+3 \cdot 2^{n-1})\epsilon^2 + i\epsilon - (2^n-2)p_n(2^n-2|i+1) & 2^{n-1} \leq i \leq 2^n-2 \text{ and } i \text{ even} \\ -(2^n-2)(1-\epsilon)^n + (-2^{n-1}+2)\epsilon^3 + (2^{n+1}-6)\epsilon^2 + (-5 \cdot 2^{n-1}+6)\epsilon + 2^n-2 & i = 2^n-1 \end{cases} \quad (45)$$

Proof. If $i = 0$, then

$$\begin{aligned} & \sum_{\substack{j=2^{n-1} \\ j \text{ even}}}^{2^n-4} j p_n(j|\pi_n^{(\text{CNC})}(i)) = \sum_{\substack{j=2^{n-1} \\ j \text{ even}}}^{2^n-4} j p_n(j|0) \\ &= \sum_{\substack{j=0 \\ j \text{ even}}}^{2^{n-1}-2} (j+2^{n-1}) p_n(j+2^{n-1}|0) - (2^n-2) p_n(2^n-2|0) \\ &= \epsilon \sum_{\substack{j=0 \\ j \text{ even}}}^{2^{n-1}-2} (j+2^{n-1}) p_{n-1}(j|0) - (2^n-2)(1-\epsilon)\epsilon^{n-1} \\ &= 2\epsilon(1-\epsilon) \sum_{j=0}^{2^{n-2}-1} j p_{n-2}(j|0) + 2^{n-1}\epsilon(1-\epsilon) \\ & \quad \times \sum_{j=0}^{2^{n-2}-1} p_{n-2}(j|0) + (2^n-2)(\epsilon^n - \epsilon^{n-1}) \\ &= (2^n-2)\epsilon^n - (2^n-2)\epsilon^{n-1} + (-2^{n-1}+2)\epsilon^3 - 2\epsilon^2 + 2^{n-1}\epsilon. \end{aligned}$$

If $1 \leq i \leq 2^{n-1} - 1$ and i odd, then

$$\begin{aligned} & \sum_{\substack{j=2^{n-1} \\ j \text{ even}}}^{2^n-4} j p_n(j|\pi_n^{(\text{CNC})}(i)) = \sum_{\substack{j=2^{n-1} \\ j \text{ even}}}^{2^n-4} j p_n(j|i) \\ &= \epsilon \sum_{\substack{j=0 \\ j \text{ even}}}^{2^{n-1}-2} j p_{n-1}(j|i) + 2^{n-1}\epsilon \sum_{\substack{j=0 \\ j \text{ even}}}^{2^{n-1}-2} p_{n-1}(j|i) - (2^n-2) p_n(2^n-2|i) \\ &= 2\epsilon^2 \sum_{j=0}^{2^{n-2}-1} j p_{n-2}\left(j \middle| \frac{i-1}{2}\right) + 2^{n-1}\epsilon^2 \sum_{j=0}^{2^{n-2}-1} p_{n-2}\left(j \middle| \frac{i-1}{2}\right) \\ & \quad - (2^n-2) p_n(2^n-2|i) \\ &= (2^{n-1}-2i)\epsilon^3 + (2^{n-1}+i-1)\epsilon^2 - (2^n-2) p_n(2^n-2|i). \end{aligned}$$

The proof of the other two cases is similar to the above. \square

Denote $b(i) \triangleq \sum_{j \in E^c} j^2 p_n(\pi_n^{(\text{CNC})}(j) | \pi_n^{(\text{CNC})}(i)) + \square \sum_{j \in E} p_n(\pi_n^{(\text{CNC})}(j) | \pi_n^{(\text{CNC})}(i))$.

Corollary 9.

