한국정보전자통신기술학회논문지(jkiiect)'19-2, Vol.12 No.1
논문 19-12-01-37 http://dx.doi.org/10.17661/jkiiect.2019.12.1.37

소프트 값을 이용한 해밍 부호의 개선된 복호 방식

정 호 영*

An Improved Decoding Scheme of Hamming Codes using Soft Values

Ho-Young Cheong*

요 약 본 논문에서는 부호 길이 내에서 1개의 오류를 정정할 수 있는 해밍 부호(Hamming code)에 대해 2 개의 오류를 정정할 수 있는 신드롬 복호 기법을 제안하였다. 본 논문에서 제안한 복호 기법은 복호 복잡도를 거의 증가시키지 않으면서도 다중 오류를 정정함으로서 복호 복잡도 대비 오율 성능을 크게 개선할 수 있는 장점을 갖는다. 본 논문에서 제안한 복호 기법은 IoT 기기들 간의 통신이나 분자 통신과 같이 부/복호기에서 에너지의 사용이 극히 제한된 환경에서 부·복호기의 에너지 사용이 적으면서도 우수한 오율 성능이 요구되는 경우에 적합하다. 본 논문에서 제안한 복호 기법이 적용된 해밍 부호의 오율 성능이 개선되었음을 보이기 위해 BPSK 변조 방식과 AWGN 채널 환경에서 부호 길이가 짧은 다수의 해밍 부호에 대해 시뮬레이션을 수행하였으며, 수행 결과 해밍 부호의 부호 길이에 무관하게 제안한 복호 방식은 기존 복호 방식에 비해 약 1.1[dB] ~ 1.2[dB] 정도의 성능 개선을 확인할 수 있었다.

Abstract In this paper, we propose a syndrome decoding scheme that can correct two errors for single error correcting Hamming codes within a code length. The decoding scheme proposed in this paper has the advantage of significantly improving the error rate performance compared to the decoder complexity by correcting multiple errors without substantially increasing the decoding complexity. It is suitable for applications in which the energy use of encoder/decoder is extremely limited and the low error rate performance is required, such as IoT communications and molecular communications. In order to verify the improvement of the error rate performance of the Hamming code with the proposed decoding scheme, we performed simulation on Hamming codes with short code length in the AWGN and BPSK modulation environments. As a result, compared with the conventional decoding method, the proposed decoding scheme showed performance improvement of about 1.1 ~ 1.2[dB] regardless of the code length of the Hamming code.

Key Words : Error rate performance, Decoding complexity, Syndrome decoding, Energy-limited environment, Multiple error correction

1. 서론

IoT 기기 간에 이루어지는 통신은 단말에서 사용할 수 있는 에너지가 한정되어 있으므로 정보 데이터를 전송하는데 있어서 가능한 한 적은 에너지를 소비해야 하는 한편 전송 신뢰도 또한 담보되어야 한다. 대부분의 IoT 기기들은 길이가 짧은 정보 데이터를 끊임없이 생성하는 경우가 많으므로 짧은 길이의 채널 부호가 필요하다. 해밍 부호(Hamming code)는 부/복호기 구조가 간단하여 적은 에너지를 소비하며 짧은 길이의 부호가 많아 IoT 환경에 적합한 부호라고 할 수 있다[1][2].

해밍 부호의 복호기는 조합논리회로 구현할 수 있어서 복호 지연이 거의 없고 에너지 소비가 적다. 따라서

Funding for this paper was provided by Namseoul University in 2018
*Department of Information and Communication Engineering, Namseoul University
Received January 04, 2019 Revised January 15, 2019 Accepted January 17, 2019

에너지 자원이 제한되어 있는 IoT 기기 간의 통신에 적합하다[3].

그러나 해밍 부호는 터보 부호(turbo code)나 LDPC(low density parity check) 부호와 비교할 때 오율 성능이 크게 떨어지는 단점이 있다. 모든 해밍 부호는 최소 해밍 거리가 3이므로 1개의 오류를 정정할 수 있고 오류 검출 능력이 2개 이하인 부호이기 때문이다[4][5].

