

# LDPC 부호화 시스템을 위한 천공 비트 선복구 기법

박 상 준\*

## Pre-Recovery of Punctured Bits for LDPC Coded Systems

Sangjoon Park\*

### 요 약

본 논문에서는 LDPC 부호가 적용된 시스템을 위한 천공 부호 비트들의 선복구 기법을 제안한다. 제안 기법은 천공된 비트 노드들의 생존 체크 노드를 이용한 단계 복구 순서에 따른 순차적인 비트 메시지 계산 과정을 통해 모든 천공 비트 노드의 비트 메시지를 반복 복호 과정이 시작되기 이전에 복구한다. 따라서 메시지 전파 과정에서 천공 비트의 복구로 인한 메시지 전파 지연을 개선할 수 있다. 모의실험 결과 제안 기법이 적용된 경우 천공량이 증가할수록 일반적인 반복 복호 기법 대비 오류 성능 및 복호 수렴 속도가 개선됨을 확인하였다.

**Key Words** : LDPC, Puncturing, Recovery, Step-Recovery, Iterative Decoding

### ABSTRACT

In this letter, a pre-recovery scheme for punctured coded bits is proposed for LDPC coded systems. Before the iterative decoding process, the proposed scheme recovers bit messages of all punctured bit nodes according to the step recovery order through the sequential recovery procedures using survived check nodes of punctured bit nodes. Therefore, the proposed scheme can enhance the message propagation delay caused by the recovery of punctures bit nodes during the iterative decoding process. Simulation results show that, compared

with the conventional iterative decoding without any pre-recovery procedures, the proposed scheme can enhance the error performance and decoding convergence speed as the number of punctured bit nodes increases.

## 1. 서 론

LDPC (Low-Density Parity-Check, 저밀도 패리티 검사) 부호는 Shannon 한계에 근접하는 오류 정정 능력 및 반복 복호 과정의 낮은 복호 연산량을 갖는 오류 정정 부호로 상용 이동통신 시스템 등에 채택되어 널리 이용되고 있다.<sup>[1-3]</sup> 이러한 LDPC 부호를 사용한 시스템에서 부호율 (code rate) 조절을 위해 천공 (puncturing) 기법을 적용할 수 있다.<sup>[4-6]</sup> 하지만 LDPC 부호의 반복 복호 과정에서 천공 비트 노드가 non-zero LLR (Log-Likelihood Ratio)을 갖기 이전, 즉 천공 비트 노드의 복구 (recovery) 이전에는 해당 천공 비트 노드와 동일한 체크 노드에 연결된 비트 노드들의 LLR이 갱신되지 않는다. 따라서 LDPC 부호에 천공 기법이 적용된 경우 천공 비트의 수의 증가에 따라 복호 수렴 속도의 저하 및 반복 복호 (iteration) 수에 따른 오류 성능 저하가 발생할 수 있다. 이를 해결하기 위해 Layered BP (Belief Propagation) 및 Shuffled BP 등 체크 노드 또는 비트 노드를 여러 그룹으로 나누어 순차적으로 복호하는 방식을 이용하여, 천공 비트의 단계 복구 (step recovery, SR)에 따른 노드 그룹 설정을 통해 천공된 부호어의 복호 수렴 속도를 가속화할 수 있는 복호 기법들이 연구되어 왔다.<sup>[7-9]</sup> 하지만 이러한 기법들은 노드 그룹간의 순차적인 메시지 전달 과정으로 인해 동일한 반복 복호 횟수를 갖는 일반적인 BP 방식 대비 높은 복호 지연 시간 (decoding latency)을 요구한다. 따라서 천공 비트 수 증가에 따라 노드 그룹 수가 증가하거나 반복 복호 횟수가 증가할 경우 보다 많은 복호 지연 시간의 증대를 불러올 수 있다.

이에 본 논문에서는 LDPC 부호화 시스템에서 천공 기법이 적용되었을 때 복호 수렴 속도를 개선하기 위한 천공 부호 비트들의 선복구 기법을 제안한다. 제안 기법에서는 각 천공 비트 노드들을 복구시켜주는 체크 노드인 생존 체크 노드 (survived check node)들

\* 이 논문은 2019년도 과학기술정보통신부의 재원으로 한국연구재단의 지원을 받아 수행된 연구임(No. NRF-2019R1C1C1003202).

