

Operating LDPC Codes with Zero Shaping Gap

Georg Böcherer and Rudolf Mathar

Institute for Theoretical Information Technology

RWTH Aachen University, 52056 Aachen, Germany

Email: {boecherer, mathar}@ti.rwth-aachen.de

Abstract—Unequal transition probabilities between input and output symbols, input power constraints, or input symbols of unequal durations can lead to non-uniform capacity achieving input distributions for communication channels. Using uniform input distributions reduces the achievable rate, which is called the shaping gap. Gallager’s idea for reliable communication with zero shaping gap is to do encoding, matching, and jointly decoding and dematching. In this work, a scheme is proposed that consists in matching, encoding, decoding, and dematching. Only matching is channel specific whereas coding is not. Thus off-the-shelf LDPC codes can be applied. Analytical formulas for shaping and coding gap of the proposed scheme are derived and it is shown that the shaping gap can be made zero. Numerical results show that the proposed scheme allows to operate off-the-shelf LDPC codes with zero shaping gap and a coding gap that is unchanged compared to uniform transmission.

I. INTRODUCTION

The ultimate rate for reliable communication over a noisy channel is given by the maximum mutual information between input and output. A channel is called non-uniform if the input distribution that achieves this maximum is non-uniform. The non-uniformity can result from different factors, examples are channels that are not symmetric [1, Theorem 8.2.1], average power constraints on input symbols [2, Sec. IV], input symbols of unequal duration [3], and multi-user channels with cross-talk [4]. Conventional communication systems restrict the channel input to be uniformly distributed. The resulting penalty in terms of achievable rates is called the shaping gap.

One approach to close the shaping gap goes back to Gallager [5, p. 208] and is nicely explained by McEliece in [6, Sec. 5]. The idea is to first encode the data and then to use a fixed-to-fixed length mapper from the codewords to the channel symbols. The mapping is chosen such that the capacity achieving probability mass function (pmf) is approximated. To achieve this, the mapping is many-to-one. For iterative decoding, a probabilistic demapper needs to be incorporated into the factor graph of the decoder [6, Fig. 8]. This idea was extended to non-binary low-density parity-check (LDPC) codes by Ratzert and MacKay in [7] and by Bennatan and Burshtein in [8].

Franceschini *et al.* conjectured in [9] that LDPC codes have universal properties, which basically means the following: *if an LDPC code is designed for some channel A, it can be used for a different channel B, as long as the mutual information between input and output are the same both for channel A*

and channel B. See [10] and references therein for analytical support of this conjecture.

In this work, we use the conjecture of Franceschini *et al.* as a design paradigm. That is, instead of designing LDPC codes for specific non-uniform channels, *we show how existent LDPC encoders/decoders that were originally designed for uniform channels can be operated with zero shaping gap on non-uniform channels.*

We propose the following transmission scheme. First, the binary equiprobable data stream is matched to the channel by a prefix-free code [11]. Then, the matched bits are encoded by a systematic code. The check bits are iteratively matched to the channel and encoded such that all transmitted bits are matched to the channel. At the receiver side, the packets are first decoded and then dematched. The central property of our scheme is that shaping (matcher) and error correction (encoder/decoder) can be designed independently, in the sense that the matcher only influences the *parameters* passed to the encoder (i.e., data symbols) and decoder (i.e., log-likelihood ratios (LLR) and priors). This is a fundamental difference to Gallager’s scheme, where decoding and dematching has to be performed jointly. For binary channels with unequal symbol durations, we provide analytical formulas for the shaping gain and the coding gain of our scheme. We show that capacity achieving matching codes can efficiently be found using Geometric Huffman Coding [11]. Our results can easily be extended to arbitrary discrete input memoryless channels. Finally, we apply our scheme to a binary symmetric channel (BSC) with symbol durations $w_0 = 1$ and $w_1 = 5$ and varying error probability ϵ . We use off-the-shelf LDPC codes, namely the codes used in DVB-S2 as implemented in the Matlab Communications Toolbox. For uniform transmission, we observe a shaping gap of 20% while for matched transmission, the shaping gap is virtually zero. For both schemes, the coding gap is almost equal, which is in perfect accordance with Franceschini’s conjecture.

