

# 낮은 복잡도의 극 부호 복호 기법

최수용\*, 이충수\*, 박찬수\*, 오왕록<sup>o</sup>

## Low Complexity Hybrid Decoding Scheme for Polar Codes.

Suyong Choi\*, Chungsoo Lee\*, Chansoo Park\*, Wangrok Oh<sup>o</sup>

### 요약

극 부호 (polar code)의 복호 기법인 연속 제거 리스트 (successive cancellation list, SCL) 복호 기법은 비트 오류율 (bit error rate, BER) 측면에서 연속 제거 복호 (successive cancellation, SC) 기법에 비해 성능이 우수하나 리스트 (list)의 크기에 비례하여 복잡도가 증가하는 문제점이 있다. 연속 제거 복호 기법에서 후반부에 복호되는 비트들은 소수의 비트들을 제외하면 해당 비트 채널의 신뢰도가 높아 복호 결과의 신뢰도가 높다. 본 논문에서는 이러한 극 부호 및 연속 제거 복호 기법의 특성을 이용한 낮은 복잡도의 복호 기법을 제안한다. 제안하는 기법은 복호 순서 앞쪽에 위치한 신뢰도가 비교적 낮은 부호 비트들에 대해서는 연속 제거 리스트 복호를 수행하고 이후의 부호 비트에 대해서는 남아있는 경로 중 일부 경로에 대해 연속 제거 복호를 수행한다. 제안하는 기법은 연속 제거 리스트 복호와 비교했을 때 성능 열화 없이 복호 복잡도를 낮출 수 있다.

**Key Words** : Polar code, Successive cancellation decoding, Successive cancellation list decoding, Channel coding, Log-likelihood ratio

### ABSTRACT

Successive Cancellation List (SCL) decoding algorithm for polar codes offers a better performance in terms of bit error rate (BER) compared to SC (successive cancellation) decoding algorithm at the expense of high complexity. On the other hand, bit channels late in order to decode have relatively high reliability in general. Based on this observation, we propose a low complexity hybrid decoding scheme for polar codes. In the proposed scheme, SCL decoding is performed up to a predetermined bit channel and then, the remaining bit channels are decoded with SC decoding. The proposed decoding scheme offers a virtually identical BER performances to that of SCL with relatively low decoding complexity.

### I. 서론

5G NR (new radio) 시스템의 PBCH (physical broadcast channel)의 채널 부호로 사용되고 있는 극 부호 (polar code)는 이진 입력 이산 무기억 대칭 채널 (binary input discrete memoryless symmetric channel, BI-DMS)에서 채널 용량 (channel capacity)

을 달성할 수 있음이 이론적으로 증명된 오류 정정 부호이다. 극 부호는 설계 과정에서 채널 양극화 (channel polarization)를 통해 각 부호 비트 (bit)가 전송되는 비트 채널 (bit channel)의 신뢰도 (reliability)를 구한다. 이 중 신뢰도가 높은 채널은 정보 비트 (information bit)를 매핑 (mapping)하고 신뢰도가 낮은 채널은 고정 비트 (frozen bit)를 매핑한다<sup>1-3</sup>.

※ 본 논문은 2023년도 교육부의 재원으로 한국연구재단의 지원을 받아 수행된 지자체-대학 협력기반 지역혁신 사업의 결과입니다. (2021RIS-004)

• First Author : Dept. of Information Communications Eng., Chungnam National University, tndyd4024@o.cnu.ac.kr, 학생회원

◦ Corresponding Author : Dept. of Information Communications Eng., Chungnam National University, kingrock@cnu.ac.kr, 종신회원

\* Dept. of Information Communications Eng., Chungnam National University, lcs13@o.cnu.ac.kr, 학생회원; pcs2106@o.cnu.ac.kr

논문번호 : 202301-009-A-RN, Received January 20, 2023; Revised March 20, 2023; Accepted April 4, 2023

Arikan은 극 부호의 복호 방법으로 연속 제거 (successive cancellation, SC) 복호를 제안하였다<sup>[1]</sup>. 그러나 연속 제거 복호는 오류 전파 (error propagation) 등의 특성으로 인하여 성능이 열화되는 단점이 있다. 이러한 단점을 극복하기 위해 연속 제거 리스트 (successive cancellation list, SCL) 복호가 제안되었다<sup>[4]</sup>. 연속 제거 리스트 복호는 연속 제거 복호에 비해 향상된 성능을 보이지만 복잡도가 리스트 (list) 크기  $L$ 에 비례하여 증가하고 오류 마루 (error flow) 현상이 발생하는 문제점이 있다. 오류 마루 현상을 해결하기 위해 순환중복검사 (cyclic redundancy check) 부호를 이용한 연속 제거 리스트 복호 (CRC-aided SCL, CA-SCL)가 제안되었으며 이를 통해 오류 마루 현상을 낮출 수 있다<sup>[5]</sup>.

