PAPER *Special Section on Signal Design and Its Applications in Communications*

# Rate Allocation for Component Codes of Plotkin-Type UEP Codes*

**Jinsoo PARK**[†a)], ***Nonmember and*** **Hong-Yeop SONG**[†b)], ***Member***

**SUMMARY** In this paper, we propose a framework to allocate code rates of component codes in a Plotkin-type unequal error protection (UEP) code. We derive an equivalent noise variance for each component code using structure of the Plotkin construction and Gaussian assumption. Comparing the equivalent noise variance and Shannon limit, we can find a combination of the code rates for the component codes. We investigate three types of code rate combinations and analyse their UEP performance. We also estimate a *performance crossing signal to noise ratio (SNR)* of the Plotkin-type UEP code. It indicates that which code has better performance for a given SNR. We confirm that the proposed framework is appropriate to obtain a desired UEP capability.

***key words:*** *unequal error protection, Plotkin construction*

## 1. Introduction

Multimedia streaming services on the wireless devices are becoming more and more popular recently, and many multimedia packets are distinguished into some different levels according to their roles with different importance. This motivates allocation of different protection level for each part of codeword according to their importance. To provide the different protection level, the unequal error protection (UEP) channel coding schemes have been studied.

UEP-turbo codes with non-uniform puncturing were proposed in [1]. Irregular UEP low-density parity-check (LDPC) codes using faster convergence property of the higher degree nodes have been discussed in [2], and a further optimized result was presented in [3].

Another interesting approach is the Plotkin-type UEP codes which are suggested in [4]–[6]. They are constructed from component codes $C_1$ and $C_2$. The Plotkin construction is a serial concatenation of $C_1$ and $C_1 + C_2$, and it is represented as $|C_1|C_1 + C_2|$. It gives different channel quality for each component codes, and then makes different performance of each code. In [4], the Plotkin construction $|C_1|C_1 + C_2|$ is exploited to obtain an UEP property. In [5], the authors proposed the Plotkin-type construction based UEP-LDPC codes. They analysed some characteristics of the Plotkin-type construction using the concept of equivalent channel model. In addition, a modified construction of Plotkin-type UEP codes using interleavers was proposed in [6].

In this paper, we will give a framework on how to allocate the rate of each component code of the Plotkin-type construction. For simplicity, we will consider only two protection levels. Also, we will define three cases of their rate combinations. For each case, we analyse the UEP performance, and compare with an equal error protection (EEP) code. Our contribution can be applied to larger number of protection levels. In addition, we will propose an estimation of *performance crossing signal to noise ratio (SNR)*. This indicates which code has better performance at a given SNR.

This paper is organized as follows. In Sect. 2, we review the Plotkin-type UEP codes with encoding scheme, multistage decoding algorithm, and some properties with the notion of equivalent channel noise. In Sect. 3, we give an analysis for the rate allocation for component codes of the Plotkin-type UEP codes using equivalent channel models based on Gaussian assumption. We also estimate the performance crossing SNR. In Sect. 4, we show some simulation results of the Plotkin-type codes which are designed using the proposed rate allocation. Conclusion is given in Sect. 5.

## 2. Plotkin-Type UEP Codes

In this section, we briefly review the Plotkin-type UEP codes described in [5].

### 2.1 Construction

Plotkin-type code $C_p$ is defined as follows:

$$C_p = \{|\mathbf{u}|\mathbf{u} + \mathbf{v}|\,|\mathbf{u} \in C_1, \mathbf{v} \in C_2\}, \tag{1}$$

where $C_1$ and $C_2$ are $[n, k_1, d_1]$ and $[n, k_2, d_2]$ binary linear codes, respectively, and $\mathbf{u} + \mathbf{v}$ over GF(2) is performed component-wise. The codes $C_1$ and $C_2$ are named *component codes*. Letting $\mathbf{w} = \mathbf{u} + \mathbf{v}$ over GF(2), one transmits $\mathbf{u}$ in the first time slot and $\mathbf{w}$ in the second. In [5], they used the LDPC codes as the component codes, and called the codes as Plotkin-type UEP-LDPC codes.