The remainder of this paper is organized as follows. In Section II, we define the binary channel with unequal symbol durations. In Section III, we develop the proposed scheme. Formulas for the shaping and coding gain are derived in Section IV. Capacity achieving matchers are defined in Section V. Finally, we provide numerical results in Section VI.

This work has been supported by the UMIC Research Center, RWTH Aachen University.

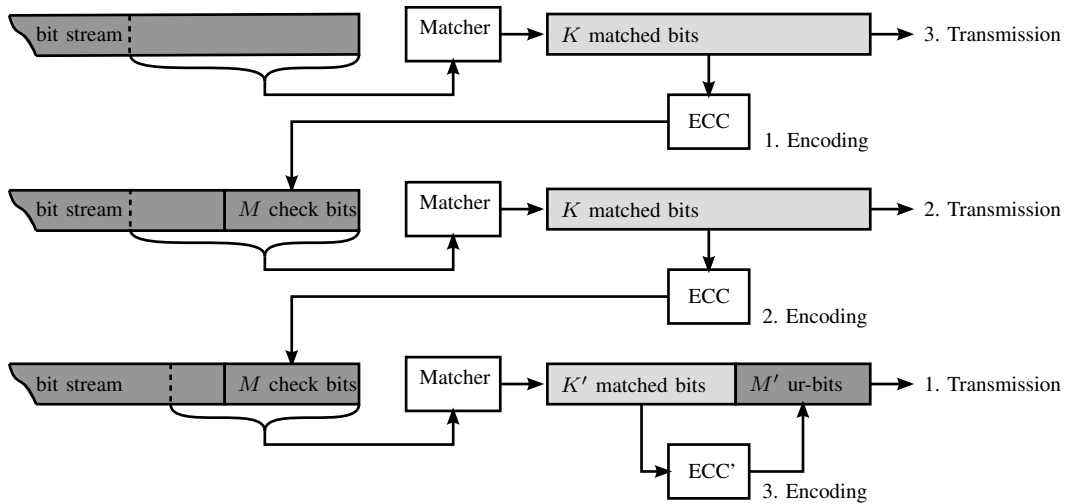


Fig. 1. Bootstrap scheme. The acronym ECC stands for error correction coding.

II. PROBLEM STATEMENT

A binary channel (BC) is specified by a matrix of transition probabilities (h_{ji}) . An input PMF \mathbf{p} relates to its corresponding output PMF \mathbf{r} as

$$\mathbf{r} = \begin{pmatrix} r_0 \\ r_1 \end{pmatrix} = \begin{pmatrix} h_{00} & h_{01} \\ h_{10} & h_{11} \end{pmatrix} \begin{pmatrix} p_0 \\ p_1 \end{pmatrix}. \quad (1)$$

The mutual information between input and output is according to Gallager [12, Eq. (8.73)] given by

$$\mathcal{I}(\mathbf{p}) = \sum_i p_i \sum_j h_{ji} \log \frac{h_{ji}}{r_j}. \quad (2)$$

We consider BCs where the input symbols 0 and 1 have unequal durations $\mathbf{w} = (w_0, w_1)^T$. According to [13, Theorem 2], the capacity of such a BC is given by

$$C = \max_{\mathbf{p}} \frac{\mathcal{I}(\mathbf{p})}{\mathbf{w}^T \mathbf{p}}. \quad (3)$$

Jimbo and Kunisawa [3] proposed a variation of the Blahut-Arimoto algorithm [2], [14] to efficiently find the capacity achieving distribution \mathbf{p}^* that maximizes (3). We denote the uniform pmf by $\mathbf{u} = (0.5, 0.5)^T$. The shaping gain that results from using \mathbf{u} instead of \mathbf{p}^* is given by

$$\frac{\mathcal{I}(\mathbf{u})}{\mathbf{w}^T \mathbf{u}} \frac{1}{C}. \quad (4)$$