연속 제거 리스트 복호의 복잡도를 개선하기 위해 신뢰도가 높은 정보 비트들을 복호할 때 복잡도가 낮은 연속 제거 복호를 사용하는 기법이 제안되었다<sup>[6]</sup>. 해당 복호 기법은 연속 제거 리스트 복호 과정에서 필요한 경로 선택과 관련된 복잡도를 줄일 수 있다는 장점이 있지만 복호 과정에서 요구되는 로그 우도비 (log-likelihood ratio, LLR) 계산 복잡도는 기존의 연속 제거 리스트 복호 기법과 동일하다.

본 논문에서는 경로 선택과 관련된 복잡도뿐만 아니라 LLR 계산 복잡도를 낮출 수 있는 복호 기법을 제안한다. 제안하는 기법은 특정 비트까지는 연속 제거 리스트 복호를 수행하고 해당 비트 이후부터는 남아있는  $L$ 개의 경로 중  $T, T < L$ 개의 경로에 대해 각

각 연속 제거 복호를 수행한다. 제안하는 기법은 연속 제거 리스트 복호와 비교했을 때 성능 열화가 거의 없이 경로 선택 및 LLR 계산에 요구되는 복잡도를 낮출 수 있다.

본 논문의 구성은 다음과 같다. 먼저 II장에서는 시스템 모델을 기술하고 연속 제거 복호와 연속 제거 리스트 복호의 복잡도를 기술한다. III장에서 제안하는 복호 기법을 설명하며 IV장에서는 제안하는 기법의 성능과 복잡도를 확인한다. 마지막으로 V장에서 결론을 맺는다.

## II. 시스템 모델

극 부호는 설계 과정에서 가우시안 근사화 (gaussian approximation, GA)를 통해 채널 양극화를 진행한다. 그림 1은 부호어 길이  $N=4$ 인 경우 가우시안 근사화 과정을 나타낸다. 그림 1에서  $R_i^j$ 는  $i=0, 1, \dots, \log_2 N$  단계 (stage)에서  $j=0, 1, \dots, N-1$ 번째 노드의 신뢰도를 나타내며 실선과 점선은 식 (1),(2)에 나타난 노드 연산  $\beta(\cdot)$ 와  $\gamma(\cdot)$ 를 각각 나타낸다. 여기에서 식 (1)의  $\phi(x)$ 는 식 (3)과 같이 표현된다. 초기값  $R_0^j, j=0, 1, \dots, N-1$ 은 모두  $\frac{2}{\sigma_d^2}$ 로 설정되며  $1 \leq i \leq \log_2 N$ 인 경우에  $R_i^j$ 는 식 (4)에 나타난 노드 연산을 통해 계산된다. 이때 극 부호 설계 신호대 잡음비 (designed signal to noise ratio)가  $\eta$  [dB]일 경우  $\sigma_d^2 = 10^{\frac{\eta}{10}}$ 로 주어진다. 채널 양극화 과정을 통해 얻어낸  $R_{\log_2 N}^j$ 를  $j$ 번째 비트 채널의 신뢰도로 정의한다.  $R_{\log_2 N}^j, j=0, 1, \dots, N-1$  중 크기가 큰  $K$ 개 비트 채널 인덱스  $j$ 들로 이루어진 집합을  $I$ , 나머지  $N-K$ 개의 비트 채널 인덱스 집합을  $I^c$ 로 정의한다. 길이  $N$ 이 2의 승수로 주어지는 벡터  $u_0^{N-1} = [u_0, u_1, \dots, u_{N-1}]$ 의 요소 중  $u_i, i \in I$ 에는 정보 비트와 순환중복검사 부호를 매핑하고  $u_i, i \in I^c$ 에는 고정비트 '0'을 매핑한다. 정보 및 고정 비트를 매핑한  $u_0^{N-1}$ 을 식 (5)를 통해 부호어 벡터  $c_0^{N-1}$ 로 변환한다. 식 (5)에서  $G_N$ 은 하삼각 정방 행렬

