Modified Robust Soliton Distribution (MRSD) with Improved Ripple Size for LT Codes

Kuo-Kuang Yen, Yen-Chin Liao, Chih-Lung Chen, and Hsie-Chia Chang

***Abstract*—In this letter, we propose a scheme to modify robust Soliton distribution (RSD) with respect to the expected *ripple size*. We only adjust the proportion of degree 1, degree 2 and the maximum degree to derive the modified RSD (MRSD). Thus the proposed scheme contains only two variables and its complexity does not increase with the code length. Our objective is to increase the mean of the expected ripple size while decreasing its variance at the same time. Furthermore, *sequential quadratic programming* is introduced to maximize the objective function under certain constraints. Simulation results show that, with different code lengths, MRSD saves 2% to 5.8% overhead to decode entire input symbols, compared to RSD.**

***Index Terms*—LT code, degree, ripple.**

I. Introduction

The performance of LT codes [1], [2] mainly depends on the degree distribution. The state of the art robust Soliton distribution (RSD) was designed for asymptotic optimality and used in various applications. However, it has been discovered that the performance of RSD is not necessarily optimal for finite code length, in particular, when the code length is small [3], [4]. Many approaches have been introduced to construct degree distributions [5]–[9], which is a high-dimensional problem. The complexity of these approaches increases with the number of degrees to be handled.

Typically, good degree distributions share certain characteristics [1], [10], [11]. Superior performance can be achieved by preserving these characteristics [12]. In this letter, we propose a scheme to modify RSD with respect to the expected *ripple size*. We only adjust the proportion of degree 1, degree 2, and the maximum degree that dominate the performance of LT codes. The term *ripple* in this letter is defined as the set of degree-1 output symbols [13]. Although the ripple has different definitions in [1] and [13], the basic properties of a good distribution mentioned in [1] apply to both definitions. That is, the ripple size must be sufficiently large and stable throughout the decoding process. We transfer these properties into a maximization problem and introduce *sequential quadratic programming* to search for solutions. Despite different code lengths, our objective function contains only two variables τ_1 and τ_2. Thus the complexity of our scheme does not increase with the code length. Compared to RSD, the modified RSD (MRSD) increases the mean of the expected ripple size [13], [14] while decreasing its variance. Then the decoding process becomes less likely to terminate due to the disappearance of the ripple. Moreover, after replacing RSD with MRSD, output symbols are still generated in the same way as LT encoding process. Thus the performance of LT codes remains independent of the channel erasure rate.

Manuscript received January 17, 2013. The associate editor coordinating the review of this letter and approving it for publication was A. Burr.

The authors are with the Department of Electronics Engineering & Institute of Electronics, National Chiao Tung University, Hsinchu, Taiwan, R.O.C. (e-mail: dannies@oasis.ee.nctu.edu.tw, ycliaoee92g@gmail.com, lung@si2lab.org, hcchang@mail.nctu.edu.tw).

This work was supported by the National Science Council of Taiwan, R.O.C., under grant NSC 101-2220-E-009 -060 - and NSC 101-2628-E-009 -013 -MY3.

Digital Object Identifier 10.1109/LCOMM.2013.040913.130136

II. LT Codes

During LT encoding, a set of output symbols $C = \{c_l | l = 1, 2, \cdots\}$ is generated from a set of k input symbols $I = \{i_j | j = 1, 2, \cdots, k\}$ via exclusive-or operations [1]. The degree distribution is denoted by $\Omega(x) = \sum_{d=1}^{d_{max}} \Omega_d x^d$, where Ω_d is the proportion of degree-d output symbols, d_{max} is the maximum degree and $\sum_{d=1}^{d_{max}} \Omega_d = 1$. If two received degree-1 output symbols connect to the same input symbol, the one with larger index is referred to as a repeated degree-1 output symbol. A successful decoding means that all k input symbols are decoded. The run time of the encoder and decoder is measured in terms of symbol operations [1]. A symbol operation is either an exclusive-or of one symbol into another or a copy of one symbol to another. The state of the art RSD is as follows:

Definition 1: RSD can be represented by $\Omega^r(x) = \sum_{d=1}^{d_{max}} \Omega^r_d x^d$, where

$$\Omega^r_d = \frac{\zeta(d) + \gamma(d)}{\sum_{d=1}^{k} \zeta(d) + \gamma(d)},$$

$$\zeta(d) = \begin{cases} \dfrac{\mu}{kd} & for\ d = 1, 2, \cdots, \lfloor \dfrac{k}{\mu} \rfloor - 1 \\ \dfrac{\mu}{k} ln(\dfrac{\mu}{\delta}) & for\ d = \lfloor \dfrac{k}{\mu} \rfloor \\ 0 & else \end{cases},$$

$$\gamma(d) = \begin{cases} \dfrac{1}{k} & for\ d = 1 \\ \dfrac{1}{d(d-1)} & for\ d = 2, 3, \cdots, k \end{cases}$$

and $\mu \equiv c \cdot ln(\frac{k}{\delta})\sqrt{k}$ is the average number of degree-1 output symbols with parameters c and δ for adjustment.

III. Proposed Scheme

A. Important characteristics of degree distribution

It has been found that the proportion of certain degrees in a degree distribution dominates the decodability of LT codes. By preserving the characteristics of these degrees, LT codes are able to achieve good performance. First, the degree-2 proportion is the maximum in a degree distribution such that $\forall_{d \neq 2} \Omega_2 \geq \Omega_d$ [12]. Especially when k approaches to infinity, the proportion of degree 2 is close to 0.5 [11]. Second, the degree-1 proportion must be small. Through the analysis of the classical balls and bins process, the degrees of the received output symbols must be at least $k \cdot ln(k/\delta)$ so that each input symbol can be connected at least once with probability $1-\delta$. As the degree-1 proportion increases, the average degree per output symbol becomes smaller. Then a receiver must collect more output symbols to ensure that the total degree is sufficiently large, resulting in lower transmission efficiency. The property $\Omega_1 \ll \Omega_2$ is common in good degree distributions, and it has been shown that the case $\Omega_1 > \Omega_2$ results in poor performance [8]. Third, a spike at the highest existing degree d_{max} is necessary, which ensures sufficient degrees of received output symbols to keep the belief-propagation (BP) decoding alive at the end of the process. In other words, the existence of output symbols of the maximum degree increases the opportunity that input symbols are fully connected.

B. Ripple size

The expected ripple size [13] is given by

$$R_\Omega(\rho) = (1+\epsilon)(k-\rho)\left(\Omega'(\frac{\rho}{k}) + \frac{1}{1+\epsilon} ln\frac{k-\rho}{k}\right) + O(1)$$
$$=(1+\epsilon)\left(\sum_{d=1}^{d_{max}} \Omega_d f(\rho,d) + \frac{k-\rho}{1+\epsilon} ln\frac{k-\rho}{k}\right) + O(1), \quad (1)$$

where$f(\rho,d) = (k-\rho)d(\frac{\rho}{k})^{d-1}$ and ρ is the number of decoded input symbols. There are two basic properties of a good distribution in terms of the expected ripple size [1]. First, output symbols enter the ripple at the same rate as they are removed from it. Large variation in the ripple size is not desired during the decoding process. This is because the decoding process is more likely to terminate when the expected ripple size is relatively small [13], while on average, a larger ripple contains more repeated degree-1 output symbols. Second, the ripple size should be kept large enough so that the decoding process does not terminate due to the disappearance of the ripple. As a counterexample, the expected ripple size of the ideal Soliton distribution is one. Thus the ripple disappears even with the smallest variation, leading to premature BP decoding termination.

C. Modification of RSD

RSD satisfies the characteristics of a good distribution mentioned in Section III-A. However, with short code lengths, the expected ripple size varies greatly during BP decoding process when RSD is employed. In particular, the ripple size is relatively small at early decoding stage where the decoding process terminates frequently.

We modify RSD to increase the mean of its expected ripple size over $\rho = 0$ to k, while decreasing its variance. The modification can be viewed as cutting the ripple in the areas where it is relatively large to compensate for those areas with smaller ripple size. We only adjust the proportion of degree 1, degree 2 and the maximum degree since output symbols of these degrees dominate the decodability of LT codes. The proportion of other degrees remain unchanged after the modification so that they are consistent with the result of the AND-OR tree analysis that the proportion of higher degree declines monotonically [3].

