Use of a $d$-Constraint During LDPC Decoding in a Bliss Scheme

Andries P. Hekstra, Member, IEEE

Abstract—Bliss schemes of a run length limited (RLL) codec in combination with an LDPC codec, generate LDPC parity bits over a systematic sequence of RLL channel bits that are inherently redundant as they satisfy e.g. a $d = 1$ minimum run length constraint. That is the subsequences consisting of runs of length $d = 1$, viz. $\ldots 010 \ldots$ and $\ldots 101 \ldots$, cannot occur. We propose to use this redundancy during LDPC decoding in a Bliss scheme by introducing additional $d$-constraint nodes in the factor graph used by the LDPC decoder. The messages sent from these new nodes to the variable or codeword bit nodes exert a “force” on the resulting soft-bit vector coming out of the LDPC decoding that give it a tendency to comply with the $d$-constraints. This way, we can significantly reduce the probability of decoding error.

Index Terms—Bliss schemes, RLL codes, LDPC codes, factor graph, modified concatenation.

I. INTRODUCTION

BLISS schemes [1], also called modified concatenation schemes by Fan [2], place a modulation encoder and decoder as outer parenthesis and a systematic ECC encoder and decoder as inner parenthesis around the storage channel. This way, the modulation decoder follows the ECC decoder, rather than precedes it, thus, avoiding error propagation by the modulation decoder at the input of the ECC decoder as in the standard concatenation scheme [2]. As shown in Fig. 1, the parity bits generated from the sequence of modulation encoded channel bits are encoded with a second modulation encoder, where both modulation encoders are of the run length limited (RLL) type [3].

The so called 1T-precoder in Fig. 1 performs an integration modulo-2 in order to convert differential bits, also called $d,k$-bits within the context of RLL coding, to so called unipolar bits, that are indicative of the type of run on the storage medium. The channel SISO detector is shared between the systematic and the parity part of the LDPC codeword and produces soft-decision values for the unipolar bits. The equalizer filter shortens the channel impulse response. The SISO RLL decoder operates on these soft-decision unipolar bits produced by the channel SISO detectors. Hence, the number of states of the RLL SISO decoder is twice the number expected if differential bits were used, as the state incorporates an additional polarity bit\(^1\).

Bliss schemes have pros and cons in comparison with a standard concatenation scheme:

- **Pro:** In the Bliss scheme, the bit error rate (BER) at the input of the LDPC decoder, at least for the dominating systematic part, is the bare BER after bit-detection, which has not been multiplied by the error propagation factor of the RLL SISO decoder.
- **Con:** The error correction capability of the LDPC code is characterized by its code rate $R_{LDPC}$ (and secondarily by its block length $N$). The RLL encoding of the systematic part increases the number of bits into the ECC parity generator. For instance, for the same LDPC code rate $R_{LDPC}$ with a rate 2/3 RLL code, as applies for the $d = 1$ case, the parity generator produces 1.5 times more parity bits, due to the RLL encoding of the systematic bit sequence. This additional amount of parity lowers the net rate of the Bliss scheme.

In order to mitigate the rate loss of Bliss schemes due to the parity generation over redundant data as mentioned above, Im-mink [7] introduced a compression codec in Bliss’ scheme. For storage channels with jitter noise on the transitions between run lengths, Zhang et al. [8] reduced the error propagation of the compression decoder for a $d = 5$ RLL code by the use of Gray labeling. In this contribution, we take a different approach. We accept the redundancy of the systematic channel bit sequence, and modify the LDPC decoder to exploit this redundancy during its decoding iterations, in order to improve its error correction performance.

Our interest is in the use of the same RLL codes by both modulation encoders and low density parity check (LDPC) codes [4] as ECC codes with a high code rate (e.g. $R_{LDPC} = 0.90 - 0.95$). Especially, in near-field optical storage as envisaged for the fourth generation of optical recording, RLL constraints with a minimum run length constraint of $d = 1$ are still quite popular as evidenced by the recent RLL code design of Coene et al. [5]. In [6], we discussed a stitching technique to connect a systematic part of a codeword to its adjacent parity part.