$$b_n(i) = \begin{cases} -(2^n - 1)^2 \epsilon^n + (2^n - 1)^2 \epsilon^{n-1} + \left(\frac{2^{2n+1}}{3} - 5 \cdot 2^n + \frac{28}{3}\right) \epsilon^3 \\ \quad + \left(\frac{2^{2n}}{3} + 2^n - \frac{19}{3}\right) \epsilon^2 + \epsilon & i = 0 \\ \left[-\frac{2^{2n+1}}{3} + 2^n - \frac{4}{3} - 4i^2 + (2^{n+2} - 4)i\right] \epsilon^3 \\ + \left[\frac{4^n+5}{3} + 8i^2 + (-3 \cdot 2^{n+1} + 6)i\right] \epsilon^2 + \left[\frac{2^{2n}-1}{3} - 2^n - 4i^2 + (2^{n+1} - 2)i\right] \epsilon \\ \quad + i^2 - i^2 p_n(0|i) + [(2^n - 1)^2 - i^2] p_{(2^n-2|i)} & 1 \leq i \leq 2^{n-1} - 1 \text{ and } i \text{ odd} \\ \left[-\frac{2^{2n+1}}{3} + 2^n - \frac{4}{3} - 4i^2 + (2^{n+2} - 4)i\right] \epsilon^3 \\ + \left[\frac{4^n+5}{3} + 8i^2 + (-3 \cdot 2^{n+1} + 6)i\right] \epsilon^2 + \left[\frac{2^{2n}-1}{3} - 2^n - 4i^2 + (2^{n+1} - 2)i\right] \epsilon \\ \quad + i^2 - i^2 p_n(0|i+1) + [(2^n - 1)^2 - i^2] p_{(2^n-2|i+1)} & 2^{n-1} \leq i \leq 2^n - 2 \text{ and } i \text{ even} \\ (2^n - 1)^2 \epsilon^n - (2^n - 1)^2 \epsilon^{n-1} + \left(\frac{2^{2n+1}+28}{3} - 5 \cdot 2^n\right) \epsilon^3 \\ \quad + \left(-\frac{5 \cdot 4^n+37}{3} + 9 \cdot 2^n\right) \epsilon + (-2^{n+1} + 3) \epsilon + (2^n - 1)^2 & i = 2^n - 1 \end{cases} \quad (46)$$

Proof.

$$\begin{aligned} & \sum_{j \in E^c} j^2 p_n(\pi_n^{(\text{CNC})}(j) | \pi_n^{(\text{CNC})}(i)) + i^2 \sum_{j \in E} p_n(\pi_n^{(\text{CNC})}(j) | \pi_n^{(\text{CNC})}(i)) \\ &= \sum_{j=0}^{2^n-1} j^2 p_n(\pi_n^{(\text{CNC})}(j) | \pi_n^{(\text{CNC})}(i)) - \sum_{j=2}^{2^{n-1}-2} j^2 p_n(\pi_n^{(\text{CNC})}(j) | \pi_n^{(\text{CNC})}(i)) \\ & \quad - \sum_{\substack{j=2^{n-1}+1 \\ j \text{ odd}}}^{2^n-3} j^2 p_n(\pi_n^{(\text{CNC})}(j) | \pi_n^{(\text{CNC})}(i)) \\ & \quad + i^2 \left[\sum_{j=2}^{2^{n-1}-2} p_n(\pi_n^{(\text{CNC})}(j) | \pi_n^{(\text{CNC})}(i)) \right. \\ & \quad \left. + \sum_{\substack{j=2^{n-1}+1 \\ j \text{ odd}}}^{2^n-3} p_n(\pi_n^{(\text{CNC})}(j) | \pi_n^{(\text{CNC})}(i)) \right] \\ &= \sum_{j=0}^{2^n-1} j^2 p_n(\pi_n^{(\text{CNC})}(j) | \pi_n^{(\text{CNC})}(i)) - \sum_{j=2}^{2^{n-1}-2} j^2 p_n(j | \pi_n^{(\text{CNC})}(i)) \\ & \quad - \sum_{j=2}^{2^n-4} (j+1)^2 p_n(j | \pi_n^{(\text{CNC})}(i)) \\ & \quad + i^2 \left[\sum_{j=2}^{2^{n-1}-2} p_n(j | \pi_n^{(\text{CNC})}(i)) + \sum_{j=2^{n-1}}^{2^n-4} p_n(j | \pi_n^{(\text{CNC})}(i)) \right] \\ &= \sum_{j=0}^{2^n-1} j^2 p_n(\pi_n^{(\text{CNC})}(j) | \pi_n^{(\text{CNC})}(i)) - \sum_{j=2}^{2^{n-1}-2} j^2 p_n(j | \pi_n^{(\text{CNC})}(i)) \\ & \quad - \sum_{j=2}^{2^n-4} (j+1)^2 p_n(j | \pi_n^{(\text{CNC})}(i)) \\ & \quad + i^2 \left[\sum_{j=2}^{2^{n-1}-2} p_n(j | \pi_n^{(\text{CNC})}(i)) + \sum_{j=2^{n-1}}^{2^n-4} p_n(j | \pi_n^{(\text{CNC})}(i)) \right] \\ &= \sum_{j=0}^{2^n-1} j^2 p_n(\pi_n^{(\text{CNC})}(j) | \pi_n^{(\text{CNC})}(i)) - \sum_{j=2}^{2^{n-1}-2} j^2 p_n(j | \pi_n^{(\text{CNC})}(i)) \\ & \quad - \sum_{j=2^{n-1}}^{2^n-4} (2j+1) p_n(j | \pi_n^{(\text{CNC})}(i)) \\ & \quad + i^2 \sum_{j=2}^{2^n-4} p_n(j | \pi_n^{(\text{CNC})}(i)) \\ &= \sum_{j=0}^{2^n-1} j^2 p_n(\pi_n^{(\text{CNC})}(j) | \pi_n^{(\text{CNC})}(i)) - \sum_{j=2}^{2^n-4} j^2 p_n(j | \pi_n^{(\text{CNC})}(i)) \\ & \quad - \sum_{j=2}^{2^n-4} (2j+1) p_n(j | \pi_n^{(\text{CNC})}(i)) \\ & \quad + i^2 \sum_{j=2}^{2^n-4} p_n(j | \pi_n^{(\text{CNC})}(i)) \end{aligned}$$

