Finite-Geometry Low-Density Parity-Check Codes for Channels with Stuck-at Defects

Guang-Chong Zhu, Wei Yu and Frank R. Kschischang
Department of Electrical and Computer Engineering
University of Toronto
10 King's College Road
Toronto, ON, Canada, M5S 3G4
Email: {zhugc, weiyu, frank}@comm.utoronto.ca

ABSTRACT

We investigate the use of finite-geometry low-density parity-check (FG-LDPC) codes for channels with stuck-at defects. Such a channel is corrupted by a stuck-at defect pattern in addition to the usual channel-induced noise. When the defect pattern is known to the encoder but not to the decoder, the capacity of the channel is the same as if the defect pattern were also revealed to the decoder. Capacity-achieving codes for such channels require a good quantization code embedded inside a good error-correcting code. The main idea of this paper is that such an embedding may be realized by taking advantage of the cyclic or quasi-cyclic structure of FG-LDPC codes, which allows a quantization codes with low trellis complexity to be constructed. Combining this with the good error-correcting capability of low-density parity-check codes, we demonstrate that FG-LDPC codes offers good performance on channels with stuck-at defects.

1. Introduction

This paper investigates the coding problem for channels with stuck-at defects. Such channels are a realistic model for computer memory or CD-ROMs in which defect pattern is known at the encoder but not at the decoder. The coding problem for these channels has many similarities with the well-known dirty-paper coding problem [1] and is therefore of great theoretical interest.

The model of channels with stuck-at defects was first proposed by Kusnetsov and Tsybakov [2], and then was mainly studied in the Russian literature (e.g., [3], [4], etc.). In [5], the capacity of such channels was derived by Heegard and El Gamal; Heegard then proposed the idea of "partitioned linear block codes" [6], which was the first attempt on explicit construction of codes for such channels aiming at approaching the capacity. Another related work is by Borden and Vinck [7], who derived bounds on the resulting error probability when block and convolutional codes are used for such channels.

Low-density parity-check (LDPC) codes [8] are currently the subject of much research due to their excellent error-correcting performance under iterative decoding. In this work, we investigate the design of LDPC codes for channels with stuck-at defects. In particular, we take advantage of the cyclic structure of a special family of LDPC codes, namely finite-geometry LDPC (FG-LDPC) codes [9], and show that a low-complexity quantization code may be embedded in the LDPC codes to provide good overall performance for channels with stuck-at defects.

2. Channel Model

Imagine that a CD-ROM has some scratches, or a computer memory has a fraction of cells damaged prior to the information storage process; i.e., no matter what the message is, some cells are always stuck at 1s or 0s. Besides such defects, the memory read-out may suffer from the usual channel-induced noise (due to dust, or memory cell damage that occurs after the data are stored, etc.).

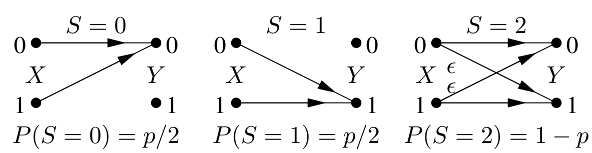

Figure 1: Channel with stuck-at defects.

As depicted in Fig. 2, there are three categories of memory cells: the stuck-at-0 cells, the stuck-at-1 cells, and the normal cells, with fractions of $p/2$, $p/2$, and $1-p$, respectively. In coding terminology, we may refer to the above three types of memory cells as three *channel states*. By introducing a random variable S (S=0, 1 and 2) for the channel state, we have

$$P(S=0)=P(S=1)=p/2, \quad P(S=2)=1-p.$$

Let the input and output of the channel be X and Y, respectively. Then for any $x \in \{0,1\}$,

$$\begin{aligned} P(Y=0|X=x,S=0) &= 1, \\ P(Y=1|X=x,S=1) &= 1, \\ P(Y \neq x|X=x,S=2) &= \epsilon. \end{aligned}$$

In other words, each memory cell has probability $p/2$ of being permanently stuck at 0 or 1, and probability $1-p$ of behaving like a binary symmetric channel (BSC) with cross-over probability ϵ. We assume that each cell is independent of other cells.

A natural question is how to design a coding system for such channels. The coding problem is trivial if the defect pattern is known at both the encoder and decoder. A more interesting situation is that the defect pattern is known only to the encoder but *not* to the decoder. We may understand this by imagining that we are given a stack of CD-ROMs with scratches (under a tolerable level). We hope that after detecting the defect pattern, the encoder (CD burner) can store as much data on these

CD-ROMs as possible. Due to the random nature of such defects, instead of avoiding writing on the defective cells (which will cause problems in reading out the data later), a more desirable strategy is to have a universal scheme that *adapts* the burning process to the defect pattern. Then at the decoder end, a "universal" CD drive can read out the data regardless of the defect pattern.

