NEW DETERMINISTIC ENCODING-SOFT DECODING APPROACH FOR SHORT LENGTH LT CODES OVER AN AWGN CHANNEL

Article *in* Ciência e Técnica Vitivinícola · January 2017

Nadhir Ibrahim Abdulkhaleq
UNIVERSITY OF INFORMATION TECHNOLOGY and COMMUNICATIONS(UoITC)

NEW DETERMINISTIC ENCODING-SOFT DECODING APPROACH FOR SHORT LENGTH LT CODES OVER AN AWGN CHANNEL

NADHIR I. ABDULKHALEQ[1,2] ORHAN GAZI[1]

C1282650@student.cankaya.edu.tr o.gazi@cankaya.edu.tr

[1] DEPARTMENT OF ELECTRONIC AND COMMUNICATION ENGINEERING -CANKAYA UNIVERSITY ANKARA-TURKEY

[2]DEPARTMENT OF ELCTRONIC TECHNIQUES-TECHNOLOGY INSTITUTE MIDDLE TECHNICAL UNIVERSITY-BAGHDAD-IRAQ

ABSTRACT

Luby transform (LT) and Raptor codes are an effectual solution for distributing bulky data files in broadcasting scenarios. For an erasure channel, like the internet communication system, these codes are used as standard codes for many applications. In this paper, we propose an LT code using deterministic degree generators. The degree values will be generated in a sequential way with a repetition period ($\boldsymbol{R_p}$), these degrees decide on the number of data packets (or symbols) which have to be combined to form the coded packets. The data packets will be truncated in segments of length ($\boldsymbol{R_p}$) and will be chosen serially. We exploit this deterministic encoding method for short length LT code over an additive white Gaussian noise (AWGN) channel and decode it using a new soft decoding algorithm based on maximum likelihood probabilities. The decoding process does not need any matrix variation which will result in decreasing the decoding complexity which is one of the important performance factors for such code length. The simulation results prove the superiority of this encoding approach over that of LT code generated using robust Soliton distribution (RSD) and decoded by belief propagation (BP) assisted by Gaussian elimination (GE)method which is one of the best decoding treatment for short length LT codes.

KEYWORDS

LT codes, deterministic degree generators, soft decoding, maximum likelihood probabilities AWGN channel.

1. INTRODUCTION

During the last decade, there has been a noticeable interest in improving the performance of rateless codes. Two well-known rateless codes are LT [1] and Raptor [2]. The first attempt to invent such codes was through the design of LT codes, which are considered to be capacity approaching erasure codes. The design of LT codes fits into the environment of an erasure channel with a high degree of reliability, while applying such codes in a noisy channel, like an AWGN, poses considerable challenges. Many types of treatment for such performance degradation

have been presented. One of the first was the treatment of Shokrollahi [2] in his invention of modified LT codes, known as Raptor codes. The core idea of his new code was to concatenate the conventional LT code with a pre-code, like low density parity-check codes (LDPC) [3] . Raptor codes perform much better than that of conventional LT codes, not only in the erasure channel, they also record a lower error floor in the noisy channels [4]. The resultant decoding matrix of Raptor codes has been suitable for applying the powerful iterative belief propagation (BP) algorithms, which are mainly designed for LDPCs [5]. The efficiency of this algorithm succeeds in deciding on the actual values for the corrupted received coded symbols, while recovering the source data symbols has to overcome the problem of early termination of the decoding process due to missing degree one coded symbols [6-8].