본 논문에서는 2 개 이상의 오류가 발생하면 오류를 정정할 수 없었던 해밍 부호의 한계를 극복하여 2 개의 오류에 대해서도 복조기의 출력 값을 이용하여 정정이 가능한 복호 알고리즘을 제안한다.

본 논문에서 제안하는 복호 기법은 복조기의 출력 값을 버리지 않고 저장한 후, 복호 과정 중에 계산된 신드롬에 해당하는 오류 패턴을 변별할 때만 사용하므로 복호기 복잡도 증가는 거의 없다.

본 논문은 다음과 같이 구성되어 있다. 2 장에서 해밍 부호에 대한 부/복호기의 동작 원리를 기술하였다. 3 장에서는 해밍 부호에 대한 기존의 복호 알고리즘의 문제점을 설명하고 이를 개선한 복호 알고리즘을 제안하였다. 4장에서는 BPSK 변조 방식과 AWGN 채널을 가정하여 해밍 부호의 비트 오율 성능을 시뮬레이션을 통해 분석하였으며, 5장에서 결론을 맺었다.

2. 해밍 부호의 신드롬 복호

(n,k) 선형 블록부호(linear block code)의 오류 정정 능력 t는 선형 블록 부호의 최소 해밍 거리 $d_{\min}$에 의해 식 (1)과 같이 정해진다[4].

$$t \leqq \left\lfloor \frac{(d_{\min}-1)}{2} \right\rfloor \tag{1}$$

이때 $\lfloor x \rfloor$는 가우스 기호를 의미하며 x보다 크지 않은 최대 정수를 말한다. 따라서 $d_{\min}=3$인 해밍 부호의 오류 정정 능력은 $t \leqq 1$이므로 부호 길이 내에 1 개의 오류 까지만 확실하게 정정할 수 있고 2 개 이상의 오류는 정정을 보장하지 않는다.

(n,k) 해밍 부호는 식 (2)와 같은 생성 행렬 $\boldsymbol{G}$를 이용하여 메시지 벡터 $\boldsymbol{m}=(m_1,m_2,...,m_k)$으로부터 $\boldsymbol{c}=(c_1,c_2,...,c_n)$를 얻는다. 부호 벡터 $\boldsymbol{c}$는 행렬 연산 $\boldsymbol{c}=\boldsymbol{m}\boldsymbol{G}$를 통해 얻을 수 있는데, $\boldsymbol{c}$의 처음 k 비트는 메시지 비트가 위치하게 되며 나머지 $(n-k)$ 비트는 $\boldsymbol{c}$의 뒷부분에 위치하여 부호어 $\boldsymbol{c}$는 체계적 형태(systematic form)를 갖는다. 식 (2)는 (15,11) 해밍 부호에 대한 생성 행렬 형태를 나타낸 것이다.

$$\boldsymbol{G}=\begin{bmatrix}1\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,1\,1\,1\,1\\0\,1\,0\,0\,0\,0\,0\,0\,0\,0\,0\,1\,1\,1\\0\,0\,1\,0\,0\,0\,0\,0\,0\,0\,0\,1\,0\,1\,1\\0\,0\,0\,1\,0\,0\,0\,0\,0\,0\,0\,1\,1\,0\,1\\0\,0\,0\,0\,1\,0\,0\,0\,0\,0\,0\,1\,1\,1\,0\\0\,0\,0\,0\,0\,1\,0\,0\,0\,0\,0\,0\,1\,1\\0\,0\,0\,0\,0\,0\,1\,0\,0\,0\,0\,0\,1\,0\,1\\0\,0\,0\,0\,0\,0\,0\,1\,0\,0\,0\,0\,1\,1\,0\\0\,0\,0\,0\,0\,0\,0\,0\,1\,0\,0\,0\,0\,1\,1\end{bmatrix} \tag{2}$$