In the case of equal symbol durations $w_0 = w_1$, this gain is at least 0.942 [15, p. 295] and there are little reasons not to use \mathbf{u} . However, for $w_0 \neq w_1$, the gain can be arbitrary close to zero. This motivates us to seek for a transmission scheme that closes the shaping gap for arbitrary symbol durations \mathbf{w} while allowing to use existent LDPC encoders/decoders.

III. BOOTSTRAP SCHEME

A. Prefix-Free Matcher Codes

The digital interface between source and channel coding is a stream of independent, identically distributed (iid) equiprobable bits. By parsing the stream by a full prefix-free code,

a dyadic pmf can be generated [11]. For example, consider the set of binary input symbol blocks of length 2, namely $\{00, 01, 10, 11\}$. Then the mapping

$$\begin{aligned} 1 &\mapsto 00 & 01 &\mapsto 01 \\ 001 &\mapsto 10 & 000 &\mapsto 11 \end{aligned} \quad (5)$$

generates the pmf $(2^{-1}, 2^{-2}, 2^{-3}, 2^{-3})$ over the set $\{00, 01, 10, 11\}$ when the stream of iid equiprobable bits is parsed by the prefix-free code $\{1, 01, 001, 000\}$. We call a device that implements this procedure a *matcher*. We will see in Section V how capacity achieving matchers can be found efficiently. When matchers are used for noisy channels, a severe problem occurs: one single bit error can lead to a complete loss of a block, since the binary input and output streams are out of sync, e.g., suppose the data bits 101001 were mapped by the matcher (5) to the block 000110 and then transmitted over the channel, and suppose 010110 was detected at the channel output. Then,

$$101001 \mapsto 000110 \quad (6)$$

$$010110 \mapsto 0101001 \quad (7)$$

i.e., matcher input and dematcher output are of different length and aligning the first two bits leads to 5 bit errors in the overlapping strings. Error-correction based on matcher input and dematcher output needs the capability of correcting insertion and deletion errors, which is difficult [16].

B. Reverse Concatenation

The above problem can be solved by interchanging the order of error-correction coding and matching. This approach is used for constrained systems by first applying constrained coding and then error correction coding, see [17] and references therein. Suppose we have applied the matcher and apply error correction coding to the matcher output. In LDPC codes, sums of bits modulo 2 are transmitted. As the number of summands grows, the resulting bit is uniformly distributed independent

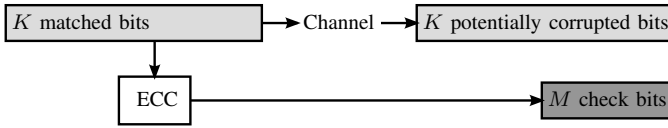


Fig. 2. Building block of the bootstrap scheme.

of the pmf of the summands, i.e.,

$$p = i_1 \oplus i_2 \oplus \dots \oplus i_m \sim U\{0, 1\}. \quad (8)$$

Thus, even if the bits i_1, \dots, i_m were matched to the channel, p is not. A first step to circumvent this problem is to use systematic codes. Here, the data bits are transmitted unchanged and the check bits are appended. Thus, the data bits can be shaped and the check bits cannot. This already gives an improvement compared to uniform transmission. This approach was suggested under the name *sparse-dense codes* by Ratzert in [7]. However, this approach does not close the shaping gap, e.g., for 1/2-rate codes, only half of the transmitted bits are matched to the channel.