After the modification, the proportion of degree 1, degree 2 and the maximum degree d_{max} are increased by τ_1, τ_2 and $\tau_{d_{max}}$, respectively. We force

$$\tau_1 + \tau_2 + \tau_{d_{max}} = 0, \quad (2)$$

so that the adjustment is confined in these three degrees. During the modification, we only search for appropriate τ_1 and τ_2 since $\tau_{d_{max}}$ can be derived through (2). Specifically, MRSD can be represented by

$$\Omega^m(x) = \sum_{d=1}^{d_{max}} \Omega_d^m x^d$$
$$=\Omega^r(x) + \tau_1 x + \tau_2 x^2 - (\tau_1+\tau_2)x^{d_{max}},$$

where $\Omega_1^m = \Omega_1^r + \tau_1$, $\Omega_2^m = \Omega_2^r + \tau_2$, $\Omega_{d_{max}}^m = \Omega_{d_{max}}^r - \tau_1 - \tau_2$ and $\Omega_d^m = \Omega_d^r$ for $d = 3, 4, \cdots, d_{max}-1$. The corresponding expected ripple size is

$$R_{\Omega^m}(\rho) = (1+\epsilon)\Bigg(\tau_1 f(\rho,1) + \tau_2 f(\rho,2)$$
$$-(\tau_1+\tau_2)f(\rho,d_{max}) + \sum_{d=1}^{d_{max}} \Omega_d^r f(\rho,d)$$
$$+\frac{k-\rho}{1+\epsilon} ln\frac{k-\rho}{k}\Bigg) + O(1).$$

with mean

$$M = \frac{1}{k}\sum_{\rho=0}^{k-1} R_{\Omega^m}(\rho)$$

and variance

$$V = \frac{1}{k}\sum_{\rho=0}^{k-1} (R_{\Omega^m}(\rho) - M)^2.$$

Our objective is to increase the mean M and reduce the variance V. At the same time, the resulting distribution must comply with the aforementioned characteristics of a good distribution. Then our objective is to maximize

$$\xi = M - \alpha \cdot V \quad (3)$$

subject to

$$-\Omega_1^r \le \tau_1 \le \Omega_2^r + \Omega_{d_{max}}^r$$
$$-\Omega_2^r \le \tau_2 \le \Omega_1^r + \Omega_{d_{max}}^r$$
$$\tau_1 + \tau_2 \le \Omega_{d_{max}}^r$$
$$\forall_{d \ne 2} \Omega_2^m \ge \Omega_d^m$$
$$W_{\Omega^m} \le \beta,$$

where parameters α and β are non-negative and W_{Ω^m} is the average number of repeated degree-1 output symbols when distribution Ω^m is used. In fact, W_{Ω^m} is proportional to Ω_1^m so that a receiver is more likely to collect repeated degree-1 output symbols as the degree-1 proportion grows. Thus Ω_1^m is upper bounded by the constraint $W_{\Omega^m} \le \beta$ and satisfies $\Omega_1^m \ll \Omega_2^m$ when β is small [12]. Besides, the proportion of each degree must not be negative. Therefore, $\tau_1 \ge -\Omega_1^r$, $\tau_2 \ge -\Omega_2^r$ and $\tau_{d_{max}} \ge -\Omega_{d_{max}}^r$. According to (2), we have $\tau_1 +$

$\tau_2 \leq \Omega^r_{d_{max}}$. Moreover, we can add an additional constraint $\sum_{d=1}^{d_{max}} d\Omega_d^m \leq \phi$ to the objective function, where $\phi \geq 1$ is a design parameter. Then the average degree per output symbol can be limited, which is linearly proportional to the number of symbol operations during the encoding and decoding process.

Different approaches can be applied to maximize ξ. In this letter, we use the technique of sequential quadratic programming [15] to iteratively solve the problem. First, a two-dimensional vector is defined as $\mathbf{t} = [\tau_1, \tau_2]$. Then, the maximization problem with the current vector $\mathbf{t}^{(j)}$ is modelled by a quadratic subproblem and the solution of this subproblem is used to find a new vector $\mathbf{t}^{(j+1)}$, where j denotes the iteration number. The entire process ends when the current vector $\mathbf{t}^{(j)}$ maximizes ξ. According to (3), large α means small tolerance of the variance V. If $\alpha = 0$, the objective is reduced to maximizing the mean of the expected ripple size. In general, appropriate values of α are located between zero and one.