부호 벡터 $\boldsymbol{c}$가 변조 신호 형태로 채널에 전송되면 수신기에서 복조 과정을 통해 수신 부호 벡터 $\boldsymbol{r}=(r_1,r_2,...,r_n)$이 복호기로 입력된다. 이때 수신 부호 벡터 $\boldsymbol{r}$은 아날로그 값을 갖는 복조기 출력 값을 경판정(hard decision)하여 얻은 이진 값으로 구성되며, 경 판정 이전의 값인 복조기 출력 값은 n 개의 아날로그 값을 갖게 되는데 이를 소프트 벡터 $\boldsymbol{q}=(q_1,q_2,...,q_n)$라고 표시하기로 한다. n 개의 소프트 값은 아날로그 값이며 경 판정을 거쳐 $\{0,1\}$으로 변환되어 수신 부호 벡터 $\boldsymbol{r}$을 형성한다. 경 판정 과정에서 대부분의 채널 정보가 소실되며 이로 인해 오류가 발생하게 되는데, 채널에서 발생한 오류들은 오류 벡터 $\boldsymbol{e}=(e_1,e_2,...,e_n)$으로 표현하기로 한다. 이 때 e_i의 값이 1이면 오류가 발생한 것을 의미하고 0이면 오류가 없는 것을 의미한다. 수신 부호 벡터 $\boldsymbol{r}$은 부호 벡터 $\boldsymbol{c}$와 오류 벡터 $\boldsymbol{e}$를 이용해 식 (3)과 같이 나타낼 수 있다.

$$\boldsymbol{r}=\boldsymbol{c}+\boldsymbol{e} \tag{3}$$

(n,k) 해밍 부호는 패리티 검사 행렬 $\boldsymbol{H}$를 이용해 복호 과정을 시작 하는데 생성 행렬 $\boldsymbol{G}$와 $\boldsymbol{H}$가 항상 식 (4)가 성립하는 관계를 갖도록 $\boldsymbol{H}$를 구성한다. 식 (5)는 식 (4)의 관계가 성립하는 (15,11) 해밍 부호의 패리티 검사 행렬을 나타낸 것이다[3].

$$GH^T = 0 \tag{4}$$

$$H = \begin{bmatrix} 1\,0\,1\,1\,1\,0\,0\,0\,1\,1\,1\,1\,0\,0\,0 \\ 1\,1\,0\,1\,1\,0\,1\,1\,0\,0\,1\,0\,1\,0\,0 \\ 1\,1\,1\,0\,1\,1\,0\,1\,1\,0\,0\,0\,0\,1\,0 \\ 1\,1\,1\,1\,0\,1\,1\,0\,0\,1\,0\,0\,0\,0\,1 \end{bmatrix} \tag{5}$$

수신 부호 벡터 r이 복호기에 입력되면 복호기는 식 (6)의 연산을 통해 $(n-k)$-비트로 구성되는 신드롬 벡터 $s = (s_1, s_2, ..., s_{n-k})$를 얻는다.

$$s = rH^T \tag{6}$$

신드롬 벡터 s는 $(n-k)$-비트로 구성되므로 신드롬 벡터의 총 수는 $2^{(n-k)}$ 이다. 예를 들어, (15,11) 부호의 경우 총 $2^{n-k} = 2^4 = 16$개의 신드롬 벡터들을 갖는다. $r = c + e$이므로 식 (6)로부터 식(7)을 얻을 수 있다.

$$\begin{aligned} s &= rH^T \\ &= (c+e)H^T \\ &= cH^T + eH^T \\ &= mGH^T + eH^T \\ &= eH^T \end{aligned} \tag{7}$$

식 (7)에서 오류가 발생하지 않으면 $e = 0$이므로 $s = 0$인 벡터가 될 것이다. 따라서 $s \neq 0$이면 오류가 발생한 경우를 의미하며 신드롬 s와 오류 벡터 e는 밀접한 관계에 있음을 알 수 있다. 여기에서 해밍 부호의 오류 정정 능력은 1개 이하이므로 정정 가능한 오류 벡터는 0 벡터를 포함하여 해밍 가중치가 1인 오류 벡터이고 이러한 오류 벡터의 수는 16개이다. 따라서 (15,11) 해밍 부호는 0 벡터를 포함한 단일 오류 벡터의 수와 신드롬 벡터의 수가 정확히 일치하는 완전 부호이며 이들 간에는 일 대 일 대응 관계를 갖게 된다. 표 1은 단일 오류 패턴과 신드롬에 대한 일대 일 대응 관계를 보여주는 (15,11) 해밍 부호의 복호 표를 나타낸 것이다[4].