\(\text{\footnote{The use of a unipolar RLL SISO as in Fig. 1 gives a moderate bit error performance improvement over a channel SISO that outputs soft differential bits and an RLL SISO that with half the number of states. Even better bit error performance is possible by combining the channel SISO and the RLL SISO in one joint SISO with the product space of the state space of the channel SISO and RLL SISO.}}\)

Andries Hekstra is with NXP Semiconductors, High Tech Campus 32, 5656 AE Eindhoven, The Netherlands. This work was done while at Philips Research. Email: andries.hekstra@nxp.com
II. MIN-SUM LDPC DECODING REVISITED

The min-sum LDPC decoding algorithm [9] is a simplification of the sum-product algorithm that uses minimum and summation operations instead of multiplication and summation operations. Both algorithms are a special case of the message passing algorithm [10] that passes messages along the edges of a so called factor graph. These messages convey soft-decision information, e.g. in the form of log-likelihood ratios (LLR). The min-sum algorithm always uses a log-likelihood representation of the messages. For a binary random variable with probability \( p_0 \) (\( p_1 \)) of taking on the value 0 (1), the log-likelihood equals \( \log(p_0/p_1) \). With a properly chosen scaling factor after the minimum operation, the performance loss of the min-sum algorithm w.r.t. the sum-product algorithm is minimal [11] [12]. The choice of the codeword bit node or “variable node” degrees in the factor graph [10] of the LDPC code seems to have only a minor influence for high LDPC code rate.

Let \( N \) be the LDPC\(^2 \) codeword length and \( M \) be the number of parity check equations. The parity check matrix \( H \) consists of \( M \) rows and \( N \) columns with elements from the binary Galois field \( GF(2) \). For a parity check equation with index \( m \), \( 0 \leq m < M \), define the set \( \mathcal{N}(m) \) of codeword bit positions that it checks, i.e.

\[
\mathcal{N}(m) = \{ n \mid n = 0, 1, \ldots, N-1, H_{mn} \neq 0 \}.
\]

Similarly, for a codeword bit position \( n \), \( 0 \leq n < N \), define the set \( \mathcal{M}(n) \) of indices of parity check equations that check the given bit position \( n \), i.e.

\[
\mathcal{M}(n) = \{ m \mid m = 0, 1, \ldots, M-1, H_{mn} \neq 0 \}.
\]

The factor graph associated with the parity check matrix \( H \) has as set of vertices \( V \), the union of the set of \( N \) bit nodes \( V_{\text{var}} \) and a set of \( M \) parity check nodes \( V_{\text{check}} \). The set of edges \( E \) consists of all edges \((m, n)\) for which \( H_{mn} \neq 0 \). For the sake of ease of discussion, from hereon, we assume that the LDPC code is regular, which means that all sets \( \mathcal{N}(m) \) have the same size \( K \), and that all sets \( \mathcal{M}(n) \) have the same size \( J \).

In general, the maximum likelihood or hard-decision estimate of a binary random variable is determined by the sign of its log-likelihood value. The absolute value of a log-likelihood value is a reliability of the hard-decision estimate. A log-likelihood value of zero corresponds to an erasure (no information).

We now discuss the various operators involved in the min-sum algorithm. The operator following \( \text{VAR} \) that is applied inside the codeword bit nodes or “variable nodes” combines a number of sources of information (log-likelihood messages) about the bit associated with a variable node. These sources of information are assumed to be statistically independent. This assumption of statistical independence\(^3 \) translates into a sum operator on the input log-likelihoods \( \Lambda_i \), \( i = 0, 1, \ldots, J - 1 \) of a variable node.

\[
\text{VAR}(\Lambda_0, \Lambda_1, \ldots, \Lambda_{J-1}) = \Lambda_0 + \Lambda_1 + \ldots + \Lambda_{J-1}.
\]

The operator \( \text{CHK} \) that is used inside the check nodes approximates the log-likelihood value of the exclusive-or “\( \oplus \)” of its presumably statistically independent input variables.

\[
|\text{CHK}(\Lambda_0, \Lambda_1, \ldots, \Lambda_{K-1})| = \alpha \min(|\Lambda_0|, |\Lambda_1|, \ldots, |\Lambda_{K-1}|) 
\]

signBit(\text{CHK}(\Lambda_0, \Lambda_1, \ldots, \Lambda_{K-1})) = \text{signBit}(\Lambda_0) \oplus \text{signBit}(\Lambda_1) \oplus \ldots \oplus \text{signBit}(\Lambda_{K-1})
\]

where \( 0 < \alpha \leq 1 \) is the aforementioned scaling factor and

\[
\text{signBit}(x) = \begin{cases} 
0 & \text{if } x \geq 0, \\
1 & \text{if } x < 0.
\end{cases}
\]

For a given iteration of the min-sum algorithm, we define the following variables. As usual, in the message passing algorithm, messages are sent along edges of the factor graph.