D. Average number of repeated degree-1 output symbols

When g received degree-1 output symbols connect to h input symbols, $g > h$, there exist $g - h$ repeated degree-1 output symbols. Therefore, using degree distribution Ω, the average number of repeated degree-1 output symbols is

$$W_\Omega = \sum_{h=1}^{min(g-1,k)} \sum_{g=2}^{(1+\epsilon)k} (g-h) A_\Omega(g) T(g,h),$$

where

$$A_\Omega(g) = \binom{(1+\epsilon)k}{g} (\Omega_1)^g (1-\Omega_1)^{(1+\epsilon)k-g}$$

is the probability of receiving g degree-1 output symbols and $T(g,h)$ is the probability that the g received degree-1 output symbols connect to h input symbols. Let $S(g,h)$ denote the number of combinations that g degree-1 output symbols connect to h input symbols and each input symbol is connected at least once. Then we have $T(g,h) = \frac{\binom{k}{h} S(g,h)}{k^g}$, where k^g is the total number of combinations that the g degree-1 output symbols connect to k input symbols. For $h = 1$, we have $S(g,1) = 1^g = 1$ since all g degree-1 output symbols connect to one input symbol. For $h = 2$, $S(g,2) = 2^g - \binom{2}{1} S(g,1) = 2^g - \binom{2}{1} 1^g$. Then for $h = 3$, $S(g,3) = 3^g - \binom{3}{2} S(g,2) - \binom{3}{1} S(g,1) = 3^g - \binom{3}{2} 2^g + \binom{3}{1} 1^g$, where 3^g is the number of all possible combinations and there are $\binom{3}{2} S(g,2)$ combinations with one input symbol unconnected and $\binom{3}{1} S(g,1)$ combinations with two input symbols unconnected. Following this regularity, we have $S(g,h) = h^g - \sum_{i=1}^{h-1} \binom{h}{i} S(g,i) = h^g - \sum_{x=1}^{h-1} (-1)^{x-h+1} \binom{h}{x} x^g$ for general g and h.

IV. Simulation Results

In this section, simulation results are based on 100000 encoding/decodings. Fig. 1 shows successful decoding rate versus the overhead for RSD and MRSD with different k. The overhead ϵ is defined as $\epsilon = \frac{n-k}{k}$, where n is the number of received output symbols. The corresponding expected ripple size is shown in Fig. 2 as a function of the ratio ρ/k of decoded input symbols. RSDs have parameters $c = 0.02$ and $\delta = 0.01$ and they are further modified into MRSDs using parameters $\epsilon = 0.1$, $\alpha = 0.25$ and $\beta = 0.001$. Parameter α is decided by searching the range between zero and one with a resolution of 0.05. Parameter $\beta = 0.001$ means that one repeated degree-1 output symbol is allowed per 1000 encoding/decodings. The resulting vectors $\mathbf{t} = [\tau_1, \tau_2]$ are listed in Table I. It can be seen that MRSD always outperforms RSD in terms of successful decoding rate. This is because compared to RSD, the expected ripple size of MRSD is increased in most regions, in particular, where the ratio ρ/k is small. Therefore, BP decoding process is less likely to terminate prematurely due to the disappearance of the ripple.

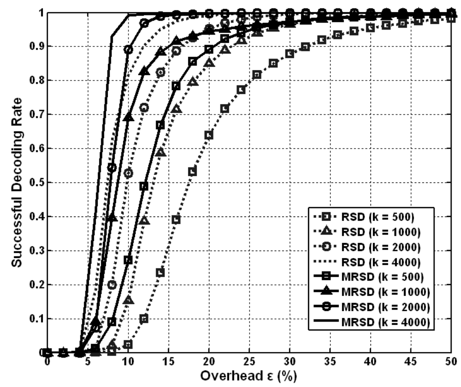

Fig. 1. Successful decoding rate versus overhead with different k. RSD of $c = 0.02$ and $\delta = 0.01$ is used and further modified into MRSD.