식 (7)로부터 신드롬 s가 0 벡터가 아닐 경우 표 1의 복호 표를 참조하여 단일 오류 패턴을 모두 정정할 수 있다. 해밍 가중치가 1인 오류 패턴이 발생한 경우 대응되는 신드롬 벡터가 유일하므로 단일 오류는 모두 정정할 수 있다. 그러나 해밍 부호의 오류 검출 능력 d는 $d \leq d_{\min} - 1$이므로 2 개의 오류가 발생한 경우에도 0이 아닌 신드롬을 얻을 수 있다. 하지만 2-오류 패턴이 발생한 경우를 고려할 경우 하나의 신드롬에 대해 중복되는 오류 패턴이 많아 신드롬에 대응되는 복호 표의 오류 패턴으로 정정할 경우 오류가 추가되는 현상을 초래할 수도 있다.

표 1. (15,11) 해밍 부호의 단일 오류 패턴과 신드롬 벡터
Table 1. 1-Error pattern and its corresponding syndrome vector of (15,11) Hamming code

오류 패턴 e	신드롬 s
[000000000000000]	[0000]
[100000000000000]	[1111]
[010000000000000]	[0111]
[001000000000000]	[1011]
[000100000000000]	[1101]
[000010000000000]	[1110]
[000001000000000]	[0011]
[000000100000000]	[0101]
[000000010000000]	[0110]
[000000001000000]	[1010]
[000000000100000]	[1001]
[000000000010000]	[1100]
[000000000001000]	[1000]
[000000000000100]	[0100]
[000000000000010]	[0010]
[000000000000001]	[0001]

예를 들어, 표 2에서 신드롬 $s = [1101]$에 해당하는 2개 이하의 오류 패턴들을 총 8 개 볼 수 있다. 따라서 하나의 부호에 2개 이하의 오류가 발생할 가능성이 많은 채널에서 수신 단의 신드롬 계산 결과가 $s = [1101]$일 경우 단순히 발생 오류 패턴을 $e = [000100000000000]$의 단일 오류 패턴으로 단정하여 오류 정정을 하는 것은 타당하지 않으며 표 2의 오류 패턴 들 중 하나라고 판단해야 할 것이다. 그러나 표 2의 오류 패턴 들 중 하나를 변별할 기준이 없어 2-오류 패턴에 대한 오류 정정을 포기하고 단일 오류 패턴으로 가정하여 오류 정정을 수행하기 때문에 오율 성능이 저하될 수밖에 없다.

본 논문에서는 표 2와 같이 하나의 신드롬 벡터에 해밍 가중치가 2개 이하인 다수의 오류 패턴이 존재하는 경우 이들 중 실제 오류 패턴을 변별할 기준을 제시하여 복호함으로서 2 개의 오류까지 정정

할 수 있는 해밍 부호의 복호 방식을 제안한다.

표 2. (15,11) 해밍 부호의 m-오류 패턴과 신드롬 벡터 ($m \leq 2$)

Table 2. m-bits error pattern and its syndrome vector of (15,11) Hamming code ($m \leq 2$)

오류 패턴 $\boldsymbol{e}$	신드롬 $\boldsymbol{s}$
[000100000000000]	[1101]
[000011000000000]	[1101]
[001000010000000]	[1101]
[010000001000000]	[1101]
[000000100001000]	[1101]
[000000000100100]	[1101]
[100000000000010]	[1101]
[000000000010001]	[1101]

3. 소프트 값을 이용한 해밍 부호의 신드롬 복호

해밍 복호기에 입력되는 수신 부호 벡터 $\boldsymbol{r}$의 비트 값들은 실수 값을 갖는 복조기 출력 신호 값에 대하여 경 판정을 이용해 이진 값으로 변환된 값들이다. 복조기 출력 값에 내재되어 있던 채널 정보들은 경 판정을 하는 순간 대부분 사라지게 되며 이로 인해 해밍 부호의 오율 성능이 저하된다.