• First and Corresponding Author : (ORCID:0000-0002-6684-9803)Kyonggi University, Department of Electronic Engineering, sj.park@kgu.ac.kr, 조교수, 정회원

논문번호 : 202009-219-A-LU, Received September 04, 2020; Revised September 23, 2020; Accepted September 25, 2020

을 이용하여 단계 복구 순서에 따라 각 천공 비트 노드의 비트 메시지를 순차적으로 복구한다. 이러한 선 복구 과정은 반복 복호 과정이 시작하기 이전 수행되며, 따라서 제안 기법은 천공량에 관계없이 반복 복호 과정에서의 LLR 미갱신 현상을 방지하고 이를 통한 복호 수렴 속도 개선 및 오류 성능 향상을 도모한다. 또한 제안 기법은 선복구 이후의 반복 복호 과정에서 그룹별 순차적인 복호 과정을 요구하지 않으므로 선 복구 과정의 진행에 따른 오버헤드 이외에 반복 복호 횟수 또는 천공 비트 수 증가에 따른 반복 복호 과정에서의 추가적인 복호 지연 시간 증대를 요구하지 않는다.

## II. 시스템 모델

본 논문에서는 메시지 비트 및 부호화 비트의 수가 각각  $K$  및  $N$ 인 LDPC 부호를 사용한다. 송신단에서는 먼저  $1 \times K$  메시지 비트 시퀀스  $\mathbf{d}$ 에 대한 LDPC 부호화 (encoding) 과정을 진행하여 모부호 (mother codeword)인  $1 \times N$  부호 비트 시퀀스  $\mathbf{c}$ 를 생성한다. 이후 부호율 조절을 위한 천공 과정 진행을 통해  $\mathbf{c}$ 로부터  $P$ 개의 부호 비트를 천공한 부호 비트 시퀀스  $\mathbf{p}$ 를 생성하며, 따라서  $\mathbf{p}$ 의 부호율은  $r = K/(N - P)$ 이다. 이렇게 생성된 천공 부호 비트 시퀀스  $\mathbf{p}$ 는 송신 심볼 시퀀스  $\mathbf{x}$ 로 변조되어 송신된다. 이 때  $i$ 번째 송신 심볼  $x_i$ 에 대한 시스템 입출력 관계식은 다음과 같다.

$$y_i = h_i x_i + n_i \quad (1)$$

식 (1)에서  $y_i$ 는  $x_i$ 에 대한 수신 신호이며,  $h_i$ 는 채널 계수,  $n_i$ 는 AWGN (Additive White Gaussian Noise, 가산 백색 가우시안 잡음) 신호이다. 모든  $x_i$ 에 대한  $y_i$ 가 수신된 이후 수신단은 반복 복호 과정을 진행하여 메시지 비트 시퀀스  $\mathbf{d}$ 에 대한 추정치  $\hat{\mathbf{d}}$ 을 얻는다.

## III. 제안 선복구 기법

### 3.1 천공 부호를 위한 반복 복호 과정

제안 기법의 설명에 앞서 본문에서는 천공 부호를 위한 일반적인 반복 복호 과정에 대하여 간략히 기술한다.<sup>[1]</sup> 편의상 본문에서는 min-sum 복호를 가정하여

기술한다. 먼저 반복 복호 과정의 진행에 앞서 초기화 과정에서는 우선 모부호에 포함된 부호 비트들의 사전 LLR (a priori LLR)  $l_n$ 을 계산한다. 이 때 천공되지 않은 부호 비트들의 경우 전송과정을 통해 얻어진 수신 신호 및 채널 정보 등에 따라 사전 LLR이 계산되며, 전송되지 않은 천공 부호 비트의 경우 사전 LLR이 0으로 초기화된다. 초기화 과정의 다음 단계에서는 각 비트 노드  $n$ 에 대해  $m$ 번째 체크 노드에 전달되는 비트 메시지  $z_{mn}$ 를 다음과 같이 초기화한다.

$$z_{mn} = l_n, \forall m \in \mathbf{M}(n) \quad (2)$$

식 (2)에서  $\mathbf{M}(n)$ 은  $n$ 번째 비트 노드에 연결된 체크 노드의 집합을 나타낸다.