Two successful manoeuvres have been utilised to enhance the performance of LT codes in noisy channels. The first was to expand the idea of Raptor codes, which led to new class of systematic LT codes. These codes were using a soft decoding approach based on calculating the logarithmic likelihood ration (LLR) that was used in LDPCs [9]. With some modification of the robust soliton degree distribution (RSD) [1], these systematic LT codes succeeded in preventing high symbol error propagation. They then recorded super error floors at low signal to noise ratios. Partial systematic encoding develops a modified systematic LT code in [10], which examine different percentages of systematic encoding for the source data. Another modification was presented by M. Zhang and S. Kim [11]. They construct the tanner graph of the LT code to contain bit nodes and check nodes, instead of encoding nodes and information nodes as in the conventional Tanner graphs for LT codes. With such adaptations, the decoding matrix will be suited for iterative soft decoding. In the case of small numbers of input symbols, just like our research case, the second type of treatment for decoding LT codes over an AWGN channel were proposed. Cheong et al [12] utilised a BP decoding algorithm assisted on-the-fly Gaussian elimination (OFG). The study tries to overcome the problem of missing degree one coded symbols by using xoring operations for certain coded symbols to reproduce a degree one coded symbol again. With such treatment, they approve the reduction of the required overhead, as well as enhance the decoding complexity. Merging both the decoding matrix edition and Gaussian elimination support was the idea presented by Amrit Kharel and Lei Cao in [13]. The decoding matrix has been constructed to be $H = [G, I]$ where $G = (N \times k)$ is the generator matrix of LT code and $I = (N \times N)$ identity matrix will be used to introduce new variable nodes that represent the unrecovered data symbols. Using this technique, the decoding matrix will again be well prepared to use the soft decoding of the LDPC iterative decoding. When the decoder stuck and the ripple box was empty they used the GE method to reproduce a degree one coded symbol and resume decoding.

It is obvious that, when dealing with LT codes in an AWGN channel, the prementioned studies [9-14] try to overcome the problems of error propagation by adapting the decoding matrix of the LT code so that it is fitted with that of LDPC

codes in order to use efficient iterative soft decoding. But this efficient decoding entails large complexity due to its iterative nature. In this study, we propose a novel solution by encoding the information symbols using a deterministic generator matrix. This encoding produces a mutual relation between coded symbols that can be used to recover data symbols using maximum likelihood probabilistic decoding. With such an idea, we claim to reduce decoding complexity and minimise the extra needed overhead for complete data recovery.

The rest of the paper will be organised as follows: in section 2 the system model and the proposed deterministic encoding with a soft decoding approach will be presented. The simulation results will be shown in section 3. Finally, the conclusion will be summarised in section 4.

2. SYSTEM MODEL

The scenario of any communication system has three main locations: the transmitter, the channel and the receiver. All these elements will be used with equal potential to compare the performance of the proposed deterministic encoding with maximum likelihood probabilistic decoding to that of LT code using random RSD encoding with soft demodulation-hard decoding. The coded symbols will be transmitted through an AWGN channel with Gaussian distribution characterised by noise samples vector of zero - mean with variance $\sigma^2 = N_o/2$, where N_o is the one-sided noise power spectral density. The following sections will be used to describe the three system elements in further detail.

2.1. THE TRANSMITTER

This section of the communication system will be used to adapt the information file so that it is suitable for transmission through the communication channel. First, the message under consideration for transmission will be sliced into packets (in this paper we deal with each packet as a single bit and named as a symbol). Second, the symbols will be encoded using a conventional LT encoding technique. Finally, these encoded symbols will be modulated using a binary phase shift keying (BPSK) scheme. Let us consider that we have truncated the file into distinct data symbols, and then the following algorithm describes the conventional LT encoding steps [1]:

Algorithm 1: LT encoding
Repeat
1: Randomly, choose a number (called a degree (d)) from a known
degree distribution.
2: Uniformly at random select d data symbols.
3: Combine the selected data symbols using XOR binary operation to
form the coded symbol.
Until Acknowledgement of full file recovery has been received

It is obvious from this algorithm that the encoder can generate an endless code length (N). This feature gives the name of a rateless code for such types of codes, since the rate of the code $(R = k/N)$ has a known data length k, while the code length will be determined after the receiver acknowledgement. The most important factor in designing an LT code is the choice of degree distribution. In this paper, we generate the coded symbols using two degree generators:

2.1.1 ROBUST SOLITON DISTRIBUTION (RSD)

The degrees for the LT code will be generated according to probability distribution function, represented by $\Omega(d)$, to be [1]:

$$\Omega(d) = \frac{\rho(d) + \tau(d)}{\beta} \tag{1}$$

$\rho(d)$ are the ideal soliton distributions (ISD) [1]:

$$\rho(d) = \begin{cases} \frac{1}{k} & d = 1 \\ \frac{1}{d(d-1)} & d = 2,3 \cdots, N \end{cases} \tag{2}$$

and the probability function $\tau(d)$ has been added to enhance the degree generation of the ISD by increasing the expected number of degree one coded symbols and inserting a high probability for high degree to ensure the contribution of all data symbols in the encoding process, $\tau(d)$ will be represented as:

$$\tau(d) = \begin{cases} \frac{S}{k}\frac{1}{d} & d = 1,2, \cdots, \frac{k}{S} - 1 \\ \frac{S}{k}\ln\left(\frac{S}{\delta}\right) & d = \frac{k}{S} \\ 0 & d > \frac{k}{S} \end{cases} \tag{3}$$

where S denotes the expected number of degree one coded symbols. It will be calculated as:

$$S = c.\ln\left(\frac{k}{\delta}\right)\sqrt{k} \tag{4}$$

and c is a positive constant such that $c \leq 1$, while δ is the probability of the decoding failure. The normalisation factor βwill be given by:

$$\beta = \sum_d \rho(d) + \tau(d) \tag{5}$$

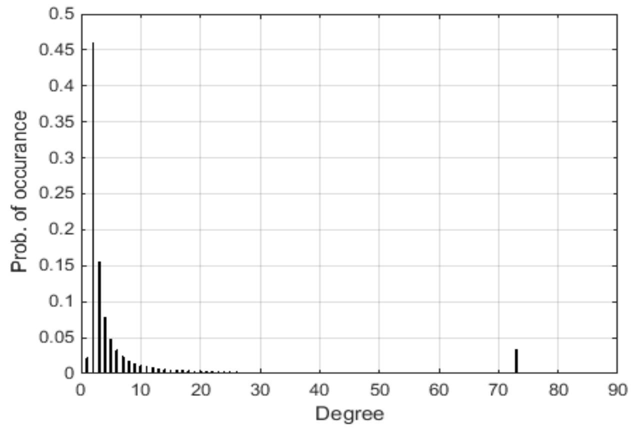

Fig. 1. The distribution $\Omega(d)$ for the case $k = 100, c = 0.02$ and $\delta = 0.1$

In Fig. 1 below, the chance of using each degree in this distribution will be determined by its probability. This distribution is mainly designed for large data sizes; however, in this paper we shall examine its performance with short data lengths like $k = 50{,}100$ and 300.

2.1.2 DETERMINISTIC ENCODING APPROACH

In this type of degree generator, we tend to overcome some problems that occur when dealing with short data lengths. As will be explained later, the degree one coded symbol is the essential trigger for starting and surviving the decoding process of LT code. Even when RSD induced its new function $\tau(d)$ with that of ISD, the performance of the RSD still faces early termination of the decoding process because of a missing degree one coded symbol. The proposed degree generator succeeds with high probability in overcoming this important decoding obstacle. The degrees of this generator will be bounded by $1 \leq d \leq R_p$, where $R_p$is a positive integer that represents the repetition period for generating the same degrees. The value of this repetition integer has to be one of the data length multiples and could be chosen within the limit $3 \leq R_p \leq k/2$. The choice of the data symbols for each degree will be done sequentially; for example, if $k = 12$ is the length of a message $(u_1\ u_2\ u_3 \cdots u_{12})$ with $R_p = 4$ the generator matrix will contain three non-overlapped triangles, each triangle has degrees (1,2,3and 4) and the data symbols for each coded symbol will be assigned as:

$$\left.\begin{aligned} &C_1 = u_1 \\ &C_2 = u_1 \oplus u_2 \\ &C_3 = u_1 \oplus u_2 \oplus u_3 \\ &C_4 = u_1 \oplus u_2 \oplus u_3 \oplus u_4 \\ &C_5 = u_5 \\ &C_6 = u_5 \oplus u_6 \\ &\qquad\quad \vdots \\ &C_{12} = u_9 \oplus u_{10} \oplus u_{11} \oplus u_{12} \end{aligned}\right\} \tag{6}$$