이를 개선하기 위해 터보 부호나 LDPC 부호에서는 연 판정 값을 이용한 반복 복호 기법을 적용하여 오율 성능을 개선하고 있다[6]. 하지만 해밍 부호에서는 기본적으로 복호 과정이 이진 값(경 판정 값)을 이용하여 이루어지므로 연 판정 값을 복호 과정에 적용할 수 없다. 또한 터보 부호나 LDPC 부호의 복호 과정과 같이 반복 복호는 많은 연산량이 요구되기 때문에 에너지가 한정되어 있고 연산 능력이 작은 IoT 통신 등에는 적합하지 않다[7][8].

본 절에서는 아날로그 값의 복조기 출력(이후 소프트 값으로 표시한다)에 대한 경 판정 이진 값과 소프트 값을 모두 복호기에 함께 입력시킨 후 소프트 값을 복호과정에 이용함으로서 오율 성능을 개선할 수 있는 새로운 복호 기법을 제안한다. 편의 상 n 개의 소프트 값으로 이루어진 벡터를 $\boldsymbol{q} = (q_1, q_2, \cdots, q_n)$으로 표시한다.

그림 1은 $E_b/N_0 = 2 \sim 8[dB]$의 채널 환경에서 BPSK 복조기 출력 중 오류가 발생한 비트와 발생하지 않은 비트에 대한 소프트 값들의 평균 전력 값을 E_b/N_0에 대해 나타낸 것이다.

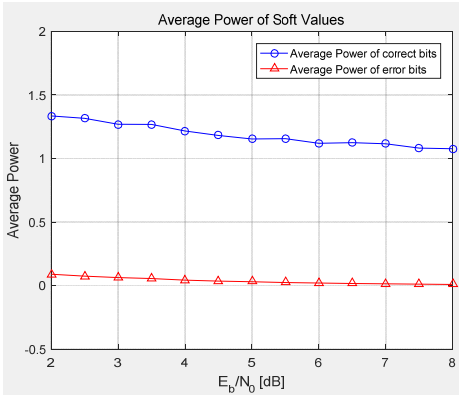

그림 1. 오류 비트와 오류가 발생하지 않은 비트에 해당하는 소프트 값의 평균 전력 대 E_b/N_0

Fig. 1. Average powers of soft values of erroneous bit and correct bit vs. E_b/N_0

해밍 부호를 사용하지 않은 상태에서 각 각의 E_b/N_0 환경에 대해 오류 비트에 해당하는 소프트 값 5,000 개와 올바른 비트에 해당하는 소프트 값 5,000 개씩을 얻은 후 이들의 평균 전력 값을 구해 나타낸 것이다. 그림 1에서 오류 비트에 대한 소프트 값의 평균 전력은 0에 가까운 작은 값을 보이며 E_b/N_0 값이 증가함에 따라 0에 수렴함을 알 수 있다. 또한, 오류가 발생하지 않은 비트의 경우 이에 해당하는 소프트 값의 평균 전력은 모두 1보다 큰 값을 가지며 E_b/N_0 값이 증가함에 따라 1에 수렴함을 알 수 있다. 이와 같은 현상은 경 판정 과정에서 복조기 출력 값이 임계 치를 넘어 반대 영역에 위치할 때 오류로 판정되기 때문에 소프트 값의 전력이 작을 수밖에 없다. 따라서 소프트 값은 오류 비트와 올바른 비트를 판별할 수 있는 정보를 담고 있다. 본 논문에서 제안하는 해밍 복호는 오류 비트의 소프트 값과 올바른 비트의 소프트 값 간의 전력 차이를 이용하여 하나의 신드롬에 해당하는 다수의 오류 패턴들 중 정확한 오류 패턴을 선별하는데 사용하여 오류 정정 능력을 높이고자 한다.

예를 들어, 표 2에서 신드롬 $\boldsymbol{s} = [1101]$에 해당하는

오류 패턴 중 2 개 이하의 오류를 포함하는 오류 패턴들은 8 가지 패턴들이 있는데 소프트 값의 전력을 활용하여 8가지 오류 패턴들 중 가장 발생 가능성이 큰 오류 패턴을 변별할 수 있다. 오류 패턴을 $\boldsymbol{e}=(e_1,e_2,\cdots,e_{15})$라고 표시하자. 오류 패턴 $\boldsymbol{e_1}=(000011000000000)$의 경우 5, 6번 째 오류 비트가 1이므로 q_5와 q_6의 전력 값이 다른 13 개의 소프트 비트 전력 값에 비해 월등히 작을 것이다. 만일 $\boldsymbol{e_1}$과 $\boldsymbol{e_2}=(100000000000010)$ 중 하나를 선택하여 오류 패턴 $\hat{\boldsymbol{e}}$를 추정하는 경우 식 (8)의 기준을 사용하면 정확한 오류 패턴을 변별할 가능성이 높다.