C. Bootstrapping the Check Bits

We now introduce the *bootstrap scheme*, which allows to match the check bits to the channel. Consider a systematic binary block code that generates M check bits per K data bits, i.e., a rate $K/(K+M)$ -code. At the transmitter side, B blocks are sequentially encoded and then transmitted in reverse order. For the first block, the matcher generates K matched bits from the equiprobable input bit stream. From these K matched bits, the encoder calculates M check bits. These check bits are approximately iid and equiprobable. They are fed back and appended to the bit stream. In the next round, the matcher again generates K matched bits from the bit stream. Since the M check bits from the first round were appended to the bit stream, they are now matched to the channel as part of the K matched bits of the second round. This procedure continues until the last but second round. In the last round, a (M', K') sparse-dense code is applied. The matcher maps the M check bits from round $B - 1$ plus data bits to K' matched bits. The encoder calculates M' check bits from the K' matched bits and appends them (unmatched) to the K' matched bits. These M' check bits of the last round are the *ur-bits* from which the decoder will start to bootstrap all B blocks. All B blocks are transmitted in reverse order such that the decoder can immediately start decoding. See Fig. 1 for an illustration.

IV. SHAPING AND CODING GAIN

The number M' of ur-bits is independent of the number of packets B . Thus, as B grows, the fraction of matched bits approaches 1. Therefore, we will focus in the rest of this work on the building block of the bootstrap scheme, which consists of the transmission of K matched bits over a channel and decoding conditioned on M perfectly known check bits. See Fig. 2 for an illustration. Note that the M perfectly known check bits are not for free: the M check bits of the next packet to be received are embedded in matched form in the K

matched bits. We will take this into account when calculating the effective transmission rate.

A. Shaping Gain

Assume the matcher uses a variable-to- k length code, i.e., blocks of k matched bits are jointly generated from the unmatched bits. Denote by \mathbf{p} the resulting block pmf. Denote by \mathbf{v} the lengths of the blocks, e.g., if block i consists of ℓ zeros and $k - \ell$ ones, the length is $v_i = \ell w_0 + (k - \ell)w_1$. The mutual information rate is directly given by

$$I_{\text{bs}}(\mathbf{p}) = \frac{\mathcal{I}(\mathbf{p})}{\mathbf{v}^T \mathbf{p}} \quad (9)$$

and the shaping gain is given by

$$\frac{I_{\text{bs}}(\mathbf{p})}{C}. \quad (10)$$

B. Coding Gain

We now calculate the effective transmission rate of the bootstrap scheme, i.e., how many information bits per block length are in the average transmitted over the channel. Denote by $\mathcal{H}(\mathbf{p})$ the entropy in bits of \mathbf{p} . The average information per matched bit is $\mathcal{H}(\mathbf{p})/k$. Thus, in the average,

$$\frac{\#\{\text{unmatched bits}\}}{\#\{\text{matched bits}\}} = \frac{1}{\frac{\mathcal{H}(\mathbf{p})}{k}} \triangleq m. \quad (11)$$

The unmatched bits consist of the information bits and the M check bits of the previous block. Thus

$$(\#\{\text{information bits}\} + M)m = \#\{\text{matched bits}\} = K \quad (12)$$

$$\Rightarrow \#\{\text{information bits}\} = \frac{K}{m} - M. \quad (13)$$

The average length of a matched bit is $\mathbf{v}^T \mathbf{p}/k$. Denote by $c = K/(K+M)$ the code rate of the applied code. Altogether, the effective transmission rate can now be written as

$$\frac{\#\{\text{information bits}\}}{K \frac{\mathbf{v}^T \mathbf{p}}{k}} = \frac{\frac{K}{m} - M}{K \frac{\mathbf{v}^T \mathbf{p}}{k}} \quad (14)$$

$$= \frac{\frac{\mathcal{H}(\mathbf{p})}{k} - \frac{M+K-K}{K}}{\frac{\mathbf{v}^T \mathbf{p}}{k}} \quad (15)$$

$$= \frac{\frac{\mathcal{H}(\mathbf{p})}{k} + 1 - \frac{1}{c}}{\frac{\mathbf{v}^T \mathbf{p}}{k}} \quad (16)$$

$$\triangleq R_{\text{bs}}(\mathbf{p}, c). \quad (17)$$

We define the *coding gain* by

$$R_{\text{bs}}(\mathbf{p}, c) = \frac{\frac{\mathcal{H}(\mathbf{p})}{k} + 1 - \frac{1}{c}}{\frac{\mathcal{I}(\mathbf{p})}{k}}. \quad (18)$$