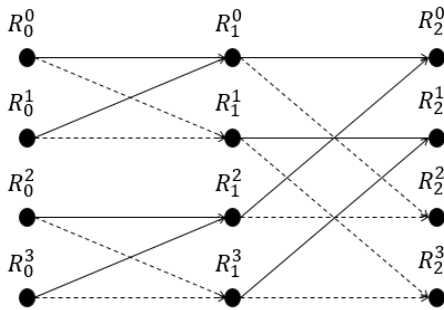


그림 1. 극 부호의 채널 양극화 과정.( $N=4$ )  
Fig. 1. Channel polarization process of polar code.

$$\beta(R_{i-1}^j, R_{i-1}^{j+2^{i-1}}) = \phi^{-1}(1 - (1 - \phi(R_{i-1}^j))(1 - \phi(R_{i-1}^{j+2^{i-1}}))). \quad (1)$$

$$\gamma(R_{i-1}^j, R_{i-1}^{j+2^{i-1}}) = R_{i-1}^j + R_{i-1}^{j+2^{i-1}}. \quad (2)$$

$F = \begin{bmatrix} 1 & 0 \\ 1 & 1 \end{bmatrix}$ 의  $n = \log_2 N$ 차 Kronecker power 연산  $F^{\otimes n}$ 을 통해 구한다<sup>[1]</sup>.

$$\phi(x) = \begin{cases} 1 - \frac{1}{\sqrt{4\pi x}} \int_{-\infty}^{\infty} \tanh \frac{u}{2} e^{-\frac{(u-x)^2}{4x}} du, & x > 0 \\ 1, & x = 0 \end{cases} \quad (3)$$

$$R_i^j = \begin{cases} f(R_{i-1}^j, R_{i-1}^{j+2^{i-1}}), & 0 \leq j \leq \frac{N}{2} - 1 \\ g(R_{i-1}^j, R_{i-1}^{j+2^{i-1}}), & \frac{N}{2} \leq j \leq N-1 \end{cases} \quad (4)$$

$$c_0^{N-1} = u_0^{N-1} G_N. \quad (5)$$

부호어 벡터  $c_0^{N-1}$ 을 BPSK (binary phase shift keying) 변조하여  $s_0^{N-1}$ 를 생성하고 생성된 변조 심볼 (symbol)은 평균이 0이고 양대역 전력 밀도가  $\frac{N_0}{2}$ 인 가산성 백색 가우시안 잡음 (additive white gaussian noise, AWGN) 채널을 통과한다. 수신된 신호  $y_0^{N-1}$ 를 복조하여  $c_0^{N-1}$ 의 로그 우도비 (log-likelihood ratio, LLR)  $L(c_0^{N-1})$ 을 구하고 이를 이용하여 극 부호를 복호한다.

연속 제거 복호는 그림 2가 반복되는 구조를 가진 다<sup>[1]</sup>. 그림 2에서  $\hat{b}_i$ 와  $\hat{b}_{i+1}$ 의 LLR  $L(\hat{b}_i)$ 와  $L(\hat{b}_{i+1})$ 을 구하는 연산은 각각 식 (6), (7)과 같이  $f(\cdot)$ 와  $g(\cdot)$ 로 정의할 수 있다. 최종 단계에서 구해진 LLR을 경판정 (hard decision)하여  $u_i$ 의 추정값  $\hat{u}_i$ 을 구한다. 연속제거 복호에서 LLR 계산 복잡도는  $O(N \log_2 N)$ 으로 나타낼 수 있다.

연속 제거 복호와 달리 연속 제거 리스트 복호는 정보 비트를 복호할 때 0과 1에 해당하는 경로를 모두 고려한다. 따라서 정보 비트를 복호할 때마다 경로의 수는 2배로 증가하며 만약 후보 경로군의 수가  $L$ 개를 초과하면 경로 매트릭이 큰 순서대로  $L$ 개의 경로만 선택되고 나머지 경로는 제거된다<sup>[4]</sup>.  $u_i = 0$ 인 경우

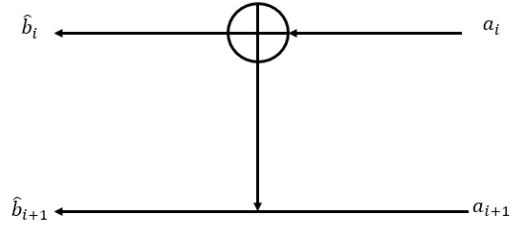


그림 2. 극 부호 복호기의 기본 구조  
Fig. 2. Basic structure of decoder for polar codes.