$$\hat{\boldsymbol{e}} = \begin{cases} \boldsymbol{e_1}, & q_5^2+q_6^2 < q_1^2+q_{14}^2 \\ \boldsymbol{e_2}, & otherwise \end{cases} \tag{8}$$

$\boldsymbol{s}$에 해당하는 오류 벡터가 다수 존재해도 소프트 값의 전력이 가장 작은 오류 패턴을 선정하여 채널에서 발생한 오류 패턴으로 결정한다.

식 (8)의 계산은 각 오류 패턴에 대해 2개 이하의 소프트 값에 대한 전력 값을 계산하므로 이로 인해 증가되는 복호 복잡 도는 크지 않다. 해밍 부호의 부호 길이가 짧으므로 저장해야 하는 소프트 값에 대한 메모리 용량도 작아 복호 복잡도의 증가는 거의 없다.

4. 시뮬레이션 및 결과 분석

본 장에서는 제안된 복호 알고리즘의 오율 성능을 확인하기 위해 BPSK 변조와 AWGN 채널을 가정하여 시뮬레이션을 수행하였다. 채널 부호는 길이가 비교적 짧은 해밍 부호를 사용하였으며 해밍 부호의 기존 복호 기법과 본 논문에서 제안된 복호 기법의 오율 성능을 비교·분석하였다.

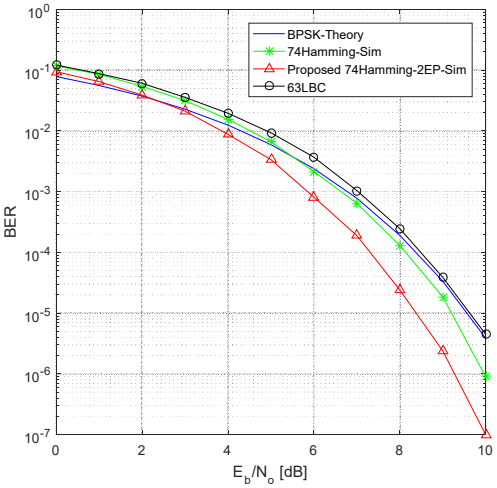

그림 2. 기존의 (7,4) Hamming 부호와 제안된 (7,4) 해밍부호의 오율 성능
Fig. 2. BER of the conventional (7,4) Hamming code and the proposed (7,4) Hamming code

그림 2는 기존의 복호 방식을 사용한 (7,4) 해밍 부호와 본 논문에서 제안한 복호 방식을 사용한 (7,4) 해밍 부호의 BER 성능을 나타낸 것이다. 또한 (7,4) 해밍 부호보다도 부호율이 낮은 (6,3) 선형 부호(63LBC)와 채널 부호를 사용하지 않은 경우 (BPSK-Theory)에 대해서도 비트 오율을 비교하여 나타내었다. 그림 2에서 제안된 해밍 부호는 기존 복호 방식과 비교할 때 10^{-4}의 오율에서 약 1~1.2[dB]의 부호 이득을 얻을 수 있음을 볼 수 있다. 또한 부호율이 낮은 (6,3) 선형 블록 부호에 대해서도 약 1.8[dB]의 부호 이득을 가짐을 알 수 있다.

그림 3은 (15,11) 해밍 부호에 대하여 기존의 복호 기법과 본 논문에서 제안된 복호 기법에 의한 BER 성능을 비교한 것이다. (15,11) 해밍 부호의 경우에도 기존의 복호 기법에 비해 본 논문에서 제안한 복호 기법의 오율 성능이 10^{-5}의 오율에서 약 1.1[dB]의 이득을 가짐을 알 수 있다.