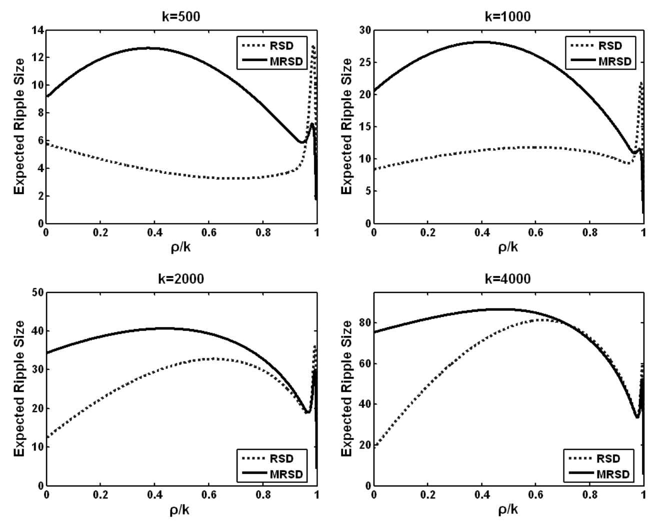

Fig. 2. Expected ripple size versus ratio ρ/k of decoded input symbols with different code length k and $\epsilon = 0.1$. RSD of $c = 0.02$ and $\delta = 0.01$ is used and further modified into MRSD.

Using the above settings, Table II shows the average overhead needed to decode all k input symbols and the average degree per output symbol λ. With different k, MRSD achieves 2% to 5.8% reduction in the average overhead, compared to RSD. In addition, MRSD has smaller λ than that of RSD for certain k. Thus the former requires smaller number of symbol operations than the latter during the encoding and decoding process because this number is linearly proportional to the average degree per output symbol [1].

TABLE I
VECTOR $\mathbf{t} = [\tau_1, \tau_2]$ OF THE MAXIMIZATION RESULT FOR DIFFERENT k

k	500	1000	2000	4000
τ_1	0.006	0.011	0.01	0.013
τ_2	0.026	0.018	-0.002	-0.009

TABLE II
AVERAGE OVERHEAD NEEDED TO DECODE ALL k INPUT SYMBOLS AND AVERAGE DEGREE PER OUTPUT SYMBOL λ FOR RSD AND MRSD

	Avg. Overhead (%)		λ	
k	RSD	MRSD	RSD	MRSD
500	20.1	14.3	11.2	7.9
1000	15.8	10.7	12.1	8.1
2000	12.0	8.9	12.9	11.5
4000	9.2	7.2	13.8	12.8

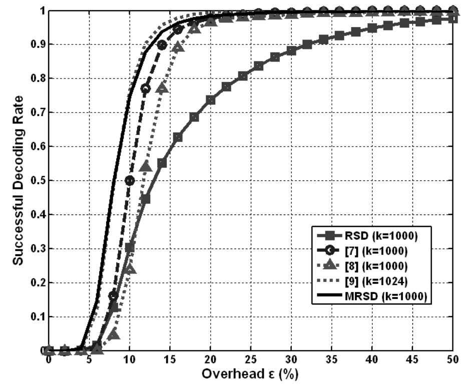

Fig. 3. Successful decoding rate versus overhead for RSD, MRSD and optimized distributions in [7], [8] and [9] with k around 1000. RSD of $c = 0.011$ and $\delta = 0.5$ is used and further modified into MRSD.

Fig. 3 shows successful decoding rate versus overhead with k around 1000. RSD has parameters $c = 0.011$ and $\delta = 0.5$, and it is further modified into MRSD using $\alpha = 0.27$, resulting in $\mathbf{t} = [0.02, -0.021]$. The other optimized distributions are from Table I in [7], equation (24) with parameters $\eta_{opt} = (0.083, 0.487, 0.032)$ in [8] and Table II in [9]. The results show that the performance of MRSD and the distribution in [9] are very close, and they both outperform RSD and the optimized distributions in [7] and [8]. Table III shows the corresponding average overhead needed to decode all k input symbols and the average degree per output symbol λ. Compared to the distributions in [7] and [8], MRSD saves an overhead of at least 1.8% $(10.8\% - 9.0\%)$, while its average degree per output symbol is close to that of the distribution in [7]. Moreover, the distribution in [9] requires an average overhead of 8.7% to decode all k input symbols, which is only 0.3% $(9.0\% - 8.7\%)$ smaller than that required by MRSD. Nevertheless, the average degree per output symbol λ of the distribution in [9] is 21.1% $(\frac{11.5-9.5}{9.5})$ larger than that of MRSD.