Manuscript received May 30, 2016.
Manuscript revised October 15, 2016.
†The authors are with Yonsei University, Korea.
*This work was supported by Institute for Information & communications Technology Promotion (IITP) grant funded by the Korea government (MSIP) (No. B0717-16-0024, Development on the core technologies of transmission, modulation and coding with low-power and low-complexity for massive connectivity in the IoT environment). This paper is a full version of 2012 ISIT proceeding paper “Rate Allocation for Component Codes of Plotkin-Type UEP Codes”.
a) E-mail: js.park09@yonsei.ac.kr
b) E-mail: hysong@yonsei.ac.kr (Corresponding author)
DOI: 10.1587/transfun.E100.A.930

### 2.2 Decoding

The basic decoding process of the Plotkin-type UEP codes is multistage (MS) decoding. Let $\mathbf{y}' = \{y_i'\}_{i=1}^n$ and $\mathbf{y}'' = \{y_i''\}_{i=1}^n$ be the first and second received vectors for $\mathbf{u}$ and $\mathbf{w}$, respectively, and $L^{y'} = \{L_i^{y'}\}_{i=1}^n$ and $L^{y''} = \{L_i^{y''}\}_{i=1}^n$ be their *log-likelihood ratios* (LLRs), respectively. The MS decoding procedure is given as follows:

1. **Calculation of LLR** $L^v = \{L_i^v\}_{i=1}^n$ **for** $\mathbf{v}$:

$$L_i^v = 2 \cdot \tanh^{-1}(\tanh(L_i^{y'}/2)\tanh(L_i^{y''}/2)). \quad (2)$$

2. **Decoding of** $\mathbf{v}$: Using $L_i^v$, decode $\mathbf{v}$ for $C_2$. Denote the result by $\hat{\mathbf{v}} = \{\hat{v}_i\}_{i=1}^n \in \{0,1\}^n$.
3. **Calculation of LLR** $L^u = \{L_i^u\}_{i=1}^n$ **for** $\mathbf{u}$:

$$L_i^u = L_i^{y'} + (-1)^{\hat{v}_i} L_i^{y''}.$$

4. **Decoding of** $\mathbf{u}$ : Using $L_i^u$, decode $\mathbf{u}$ for $C_1$.

The advantage of this algorithm is the LLR gain of the $\mathbf{u}$ in the 3rd step of the above as mentioned in [5]. This algorithm can be applied repeatedly, and it is called as the multiround-multistage (MR-MS) decoding.

### 2.3 Properties

In [5], authors used the concept of the *equivalent channel* under the Gaussian assumption [8]. If two additive white Gaussian noise (AWGN) channels are modelled as two Gaussian random variables $Z_1 \sim N(m_{z1}, \sigma_{z1}^2)$ and $Z_2 \sim N(m_{z2}, \sigma_{z2}^2)$ with $m_{z1}^2/\sigma_{z1}^2 = m_{z2}^2/\sigma_{z2}^2$, then they can be referred as *equivalent* channel, in the sense of the equivalent SNR. For an AWGN channel of Gaussian random variable $X \sim N(m_x, \sigma_x^2)$, we can calculate the equivalent channel noise variance $\sigma_{x,eq}^2 = \sigma_x^2/m_x^2$, where the equivalent channel is modelled as $N(1, \sigma_{x,eq}^2)$. Let $\sigma_1^2$ and $\sigma_2^2$ are the equivalent channel noises that $C_1$ and $C_2$ in $C_p$ experience, respectively. The brief summaries of two lemmas in [5] are listed as follows. Here, the perfect decoding of $C_2$ (i.e. $\mathbf{v}$) is assumed.

**Lemma 1:** *[5]* If the equivalent channel noise $\sigma_{ch}^2 \to \infty$, then $\sigma_1^2 \to \sigma_{ch}^2/2$ and $\sigma_2^2 \to \sigma_{ch}^4$.

**Lemma 2:** *[5]* If the equivalent channel noise $\sigma_{ch}^2 \to 0$, then $\sigma_1^2 \to \sigma_{ch}^2/2$ and $\sigma_2^2 \to \sigma_{ch}^2$.

These lemmas indicate that the UEP property of MS decoding is based on the unequal channel SNR of the component codes.

## 3. Rate Allocation of the Component Codes

In this section, we discuss the rates of the component codes and their equivalent channel. In this paper, we just consider the AWGN channels only. The following notations are fixed throughout the paper:

- $C_p$ and $R_p$: Plotkin-type UEP code and its rate.
- $C_1$ and $C_2$: two component codes of $C_p$.
- $R_1$ and $R_2$: code rate of $C_1$ and $C_2$, respectively.
- $C_a$ and $R_a$: an average EEP code with the same code rate as $C_p$.
- $C_i$ in $C_p$: usage of $C_i$ as a component code in the Plotkin-type UEP code $C_p$.
- $C_i$ only as EEP: usage of $C_i$ as an EEP code by itself.