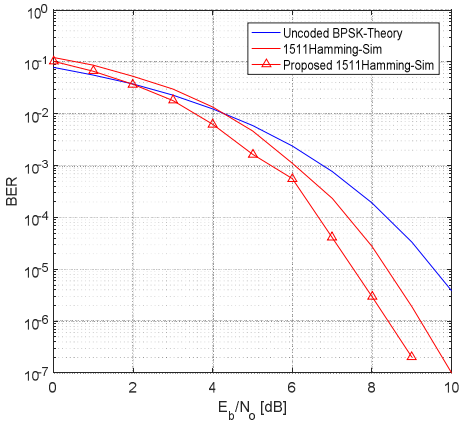

그림 3. 기존의 (15,11) Hamming 부호와 제안된 (15, 11) 해밍부호의 오율 성능

Fig. 3. BER of the conventional (15,11) Hamming code and the proposed (15,11) Hamming code

5. 결론

본 논문에서는 소프트 값을 이용하여 해밍 부호의 비트 오율 성능을 개선할 수 있는 신드롬 복호 방식을 제안하고 시뮬레이션을 통해 성능을 입증하였다. 특히 복호기의 복잡도 증가 없이 비트 오율 성능 개선할 수 있으므로 에너지가 한정되어 있고 부·복호기의 알고리즘 연산을 수행할 수 있는 신호 처리 능력이 작은 IoT 통신 환경에서 효율적인 오류 정정 부호로 활용될 수 있다.

오율 성능 개선을 확인하기 위해 BPSK 변조 방식과 AWGN 채널을 가정하여 부호 길이가 짧은 해밍 부호에 대해 시뮬레이션을 수행하였다. 수행 결과 해밍 부호 길이에 무관하게 기존의 복호 방식에 비해 제안한 복호 방식은 약 1.1~1.2 [dB]의 오율 성능 개선을 확인할 수 있었다.

REFERENCES

[1] E. Tsimbalo, X. Fafoutis, and R. J. Piechocki, ‘‘CRC error correction in IoT applications,’’ IEEE Trans. Ind. Informat., vol. 13, no. 1, pp. 361-369, Feb. 2017

[2] S. A. Alabady, and Fadi Al-Turjman, “Low Complexity Parity Check Code for Futuristic Wireless Networks Applications,” IEEE Access, Vol. 6, 2018, pp. 18398-18407.

[3] S. L. Howard, C. Schlegel, and K. Iniewski, “Error Control Coding in Low-Power Wireless Sensor Networks: When is ECC Energy -Efficient?, ”EURASIP Journal on Wireless Communication and Networking, Volume 2006, pp. 1-14.

[4] S. Lin and D. J. Costello, Jr., *Error Control Coding: Fundamentals and Applications*, 2nd ed., Prentice Hall: Englewood Cliffs, 2004.

[5] Frederic Lehmann and Gian Mario Maggio, “Analysis of the Iterative Decoding of LDPC and Product Codes Using the Gaussian Approximation,” IEEE Trans. on Information Theory, Vol. 49, No. 11, Nov., 2003.

[6] Z. Su, Q. Qiu, and H. Zhou, ‘‘Analysis and elimination of short cycles in LDPC convolutional codes,’’ in Proc. 2nd IEEE Int. Conf. Comput. Commun. (ICCC), Chengdu, China, Oct. 2016, pp. 1128-1132.

[7] G. Liva, E. Paolini, B. Matuz, S. Scalise, and M. Chiani, ‘‘Short turbo codes over high order fields,’’ IEEE Trans. Commun., vol. 61, no. 6, pp. 2201-2211, Jun. 2013.

[8] C. Y. Chen, Q. Huang, C. C. Chao, and S. Lin, ‘‘Two low-complexity reliability-based message-passing algorithms for decoding non-binary LDPC codes,’’ IEEE Trans. Comm., Vol. 58(11), pp.3140-3147, 2010.

저자약력

정 호 영(Ho-Young Cheong) **[종신회원]**

- 1987년 8월 : 연세대학교 대학원 전자공학과 (공학석사)
- 1995년 2월 : 연세대학교 대학원 전자공학과 (공학박사)
- 1995년 4월 ~ 현재 : 남서울대학교 정보통신공학과 교수

〈관심분야〉 채널부호, 분자통신