From equation (6) we can formulate a mathematical representation for the degree of the coded symbol to be:

$$deg(C_n) = \begin{cases} n, & n = 1,2,\cdots,R_p \\ n - m \cdot R_p, & n = R_p + 1,\cdots,k \end{cases} \tag{7}$$

and m is an integer denoting the number of the repeated degree group that can have a value from the integer set $1,2,\cdots,(k/R_p) - 1$. For our case of $k = 12$, the overhead coded symbols have the same order of the first k group, which can tell us that C_{15} will have a degree of 3 again and will combine the first three data symbols to get the coded symbol value.

The coded symbols that have been generated using the LT encoding algorithm 1 by using one of the prementioned degree generations will be mapped from the $(0,1)$representation to the bilateral case of $(-1,1)$ representation using BPSK modulation. A stream of encoded symbols will be transmitted as a two-sided waveform through the noisy channel.

2.2. THE CHANNEL

The main effect of the AWGN channel model is to add a white Gaussian noise process to the transmitted signal. This addition can be formulated in the discrete time vector representation as [15]:

$$r = s + n \tag{8}$$

Where r represents the received noisy signal, which is a result of adding our modulated coded symbols vector s to a noise vector n created from sampling a Gaussian distribution noise with zero mean and noise variance σ^2 and denoted by $\mathcal{N}(0, \sigma^2)$.

2.3 THE RECEIVER

In order to increase the probability of success in recovering the data file, the number of received noisy symbols N usually exceeds the data length by an extra redundant symbol denoted by ϵ. In the receiver side, two main steps will be achieved. First, a soft demodulation will be employed to compute the probability of the noisy symbol if it is 1 or 0; these probabilities are calculated as:

$$\Pr(C_i = 1\ /y_i) = (1 + e^{-\frac{y_i}{\sigma^2}})^{-1} \tag{8}$$

$$\Pr(C_i = 0\ /y_i) = 1 - \Pr(C_i = 1\ /y_i) \tag{9}$$

Second, these probabilities will be fed to the decoder, which will employ one the following procedures to recover the data symbols.

2.3.1 BELIEF PROPAGATION-ASSISTED GAUSSIAN ELIMINATION APPROACH (BP-GE)

This decoding approach will be applied for the LT code using RSD degree generators. The decoding has two processes: first, the decoder decides on the actual value of the received corrupted symbols using the maximum likelihood probability that depends on the probabilities calculated from (8) and (9); and second, the estimated coded symbols and their matrix will be fed to the BP-GE process. This process employs the GE approach described in [14] which consists of two steps: the triangularisation step and the back-substitution step. If the GE procedure succeeds in having an upper triangle identity matrix then recovering the data symbols will be undertaken using back-substitution; if not, the estimated coded symbols and their matrix will be processed using the conventional BP decoding approach explained in [1].

2.3.2 SOFT SEQUENTIAL DECODING APPROACH (SSD)

The set of equations illustrated in (6), which represents the coded symbols, can be written in another shape in order to be used when recovering the data symbols. It is quite obvious that these equations have been arranged in a repeated manner with a repetition of R_p. For instance, the first set of coded symbol equations can be used to get:

$$u_1 = C_1$$

$$u_2 = C_1 \oplus C_2$$

$$u_3 = C_2 \oplus C_3$$

$$u_4 = C_3 \oplus C_4 \tag{10}$$

With such an arrangement, if the decoder has the opportunity to collect a complete set of k coded symbols then it will be able to recover all the data symbols with only k symbol operations. Keep in mind that the first step in the decoding process was calculating coded symbol probabilities. Now, let $P_{C1}$and P_{C0}denote the $\Pr(C_i = 1\,/y_i)$ and $\Pr(C_i = 0\,/y_i)$ respectively, while P_{u1} and P_{u0} represent $\Pr(u_i = 1\,)$ and $\Pr(u_i = 0\,)$, respectively. Suppose that the decoder was able to collect a sufficient number of noisy symbols ($N \geq k$) equipped with their equations listed in the decoding matrix $G'(N{\times}k)$. Then, the soft sequential decoding can be summarised by the following algorithm:

ALGORITHM 2: SOFT-SEQUENTIAL-DECODING (SSD)

1: **compute** the likelihood probability $P_{C1}$and P_{C0}.