V. CAPACITY ACHIEVING MATCHER

We now show that the shaping gap can indeed be made as small as desired by approximating the capacity achieving pmf \mathbf{p}^* with the matching code found via Geometric Huffman Coding (GHC) [11]. We show this by first calculating the penalty that results from using an input pmf different from \mathbf{p}^* , and second, by bounding this penalty.

A. Using the Wrong PMF

The capacity achieving pmf \mathbf{p}^* is efficiently found by the Algorithm of Jimbo and Kunisawa [3]. By [3, Lemma 2],

$$\sum_j h_{ji} \log \frac{h_{ji}}{p_j^*} \leq C w_i, \quad \text{with equality if } p_i^* > 0. \quad (19)$$

Denote by \mathbf{p} some pmf with the only restriction that

$$p_i = 0, \quad \text{whenever } p_i^* = 0. \quad (20)$$

Now,

$$\mathcal{I}(\mathbf{p}) = \sum_i p_i \sum_j h_{ji} \log \frac{h_{ji}}{r_j} \quad (21)$$

$$= \sum_i p_i \sum_j h_{ji} \log \frac{h_{ji} r_j^*}{r_j r_j^*} \quad (22)$$

$$= \sum_i p_i \sum_j h_{ji} \log \frac{h_{ji}}{r_j^*} + \sum_i p_i \sum_j h_{ji} \log \frac{r_j^*}{r_j} \quad (23)$$

$$= \mathbf{C} \mathbf{w}^T \mathbf{p} - \mathbf{D}(\mathbf{r} \parallel \mathbf{r}^*) \quad (24)$$

where equality in the last line follows from (20) and (19). Dividing by $\mathbf{w}^T \mathbf{p}$ yields

$$\frac{\mathcal{I}(\mathbf{p})}{\mathbf{w}^T \mathbf{p}} = \mathbf{C} - \frac{\mathbf{D}(\mathbf{r} \parallel \mathbf{r}^*)}{\mathbf{w}^T \mathbf{p}}. \quad (25)$$

Thus, using pmf \mathbf{p} instead of the capacity achieving pmf \mathbf{p}^* results in a penalty of $\frac{\mathbf{D}(\mathbf{r} \parallel \mathbf{r}^*)}{\mathbf{w}^T \mathbf{p}}$.

B. Capacity Achieving Prefix-Free Matcher

By [1, Eq. (4.45)], the KL-distance between the output pmfs \mathbf{r} and \mathbf{r}^* is upper bounded by the KL-distance between the corresponding input pmfs, i.e., $\mathbf{D}(\mathbf{r} \parallel \mathbf{r}^*) \leq \mathbf{D}(\mathbf{p} \parallel \mathbf{p}^*)$. Thus, the penalty is upper bounded by

$$\frac{\mathbf{D}(\mathbf{p} \parallel \mathbf{p}^*)}{\mathbf{w}^T \mathbf{p}}. \quad (26)$$

By jointly approximating the capacity achieving pmf of k consecutive input symbols via GHC, by [11, Prop. 2], the penalty bound (26) goes to zero as block-length k goes to infinity. Consequently, the left-hand side of (25) approaches capacity.