- \( t_n \) – The decoder input message into variable node \( n \).
- \( u_{mn} \) – The message sent from variable node \( n \) to check node \( m \). It is obtained as a function \( \text{VAR} \) of message \( t_n \), and the last received messages \( v_{mn} \) of all check nodes \( m', m' \neq m \),

\[
u_{mn} = \text{VAR}(t_n, \{ v_{mn'} \mid m' \in \mathcal{M}(n) \setminus \{m\} \}). \tag{3}
\]

- \( v_{mn} \) – The message sent from check node \( m \) to variable node \( n \). It is obtained as a function \( \text{CHK} \) of the last received messages \( u_{mn} \) of all variable nodes \( n', n' \neq n \),

\[
v_{mn} = \text{CHK}(\{ u_{mn'} \mid n' \in \mathcal{N}(m) \setminus \{n\} \}). \tag{4}
\]

- \( w_n \) – The decoder output messages. Unlike the messages \( u_{mn} \) the decoder output message \( w_n \) uses all available information in a variable node \( n \). It is obtained as the function \( \text{VAR} \) of the message \( t_n \) and the last received messages \( v_{mn} \) of all check nodes \( m \),

\[
w_n = \text{VAR}(t_n, \{ v_{mn} \mid m' \in \mathcal{M}(n) \}). \tag{5}
\]

A classical implementation of the min-sum algorithm stores all received messages. During the first half-iteration, all messages \( u_{mn} \) are sent from all variable nodes to the check nodes. During the second half-iteration, all messages \( v_{mn} \) are sent from all check nodes to the variable nodes. A constant number of iterations can be used. The decoder output messages need not be evaluated for all iterations, but only for the final iteration.

III. \( d = 1 \)-CONSTRAINT NODES IN THE FACTOR GRAPH

Our approach is to modify the LDPC decoder so that it can use the knowledge that in the systematic channel sequence the subsequences \( 0^p0 \) and \( 1^p1 \), \( p = 1, 2, \ldots, d \) do not occur. Here, \( d \) denotes a sequence consisting of \( d \) copies of the value \( a \), \( a = 0, 1 \). This is achieved by addition of so called \( d \)-constraint nodes in the LDPC code’s factor graph. As stated

\(^2\)Our method may also apply to other, similar codes with a low density parity check matrix, such as repeat-accumulate (RA) codes.

\(^3\)Due to the presence of cycles in the factor graph, this is only an approximation.
before, our aim is to improve the error correction capability of the LDPC decoder. In the definition of a new, modified factor graph for the LDPC decoder in a Bliss scheme, it is very common that the d-constraint is a local constraint [10]. This means, that the constraint involves only a fixed, small number d + 2 of codeword bits (codeword bit nodes). This sparseness of the connectivity matrix between the nodes is essential for efficient and effective operation of the modified LDPC decoder. This is all the more true, the smaller the value of d, d > 0 is. For this reason, and for the practical importance of their applications [5], from hereon we concentrate on the d = 1-based Bliss schemes.

The new factor graph with the d-constraint nodes is shown in Fig. 3. In general, the degree of a d-constraint node equals d + 2, as these nodes need to be able to detect the presence of the subsequences 01^d0 and 10^d1 of length d + 2. Define the following additional variables in the LDPC decoding algorithm.

- $a_{mp} = \text{The input message to d-constraint node } p \text{ sent from variable node } n$.
- $b_{np} = \text{The output message of d-constraint node } p \text{ sent to variable node } n$.

Here, the index n of the variable node (LDPC codeword variable node) is implicitly restricted to the systematic bit positions as there is no d-constraint on the parity part of an LDPC codeword.