If both the encoder and the decoder know the defect pattern, the channel capacity is given by

$$C = (1-p)(1-H(\epsilon)) \text{ bits/cell.} \tag{1}$$

Interestingly, when the defect pattern is revealed only to the encoder but not to the decoder, it has been shown that the same capacity as in (1) can be achieved [5]. A similar result for the Gaussian channel, known as *writing on dirty paper*, has also been proved [1]. Coding for channels with stuck-at defects is of great interest both practically and theoretically.

As in dirty-paper coding, the proof in [5] is based on a random binning argument. Heegard then proposed partitioned linear block codes [6] as a practical binning scheme to approach the capacity. However, the codes provided in [6] have relatively short block lengths (from 7 up to 1023), thus cannot offer capacity-approaching performance. As the length grows, explicit construction of such codes becomes more complex.

3. System Design

LDPC codes are the most powerful linear block codes known to date. In this paper, we investigate the use of LDPC codes for channels with stuck-at defects. Heegard's idea of code-partitioning is adopted. For a given $(n,\ k)$ LDPC code $\mathcal{C}$, $l < k$ rows of the generator matrix G are selected to form a sub-matrix G_0; i.e.,

$$G = \begin{bmatrix} G_0 \\ G_1 \end{bmatrix}. \tag{2}$$

The sub-matrix G_0 generates a subcode $\mathcal{C}_0$ of $\mathcal{C}$. Let g_1 be an arbitrary element of the subcode $\mathcal{C}_1$ generated by G_1. Adding g_1 to $\mathcal{C}_0$ produces a coset of $\mathcal{C}_0$ in $\mathcal{C}$. The message bits are encoded in cosets of $\mathcal{C}_0$. Every element in $g_1 + \mathcal{C}_0$ is regarded as the same message by the decoder. There are 2^{k-l} cosets in total; thus, the maximum number of message bits is $k-l$.

The encoder works as follows. For a given $(k-l)$-bit message, the $(k-l)$ bits are used as the index to identify a coset of $\mathcal{C}_0$. For a given stuck-at defect pattern (which is known at the transmitter), the encoder tries to select a codeword in the coset to match as many defect locations as possible. (The unmatched locations become part of the channel noise in the decoding process.) This codeword is then transmitted through the channel, where it is corrupted by additional channel-induced noise in non-stuck-at positions. Upon receiving the corrupted codeword, the decoder tries to correct as many errors as possible. Finally, the decoder recovers the coset index from the decoded codeword.

We may understand the asymptotic capacity limit of the channel with stuck-at defects as follows. First, it turns out that for a stuck-at channel with channel parameter (p, ϵ), the code parameter (n, k, l) should be chosen so that in the encoding process the number of unmatched positions (among np stuck-at positions) is at most ϵnp. Thus, from a decoder point of view, the decoding process is equivalent to that of a usual binary symmetric channel with crossover probability ϵ. In this case, the parameters of the overall (n, k)-code must satisfy:

$$\frac{k}{n} \leq 1 - H(\epsilon). \tag{3}$$

Now, in the matching process, in order to match $np(1-\epsilon)$ locations among np stuck-at positions using an l-dimensional code $\mathcal{C}_0$, from Shannon's rate-distortion theory [10], we know that the code parameters for each coset of $\mathcal{C}_0$ must satisfy

$$\frac{l}{np} \geq 1 - H(\epsilon). \tag{4}$$

Therefore, the overall transmission rate is

$$\frac{k-l}{n} \leq (1-p)(1-H(\epsilon)) \text{ bits/cell.} \tag{5}$$

As this is also the maximum rate if the stuck-at positions were also available at the receiver, $(1-p)(1-H(\epsilon))$ is the capacity of the channel with stuck-at defects.

4. Quantization Code Embedded In Error-Correcting Code

For the above coding scheme to work well, the overall (n, k) code $\mathcal{C}$ must be a good error-correcting code. In addition, as $\mathcal{C}_0$ (or a coset of $\mathcal{C}_0$) is used to match as many defects as possible in the encoding process, $\mathcal{C}_0$ (or its cosets) must be a good *vector quantization code.* Since $\mathcal{C}_0 \subset \mathcal{C}$, in order to achieve the channel capacity, we need to have a good quantization code embedded in a good error-correcting code.