경로 매트릭  $m_i^{2^l}$ 과  $u_i = 1$ 인 경우 경로 매트릭  $m_i^{2^{l+1}}$ 은 식 (8), (9)와 같이 나타낼 수 있다. 식 (8), (9)에서  $l = 0, 1, \dots, L-1$ 은  $i-1$ 번째 비트까지 생존한 경로의 경로 인덱스를 나타내며  $L(\hat{u}_i^l)$ 은  $l$ 번째 경로에서 도출된  $u_i$ 에 대한 LLR이다.  $u_i$ 를 복호할 때  $i \in I$ 인 경우  $u_i = 0$ 과  $u_i = 1$ 을 모두 고려하므로 식 (8), (9)의 값을 모두 계산하여야 하며  $i \in I^c$ 인 경우 항상 0으로 복호되기 때문에 식 (8)의 값만 계산한다.

$$m_i^{2^l} = m_{i-1}^l + L(\hat{u}_i^l) - \ln(1 + e^{L(\hat{u}_i^l)}). \quad (8)$$

$$m_i^{2^{l+1}} = m_{i-1}^l - \ln(1 + e^{L(\hat{u}_i^l)}). \quad (9)$$

연속 제거 리스트 복호에서 LLR 계산 복잡도는  $O(LN \log_2 N)$ 으로 나타낼 수 있다. 정보 비트를 복호하는 경우  $2L$ 개의 경로 매트릭 중  $L$ 개의 경로를 선택하는 과정이 필요하며 이때  $2L$ 개의 경로 매트릭에 대해 정렬 연산이 추가적으로 요구된다. 표 1에 길이가  $N$ 이고 부호율 (code rate)이  $K/N$ 인 극 부호의 연속 제거 리스트 복호에 요구되는 복잡도를 나타냈다. 이때  $K$ 는 정보 비트의 수이며 정렬을 위해 직렬 정렬기를 사용하는 것을 가정하였다<sup>[6]</sup>. 연속 제거 리스트 복호를 사용하는 경우 연속 제거 복호에 비해 우수한 성능을 얻을 수 있지만 표 1에 기술된 바와 같이 복잡도가  $L$ 에 비례해서 증가하는 단점이 있다.

표 1에 나타난 복잡도 중 경로 매트릭 비교 연산을 줄이기 위해 연속 제거 리스트 복호 과정에서 비트 채널의 신뢰도가 높은 비트는 계산된 LLR 값을 경판정

$$L(\hat{b}_i) = f(L(a_i), L(a_{i+1})) = \tanh^{-1}(\tanh(\frac{L(a_i)}{2}) \tanh(\frac{L(a_{i+1})}{2})), i = 0, 2, 4, \dots, N-2. \quad (6)$$

$$L(\hat{b}_{i+1}) = g(L(a_i), L(a_{i+1})) = (-1)^{\hat{b}_i} L(a_i) + L(a_{i+1}), i = 0, 2, 4, \dots, N-2. \quad (7)$$

표 1. 연속 제거 리스트 복호의 복잡도  
Table 1. Complexites for SCL decoding.

| Operations           | LLR calculation | PM calculation   | Comparator                   |
|----------------------|-----------------|------------------|------------------------------|
| Number of operations | $O(LN\log_2 N)$ | $2LK + L(N - K)$ | $(K - \log_2 L)2L\log_2(2L)$ |

하여 복호하는 기법이 제안되었다<sup>7)</sup>. 해당 극 부호 복호 기법은  $N = 256$ 이고 부호율이 1/2인 극 부호를 복호할 때  $L = 4$ 인 연속 제거 리스트 복호에 비해 경로 매트릭 비교 연산을 약 72.2%까지 줄일 수 있으나 경로 매트릭 계산에 필요한 LLR 계산에 요구되는 복잡도는 기존의 연속 제거 리스트 복호 기법과 동일하며 LLR 계산의 경우 경로 매트릭 비교 연산에 비하여 높은 계산 복잡도가 요구된다<sup>8)</sup>.

### III. 제안하는 기법

그림 3에  $N = 256$ ,  $K = 136$ 이고  $\eta$ 가 2 dB일 때 가우시안 근사화 기반 극 부호 설계를 통해 얻은 비트 채널의 신뢰도를 복호 순서에 따라 나타냈다. 그림 3에서 확인할 수 있는 바와 같이 극 부호는 비트 채널의 신뢰도가 작은 비트가 대부분 복호 순서 앞부분에 위치하는 특징이 있다. 채널의 신뢰도가 낮은 정보 비트에 대해 연속 제거 복호 기법을 적용할 경우 해당 비트에서 복호 오류가 발생하고 이로 인해 오류 전파가 발생하며 성능이 열화되는 문제가 있다. 본 논문에서