2: **for** $i = 1{:}\,R_p{:}\,k$

3: construct $G1 = G'(:, i{:}\,i + (R_p - 1))$

4: **let** $d_C = set\ of\ degree\ in\ G1$

5: **re-arrange** the coded symbols sequentially according to their degrees

6: **if** any two coded symbols are identical

7: averaging their probabilities and delete the redundant

7: **end if**

8: after re-arrangement inspect each triangle in G' to decide on the data symbol value

9: **if** the coded symbol has degree one

10: copy its probabilities to its connected data symbol

11: **else**

12: $P_{u1}\big(u(i)\big) = (P_{C1}\big(C(i)\big)\times P_{C0}\big(C(i-1)\big) + (P_{C0}\big(C(i)\big)\times P_{C1}\big(C(i-1)\big)$

13: **end if**

14: $P_{u0}(u(i)) = 1 - P_{u1}(u(i))$

After calculating the likelihood probabilities of the data symbols (P_{u1} and P_{u0}), the estimation will be taken on the maximum probability. Algorithm 2 has two main steps: first, the received symbols in the original sequential sequence must be rearranged; and second, the sequential relations between the data symbols and the coded symbols listed in (10) for calculating the probabilities mentioned in algorithm 2 must be employed.

3. SIMULATION RESULTS

In order to test the performance of our sequential encoding approach using the proposed soft decoding, a MATLAB simulation based on algorithm 2 was applied for a data length of ($k = 50{,}100{,}300$). The results gained from our new LT code have been compared with that of an LT code generated using RSD with parameters of ($c = 0.02$ and $\delta = 0.1$). The decoder for the RSD case applies the belief propagation approach assisted by the powerful mean of the Gaussian-Jordon elimination method [14] as a significant tool to enhance the performance of LT code with short length messages. The first test for the two LT codes was done by allowing the receiver to collect an extra coded symbol, then for each SNR point on the graph we recorded the bit error rate (BER) after recording 100 error frames. It is shown in Fig. 2 that the proposed SSD approach outperforms the records of BER for the RSD, even when allowing the former to get 100% extra overhead. It is also clear that the best BER records have been gained from the proposed approach with only 60% overhead symbols.

In Fig. 3, we examine the LT codes with both generator methods by fixing the SNR at two values of 3dB and 5dB, and permitting the receiver to collect a variable number of overhead symbols up to 100%. It is also clear that our proposed SSD approach defeats that of the RSD using BP assisted by the GE approach. The successful decoding ratio has been illustrated in Fig. 4. We use deferent data lengths of 50,100 and 300 by sending 500 frames and test the ability of the decoder to recover all the data symbols by recording the average statistics of the successful recovery. The proposed SSD with a length of $k = 50$ can recover about 45% of the sent frames even with no extra symbols needed and at $SNR = 0dB$, while the LT code with RSD assisted by GE method has null recovery at this point on the graph. As the length of the code increases, the chance of an error propagating will also increase. That is obvious for the case of $k = 100$ and 300; however, with both lengths the SSD approach can achieve 100% recovery at approximately $7dB$.

These records of the successful decoding ratios have a certain number of extra needed symbols. For instance, at $SNR = 6d$B, by comparing the average results from both Fig. 4 and Fig. 5, we found that for the case of $k = 50$,SSD successes in recovering all the sent frames with only need of 0.2 overhead compared to a 87% recovery of the sent frames with the need of 0.58 overhead recorded by RSD-BP-GE. While for the data length of ($k = 100$ and 300) the successful decoding of (99%) of all the sent frames with an overhead of (0.21 and 0.35) extra needed symbols.