The design of good vector-quantization codes is a much harder problem as compared to the design of good error-correcting codes. To approach the rate-distortion limit, the best decoding algorithms known to date are complete decoders such as the classic Viterbi decoder. However, Viterbi algorithm is feasible only when the trellis complexity of the code is relatively low. Thus, one way to design good coding schemes for channels with stuck-at defects is to start with a good error-correcting code $\mathcal{C}$ and then to find a good quantization subcode $\mathcal{C}_0$ within $\mathcal{C}$ with manageable trellis complexity.

In general, when partitioning the generator matrix G into G_0 and G_1, the trellis complexity of G_0 increases exponentially with the number of rows of G_0. The trellis complexity of a linear block code may be characterized using the methods of [11]. First, the generator matrix of a given code is reduced into a *trellis-oriented form* via elementary row operations; i.e., no two rows start or end at the same position. Once the generator matrix is in trellis-oriented form, a trellis for this linear block code can then be constructed by taking the product of all the elementary trellises corresponding to every row of the generator matrix [11]. The complexity of the overall trellis is therefore roughly exponential in the number of overlapping elementary trellises. As will be explained in more detail in the next section, this means of measuring trellis complexity gives us a way to intelligently select a low-complexity embedded subcode for an LDPC code when the LDPC code has a certain cyclic structure.

5. FG-LDPC Codes for Channels with Stuck-at Defects

FG-LDPC codes are a special family of LDPC codes constructed from the lines and points of finite Euclidean or projective geometries over finite fields. An m-dimensional Euclidean geometry over $GF(2^s)$ consists of 2^{ms} m-tuples whose components are elements defined on $GF(2^s)$. The construction of a projective geometry from the elements of a Galois field is detailed in [12]. A finite (Euclidean or projective) geometry with n points and r lines satisfies some structural properties [9]. For the sake of completeness, we briefly quote them from [9] as follows: 1) every line consists of ρ points; 2) any two points are connected by one and only one line; 3) every point is intersected by γ lines; 4) any two lines either are parallel or intersect at one and only one point.

Such a finite geometry can be represented by a $r \times n$ binary matrix $H = [h_{i,j}]$ with every row corresponding to a line, and every column to a point. If the j-th point is contained in the i-th line, then $h_{i,j} = 1$; otherwise, $h_{i,j} = 0$ [9]. A linear block code can be generated whose parity check matrix is H. When ρ and γ are relatively small compared to r and n, the codes produced by such H's are a subclass of LDPC codes, named FG-LDPC codes.

FG-LDPC codes possess many attractive properties that other LDPC codes generally do not have [9, 12]. First, these codes are either cyclic or quasi-cyclic, which means that linear-time encoding can be implemented by using shift registers based on their generator polynomials. Second, the minimum distances of FG-LDPC codes are relatively large. Third, the Tanner graphs of FG-LDPC codes do not contain cycles of length 4. The second and third properties imply good error-correcting capability; however, the cyclic or quasi-cyclic structural property is of particular interest to us.

When the code is cyclic or quasi-cyclic, there exists a generator polynomial $g(x)$ such that every row in the generator matrix is a shifted version of $g(x)$. It then follows [11] that the generator matrix is already in a trellis-oriented form because no two rows start or end at the same position. Furthermore, being cyclic or quasi-cyclic, the trellis-oriented generator matrix has a regular structure, making the selection of rows to form a G_0 with low trellis complexity particularly easy. For example, we may choose the first and the last rows in G, and the remaining $l-2$ rows which are evenly spaced between the first and the last rows. (This can be easily done if $k-1$ is a multiple of $l-1$, otherwise we adjust to make them as much evenly spaced as possible). Since the trellis complexity is determined by the overlapping part of the elementary trellises of all the rows, a generator matrix G_0 constructed this way has minimal overlaps among all the rows, forming a quantization code with low trellis complexity. For FG-LDPC codes, the span (the length between the starting and ending position) of each row is exactly $n-k$. In the generator matrix G_0 constructed as above, every row is obtained by shifting the previous row by $\lfloor (k-1)/(l-1) \rfloor$ positions (neglecting the adjustment when $(k-1)$ is not a multiple of $(l-1)$). If

$$(j-1)\left\lfloor \frac{k-1}{l-1} \right\rfloor < n-k \leq j \left\lfloor \frac{k-1}{l-1} \right\rfloor,$$

for some integer j, then the maximum number of rows that overlap is j; therefore, the maximum number of states in the trellis is 2^j.