If $i = 0$, then

$$\begin{aligned} b_n(0) &= \sum_{j=0}^{2^n-1} j^2 p_n(\pi_n^{(\text{CNC})}(j) | 0) - \sum_{j=2}^{2^n-4} j^2 p_n(j | 0) \\ & \quad - \sum_{j=2^{n-1}}^{2^n-4} (2j+1) p_n(j | 0) \\ &= \sum_{j=0}^{2^n-1} j^2 p_n(\pi_n^{(\text{CNC})}(j) | 0) - \sum_{j=2}^{2^n-4} j^2 p_n(j | 0) - 2 \sum_{j=2^{n-1}}^{2^n-4} j p_n(j | 0) \\ & \quad - \left[\epsilon(1-\epsilon) \sum_{j=0}^{2^{n-2}-1} p_{n-2}(j | 0) - p_n(2^n - 2 | 0) \right] \\ &= -(2^n - 1)^2 \epsilon^n + (2^n - 1)^2 \epsilon^{n-1} + \left(\frac{2^{2n+1}}{3} - 5 \cdot 2^n + \frac{28}{3}\right) \epsilon^3 \\ & \quad + \left(\frac{2^{2n}}{3} + 2^n - \frac{19}{3}\right) \epsilon^2 + \epsilon. \end{aligned}$$

If $1 \leq i \leq 2^{n-1} - 1$ and i odd, then

$$\begin{aligned}
 b_n(i) &= \sum_{j=0}^{2^n-1} j^2 p_n(\pi_n^{(\text{CNC})}(j|i)) - \sum_{\substack{j=2 \\ j \text{ even}}}^{2^n-4} j^2 p_n(j|i) \\
 &\quad - \sum_{\substack{j=2^{n-1} \\ j \text{ even}}}^{2^n-4} (2j+1) p_n(j|i) + i^2 \sum_{\substack{j=2 \\ j \text{ even}}}^{2^n-4} p_n(j|i) \\
 &= \sum_{j=0}^{2^n-1} j^2 p_n(\pi_n^{(\text{CNC})}(j|i)) - \sum_{\substack{j=2 \\ j \text{ even}}}^{2^n-4} j^2 p_n(j|i) \\
 &\quad - 2 \sum_{\substack{j=2^{n-1} \\ j \text{ even}}}^{2^n-4} j p_n(j|i) - \left[\epsilon^2 \sum_{j=0}^{2^{n-2}-1} p_{n-2}\left(j \left| \frac{i-1}{2} \right.\right) \right. \\
 &\quad \left. - p_n(2^n-2|i) \right] + i^2 \sum_{\substack{j=2 \\ j \text{ even}}}^{2^n-4} p_n(j|i).
 \end{aligned}$$

The proof of the other two cases is similar to the above. \square

Clearly,

$$\beta'_n(i, k) = b_n(i) - b_n(k). \quad (47)$$

After substituting (46) for $b_n(i)$ and $b_n(k)$ in (47) and merging like terms, $\beta'_n(i, k)$ is simplified.

Competing interests

The authors declare that they have no competing interests.

Acknowledgements

This work was supported in part by the National Natural Science Foundation of China under Grant 81101127 and in part by the Shenzhen Basic Research Funds under Grant JCYJ20120615140419045.

Author details

¹Shenzhen Institutes of Advanced Technology, Chinese Academy of Sciences, Shenzhen, China. ²Huawei Technologies Co., Ltd., Shenzhen, China.