### 3.1 Average Code

The code rate of $C_p$ is the average of $R_1$ and $R_2$:

$$R_p = \frac{R_1 + R_2}{2} = \frac{k_1 + k_2}{2n}. \quad (3)$$

Now we consider the average EEP code $C_a$ which achieves the channel capacity and has rate $R_a$. We further assume that its rate and length are the same as those of $C_p$, i.e. $R_a = R_p$. Throughout this paper, *threshold* refers to the equivalent noise variance corresponding to the Shannon limit of a given code rate. Let $\sigma_{a,th}^2$ be the threshold of $C_a$ in BPSK modulation system under AWGN channel [9]. Then,

$$R_a = f(\sigma_{a,th}^2), \quad (4)$$

where $f(\sigma^2)$ is a capacity function of binary phase shift keying (BPSK) given as

$$f(\sigma^2) = \int \frac{\exp\left(-\frac{(x-1)^2}{2\sigma^2}\right)}{\sqrt{2\pi\sigma^2}} \log_2\left(\frac{2}{1+\exp\left(-\frac{2x}{\sigma^2}\right)}\right) \mathrm{d}x.$$

Let $\sigma_{1,th}^2$ and $\sigma_{2,th}^2$ be the threshold of $C_1$ and $C_2$ in the equivalent noise variance form, respectively. Then we can rewrite the code rate relation as follows,

$$f(\sigma_{a,th}^2) = \frac{f(\sigma_{1,th}^2) + f(\sigma_{2,th}^2)}{2}, \quad (5)$$

$$\sigma_{2,th}^2 = f^{-1}(2f(\sigma_{a,th}^2) - f(\sigma_{1,th}^2)). \quad (6)$$

### 3.2 Equivalent Channel Models and Thresholds of Component Codes

By the relation (3), if we select $R_a$ and $R_1$, then $R_2$ will be fixed. The Gaussian assumption enables the approximation of equivalent noise variance $\sigma_2^2$ that is experienced by the decoder of $C_2$ in $C_p$. We assume $C_p$ and $C_a$ are transmitted over the equivalent channel with $\sigma_{ch}^2$. For a fair comparison and design purpose, we may put $\sigma_{ch}^2 = \sigma_{a,th}^2$. Then, the LLRs of received vectors $\{L_i^{y'}\}_{i=1}^n$ for $C_1$ in $C_p$ and $\{L_i^{y''}\}_{i=1}^n$ for $C_1 + C_2$ in $C_p$ both follow independent $N(2/\sigma_{a,th}^2, 4/\sigma_{a,th}^2)$.

The first step of the MS decoding calculates $L_i^v$ using the relation in (2). By using Monte Carlo simulation and Gaussian assumption, we can obtain $E\left[L_i^v\right]$ and $Var\left[L_i^v\right]$.

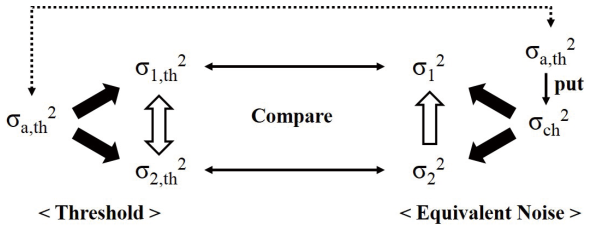



**Fig. 1** Diagram of the analysis process. Left: threshold of each codes. Right: equivalent channel noise of each codes.

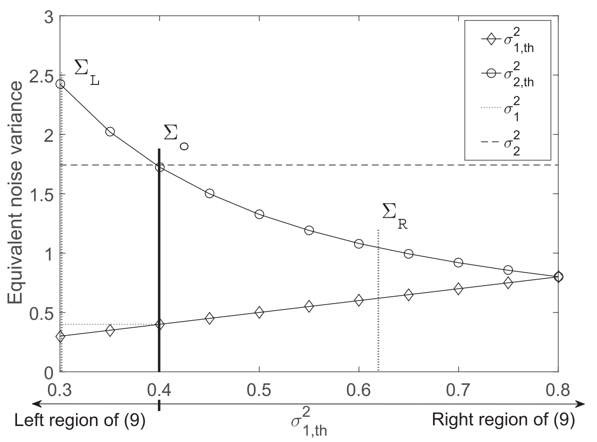



**Fig. 2** Case $\sigma^2_{a,th} = 0.8$ and hence $R_a = 0.5604$.

And hence, the equivalent noise $\sigma_2^2$ of the channel for $C_2$ in $C_p$ is determined as