TABLE III
AVERAGE OVERHEAD NEEDED TO DECODE ALL k INPUT SYMBOLS AND AVERAGE DEGREE PER OUTPUT SYMBOL λ

	[7]	[8]	[9]	RSD	MRSD
Avg. Overhead (%)	10.8	12.1	8.7	19.7	9.0
λ	9.6	7.7	11.5	9.0	9.5

V. CONCLUSION

In this letter, we propose a scheme to modify RSD with respect to the expected ripple size. We only adjust the most important proportion of degree 1, degree 2 and the maximum degree. Thus the complexity of our method stays unchanged as the code length grows. The objective of increasing the mean of the expected ripple size while decreasing its variance is transferred into a maximization problem. Sequential quadratic programming algorithm is introduced to solve the objective function. Simulation results show that by carefully choosing parameter α, MRSD is able to outperform RSD in terms of successful decoding rate. Moreover, compared to other optimized distributions, MRSD either has lower average degree per output symbol or requires smaller overhead to decode all k input symbols.

REFERENCES

[1] M. Luby, "LT codes," in *Proc. 2002 IEEE Symposium on Foundations of Computer Science*, pp. 271–280.
[2] A. Shokrollahi, "Raptor codes," *IEEE Trans. Inf. Theory*, vol. 52, no. 6, pp. 2551–2567, 2006.
[3] D. Sejdinovic, R. Piechocki, and A. Doufexi, "AND-OR tree analysis of distributed LT codes," in *Proc. 2009 IEEE Information Theory Workshop on Networking and Information Theory*, pp. 261–265.
[4] E. A. Bodine and M. K. Cheng, "Characterization of Luby Transform codes with small message size for low-latency decoding," in *Proc. 2008 IEEE International Conference on Communications*, pp. 1195–1199.
[5] E. Hyytia, T. Tirronen, and J. Virtamo, "Optimal degree distribution for LT codes with small message length," in *Proc. 2007 IEEE International Conference on Computer Communications*, pp. 2576–2580.
[6] C.-M. Chen, Y.-P. Chen, T.-C. Shen, and J. K. Zao, "Optimizing degree distributions in LT codes by using the multiobjective evolutionary algorithm based on decomposition," in *Proc. 2010 IEEE Congress on Evolutionary Computation*, pp. 1–8.
[7] ——, "On the optimization of degree distributions in LT code with covariance matrix adaptation evolution strategy," in *Proc. 2010 IEEE Congress on Evolutionary Computation*, pp. 1–8.
[8] E. Hyytia, T. Tirronen, and J. Virtamo, "Optimizing the degree distribution of LT codes with an importance sampling approach," in *Proc. 2006 International Workshop on Rare Event Simulation*, pp. 2576–2580.
[9] P. P. Jesper, H. Sorensen, and J. Ostergaard, "Design and analysis of LT codes with decreasing ripple size," *IEEE Trans. Commun.*, vol. 60, no. 11, pp. 3191–3197, 2012.
[10] D. J. C. MacKay, "Fountain codes," in *IEE Proc. Commun.*, vol. 152, pp. 1062–1068, Dec. 2005.
[11] O. Etesami and A. Shokrollahi, "Raptor codes on binary memoryless symmetric channels," *IEEE Trans. Inf. Theory*, vol. 52, no. 5, pp. 2033–2051, 2006.
[12] A. Liau, S. Yousefi, and I.-M. Kim, "Binary soliton-like rateless coding for the Y-network," *IEEE Trans. Commun.*, vol. 59, no. 12, pp. 3217–3222, 2011.
[13] G. Maatouk and A. Shokrollahi, "Analysis of the second moment of the LT decoder," in *Proc. 2009 International Symposium on Information Theory*, pp. 2326–2330.
[14] R. Karp, M. Luby, and A. Shokrollahi, "Finite length analysis of LT codes," in *Proc. 2004 International Symposium on Information Theory*, p. 39.
[15] J. Nocedal and S. Wright, *Numerical Optimization*, ser. Springer Series in Operations Research. Springer, 1999.