As a general principle in message passing algorithms, the output message along a certain arc from a certain node is only allowed to depend on the (most recently) received input messages via all other arcs into that node. Hence, for d = 1-constraint node with index n, and most recently received log-likelihood messages $a_{n-1,n}$, $a_{n,n}$ and $a_{n+1,n}$, we need to specify three output functions $CO_0, CO_1$ and $CO_2$ in order to generate as many log-likelihood output messages $b_{n-1,n}, b_{n,n}, b_{n+1,n}$, such that

$$b_{n-1,n} = CO_0(a_{n,n}, a_{n+1,n})$$
$$b_{n,n} = CO_1(a_{n-1,n}, a_{n+1,n})$$
$$b_{n+1,n} = CO_2(a_{n-1,n}, a_{n,n}).$$

Observe that, if one of the input log-likelihood values $a_{n-1}, a_{n}, a_{n+1}$ of a d = 1-constraint node is zero, there is no indication that the d = 1-constraint is violated, as it is not possible to conclude to a violation from the knowledge of fewer than d + 2 = 3 hard-decision estimates. Then, the output log-likelihood messages are to be zero. The output messages do not need to exert a force on the solution of the LDPC decoder in that case. On the contrary, if the input log-likelihoods have large absolute values, and their hard-decision values indicate a violation of the d-constraint, the output values also should have large absolute values. However, if the hard-decision values indicate compliance with the d-constraint, the output log-likelihoods should be zero. Hence, in the spirit of the min-sum algorithm we choose to let

$$|CO_r(x, y)| \in \{0, \min(|x|, |y|)\}, \ r = 0, 1, 2.$$  

Similar to Eq. (2), one can use a scaling factor to post-multiply the above minimum values. 

Also the sign of $CO_0$ is not allowed to depend on $a_{n-1,n}$, etc. The signs are chosen such that they enforce the disappearance of violations of the d = 1 constraint in the decoded LDPC codeword, see Tables I, II and III. As stated before, a zero entry in these tables applies when there is no violation of the d-constraint.

IV. MIN-SUM LDPC DECODING WITH d = 1-CONSTRAINT NODES IN THE FACTOR GRAPH

With reference to the above definitions of log-likelihood messages, one iteration of the extension of the min-sum LDPC decoding algorithm with d = 1-constraint nodes in the factor graph is given by the following equations.

$$u_{mn} = \text{VAR}(t_n, (v_{m'n'} | m' \in \mathcal{M}(n) \setminus \{m\}),$$
$$b_{n-1,n}, b_{n,n}, b_{n+1,n})$$
$$a_{n-1,n} = \text{VAR}(t_n, (v_{m'n'} | m' \in \mathcal{M}(n)),$n,n, b_{n+1,n})$$
$$a_{n,n} = \text{VAR}(t_n, (v_{m'n'} | m' \in \mathcal{M}(n)),$$
$$b_{n-1,n}, b_{n+1,n})$$
$$a_{n+1,n} = \text{VAR}(t_n, (v_{m'n'} | m' \in \mathcal{M}(n)),$n,n, b_{n+1,n})$$
$$b_{n-1,n} = CO_0(a_{n,n}, a_{n+1,n})$$
$$b_{n,n} = CO_1(a_{n-1,n}, a_{n+1,n})$$
$$b_{n+1,n} = CO_2(a_{n-1,n}, a_{n,n})$$
$$v_{mn} = \text{CHK}(v_{m'n'} | n' \in \mathcal{N}(m) \setminus \{n\})$$
$$w_n = \text{VAR}(t_n, (v_{m'n'} | m' \in \mathcal{M}(n))$$

(6)

V. SIMULATION RESULTS

We experimented with two pseudo-random regular LDPC codes. The codeword bit nodes or variable bit nodes all had degree $J = 3$. The shorter LDPC code had a code rate of $R_{LDPC} = 0.906$ and a code length of $N = 1728$. The bit error rates after LDPC decoding obtained from simulations using this shorter code are shown in Figs. 4 and 5. In Fig. 4, the peak signal-to-noise ratio (PSNR) of white additive noise at the channel output is varied for a fixed channel bit length of 53 nm, scaled to the numerical aperture and wavelength of Blu-ray disc [13]. The user capacity of the disc scaled to the physics of Blu-ray disc then equals 30.5 GB. The simulations used a channel model of the optical storage readout channel using the channel modulation transfer function from the Brault-Hopkins formalism [14]. The channel bit length is scaled to the physical readout parameters of Blu-ray disc. In Fig. 5, channel bit length is varied, at a fixed PSNR of 35 dB. The $d = 1$ RLL code of [5] is used, that has code rate $R_{RLL} = 2/3$. 