$$\sigma_2^2 = \frac{Var\left[L_i^v\right]}{E^2\left[L_i^v\right]}. \tag{7}$$

The two component codes $C_1$ and $C_2$ in $C_p$ should be error free to decode $C_p$ perfectly. If the equivalent channel $\sigma^2_{a,th}$ satisfies the error free condition of $C_2$ in $C_p$, then we can easily claim that the equivalent channel noise of $C_1$ in $C_p$ is

$$\sigma_1^2 = \frac{\sigma^2_{a,th}}{2}. \tag{8}$$

It is clear that $C_p$ will be error free if the following inequalities (9) hold:

$$\sigma_1^2 \le \sigma^2_{1,th} \quad \text{and} \quad \sigma_2^2 \le \sigma^2_{2,th}. \tag{9}$$

Now, we need to determine the values of $\sigma^2_{1,th}$ and $\sigma^2_{2,th}$ satisfying (9) for a given $R_a$ and hence $\sigma^2_{a,th}$. Then this will determine the rate $R_1$ and $R_2$ by the similar relations as (4) or (5). This process is shown in Fig. 1. Note that we set $\sigma^2_{ch} = \sigma^2_{a,th}$.

Figure 2 shows the result of all the computations above for the various values of $\sigma^2_{1,th}$ for a given value of $\sigma^2_{a,th} = 0.8$ and hence $R_a = 0.5604$. Here, the solid lines of $\sigma^2_{1,th}$ and $\sigma^2_{2,th}$ correspond to the various values of the threshold of the each component code where $\sigma^2_{1,th}$ is set to be the x-axis. Note that $\sigma^2_{2,th}$ is determined from $\sigma^2_{a,th}$ and $\sigma^2_{1,th}$. The dashed horizontal line is the equivalent channel noise $\sigma_2^2$ for the decoding of $C_2$ and the dotted horizontal line is $\sigma_1^2$ for the decoding of $C_1$, respectively, where the given channel condition is $\sigma^2_{a,th}$. Here, we obtained $\sigma_1^2$ and $\sigma_2^2$ using (8) and (7), respectively.

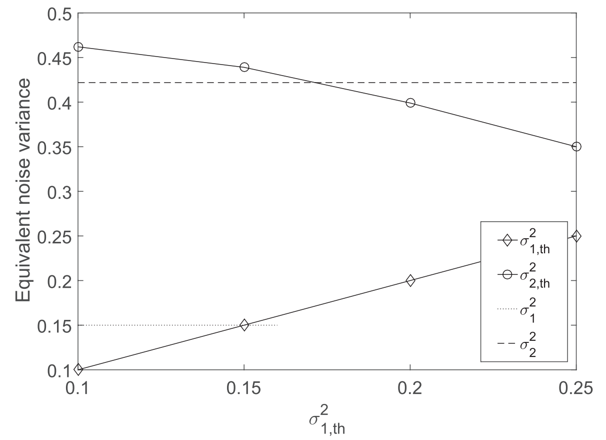



**Fig. 3** Case $\sigma^2_{a,th} = 0.3$ and hence $R_a = 0.8724$.

In the figure, $\sigma^2_{1,th} = 0.4$ and $\sigma^2_{2,th} = 1.724$ satisfy the inequalities (9), if we allow small errors in the numerical integration in (9) or Monte Carlo simulation of $L_i^v$. The thresholds $\sigma^2_{1,th}$ and $\sigma^2_{2,th}$ are designated by the solid vertical line with mark $\Sigma_o$. They determine the rates of $C_1$ and $C_2$ to be $R_1 = 0.79$ and $R_2 = 0.33$. From this, we may foresee that both of $C_1$ and $C_2$ with these rates will perform (ideally) error-free in $C_p$. In practical situation, $C_p$ will perform almost same as $C_a$. To the left of (9) lies the region in which the following holds:

$$\sigma_1^2 > \sigma^2_{1,th} \quad \text{and} \quad \sigma_2^2 \le \sigma^2_{2,th}.$$

This implies that $C_1$ works poorly even though $C_2$ might work with acceptable level of performance. In the right of (9), the dotted horizontal line for $\sigma_1^2$ will go up passing the vertical line $\Sigma_o$ due to the poor performance of $C_2$ which can obviously be seen by the fact that $\sigma_2^2 > \sigma^2_{2,th}$. The same as $\Sigma_o$, we can choose a pair of $\sigma^2_{1,th}$ and $\sigma_{2,th}$ in both (left and right of $\Sigma_o$) ranges, respectively. In the figure, $\Sigma_L$ and $\Sigma_R$ represent two examples of the pairs for the left and right regions, respectively. They are described with dotted vertical lines in Fig. 2.