초기화 과정이 종료된 이후에는 반복 복호 과정을 진행하여 체크 및 비트 메시지를 반복적으로 갱신한다. 먼저  $m$ 번째 체크 노드에서  $n$ 번째 비트 노드로 전달되는 체크 메시지  $\epsilon_{mn}$ 을 다음과 같이 계산한다.

$$\epsilon_{mn} = \min_{\mathbf{N}_n(m)} |z_{mn'}| \cdot \prod_{\mathbf{N}_n(m)} \text{sign}(z_{mn'}) \quad (3)$$

이 때  $\mathbf{N}_n(m)$ 은  $n$ 번째 비트 노드를 제외한  $m$ 번째 체크 노드에 연결된 비트 노드들의 집합이다. 갱신된 체크 메시지를 이용하여 비트 메시지  $z_{mn}$ 를 다음과 같이 갱신하며 한 번의 반복 복호 과정이 종료되며, 이러한 반복 복호 과정은 패리티 검사 통과 등 주어진 조건이 만족될 때까지 반복된다.

$$z_{mn} = l_n + \sum_{\mathbf{M}_m(n)} \epsilon_{m'n} \quad (4)$$

이 때  $\mathbf{M}_m(n)$ 은  $m$ 번째 체크 노드를 제외한  $n$ 번째 비트 노드에 연결된 체크 노드들의 집합이다.

만약  $\epsilon_{mn}$ 의 계산 과정에서 사용되는 하나 이상의  $z_{mn'}$ 이 0인 경우, 식 (3)에 의해  $n' \in \mathbf{N}_n(m)$ 인 비트들의 체크 메시지는  $\epsilon_{mn'} = 0$ 로 계산된다. 반복 과정의 시작에서  $z_{mn} = l_n$ 으로 초기화되므로,  $c_n$ 이  $k$ 번째 반복 복호 과정에서 처음으로 non-zero 비트 메시지  $z_{mn}$ 을 갖는  $k$ -SR 천공 비트 노드일 경우,  $k$ 번째 반복 과정까지  $\mathbf{N}_n(m)$ 에 속한 다른 비트노드들에 대한  $\epsilon_{mn'}$  또한 0으로 계산된다. 즉 통상적인

반복 복호 과정의 진행시  $k$ 번째 반복과정이 될 때까지  $k$ -SR 비트 노드와 같은 체크 노드를 공유하는 비트 노드들은 해당 체크 노드로부터 non-zero 메시지를 받지 못하여 이에 따른 복호 수렴의 지연이 발생한다. 또한 통상적으로 천공량  $P$ 가 증가할수록 0으로 초기화된  $z_{mn}$ 의 수가 증가함과 동시에 각 비트들의 복구 단계  $k$ 가 증가할 수 있으므로, 이러한 복호 수렴의 지연은 천공량 증대에 따라 더욱 가속화할 수 있다.

### 3.2 제안 선복구 기법

이러한 천공 LDPC 부호에서의 복호 지연 방지를 위해, 본 논문에서는 천공 비트 노드들에 대한 선복구 기법을 제안한다.  $\mathbf{M}_k^*$ 를  $k$ -SR 천공 비트 노드들을 복구시키는 생존 체크 노드들의 집합으로,  $k^*$ 를 천공 비트들의 최대 복구 단계로 정의한다. 이 때 제안 기법은 식 (2)가 수행된 이후 식 (3)부터의 반복 복호 과정이 시작되기 이전  $k = 1$ 부터  $k = k^*$ 까지 다음 단계 1) 및 2)를 순차적으로 반복한다.

1) [체크 메시지 갱신]  $k$ -SR 천공 비트 노드에 대한 생존 체크 노드  $m \in \mathbf{M}_k^*$  및 천공 비트 노드  $n \in \mathbf{N}(m)$ 에 대해  $\epsilon_{mn}$ 을 갱신한다.

$$\epsilon_{mn} = \min_{\mathbf{N}_n(m)} |z_{mn'}| \cdot \prod_{\mathbf{N}_n(m)} \text{sign}(z_{mn'}) \quad (5)$$

2) [비트 메시지 갱신]  $k$ -SR 천공 비트 노드  $n$ 에 대해 모든  $z_{mn} (\forall m \in \mathbf{M}(n))$ 을 갱신한다.