서는 비트 채널 신뢰도가 미리 설정된  $\alpha$ 보다 작은 비트 채널들의 인덱스 중 가장 큰 값을 찾고 해당 인덱스에 매핑되는 비트까지는 연속 제거 리스트 복호를 수행하고 이후 비트부터는 남아있는  $L$ 개의 경로 중 경로 매트릭이 가장 큰  $T$ 개의 경로에 대해 연속 제거 복호를 수행하는 기법을 제안한다. 제안하는 기법을 도식화하면 그림 4와 같다. 그림 4는  $L = 4$ ,  $T = 2$ 인 경우이며 그림 4에 나타난 바와 같이  $M$ 번째 부호 비트까지는 연속 제거 리스트 복호를 수행하고  $M + 1$ 번째 부호 비트부터는 남아있는  $L = 4$ 개의 경로 중  $T = 2$ 개의 경로에 대해 각각 연속 제거 복호를 수행하며 모든 비트에 대한 복호가 완료된 후  $T$ 개의 경로 중 CRC 검사를 통과한 경로를 부호어로 선택한다. 제안하는 기법은  $M + 1$ 번째 부호 비트부터는  $T$ 개의 경로에 대해서만 독립적으로 연속 제거 복호를 진행하므로 LLR 계산이 더 이상 필요하지 않을 뿐만 아니라 경로 매트릭 계산 및 경로 매트릭 비교 연산이 필요하지 않다. 제안하는 기법의 복잡도를 요약하면 표 2와 같다. 여기에서  $N_i$ 는 연속 제거 리스트 복호로 복호되는  $M$ 개의 비트들 중 정보 비트의 수를 나타낸다. 표 2를 통해 알 수 있듯이  $M$ 이 작아질수록  $N_i$  또한 작아

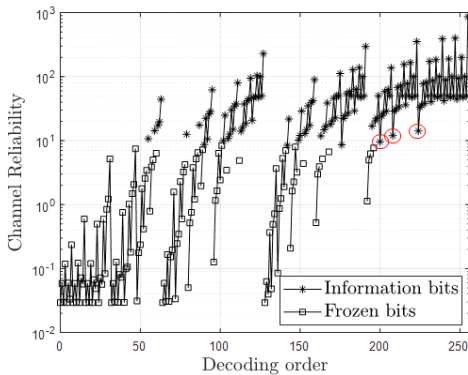


그림 3. 복호 순서에 따른 비트 채널의 신뢰도  
Fig. 3. Reliability of bit channels versus decoding order.

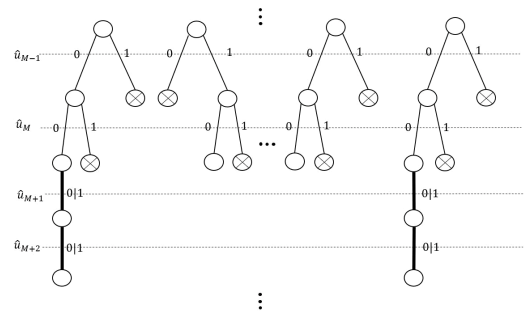


그림 4.  $L = 4$ ,  $T = 2$ 인 제안하는 복호 기법  
Fig. 4. Proposed scheme ( $L = 4$ ,  $T = 2$ ).

표 2. 제안하는 기법의 복잡도  
Table 2. Complexity of proposed scheme.

| Operations           | LLR calculation                    | PM calculation       | Comparator                     |
|----------------------|------------------------------------|----------------------|--------------------------------|
| Number of operations | $O(LM\log_2 N + T(N - M)\log_2 N)$ | $2LN_i + L(M - N_i)$ | $(N_i - \log_2 L)2L\log_2(2L)$ |

지므로 기존 기법 대비 복잡도를 크게 낮출 수 있다.

추가적으로 복잡도를 낮추기 위해 극 부호 설계 과정에서 복호 순서 후반부에 있는 신뢰도가 낮은 비트 채널에 정보 비트 대신 고정 비트를 할당하고 해당 정보 비트들을 고정 비트가 할당된 비트 채널 중 신뢰도가 높은 비트 채널에 할당하는 극 부호 설계 기법을 제안한다. 그림 3에 나타난  $N=256, K=136$ 인 경우의 채널 신뢰도 결과를 바탕으로 200, 208, 224번째로 복호되는 비트 채널에 고정 비트를 할당하고 고정 비트가 할당되어 있던 비트 채널 중 신뢰도가 가장 높은 47, 147, 101번째 비트 채널에 해당 정보 비트를 할당하여 극 부호를 설계할 경우 제안하는 기법에서  $M$ 을 크게 줄일 수 있어 복호 과정에서 요구되는 복잡도를 크게 낮출 수 있다. 또한, 고정 비트로 전환되는 비트들과 정보 비트로 전환되는 비트들의 채널 신뢰도 값에 큰 차이가 없어 극 부호 설계 기법 변경으로 인한 성능 열화는 크지 않다.