6. Theoretical Performance Limit

In this section, we derive the theoretical performance limit that an optimal error-correcting code with an embedded optimal quantization code can achieve. The theoretical limit is characterized by the (p, ϵ) pair, i.e., the stuck-at probability and the cross-over probability of the channel that a code with parameters (n, k, l) may attain.

Given n, k and l, inequality (5) directly gives an upper bound for the (p, ϵ) pair. When $p = 0$, the largest value of ϵ for reliable communication is given by

$$\epsilon^* = H^{-1}\left(1 - \frac{k-l}{n}\right). \tag{6}$$

However, the bound given by (5) becomes very loose as p increases. In the following, we derive a tighter bound for the (p, ϵ) pair.

In the encoder, since we are using an l-dimensional binary vector generated by G_0 to quantize pn defective positions, the rate of the embedded quantization code is l/pn. To quantize a Bernoulli-1/2 source at a desired distortion level D, Shannon's rate-distortion theory [10] states that the rate R and the distortion D must satisfy the following relation:

$$R \geq 1 - H(D).$$

Therefore, given p, n and l, the distortion D is lower bounded by

$$D \geq H^{-1}\left(1 - \frac{l}{pn}\right).$$

Note that the condition $pn \geq l$ is implied in the above inequality. Intuitively, when $pn \leq l$, no distortion has to be introduced if the quantization code is properly designed.

Under the Hamming distortion measure, D is exactly the bit error rate (BER). Note that here D is the BER averaged over the pn positions. A more useful parameter here would be the residual quantization error rate, ϵ_q, which is the fraction of unmatched positions after quantization averaged over the entire sequence; i.e.,

$$\epsilon_q = \frac{pnD}{n} = pD \geq \begin{cases} pH^{-1}\left(1 - \frac{l}{pn}\right), & p > \frac{l}{n}, \\ 0, & p \leq \frac{l}{n}. \end{cases} \tag{7}$$

Given the largest possible cross-over probability ϵ^* for reliable communication over the channel, the actual channel cross-over probability ϵ and the residual quantization error rate ϵ_q must satisfy

$$(1 - \epsilon_q) \cdot (1 - \epsilon) \geq 1 - \epsilon^*.$$

Thus, the channel cross-over probability ϵ is bounded by

$$\epsilon \leq 1 - \frac{1 - \epsilon^*}{1 - \epsilon_q}. \tag{8}$$

Combining (8) with (6) and (7) yields the following relation between p and ϵ:

$$\epsilon \leq \begin{cases} 1 - \frac{1 - H^{-1}\left(1 - \frac{k-l}{n}\right)}{1 - pH^{-1}\left(1 - \frac{l}{pn}\right)}, & p > \frac{l}{n}, \\ H^{-1}\left(1 - \frac{k-l}{n}\right), & p \leq \frac{l}{n}. \end{cases}$$

7. Simulation Results

We now provide some simulation results using FG-LDPC codes for channels with stuck-at defects. The code is provided in [9], which is a (4095, 3367) cyclic code. We choose $l = 15$ rows in the generator matrix G to form G_0. Thus, the optimum rate achievable is $(k - l)/n = 0.8186$. To keep these rows as equally spaced apart as possible while every position is covered by at least one codeword, the row numbers chosen are $j = 1+(i-1)\times 240$, where $i = 1, 2, \cdots, 14$, and $j = 3367$, the last row. The complexity of the trellis is very low; the maximum number of states is $2^3 = 8$. A Viterbi decoder is used for the quantization step at the encoder. An iterative sum-product algorithm is used for the error-correcting step at the decoder. The number of iterations is 20.

In Fig. 2, the maximum (p, ϵ) region for which the (4095, 3367, 15) cyclic code is able to sucessfully encode and decode with an overall word error rate (WER) of 10^{-2} is plotted. The performance is compared with the theoretical limit as computed in Section 6. We can see that at the block length of only 4095 bits, our simulation results are reasonably close to the theoretical limit. We believe that the performance loss is mainly due to the relatively weak error-correcting capability of the (4095, 3367) FG-LDPC codes. When more powerful LDPC codes with longer block lengths are used, we expect performance that is closer to theoretical limits.