$$z_{mn} = \sum_{\mathbf{M}_m(n)} \epsilon_{m'n} \quad (6)$$

식 (5)에서는  $k$ -SR 천공 비트 노드에게 전달되는 생존 체크 노드에서의 체크 메시지를 계산하며, 식 (6)에서는 갱신된 체크 메시지를 통해 다른 체크 노드로 전달될 비트 메시지를 갱신한다. 즉 제안 선복구 기법은 천공 비트 노드들에 대한 반복 복호 과정만을 순차적으로 수행하여 본격적인 반복 복호 과정이 시작되기 이전 모든 천공 비트 노드들에 대한 non-zero 비트 메시지를 생성한다. 구체적으로,  $n$ 번째 비트 노드가 1-SR 천공 비트이며 생존 체크 노드가  $m$ 일 때, 모든  $z_{mn'} (\forall n' \in \mathbf{N}_n(m))$ 은 non-zero이다. 따라서

$k = 1$ 에 대한 식 (5) 수행 시 non-zero 체크 메시지  $\epsilon_{mn}$ 을 얻으며, 이후 식 (6)의 수행을 통해 모든  $z_{m'n} (\forall m' \in \mathbf{M}_m(n))$  또한 non-zero 메시지를 갖게 된다. 이렇게 모든 1-SR 천공 비트 노드에 대한 비트 메시지가 non-zero로 갱신되면,  $k = 2$ 에 대한 선복구 과정에서 2-SR 천공 비트 노드 또한 non-zero 체크 및 비트 메시지를 얻게 된다. 이러한 방식으로 제안 선복구 기법은 복구 단계에 따라  $k^*$ -SR 천공 비트 노드까지 모든 천공 비트들에 대한 non-zero 비트 메시지  $z_{mn}$ 을 생성한다. 이를 통해 천공 비트의 존재로 인한 복호 수렴 지연을 방지할 수 있다.

제안 선복구 기법에서는 천공 비트 노드들에 대한 메시지 갱신 작업만을 수행된다. 따라서 제안 기법은  $P$ 개의 체크 메시지에 대한 계산 과정을 추가로 요구한다. 한 번의 복호 과정 (식 (3) 및 (4))에서 계산되는 체크 메시지의 수는  $\sum_{m=1}^M |N(m)|$ 이고  $P$ 는 체크 노드의 수  $M$ 보다 작으므로, 제안 기법에 요구되는 연산량은 한 번의 복호 과정에 요구되는 연산량 대비 항상 작다. 구체적으로,  $(w_c, w_r)$ 인 정규 (regular) LDPC 부호를 가정할 때, 한 번의 반복 복호 과정에서 계산되는 체크 메시지 수는  $Mw_r = (Nw_c/w_r)w_r = Nw_c$ 이며, 제안 기법 및 한 번의 복호 과정에서 계산된 체크 메시지 수의 비는  $P/(Nw_c) < M/(Nw_c) = 1/w_r$ 이다. 즉 정규 LDPC 부호에서 제안 기법은 한 번의 복호 과정 대비 최대 약  $1/w_r$ 의 연산량만을 요구한다.

그림 1에서는 AWGN 채널을 가정하여 천공된 (3,6) 정규 LDPC 부호에서 가우시안 근사화 (Gaussian approximation) 기반 밀도 진화 (density evolution)를 통해 구해진 반복 복호 횟수에 따른 체크 노드 메시지의 평균을 나타내고 있다.<sup>[4]</sup> 이 때 채널에서 얻어진 LLR 메시지의 평균은 10으로 가정하였다. 그림 1을 통해 제안 선복구 기법의 적용이 미적용 경우 대비 동일한 반복 복호 횟수에서 보다 높은 평균 값을 달성하여 복호 수렴을 향상시키며, 향상 정도는 천공량(부호율) 증가에 따라 점진적으로 증대함을 확인할 수 있다. 그림 1에서 예측된 제안 기법의 복호 수렴의 향상 정도는 최대 한 번의 복호 과정 가량으로 나타나며, 제안 기법이 한 번의 복호 과정 대비 크게 낮은 연산량을 요구함을 고려할 때 제안 기법은 추가되는 복잡도 대비 효율적으로 천공 부호에 대한 복호

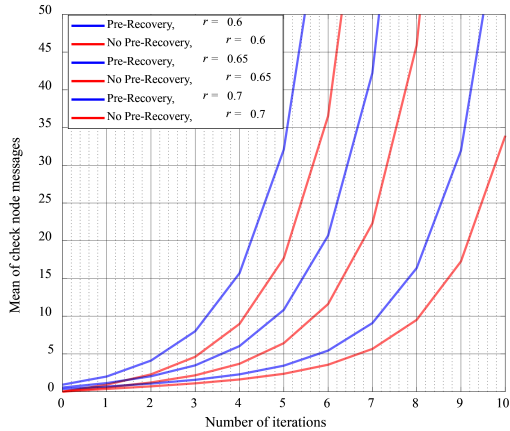


그림 1. 반복 복호 횟수에 따른 체크노드 메시지의 평균  
Fig. 1. Mean of check node messages according to the number of iterations