#### IV. 전산 실험 결과

정보 비트가 할당된 일부 비트 채널에 고정 비트를 할당하고 고정 비트가 할당된 비트 채널 중 신뢰도가 높은 비트 채널에 해당 정보 비트를 할당하는 방식으로 수정된 극 부호의 성능을 기존 설계 기법 중 가우시안 근사화 기법으로 설계된 경우의 성능과 비교하였으며 그 결과는 그림 5와 같다<sup>3)</sup>. 이때 사용된 주요 파라미터는 표 3과 같으며 복호 기법으로는  $L=4$ 인 연속 제거 리스트 복호를 사용하였다. 그림 5를 통해 확인할 수 있는 바와 같이 제안하는 복호 기법의 복잡도를 더욱 낮추기 위하여 제안한 수정된 극 부호 설계

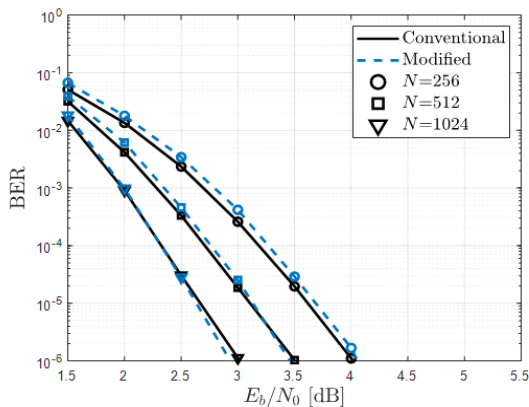


그림 5. 수정된 극 부호 설계 기법의 BER 성능  
Fig. 5. BER performance of proposed polar code construction.

표 3. 전산 실험 파라미터  
Table 3. Simulation parameters.

| Parameters               | Values              |
|--------------------------|---------------------|
| Channel model            | AWGN channel        |
| $\eta$                   | 2 dB                |
| $N$                      | 256, 512, 1024      |
| $K$                      | 136 (CRC 8 bits)    |
| $L$                      | 4                   |
| CRC generator polynomial | $x^8 + x^2 + x + 1$ |

표 4.  $N$ 에 따른  $\alpha$ 와  $M$ 의 후보군  
Table 4. Candidates of  $\alpha$  and  $M$  versus  $N$ .

| $N$  | $\alpha$ | $M$ |
|------|----------|-----|
| 256  | 9        | 176 |
|      | 17       | 195 |
|      | 25       | 201 |
| 512  | 21       | 402 |
|      | 25       | 418 |
|      | 30       | 450 |
| 1024 | 17       | 804 |
|      | 20       | 834 |
|      | 24       | 898 |

로 인한 비트 오류 (bit error rate, BER)  $10^{-5}$  영역에서의 성능 열화는 크지 않음을 확인할 수 있다.

제안하는 기법의 성능을 확인하기 위해 표 3 및 4에 나타난 파라미터를 바탕으로 전산 실험을 수행하였으며 그 결과는 그림 6-8과 같다. 모든 전산 실험에서는 성능 확인을 위하여 주어진  $E_b/N_0$ 에 대한 1,000개 이상의 비트 오류를 관찰하였다. 표 4의  $M$ 은 수정된 극 부호 설계 기법을 이용했을 때 비트 채널 신뢰도가  $\alpha$ 보다 작은 비트 채널들의 인덱스 중 가장 큰 값이며 해당 값이 작을수록 제안하는 기법을 통한 복잡도 감소 효과가 크다.  $N=256$ 인 경우  $\alpha$ 는 그림 3에 붉은 원으로 표시한 인덱스와 같이 복호 순서 후반부에 위치한 비트 채널 중 다른 비트 채널에 비해 신뢰도가 낮은 비트 채널의 신뢰도 값을 기준으로 설정하였으며  $N=512, 1024$ 의 경우도 같은 방법으로  $\alpha$ 를 설정하였다.  $\alpha$ 를 조절함으로써 제안하는 기법의 복호 성능과 복잡도 간 최적의 트레이드 오프 (trade-off)점을 선택할 수 있다. 그림 6에서 확인할 수 있는 바와 같이  $N=256$ 일 때 제안하는 기법은  $T \geq 2$ 인 경우 고려한 모든  $M$ 에 대해  $BER = 10^{-6}$  지점에서 기존 기법 대비 BER 성능 열화가 약 0.3 dB