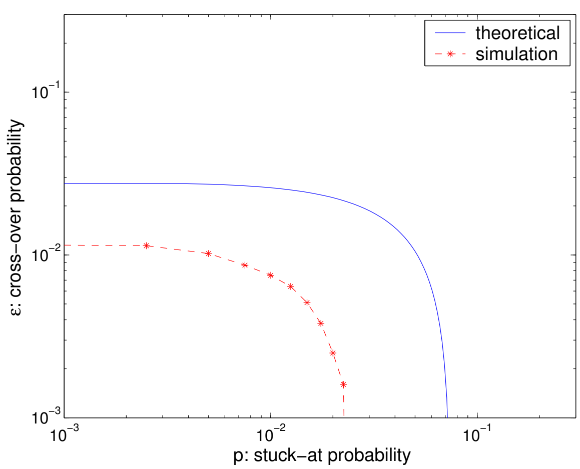

Figure 2: Performance of the FG-LDPC codes on channels with stuck-at defects. The code parameters are $(n, k, l) = (4095, 3367, 15)$. The maximum theoretically attainable (p, ϵ) and the actual (p, ϵ) at WER $= 10^{-2}$ are plotted.

Our result is not directly comparable to that of Heegard's or Borden and Vinck's. In [6], the error-correcting and defect-matching capability of Heegard's codes are measured by the minimum distances of the partitioned subcodes. In [7], bounds on the probability of error are derived. Simulation results are not provided in [6] and [7]. On the other hand, FG-LDPC codes do not allow flexibility in their sizes and dimensions. For a finite geometry, once the number of points n is given, the number of lines $r = n - k$ is immediately determined. Therefore, only certain rates are available among the FG-LDPC codes. The sizes and dimensions (and hence the rates) of FG-LDPC codes do not match those of the codes in [6] and [7].

8. Conclusion

In this work, we consider the coding problem for a class of channels with stuck-at defects as well as channel-induced noise. The most interesting case is when the defect pattern is revealed only to the encoder but not to the decoder; the channel capacity is exactly the same as if the decoder also knew the defect pattern, reminiscent of the classic dirty-paper coding problem. The major challenge of coding for such channels lies in the need of a good quantization code embedded in a good error-correcting code. The recently discovered FG-LDPC codes are shown to be a suitable candidate, because the construction of a good quantization code with low trellis complexity becomes particularly easy due to the cyclic or quasi-cyclic structure of FG-LDPC codes. Simulation results show that FG-LDPC codes are promising in approaching the theoretical performance limit.

Acknowledgments

The authors wish to thank Jun Xu, Heng Tang and Yu Kou for their assistance in providing the FG-LDPC codes.

References

[1] M. Costa, "Writing on dirty paper," *IEEE Trans. Inform. Theory*, vol. 29, pp. 439-441, May 1983.

[2] A. V. Kusnetsov and B. S. Tsybakov, "Coding in a memory with defective cells,"translated from *Prob. Peredach. Inform.*, vol. 10, no. 2, pp. 52–60, Apr.–Jun. 1974.

[3] B. S. Tsybakov, "Additive group codes for defect correction," translated from *Prob. Peredach. Inform.*, vol. II, no. 1, pp. 111–113, Jan.–Mar. 1975.

[4] V. V. Losev, V. K. Konopel'ko, and Yu. D. Daryakin, "Double-and-triple-defect-correcting codes," translated from *Prob. Peredach. Inform.*, vol. 14, no. 4, pp. 98–101, Oct.–Dec. 1978.

[5] C. Heegard and A. A. El Gamal, "On the capacity of computer memory with defects," *IEEE Trans. Inform. Theory*, vol. 29, no. 5, pp. 731–739, Sept. 1983.

[6] C. Heegard, "Partitioned linear block codes for computer memory with 'stuck-at' defects," *IEEE Trans. Inform. Theory*, vol. 29, no. 6, pp. 831–842, Nov. 1983.

[7] J. M. Borden and A. J. Vinck, "On coding for 'stuck-at' defects," *IEEE Trans. Inform. Theory*, vol. IT-33, pp. 729–735, Sept. 1987.

[8] R. G. Gallager, "Low density parity check codes," *IRE Trans. Inform. Theory*, vol. IT-8, pp. 21-28, Jan. 1962.

[9] Y. Kou, S. Lin, and M. P. C. Fossorier, "Low-density parity-check codes based on finite geometries: a rediscovery and new results," *IEEE Trans. Inform. Theory*, vol. 47, no. 7, pp. 2711–2736, Nov. 2001.

[10] C. E. Shannon, "Coding theorems for a discrete source with a fidelity criterion," *IRE Nat. Conv. Rec.*, pt. 4, pp. 142–163, Mar. 1959.

[11] F. R. Kschischang and V. Sorokine, "On the trellis structure of block codes," *IEEE Trans. on Inform. Theory*, vol. 41, no. 6, pp. 1924-1937, 1995.

[12] S. Lin and D. J. Costello, *Jr.*, *Error Control Coding: Fundamentals and Applications*, Prentice-Hall, 1983.