VI. APPLICATION TO A BSC WITH $\mathbf{w} = (1, 5)^T$

We now consider a binary symmetric channel (BSC) where the input symbols have the weights $w(0) = 1$ and $w(1) = 5$. We vary the bit error probability ϵ between 0.005 and 0.057. As LDPC codes, we use the DVB-S2 codes with rates 8/9, 5/6, 4/5, and 3/4. The DVB-S2 codes are systematic and amenable for simulation, since an implementation is readily available in the Communications System Toolbox of Matlab. Because the simulation is very time-intensive, we evaluate the performance around block-error rates p_b of 10^{-2} . For each value of ϵ , capacity achieving pmf \mathbf{p}^* and capacity \mathbf{C} are calculated using the Jimbo-Kunisawa Algorithm [3]. Both for uniform and matched transmission, we use exactly the same parity check matrices, encoders, and decoders, namely the unchanged implementation in Matlab. The differences between uniform and matched transmission are detailed next.

00 \mapsto 0000	010 \mapsto 0001	011 \mapsto 0010	10111 \mapsto 0011
100 \mapsto 0100	10110 \mapsto 0101	10100 \mapsto 0110	1010111 \mapsto 0111
110 \mapsto 1000	11101 \mapsto 1001	11100 \mapsto 1010	101011 \mapsto 1011
11110 \mapsto 1100	111110 \mapsto 1101	111111 \mapsto 1110	1010100 \mapsto 1111

Fig. 3. Matching code found by GHC. The same matching code was found for the whole considered range of the error probability ϵ .

A. Uniform Transmission

1) *Matching*: K iid equiprobable data bits are generated. No matching is performed.

2) *Transmission*: Both the K data bits and the M check bits are transmitted over a BSC with parameter ϵ .

3) *Decoder parameters*: Since we have uniform priors, we pass to the decoder for each of the $K = K + M$ received bits the LLRs

$$\text{LLR}(0) = \ln \frac{(1-\epsilon)}{\epsilon}, \quad \text{if received bit } y = 0$$

$$\text{LLR}(1) = \ln \frac{\epsilon}{(1-\epsilon)}, \quad \text{if received bit } y = 1.$$

4) *Evaluation*: For each ϵ , the shaping gain is according to (4) calculated as

$$\frac{\mathcal{I}(\mathbf{u})}{\mathbf{w}^T \mathbf{u}} \frac{1}{\mathbf{C}} \quad (27)$$

where \mathbf{u} is the uniform pmf. For each coding rate c and channel parameter ϵ , the coding gain calculates as

$$\frac{c}{\mathcal{I}(\mathbf{u})}. \quad (28)$$

B. Matched Transmission

1) *Matching*: For the capacity achieving joint pmf \mathbf{p}^{*4} of 4 subsequent bits, the dyadic approximation \mathbf{p} is calculated using the implementation [18] of GHC. The prefix-free matching code induced by \mathbf{p} is displayed in Fig. 3. For the whole considered range of ϵ , the optimal dyadic approximation remains unchanged. A sufficiently long data stream of iid equiprobable bits is generated. By parsing the stream by the matcher code, a sequence of $K/4$ blocks consisting each of 4 bits is generated. These blocks are iid according to \mathbf{p} .

2) *Transmission*: We transmit the K matched data bits over a BSC with parameter ϵ . The M check bits remain unchanged.

3) *Decoder*: Since we have non-uniform priors $\mathbf{p}^* = (\pi_0, \pi_1)^T$, we pass to the decoder for the K received bits

$$\text{LLR}(0) + \ln \frac{\pi_0}{\pi_1} = \ln \frac{(1-\epsilon)\pi_0}{\epsilon\pi_1}, \quad \text{if received bit } y = 0$$

$$\text{LLR}(1) + \ln \frac{\pi_0}{\pi_1} = \ln \frac{\epsilon\pi_0}{(1-\epsilon)\pi_1}, \quad \text{if received bit } y = 1$$