수렴을 가속화함을 확인할 수 있다. 또한 그림 1에서는 천공률에 관계없이 최대 복구 단계  $k^*$ 가 항상 1로 설정되었으므로, 천공률에 따라  $k^*$ 가 지속적으로 증가할 수 있는 실제 천공 부호어에 대한 복호 상황을 고려할 때 제안 기법 사용에 의한 성능 향상 정도는 더욱 증가할 수 있음을 예측할 수 있다.

#### IV. 모의실험 결과

모의실험을 위해  $K = 1152$  및  $N = 2304$ 인 Block LDPC 부호를 모부호로 고려하였다. 변조 방식으로 QPSK (Quadrature Amplitude Shift Keying) 변조를 고려하였으며 채널 모델로 독립적 레일리 감쇄 채널을 고려하였다. 반복 복호 과정에서의 최대 반복 복호 횟수  $I^*$ 는 10 및 20이며, 천공 LDPC 부호의 부호율  $r$ 은  $8/9$  ( $P = 1008$ ) 및  $4/5$  ( $P = 864$ )를 고려하였다.

그림 2는 평균 BER (Bit Error Rate) 성능을 비교하고 있다. 해당 결과로부터 제안 선복구 기법이 적용된 경우 통상적인 반복 복호 기법 대비 성능 이득을 달성할 수 있음을 확인할 수 있다. 평균 BER  $10^{-5}$ 을 기준으로 할 때,  $r = 8/9$ 인 경우  $I^* = 10$ 일 때 약 0.9 dB,  $I^* = 20$ 일 때 약 0.2 dB의 SNR (Signal-to-Noise Ratio) 이득을 달성하며,  $r = 4/5$ 인 경우  $I^* = 10$ 일 때 약 0.3 dB,  $I^* = 20$ 일 때 약 0.05 dB의 SNR 이득을 달성한다. 즉, 최대 반복 복호 횟수가 작거나 천공률이 증가할수록 통상적인 반복

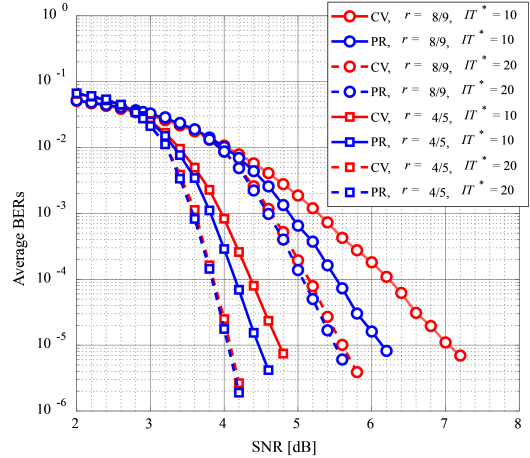


그림 2. 부호율에 따른 평균 BER 비교  
Fig. 2. Average BERs according to the code rate

복호 기법에서의 메시지 갱신 지연이 증가하여 그를 방지하는 제안 선복구 기법에 의한 성능 이득이 증가함을 관찰할 수 있다.

그림 3은 패리티 검사를 통과할 때까지 요구되는 평균 반복 복호 횟수를 비교하고 있다. 그림 3의 결과를 통해 제안 선복구 기법이 통상 기법에서의 복호 수렴을 가속화함을 확인할 수 있다. 구체적으로, 제안 기법 적용 시 평균 반복 복호 횟수는 통상 반복 복호 기법 대비  $r = 8/9$ 인 경우  $I^* = 10$ 일 때 최대 약 39%,  $I^* = 20$ 일 때 최대 약 29% 감소하며,  $r = 4/5$ 인 경우  $I^* = 10$ 일 때 최대 약 22%,  $I^* = 20$ 일 때 최대 약 19% 감소한다. 그림 2의 평균