\[ \text{Note: a } d = 1 \text{ constraint itself does not imply a non-trivial Hamming distance } D, \ D > 1. \text{ For instance the } d = 1 \text{ constrained sequences } 00...01100...00 \text{ and } 00...01110...00 \text{ evidently have Hamming distance } D = 1. \text{ However, for the kind of intermediate range of target bit error rates after LDPC decoding in our storage application a high minimum distance of the LDPC code is not required for good or even superior error correction performance, anyway.} \]
The number of LDPC iterations was set at 16. We use the schedule of Yeo et al. [15], wherein a single check node or d-constraint node update is followed by variable updates of the connected variable nodes as a kind of “mini decoding iterations.” We process the check nodes and the d-constraint nodes in sequence. An LDPC decoding iteration is then complete when all check nodes and all d-constraint nodes have undergone an update. This update schedule about halves the number of iterations that need to be performed in a decoder. Up to half a million LDPC codeword have been simulated per point of the graph.

In all graphs presented in this section, the top curve shows the BER after bit-detection. The middle curve shows the BER after standard LDPC decoding. The bottom curve shows the BER with decoding using the additional RLL-constraint nodes in the factor graph.

The longer LDPC code has a code rate of $R_{LDPC} = 0.955$ and a code length of $N = 6912$. The BER simulation results for this longer code are shown in Fig. 6 and 7.

For a physical sector size of the storage medium of approx. 1 Mbit, the shorter LDPC code length allows combination with an 10-bit outer RS code with about maximal length $(2^{10} - 1)$ symbols. The longer LDPC code allows combination with an 8-bit outer RS code of about maximal length $(2^8 - 1)$.

VI. CONCLUSION

We conclude that the use of a $d = 1$-constraint during the decoding of an LDPC-based Bliss scheme, brings an advantage in PSNR of around 0.25 dB. Using this technique, the bit length of a simulated model of an optical storage channel can be decreased by ca. 0.6 %. This suggests that the use of $d = 1$-constraint during LDPC decoding can enable an increase in the storage density of around 0.6 %.

ACKNOWLEDGMENT

These results were obtained during a joint project between Philips Research and Sony. The author is indebted to his former colleague Haibin Zhang, now with TNO Telecom, for the construction of the LDPC codes used in this research. Furthermore, the author would like to acknowledge stimulating discussions with his former colleagues Stan Baggen, Wim Coene and Bin Yin as well as his project members from Sony, Seji Kobajashi, Toshi Horigome, Makato Noda and Hiroyuki Yamagishi.

REFERENCES

[1] W. G. Bliss, “Circuitry for performing error correction calculations on baseband encoded data to eliminate error propagation,” IBM Techn. Discov. Bull., vol. 23, pp. 4633-4634, 1981.

[2] J. L. Fan, Constrained Coding and Soft Iterative Decoding, Norwell, MA: Kluwer Academic Publishers, 2001, pp. 144-145.

[3] K. A. S. Immink, Codes for mass data storage systems, 2nd ed., Eindhoven, The Netherlands: Shannon Foundation Publishers, 2004, pp. 51-64.

[4] R. G. Gallager, Low-Density Parity-Check Codes, Cambridge, MA: MIT Press, 1963.

[5] W. M. J. Coene, A. Hekstra, B. Yin, H. Yamagishi, M. Noda, A. Nakaoki, and T. Horigome, “A new d=1, k=10 soft-decodable RLL code with r=2 RMTR-constraint and a 2-to-3 PCWA mapping for DC-control,” in Proc. of SPIE, Vol 6282, Optical Data Storage 2006, R. Katayama, R. Schlesinga, editors.

[6] A.P. Hekstra, W.M.J. Coene, R.I.W Debets, “A Stitching Technique for Bliss Schemes,” submitted to IEEE Tr. on Magn., May 2007.

[7] K. A. S. Immink, “A practical method for approaching the channel capacity of constrained channels,” IEEE Trans. Inform. Theory, vol. 43, no. 5, pp. 1389-1399, Sept. 1997.

[8] H. Zhang, A. P. Hekstra, W.M.J. Coene and B. Yin, “Performance investigation of soft-decodable RLL codes in high density optical recording,” accepted for publication in IEEE Trans. on Magn., 2007. pp. 879-887, Aug. 2006.

[9] M. Fossorier, M. Mihaljevic, H. Imai, “Reduced complexity iterative decoding of low-density parity check codes based on belief propagation,” IEEE Trans. on Commun., Vol. COM-47, No. 5, pp. 673-680, May 1999.