The results of high and low $R_a$ are represented in Fig. 3 and Fig. 4, respectively. For each figure, we can find a region of (9). We think that the region exists over wide range of $R_a$.

For the performance simulation, we have selected three cases from Fig. 2, one from each region, and performed BER simulation of involved codes in order to confirm the analysis and predicted behavior of the component codes described so far. The parameters are listed in Table 1, and the simulation results will show that they work as expected in Sect. 4.

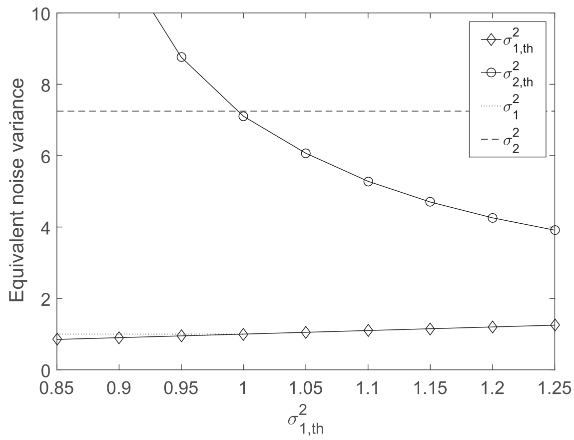



**Fig. 4** Case $\sigma^2_{a,th} = 2.0$ and hence $R_a = 0.2905$.

**Table 1** Code rates from three cases.

| | $R_a$ ($\sigma^2_{a,th}$) | $R_1$ ($\sigma^2_{1,th}$) | $R_2$ ($\sigma^2_{2,th}$) |
|---|---|---|---|
| $\Sigma_L$ | 0.56 (0.8) | 0.87 (0.3) | 0.25 (2.425) |
| $\Sigma_o$ | 0.56 (0.8) | 0.79 (0.4) | 0.33 (1.724) |
| $\Sigma_R$ | 0.56 (0.8) | 0.65 (0.619) | 0.47 (1.051) |

### 3.3 Performance Crossing SNR of $C_2$ in $C_p$

In this section, we analyse the *performance crossing SNR* which gives the same performance for $C_2$ in $C_p$ and $\bar{C}_2$, where $\bar{C}_2$ denotes $C_2$ as an EEP code. Because of the purpose of UEP, the performance crossing SNR becomes an important parameter. For a given SNR, if the performance crossing SNR is smaller than the given SNR, then $C_2$ in $C_p$ has better performance than $\bar{C}_2$. Otherwise, $\bar{C}_2$ has better performance than $C_2$ in $C_p$. Investigating the performance crossing SNR will be helpful to decide whether we will exploit the Plotkin-type UEP code or not.

The performance crossing SNR of $C_2$ in $C_p$ can be calculated for every Plotkin construction with the same length $C_1$ and $C_2$. For a given $E_b/N_0$, $\bar{C}_2$ and $C_p$ have $E_s/N_0$ given by

$$E_s/N_0^{\bar{C}_2} = R_2(E_b/N_0) \quad \text{and} \quad E_s/N_0^{C_p} = R_p(E_b/N_0),$$

respectively. If $E_s = 1$, then the relations can be stated as follows

$$\frac{1}{R_2(E_b/N_0)} = N_0^{\bar{C}_2} = 2\bar{\sigma}_2^2$$
$$\frac{1}{R_p(E_b/N_0)} = N_0^{C_p} = 2\tilde{\sigma}_a^2,$$

where $\bar{\sigma}_2^2$ and $\tilde{\sigma}_a^2$ are equivalent noise variance of $\bar{C}_2$ and $C_p$, respectively. From Sect. 3.2, we can approximate the equivalent noise variance $\sigma_2^2$ that $C_2$ in $C_p$ experiences through the channel by setting $\sigma^2_{ch} \leftarrow \tilde{\sigma}_a^2$.