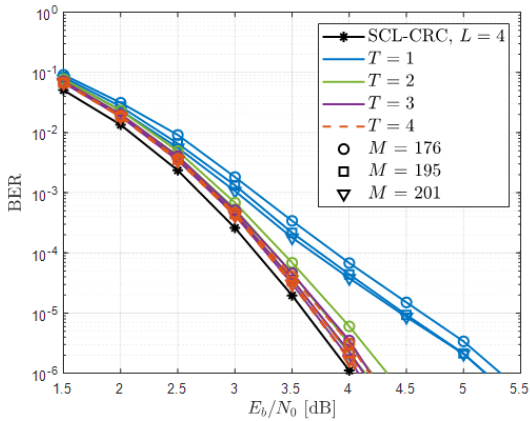


그림 6.  $N=256$ 일 때 제안하는 기법의 BER 성능  
Fig. 6. Performance of the proposed scheme ( $N=256$ ).

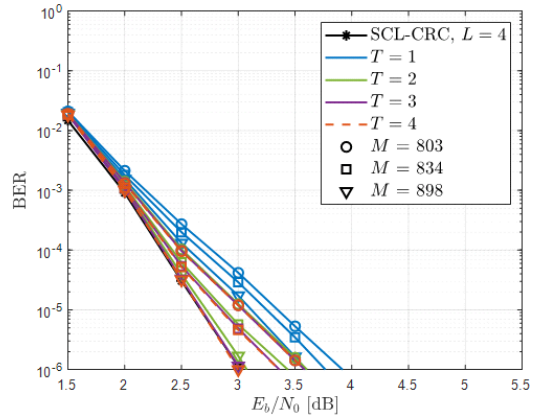


그림 8.  $N=1024$ 일 때 제안하는 기법의 BER 성능  
Fig. 8. Performance of the proposed scheme ( $N=1024$ ).

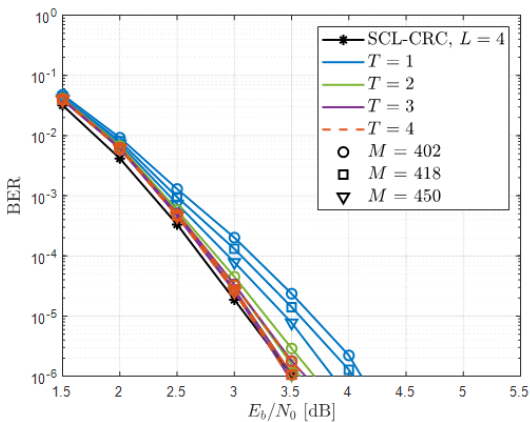


그림 7.  $N=512$ 일 때 제안하는 기법의 BER 성능  
Fig. 7. Performance of the proposed scheme ( $N=512$ ).

보다 작다. 특히  $T \geq 2, M \geq 195$ 에 대해서는 기존 기법 대비 BER 성능 열화가 0.1 dB 이내로 매우 작다. 이 중에서  $T=4$ 는 기존 기법 대비 성능 열화가 거의 없으나 LLR 계산에 요구되는 복잡도는 줄일 수 없는 경우에 해당한다.  $N=512$ 일 때 제안하는 기법의 성능은 그림 7을 통해 확인할 수 있다. 그림 7을 통해 확인할 수 있는 바와 같이  $N=512$ 일 때  $T \geq 2$ 이고  $M \geq 418$ 인 경우 BER =  $10^{-6}$  지점에서 기존 기

법과 거의 동일한 비트 오류 성능을 보인다.  $N=1024$ 의 경우에도 이전 경우와 유사하게  $T \geq 2$ 이고  $M \geq 898$ 인 경우 비트 오류  $10^{-6}$ 영역에서 기존 기법 대비 BER 성능 열화가 거의 없는 것을 확인할 수 있으며 이는 그림 8을 통해 확인할 수 있다.

표 1과 2를 통해 고려한  $T$  및  $M$ 에 따른 복잡도 감소량을 구할 수 있으며 이를 표 5에 나타냈다. 표 5에서  $T, M$ 은 전산 실험 결과를 바탕으로 성능 열화가 거의 없이 복호 복잡도를 줄일 수 있는 값이다. 표 5를 통해 알 수 있는 바와 같이 제안하는 기법은 성능 열화가 거의 없이 기존 기법보다 LLR 계산, 경로 매트릭 계산, 경로 매트릭 비교 연산을 각각 최대 11.72%, 27.55%, 38.81% 줄일 수 있다.