[10] F.R. Kschischang, B. J. Frey, H.A. Loeliger, “Factor Graphs and the Sum-Product Algorithm,” IEEE Trans. Inf. Th., Vol. IT-47, No. 2, Feb. 2001.

[11] J. Chen, M. Fossorier, “Near optimum universal belief propagation based decoding of low density parity check codes,” IEEE Trans. on Commun., Vol. COM-50, No. 3, pp. 406-414, March 2002.

[12] J. Hess, “Analysis of scaling soft information on low density parity check code,” Electron. Lett., vol. 39, no. 2, pp. 219-221, Jan. 2003.

[13] T. Narahara, S. Kobayashi, M. Hattori, Y. Shimpu, K. van den Enden, J. Kahlman, M. van Dijk, R. van Woudenberg, “Optical disc system for digital video recording,” Proc. ISOM/ODS 1999, Hawaii, Jul. 1999, SPIE Vol. 3864, pp. 50-52.

[14] J. Braat, Principles of Optical Disc Systems: Chapter2, Read-out of Optical Discs, Adam Hilger, Ltd, 1985.

[15] E. Yeo, B. Nikolic, V. Anantharam, “Architectures and implementations of low-density parity check decoding algorithms,” The 2002 45th Midwest Symp. on Circuits and Systems, Vol. 3, Aug. 2002, pp. III-437-III-440.

Andries P. Hekstra (M’00) was born in Breda, the Netherlands in 1961. He received the "Ingenieur" degree in Electrical Engineering from Eindhoven University, the Netherlands, summa cum laude in 1985 with a specialization in multi-user information theory. In 1985-86 he was a Young Graduate Trainee at the European Space Agency where he worked on spread spectrum telemetry systems in Darmstadt, Germany. Successfully, he was a Ph.D. student at the Electrical Engineering Department of Cornell University, Ithaca, USA. There, he studied abstractions of VLSI packing problems. In 1990 he joined KPN Research in Leidschendam, the Netherlands, where he finished his Ph.D. in 1994 at Eindhoven University of Technology with his professor from Cornell University as second promoter. From 1995 to 2000 he investigated automatic assessment of video and speech quality using models of human perception and cognition. During 2001-2006, he worked at Philips Research, Eindhoven, mainly on error correction for optical storage systems. Since Sept. 2006, he works on applied communication and information theory at large within the research department of NXP Semiconductors, Eindhoven.
Fig. 1. Block diagram of a Bliss scheme.

Fig. 2. Factor graph of an LDPC code. Variable nodes are located in the middle. At the left are the input nodes, representing the channel output messages. Check nodes are situated at the right.

Fig. 3. Factor graph of an LDPC code with additional $d$-constraint nodes for the case of $d = 1$.

Fig. 4. Simulated BER curves for an LDPC-based Bliss scheme with the $J = 3$-regular LDPC code is $R_{LDPC} = 0.906$ and the code length $N = 1728$. 

### TABLE I

| $a_{n-1,n}$ | $a_{n,n}$ | $b_{n,n} = CO_1(a_{n-1,n}, a_{n+1,n})$ |
|-------------|-----------|-----------------------------------|
| +           | +         | 0                                 |
| +           | -         | 0                                 |
| -           | +         | 0                                 |
| -           | -         | 0                                 |

### TABLE II

| $a_{n-1,n}$ | $a_{n,n}$ | $b_{n+1,n} = CO_2(a_{n-1,n}, a_{n,n})$ |
|-------------|-----------|-----------------------------------|
| +           | +         | 0                                 |
| +           | -         | -                                 |
| -           | +         | +                                 |
| -           | -         | 0                                 |

### TABLE III

| $a_{n,n}$ | $a_{n+1,n}$ | $b_{n-1,n} = CO_0(a_{n,n}, a_{n+1,n})$ |
|-----------|-------------|-----------------------------------|
| +         | +           | 0                                 |
| +         | -           | -                                 |
| -         | +           | +                                 |
| -         | -           | 0                                 |
Fig. 5. Similar graph as Fig. 4, where the peak signal-to-noise ratio (PSNR) is fixed at 35 dB, and the bit-length of the channel bits is varied.

Fig. 6. Similar graph as Fig. 4, with a longer LDPC code (length 6912 bits) of a higher code rate 0.955.

Fig. 7. Similar graph as Fig. 5, also with a longer LDPC code (length 6912 bits) of a higher code rate 0.955, as in Fig. 6.