It is worth mentioning that, when compared to that of the BP-GE used for the case of RSD, all these superiorities in performance for the proposed SSD approach have been achieved with a noticeable simplicity in the encoding part and less symbol operations while decoding. This simplicity came from the deterministic manner of degree generation with a pre-determined allocation of the data symbols combined to form the coded symbols; however, in the encoding part we have to compromise the value of R_p between a higher value which is better for contributing a greater part of the data frame in the encoding operation and a smaller value in order to get a minimum average degree. It is obvious that the complexity in the decoding part of the SSD requires fewer symbol operations compared to that of the BP-GE. In order to recover k data symbols from an estimated N coded symbols, the SSD only requires $O(k + (N \times (N - 1))$ to recover the data symbols from the sequential order estimated coded symbols, compared to that of $O(N^3)$ required for the BP-GE method.

4. CONCLUSION

A new sequential soft decoding (SSD) method for a deterministic encoding LT code over AWGN channel has been presented in this paper. The approach has been applied for short length data frames of sizes 50,100 and 300. The equation format of the coded symbols provides an interesting, sequential relation between data symbols and the estimated coded symbols which is used perfectly to simplified the decoding part. In the decoding part, we use two main operations. These are

calculating the likelihood probability for the received corrupted coded symbols and then employing these probabilities with the sequential relations to decide on the estimated data symbols. Our results support the superiority of the proposed deterministic encoding with SSD approach by recording lower BER floor and higher successful decoding ratios with less overhead extra needed symbols when comparing its performance with an LT code using RSD for the encoding and the powerful BP-GE method in the decoding part. Also, these achievements have been made with less decoding complexity gained from the simplicity of the estimated data symbols calculations. The complexity price for our proposed approach was $O(k + (N \times (N - 1))$ compared to $O(N^3)$ needs to decode an LT code using RSD and decoded by BP-GE approach.

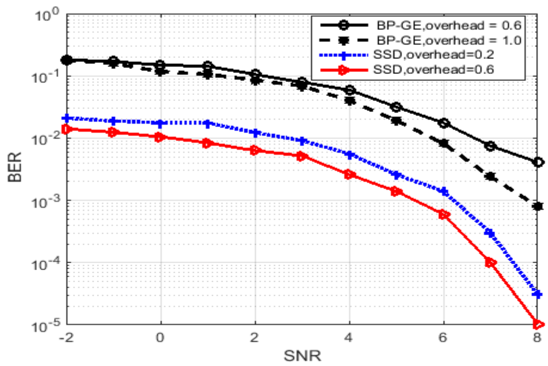

Fig. 2. BER versus SNR results for SSD approach and RSD using BP-GE method, for k=100

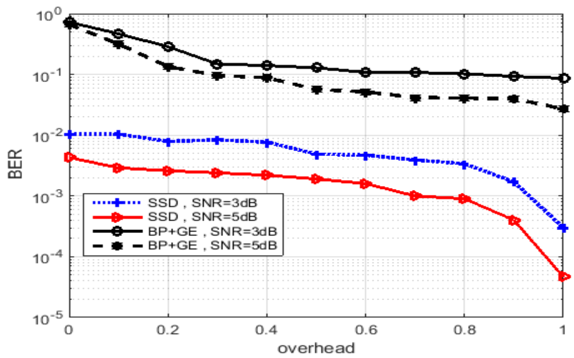

Fig. 3. BER versus overhead results for SSD approach and RSD using BP-GE method, for k=100 for fixed SNR

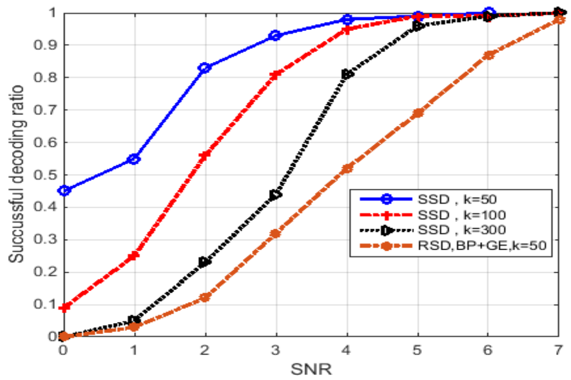

Fig. 4. Comparison of successful decoding ratio, with $k = 50{,}100$ and 300 both approaches using $N = 2k$ (records are the average of 500 runs)