Now, it is obvious that the value of $E_b/N_0$ such that

**Table 2** Degree distributions of $C_a$, $C_1$, and $C_2$ for simulation.

| | | |
|---|---|---|
| | $C_a$ | $\tilde{\lambda}(x) = 0.634x + 0.088x^2 + 0.278x^3$<br>$\tilde{\rho}(x) = 0.030909x^4 + 0.929091x^5 + 0.04x^6$ |
| $\Sigma_o$ | $C_1$ | $\tilde{\lambda}(x) = 0.7908x + 0.2092x^2$<br>$\tilde{\rho}(x) = 0.485714x^9 + 0.508571x^{10} + 0.005714x^{11}$ |
| | $C_2$ | $\tilde{\lambda}(x) = 0.5876x + 0.2352x^2 + 0.1772x^9$<br>$\tilde{\rho}(x) = 0.54806x^4 + 0.45194x^5$ |
| $\Sigma_L$ | $C_1$ | $\tilde{\lambda}(x) = 0.5768x + 0.1308x^3 + 0.2924x^4$<br>$\tilde{\rho}(x) = 0.855385x^{23} + 0.144615x^{24}$ |
| | $C_2$ | $\tilde{\lambda}(x) = 0.6952x + 0.0976x^3 + 0.1668x^4$<br>$+0.0276x^{12} + 0.0128x^{13}$<br>$\tilde{\rho}(x) = 0.796267x^3 + 0.203733x^4$ |
| $\Sigma_R$ | $C_1$ | $\tilde{\lambda}(x) = 0.604x + 0.0872x^2 + 0.3088x^3$<br>$\tilde{\rho}(x) = 0.278857x^6 + 0.714286x^7 + 0.006857x^8$ |
| | $C_2$ | $\tilde{\lambda}(x) = 0.5328x + 0.2988x^2 + 0.1276x^6 + 0.0408x^7$<br>$\tilde{\rho}(x) = 0.036226x^4 + 0.924528x^5 + 0.039245x^6$ |

$$\bar{\sigma}_2^2 = \sigma_2^2 \tag{10}$$

is the performance crossing SNR that gives the same performance for $C_2$ in $C_p$ and $\bar{C}_2$. These results are confirmed by the following simulation results.

## 4. Simulations and Discussions

For BER performance simulations, we used LDPC codes for the component codes. Table 1 shows the target code rates for the simulation of the Plotkin-type UEP-LDPC codes. $C_a$, $C_1$ and $C_2$ are optimized LDPC codes using Gaussian approximation [8] and progressive edge growth (PEG) [7]. The length of $C_a$ and $C_p$ are 5000, those of $C_1$ and $C_2$ are 2500. We decoded the codes $C_p$ (and thus $C_1$ and $C_2$ in $C_p$) by using the MS decoding in Sect. 2. For the decoding of the component codes as EEP codes, the belief propagation (BP) decoding [10] and 100 iterations are applied.

We would like to compare the performance difference between $C_p$ and $C_a$, where $C_p$ is determined by $C_1$ and $C_2$ whose rates are determined by the vertical lines $\Sigma_o$, $\Sigma_L$, and $\Sigma_R$ in Fig. 2. Their code rates are shown in Table 1. In Figs. 5, 6, and 7, the curve for $C_p$ is the average of $C_1$ and $C_2$ in $C_p$. $C_a$ is the code that has the same length and code rate as $C_p$ and optimized by itself. The degree distribution of $C_a$, $C_1$, and $C_2$ for $\Sigma_o$, $\Sigma_L$, and $\Sigma_R$ are shown in Table 2. $\tilde{\lambda}(x)$ and $\tilde{\rho}(x)$ represent the node perspective degree distribution of variable and check nodes, respectively. $C_a$ is the same throughout the simulations in Figs. 5, 6, and 7. The girth of every code becomes 10 by PEG, except for $C_1$ in $\Sigma_L$ which becomes girth 6.