Received: 20 May 2013 Accepted: 26 May 2014

Published: 18 June 2014

References

1. A Gersho, R Gray, *Vector Quantization and Signal Compression* (Kluwer, Norwell, MA, 1991)
2. B Hochwald, K Zeger, Tradeoff between source and channel coding. *IEEE Trans. Inform. Theory.* **43**(5), 1412–1424 (1997)
3. S Matloub, T Weissman, Universal zero-delay joint source channel coding. *IEEE Trans. Inform. Theory.* **52**(12), 5240–5250 (2006)
4. H Wang, S Tsiftaris, A Katsaggelos, Joint source-channel coding for wireless object-based video communications utilizing data hiding. *IEEE Trans. Image Process.* **15**(8), 2158–2169 (2006)
5. D Qiao, Y Li, Y Zhang, Energy efficient video transmission over fast fading channels. *EURASIP J. Wireless Commun. Netw.* **2010**, 1–12 (2010)
6. Li Ye, M Reisslein, C Chakrabarti, Energy-efficient video transmission over a wireless link. *IEEE Trans. on Vehicular Technol.* **58**(3), 1229–1244 (2009)
7. B Farber, K Zeger, Quantizers with uniform decoders and channel-optimized encoders. *IEEE Trans. Inform. Theory.* **52**(2), 640–661 (2006)

8. B Farber, K Zeger, Quantizers with uniform encoders and channel optimized decoders. *IEEE Trans. Inform. Theory.* **50**(1), 62–77 (2004)
9. H Kumazawa, M Kasahara, T Namekawa, A construction of vector quantizers for noisy channels. *Electron. Eng. Jpn.* **67-B**(4), 39–47 (1984)
10. A Kurtenbach, P Wintz, Quantizing for noisy channels. *IEEE Trans. Commun. Technol.* **COM-17**(4), 291–302 (1969)
11. J Zheng, D Rao, Analysis of vector quantizers using transformed codebooks with application to feedback-based multiple antenna systems. *EURASIP J. Wireless Commun. Netw.* **2008**, 1–13 (2008)
12. J Dunham, RM Gray, Joint source and noisy channel trellis encoding. *IEEE Trans. Inform. Theory.* **IT-27**(4), 516–519 (1981)
13. J Ho, E-H Yang, Designing optimal multiresolution quantizers with error detecting codes. *IEEE Trans. Wireless Com.* **12**(7), 3588–3599 (2013)
14. R Hagen, P Hedelin, Robust vector quantization by a linear mapping of a block code. *IEEE Trans. Inform. Theory.* **45**(1), 200–218 (1999)
15. M Skoglund, On channel-constrained vector quantization and index assignment for discrete memoryless channels. *IEEE Trans. Inform. Theory.* **45**(6), 2615–2622 (1999)
16. TR Crippins, HM Horwitz, CJ Palermo, RV Palermo, Minimization of mean-square error for data transmitted via group codes. *IEEE Trans. Inform. Theory.* **IT-15**(1), 72–78 (1969)
17. SW McLaughlin, DL Neuhoff, JJ Ashley, Optimal binary index assignments for a class of equiprobable scalar and vector quantizers. *IEEE Trans. Inform. Theory.* **41**(6), 2031–2037 (1995)
18. P Knagenhjelm, E Agrell, The Hadamard transform - a tool for index assignment. *IEEE Trans. Inform. Theory.* **42**(4), 1139–1151 (1996)
19. A Mehes, K Zeger, Binary lattice vector quantization with linear block codes and affine index assignments. *IEEE Trans. Inform. Theory.* **44**(1), 79–94 (1998)
20. A Mehes, K Zeger, Randomly chosen index assignments are asymptotically bad for uniform sources. *IEEE Trans. Inform. Theory.* **45**(2), 788–794 (1999)
21. X Yu, H Wang, E-H Yang, Design and analysis of optimal noisy channel quantization with random index assignment. *IEEE Trans. Inform. Theory.* **56**(11), 5796–5804 (2010)
22. N Farvardin, VA Vaishampayan, Optimal quantizer design for noisy channel: an approach to combined source-channel coding. *IEEE Trans. Inform. Theory.* **IT-22**(6), 827–838 (1987)
23. N Farvardin, VA Vaishampayan, On the performance and complexity of channel-optimized vector quantizers. *IEEE Trans. Inform. Theory.* **37**(1), 155–160 (1991)
24. DM Rankin, TA Gulliver, Single parity check product codes. *IEEE Trans. Commun.* **49**(8), 1354–1362 (2001)

doi:10.1186/1687-1499-2014-99

Cite this article as: Qiao and Li: Channel-optimized scalar quantizers with erasure correcting codes. *EURASIP Journal on Wireless Communications and Networking* 2014 **2014**:99.

Submit your manuscript to a SpringerOpen journal and benefit from:

- Convenient online submission
- Rigorous peer review
- Immediate publication on acceptance
- Open access: articles freely available online
- High visibility within the field
- Retaining the copyright to your article

Submit your next manuscript at ► springeropen.com