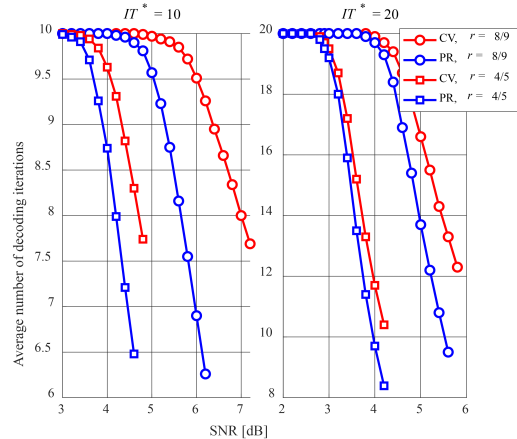


그림 3. 부호율에 따른 평균 반복 복호 횟수  
Fig. 3. Average decoding iterations according to the code rate

BER 결과와 마찬가지로, 이러한 복호 수렴 속도 향상 정도는 최대 반복 복호 횟수가 작거나 천공량이 증가할수록 보다 증가함을 관찰할 수 있다. 또한  $r = 4/5$  이며  $L^* = 20$  일 때, 즉 천공량이 적고 최대 반복 복호 횟수가 큰 경우 평균 BER의 향상 정도는 미미하였으나 복호 수렴 속도 관점에서는 상당한 성능 이득을 얻을 수 있음을 확인할 수 있다.

## V. 결 론

본 논문에서는 천공된 LDPC 부호어의 복호 수렴 속도를 향상시키기 위한 선복구 기법을 제안하였다. 모의실험 결과를 통해 제안 기법을 사용할 경우 복호 수렴 속도 및 오류 정정 성능이 향상됨을 확인하였으며, 특히 천공량이 증가하거나 최대 반복 복호 횟수가 작을수록 복호 수렴 속도 및 오류 정정 성능의 향상 정도가 증가함을 확인하였다.

본 논문에서는 서술의 편의상 min-sum 복호에 기반을 둔 선복구 기법을 기술하였으나, BP 복호 등 다른 그래프 기반 LDPC 복호 방식을 통해서도 유사하게 구현될 수 있다. 또한 min-sum 반복 복호 과정에 앞선 BP 기반 선복구 기법의 적용 등 선복구 기법 및 반복 복호 과정을 서로 다른 방식으로 설계할 수 있으며, 복구 단계 등에 따라 천공 비트별로도 서로 다른 방식의 선복구 방식을 적용할 수 있다. 이러한 반복 복호 방식 및 천공 비트들의 복구 단계 등에 따른 선복구 기법의 최적 운영 방식은 향후 과제로 남는다.

## References

[1] M. Jang, et al., "Decoding method of LDPC codes in IEEE 802.16e standards for improving the convergence speed," *J. KICS*, vol. 31, no. 12C, pp. 1143-1149, Dec. 2006.

[2] M. Li, et al., "Area and energy efficient 802.11ad LDPC decoding processor," *Electron. Lett.*, vol. 51, no. 4, pp. 339-341, Feb. 2015.

[3] H. Wu and H. Wang, "A high throughput implementation of QC-LDPC codes for 5G NR," *IEEE Access*, vol. 7, pp. 185373-185384, Dec. 2019.

[4] J. Ha, et al., "Rate-compatible puncturing of low-density parity-check codes," *IEEE Trans.*

*Inf. Theory*, vol. 50, no. 11, pp. 2824-2836, Nov. 2004.

[5] J. Ha, et al., "Rate-compatible punctured low-density parity-check codes with short block lengths," *IEEE Trans. Inf. Theory*, vol. 52, no. 2, pp. 728-738, Feb. 2006.

[6] H. Park, et al., "Efficient puncturing method for rate-compatible low-density parity-check codes," *IEEE Trans. Wirel. Commun.*, vol. 6, no. 11, pp. 3914-3919, Nov. 2007.

[7] J. Ha, et al., "Layered BP decoding for rate-compatible punctured LDPC codes," *IEEE Commun. Lett.*, vol. 11, no. 5, pp. 440-442, May 2007.

[8] J. Kwon, et al., "Fast decoding of rate-compatible punctured LDPC codes," in *Proc. ISIT*, pp. 216-220, Nice, France, Jun. 2007.

[9] S. Park, et al., "A group shuffled BP decoding for punctured low-density parity-check codes," *IEICE Electron. Express*, vol. 7, no. 19, pp. 1429-1434, Oct. 2010.