and for the M check bits, we pass

$$\infty, \quad \text{if check bit } b = 0$$

$$-\infty, \quad \text{if check bit } b = 1.$$

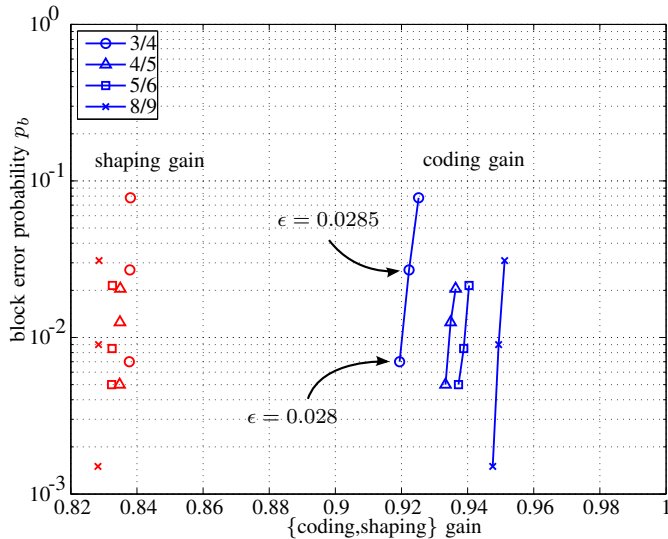


Fig. 4. Uniform transmission.

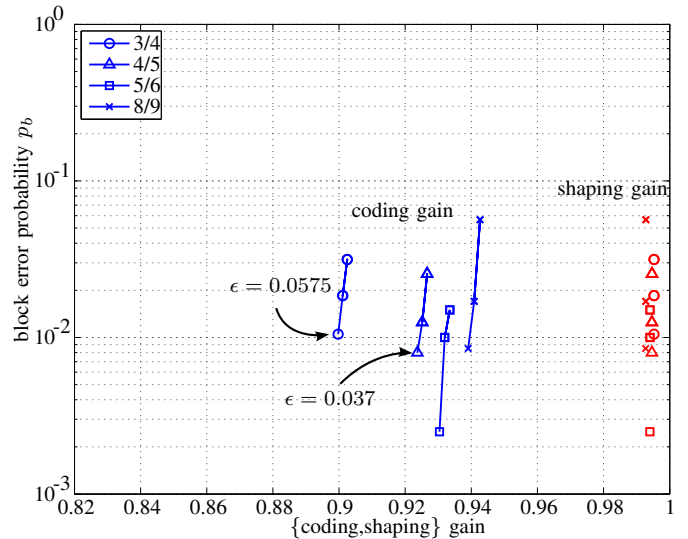


Fig. 5. Matched transmission.

4) *Evaluation:* For each channel parameter ϵ , the shaping gain is according to (10) calculated as

$$\frac{\mathcal{I}(\mathbf{p})}{\mathbf{v}^T \mathbf{p} \mathbf{C}} \quad (29)$$

where \mathbf{v} denotes the weights of the symbol blocks. For each coding rate c and channel parameter ϵ , the coding gain is according to (18) calculated as

$$\frac{\frac{\mathcal{H}(\mathbf{p})}{4} + 1 - \frac{1}{c}}{\frac{\mathcal{I}(\mathbf{p})}{4}}. \quad (30)$$

C. Discussion

The simulation results are displayed in Fig. 4 and Fig. 5. Shaping and coding gain versus block error probability p_b are displayed for 3/4, 4/5, 5/6, and 8/9 DVB-S2 codes. The different operating points were generated by varying the channel error probability ϵ . For uniform transmission, the shaping gain is around 0.83. For matched transmission, the shaping gain is larger than 0.99. The coding gain is for uniform transmission between 0.92 and 0.95. For matched transmission, it is between 0.9 and 0.94. For example, for the rate 3/4 code, the coding gain for uniform transmission is 0.92 and for matched transmission, it is 0.9, thus slightly worse. However, for uniform transmission, the 3/4 code works at $p_b = 10^{-2}$ for some ϵ between 0.028 and 0.0285, while for matched transmission, the 3/4 code works at $p_b = 10^{-2}$ for $\epsilon = 0.0575$, which is a significantly higher channel error probability. If we compare the achieved coding gains of uniform and matched transmissions at similar (p_b, ϵ) pairs, we observe a higher coding gain for matched transmission than for uniform transmission. For example, we have a coding gain of 0.921 at $(p_b = 0.03, \epsilon = 0.0285)$ for uniform transmission and a coding gain of more than 0.921 at $(p_b = 0.008, \epsilon = 0.037)$. This corresponds to a better coding gain at a lower block error rate for a higher channel error probability.