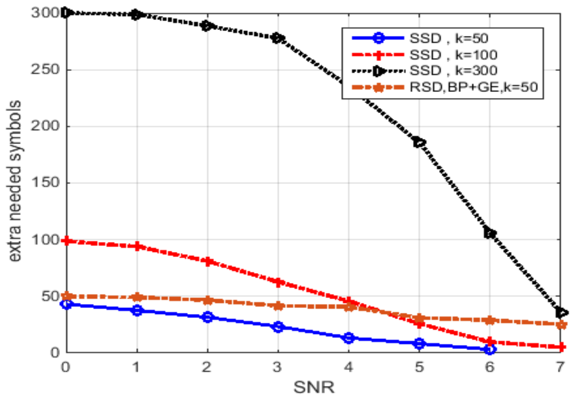

Fig. 5. Extra needed symbols for different SNRs, with $k = 50{,}100$ and 300 both approaches using $N = 2k$ (records are the average of 500 runs)

REFERENCES

[1] Luby, M. (2002). LT Codes. In: IEEE Symposium on Foundations of Computer Science, pp 271–280.

[2] Shokrollahi, A. (2006). Raptor codes. IEEE transactions on information theory, 52(6), 2551-2567.

[3] Gallager, R. (1962). Low-density parity-check codes. IRE Transactions on information theory, 8(1), 21-28.

[4] Palanki, R., & Yedidia, J. S. (2004, June). Rateless codes on noisy channels. In Information theory, 2004. ISIT 2004. Proceedings. International symposium on (p. 37). IEEE.

[5] Richardson, T. J., & Urbanke, R. L. (2001). The capacity of low-density parity-check codes under message-passing decoding. IEEE Transactions on information theory, 47(2), 599-618.

[6] Chang, L. J., Wang, C. H., & Zao, J. K. (2014, October). An error-floor reduction technique for short-length LT codes. In Information Theory and its Applications (ISITA), 2014 International Symposium on (pp. 279-283). IEEE.

[7] Zao, J. K., Hornansky, M., & Diao, P. L. (2012, June). Design of optimal short-length LT codes using evolution strategies. In Evolutionary Computation (CEC), 2012 IEEE Congress on (pp. 1-9). IEEE.

[8] Wang, X., Willig, A., & Woodward, G. (2013, April). Improving fountain codes for short message lengths by adding memory. In Intelligent Sensors, Sensor Networks and Information Processing, 2013 IEEE Eighth International Conference on (pp. 189-194). IEEE.

[9] Nguyen, T. D., Yang, L. L., & Hanzo, L. (2007, October). Systematic Luby transform codes and their soft decoding. In Signal Processing Systems, 2007 IEEE Workshop on (pp. 67-72). IEEE.

[10] Tran, V. H., Lay, K. T., & Peng, L. C. (2012, August). Modified LT coding with systematic connections. In Anti-Counterfeiting, Security and Identification (ASID), 2012 International Conference on (pp. 1-4). IEEE.

[11] Zhang, M., & Kim, S. (2014, May). A new soft decoding method for systematic LT codes. In European Wireless 2014; 20th European Wireless Conference; Proceedings of (pp. 1-6). VDE.

[12] Cheong, H., Eun, J., Kim, H., & Kim, K. J. (2016). Belief propagation decoding assisted on-the-fly Gaussian elimination for short LT codes. Cluster Computing, 19(1), 309-314.

[13] Kharel, A., & Cao, L. (2015, April). Decoding of short LT codes over BIAWGN channels with Gauss-Jordan elimination-assisted belief propagation method. In Wireless Telecommunications Symposium (WTS), 2015 (pp. 1-6). IEEE.

[14] Kim, S., Ko, K., & Chung, S. Y. (2008). Incremental Gaussian elimination decoding of raptor codes over BEC. IEEE communications letters, 12(4).

[15] Proakis, J. G. Digital communications. 1995. McGraw-Hill, New York.