In Fig. 5, $C_p$ and $C_a$ have quite similar performance as we predicted in the previous section. Figures 6 and 7 show the BER performances of rate assignment of $\Sigma_L$ and $\Sigma_R$, respectively. In both Fig. 6 and Fig. 7, $C_p$ is worse than $C_a$ as we could predict, because they do not satisfy (9). But Fig. 6

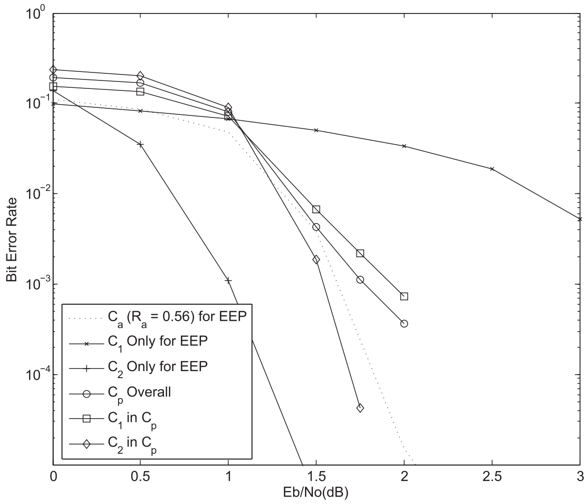



**Fig. 5** Simulation results of $\Sigma_o$.

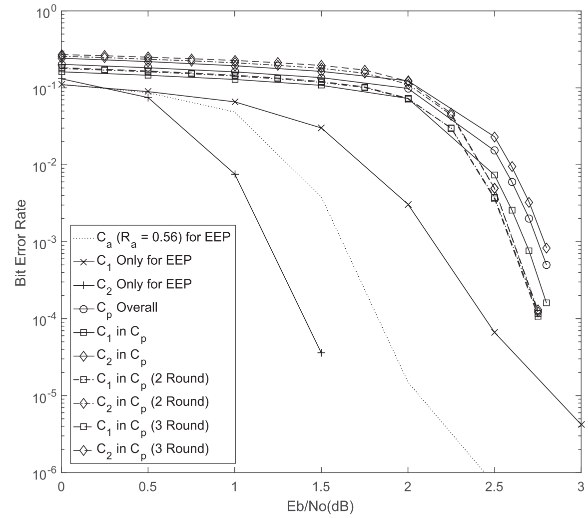



**Fig. 7** Simulation results of $\Sigma_R$.

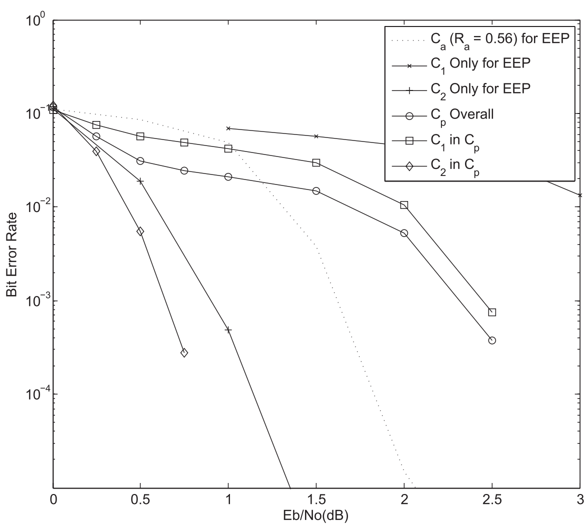



**Fig. 6** Simulation results of $\Sigma_L$.

shows good UEP capability. In other words, $C_1$ and $C_2$ in $C_p$ have wide performance gap and also their performances are better than those of $C_1$ only and $C_2$ only, respectively. These properties are good for the purpose of the UEP codes with the expense of the average performance of $C_p$. In Fig. 7, however, performance of the Plotkin-type code is much worse than $C_a$. This result is due to the poor assignment of the rate for the component codes of the Plotkin-type UEP code $C_p$. We have to avoid this configuration.

In Fig. 7, performances of MR-MS decoding of $\Sigma_R$ are shown for reference. Dashed-dot and dashed lines represent the performances of 2 and 3 round MR-MS decoding, respectively. The result shows that MR-MS decoding can improve the performance of Plotkin-type UEP codes, however it is still worse than the performance of $C_a$. Since the results of MR-MS decoding for $\Sigma_o$ and $\Sigma_L$ have almost same performance as MS decoding, they have been omitted for readability. From the results, we can confirm that the rate allocation is more important than the number of decoding round for the performance of the Plotkin-type UEP code.