REFERENCES

- [1] T. M. Cover and J. A. Thomas, *Elements of Information Theory*, 2nd ed. John Wiley & Sons, Inc., 2006.
- [2] R. Blahut, "Computation of channel capacity and rate-distortion functions," *IEEE Trans. Inf. Theory*, vol. 18, no. 4, pp. 460–473, 1972.
- [3] M. Jimbo and K. Kunisawa, "An iteration method for calculating the relative capacity," *Inf. Contr.*, vol. 43, no. 2, pp. 216–223, Nov. 1979.
- [4] E. A. Ratzler and D. J. C. MacKay, "Sparse low-density parity-check codes for channels with cross-talk," in *Proc. IEEE Information Theory Workshop*, 2003, pp. 127–130.
- [5] R. G. Gallager, *Information Theory and Reliable Communication*. John Wiley & Sons, Inc., 1968.
- [6] R. McEliece, "Are turbo-like codes effective on nonstandard channels?" *IEEE Inf. Theo. Society Newsletter*, vol. 51, no. 4, pp. 1,3–8, Dec. 2001.
- [7] E. A. Ratzler, "Error-correction on non-standard communication channels," Ph.D. dissertation, University of Cambridge, 2003.
- [8] A. Bennatan and D. Burshtein, "On the application of LDPC codes to arbitrary discrete-memoryless channels," *IEEE Trans. Inf. Theory*, vol. 50, no. 3, pp. 417–438, 2004.
- [9] M. Franceschini, G. Ferrari, and R. Raheli, "Does the performance of ldpc codes depend on the channel?" *IEEE Trans. Commun.*, vol. 54, no. 12, pp. 2129–2132, 2006.
- [10] I. Sason and B. Shuval. (2011, Jan.) On universal LDPC codes over memoryless symmetric channels. Submitted to *IEEE Trans. Inf. Theory*. [Online]. Available: http://www.ee.technion.ac.il/people/sason/universal_LDPC.pdf
- [11] G. Böcherer and R. Mathar, "Matching dyadic distributions to channels," in *Proc. Data Compression Conf.*, 2011.
- [12] R. G. Gallager, *Principles of Digital Communication*. Cambridge University Press, 2008.
- [13] S. Verdú, "On channel capacity per unit cost," *IEEE Trans. Inf. Theory*, vol. 36, no. 5, pp. 1019–1030, 1990.
- [14] S. Arimoto, "An algorithm for computing the capacity of arbitrary discrete memoryless channels," *IEEE Trans. Inf. Theory*, vol. 18, no. 1, pp. 14–20, 1972.
- [15] T. Richardson and R. Urbanke, *Modern Coding Theory*. Cambridge University Press, 2008.
- [16] M. Mitzenmacher. (2008) A survey of results for deletion channels and related synchronization channels. [Online]. Available: <http://www.eecs.harvard.edu/~michaelm/postscripts/DelSurvey.pdf>
- [17] M. Blaum, R. D. Cideciyan, E. Eleftheriou, R. Galbraith, K. Lakovic, T. Mittelholzer, T. Oenning, and B. Wilson, "High-rate modulation codes for reverse concatenation," *IEEE Trans. Magn.*, vol. 43, no. 2, pp. 740–743, 2007.
- [18] G. Böcherer, "Geometric huffman coding," <http://www.georg-boecher.de/ghc>, Dec. 2010.