## V. 결론

본 논문에서는 복호 순서 후반부에 위치한 채널의 신뢰도가 낮은 특정 비트까지는 연속 제거 리스트 복호를 수행하고 해당 비트 이후에는 연속 제거 복호를 수행하는 방식의 복호 기법을 제안하였다. 제안하는 기법을 사용하는 경우 성능 열화가 거의 없이 LLR 계산, 경로 매트릭 계산 및 경로 매트릭 비교 연산을 감소시킬 수 있다.

표 5. 연속 제거 리스트 복호 대비 제안하는 기법의 복잡도 감소량 (%)  
Table 5. Amount of complexity reductions of the proposed scheme (%).

| $N$  | $T$ | $M$ | LLR calculation | PM calculation | Comparator |
|------|-----|-----|-----------------|----------------|------------|
| 256  | 2   | 195 | 11.72           | 27.55          | 38.81      |
| 512  | 2   | 418 | 9.18            | 23.97          | 35.11      |
| 1024 | 2   | 898 | 6.15            | 16.32          | 24.32      |



References

- [1] E. Arıkan, "Channel polarization: A method for constructing capacity-achieving codes for symmetric binary-input memoryless channels," *IEEE Trans. Inf. Theory*, vol. 55, no. 7, pp. 3051-3073, Jul. 2009.  
(<https://doi.org/10.1109/TIT.2009.2021379>)
- [2] GPP TS 38.212 v15.3.0, "Multiplexing and channel coding (release 15)," Oct. 2018.
- [3] H. Li and J. Yuan, "A practical construction method for polar codes in AWGN channels," in *TENCON Spring Conf.*, pp. 223-226, Sydney, NSW, Apr. 2013.  
(<https://doi.org/10.1109/TENCONSpring.2013.6584444>)
- [4] I. Tal and A. Vardy, "List decoding of polar codes," in *Proc. IEEE ISIT*, pp. 1-5, Jul. 2011.  
(<https://doi.org/10.1109/ISIT.2011.6033904>)
- [5] K. Niu and K. Chen, "CRC-aided decoding of polar codes," *IEEE Commun. Lett.*, vol. 16, no. 12, pp. 2044-2047, 2012.  
(<https://doi.org/10.1109/LCOMM.2012.090312.121501>)
- [6] D. E. Knuth, *Art of Computer Programming, The: Sorting and Searching*, vol. 3, 2nd Ed., Addison-Wesley Professional, Apr. 1998.
- [7] P. S. Park, H. S. Ju, and S. H. Kim, "A low-complexity successive cancellation list decoding of polar codes," in *Proc. Symp. KICS*, pp. 54-55, 2021.
- [8] I. Koren, *Computer Arithmetic Algorithms*, 2nd Ed., A K Peters/CRC Press, Nov. 2001.

최수용 (Suyong Choi)



2021년 2월: 충남대학교 정보통신공학과 학사  
 2023년 2월: 충남대학교 전파정보통신공학과 석사  
 <관심분야> 통신시스템 설계 및 구현, 디지털 통신  
 [ORCID:0000-0003-4515-8904]

이충수 (Chungsoo Lee)



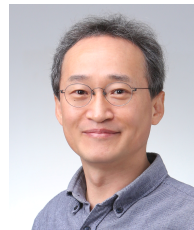
2022년 2월: 충남대학교 전파정보통신공학과 학사  
 2022년 3월~현재: 충남대학교 전파정보통신공학과 석사과정  
 <관심분야> 통신시스템 설계 및 구현, 디지털 통신  
 [ORCID:0009-0001-3281-360X]

박찬수 (Chansoo Park)



2023년 2월: 충남대학교 전파정보통신공학과 학사  
 2023년 3월~현재: 충남대학교 전파정보통신공학과 석사과정  
 <관심분야> 통신시스템 설계 및 구현, 디지털 통신  
 [ORCID:0009-0009-2044-2336]

오왕록 (Wangrok Oh)



1994년 2월: 포항공과대학교 학사  
 1997년 2월: 포항공과대학교 석사  
 2003년 8월: 포항공과대학교 박사  
 1997년~2000년: 포항공과대학교 정보통신연구소 전임연구원  
 2003년~2006년: 포항공과대학교 정보통신연구소 전임연구원  
 2006년~2010년: 충남대학교 정보통신공학과 조교수  
 2010년~2015년: 충남대학교 정보통신공학과 부교수  
 2015년~현재: 충남대학교 정보통신공학과 교수  
 <관심분야> 통신시스템 설계 및 구현, 오류정정부호, MIMO 시스템  
 [ORCID:0000-0001-8205-5432]