We now consider the performance crossing SNR and confirm its analysis. In Fig. 5, BER of $C_2$ in $C_p$ is decreased more rapidly than that of $C_2$ only. The two curves will meet at some point of SNR and this performance crossing SNR can be estimated as we discussed in the previous section. By trial and error, we can scan the value of $E_b/N_0$ which gives the value of $\sigma_2^2$ and $\bar{\sigma}_2^2$ satisfying (10). In Fig. 6, the performance of $C_2$ in $C_p$ starts to exceed for $E_b/N_0 > 0.2$ dB. For this case, we found that $E_b/N_0 = 0.2$ dB gives $\sigma_2^2 = \bar{\sigma}_a^2 = 1.9$, then we could confirm the performance crossing SNR is the same value to 0.2 dB mentioned above. In $E_b/N_0 > 0.2$ dB region, $C_2$ in $C_p$ shows better performance than $C_2$ only. In Fig. 5, $E_b/N_0 = 2.8$ dB gives $\sigma_2^2 = \bar{\sigma}_a^2 = 0.79$, but it is out of the range in the figure.

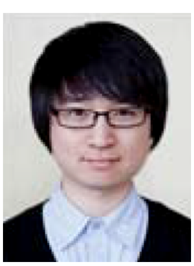
## 5. Conclusion

In this paper, we proposed a guideline for the combination of the component code rates of the Plotkin-type UEP codes to make good UEP capabilities by using the threshold and equivalent noise variance analysis. For a given average code rate, we show that one can draw a graph similar to Figs. 2, 3, and 4. From the figures, we can predict the performance of the Plotkin-type UEP codes and classify the tendency of the three regions. For a good average performance, we should take a code rate combination at the near range of (9) region. And for a good UEP capability, $\Sigma_L$ region will be a good candidate. $\Sigma_R$ region should be considered carefully, because the performance could be worse than those of the component codes. And we can easily estimate the performance crossing SNR which is an $E_b/N_0$ value that the performance of $C_2$ in $C_p$ exceeds the case that $C_2$ is used solely.

### References

[1] G. Caire and G. Lechner, "Turbo codes with unequal error protec-

tion," Electron. Lett., vol.32, no.7, pp.629–631, March 1996.
[2] N. Rahnavard and F. Fekri, "New results on unequal error protection using LDPC codes," IEEE Commun. Lett., vol.10, no.1, pp.43–45, Jan. 2006.
[3] C. Poulliat, D. Declercq, and I. Fijalkow, "Enhancement of unequal error protection properties of LDPC codes," EURASIP J. Wireless Communication and Networking, vol.2007, no.1, Article ID 92659, 2007.
[4] W.J. Van Gils, "Linear unequal error protection codes from shorter codes," IEEE Trans. Inf. Theory, vol.IT-30, no.3, pp.544–546, May 1984.
[5] V. Kumar and O. Milenkovic, "On unequal error protection LDPC codes based on Plotkin-type constructions," IEEE Trans. Inf. Theory, vol.54, no.6, pp.994–1005, June 2006.
[6] K. Huang, C. Liang, X. Ma, and B. Bai, "Unequal error protection by partial superposition transmission using low-density parity-check codes," IET Communications, vol.8, no.13, pp.2348–2355, 2014.
[7] X.-Y. Hu, E. Eleftheriou, and D.M. Arnold, "Regular and irregular progressive edge-growth tanner graphs," IEEE Trans. Inf. Theory, vol.51, no.1, pp.386–398, Jan. 2005.
[8] S.-Y. Chung, T.J. Richardson, and R.L. Urbanke, "Analysis of sum-product decoding of low-density parity-check codes using a Gaussian approximation," IEEE Trans. Inf. Theory, vol.47, no.2, pp.657–670, Feb. 2001.
[9] D.J.C. MacKay, Information Theory, Inference, and Learning Algorithms, ch. 11, Cambridge University Press, Cambridge, 2003.
[10] U. Madhow, Fundamentals of Digital Communication, Cambridge University Press, Cambridge, 2008.



**Jinsoo Park** received the B.S. degree in Electrical and Electronic Engineering from Yonsei University of Seoul, Korea, in 2009. He is currently a Ph.D. student working at Yonsei University. His area of research interest includes channel coding, anti-jamming communications, and wireless communication systems.

**Hong-Yeop Song** received his B.S. degree in Electronic Engineering from Yonsei University in 1984, MSEE and Ph.D. degrees from the University of Southern California, Los Angeles, California, in 1986 and 1991, respectively. He spent 2 years as a research associate at USC and then 2 years as a senior engineer at the standard team of Qualcomm Inc., San Diego, California. Since Sept. 1995, he has been with Dept. of electrical and electronic engineering, Yonsei University, Seoul, Korea. His area of research interest includes digital communications and channel coding, distributed storage codes, design and analysis of various pseudo-random sequences for communications and cryptography. He is a member of IEICE, IEEE, MAA, KICS, KIISC and KMS.