

論文93-30A-8-3

Reed-Solomon/길쌈 연쇄부호의 비트오율해석

(An Analysis of Bit Error Probability of Reed-Solomon/Convolutional Concatenated Codes)

李相坤*, 文相在**

(Sang Gon Lee and Sang Jae Moon)

要約

Reed-Solomon 부호와 연쇄된 길쌈부호의 더 정확한 심볼오율 상한을 사용함으로써 Reed-Solomon/길쌈 연쇄부호의 비트오율을 더 정확하게 계산할 수 있다. 이를 위해 본 논문에서는 길쌈부호의 부등심볼 오율 상한을 구하여 연쇄부호의 비트오율 계산하였다. 본 결과는 기존의 값보다 정확하였다.

Abstract

The bit error probability of Reed-Solomon/convolutional concatenated codes can be more exactly calculated by using a more approximate bound of the symbol error probability of the convolutional codes. This paper obtains the unequal symbol error bound of the convolutional codes, and applies to the calculation of the bit error probability of the concatenated codes. Our results are tighter than the earlier studied other bounds.

1. 서론

동일 부호율에서 오류제어부호의 성능은 블럭부호에서는 블럭길이를, 길쌈부호에서는 구속장을 증가시킬수록 향상된다. 하지만 복호기의 복잡도는 블럭의 길이와 구속장이 길어질수록 지수적으로 증가한다. ^[1,2] 비교적 복잡도가 낮은 두 부호를 연이어 사용하는 연쇄부호(concatenated code)는 복잡도가 낮으

며 오류제어성능이 우수하다. 특히 Viterbi복호법을 사용한 짧은 구속장의 길쌈부호와 Reed-Solomon (RS)부호로 구성된 RS/길쌈 연쇄부호는 2.0~2.5 [dB]의 낮은 비트에너지대 잡음에너지의 비에서 10^{-5} 의 비트오율을 얻을 수 있으므로 ^[3,4] 위성통신등에 널리 이용되고 있다. 그럼 1은 RS/길쌈 연쇄부호의 블럭선도이다. Viterbi 복호과정에서 생기는 오류는 주로 연접오류이다. 하나의 RS부호어 내에 오류정정능력 이상의 오류심볼이 발생하면 오류정정이 불가능하므로 인터리버를 사용하여 심볼단위로 오류를 산발화 시킨다. GF(2^b)상의 RS/길쌈 연쇄부호의 비트오율은

$$P_b = \frac{2^{b-1}}{2^b - 1} \sum_{i=t+1}^N \frac{i+t}{N} \binom{N}{i} P_{sb}^i (1 - P_{sb})^{N-i} \quad (1)$$

이다. ^[4] 여기서 N은 RS부호어의 길이, t는 RS부호의 오류정정심볼 수를 나타낸다. 그리고 P_{sb} 는 RS부

* 正會員, 昌信專門大學 電子通信科
(Dept. of Electronic Communications,
Changshin Junior Collage)

**正會員, 慶北大學校 電子工學科
(Dept. of Ele., Kyungpook National Univ.)
(* 이 연구는 '92년도 한국과학재단 연구비 지원에 의한 결과임.)

接受日字: 1993年 2月 9日

호 복호기의 입력단에서 본 b비트 RS부호의 심볼오율이며 동시에 Viterbi복호기 출력에서의 b비트 심볼오율이다.

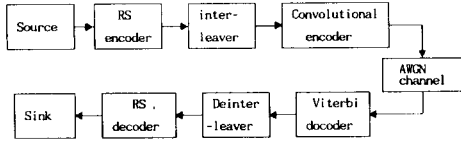


그림 1. RS/길쌈 연쇄부호의 블럭선도
Fig. 1. The block diagram of RS/convolutional concatenated code.

식 (1)에 나타난 바와 같이 길쌈부호의 심볼오율은 RS/길쌈 연쇄부호 성능의 척도가 되므로 정확해야 한다. 길쌈부호의 심볼오율은 시뮬레이션을 통하여 구할 수도 있으나 시간이 오래 걸리고 번거로우므로 해석적인 오율식이 요구된다. McEliece 등^[5]이 처음으로 부호율 $1/n$ 길쌈부호의 심볼오율 상한을 구하였으며, Perez 등^[6]은 McEliece 등의 수식을 확장 해석하여 k/n 길쌈부호의 심볼오율 상한을 구하였다. 그러나 이들은 Viterbi복호의 오류경로위에 있는 영(zero) 심볼도 오류심볼로 산입하여 상한을 구하여야 하므로 좀 덜 정확(loose)함을 알 수 있다.

부호율 $r=k/n(k > 1)$ 인 길쌈부호의 k비트 심볼은 각 심볼별로 서로 다른 오율을 나타내며 이를 길쌈부호의 부등심볼오율 특성이라 한다.^[7] 심볼별 오율을 모두 합하면 심볼오율이 되므로, 만약 심볼별 오율을 구할 수 있다면 영 심볼이 산입되지 않는 보다 정확한 심볼오율 상한을 구할 수 있다.

본 논문에서는 길쌈부호의 부등심볼오율 특성을 적용하여 Perez 등의 결과식보다 정확한 심볼오율 상한을 구하여 RS/길쌈 연쇄부호의 비트오율을 계산하였다. 연쇄부호에 적합한 부호율을 갖는 부호들을 대상으로 본 논문의 상한을 기존의 상한과 비교하였다.

II. 기존의 심볼오율 상한

부호율 $r=k/n$ 길쌈부호기는 형태에 따라 k개의 쉬프트 레지스터에 정보비트가 병렬로 입력되는 병렬형과 한개의 쉬프트레지스터에 정보비트가 직렬로 입력되는 직렬형으로 구분할 수 있다. 병렬형에서는 k개의 쉬프트레지스터중 가장 긴 길이를 기억차수(memory order) m이라 하여 (n, k, m)로 길쌈부호기를 표시하고^[8], 직렬형에서는 쉬프트레지스터의 길이를 L이라 할때 이를 구속장으로 하여 k/n, 구속장

L로 길쌈부호기를 표시한다.^[9] 그림 2(a) 와 그림 2(b)는 각각 (3, 2, 1) 병렬형과 $r=2/3, L=4$ 직렬형 길쌈부호기를 나타낸다.

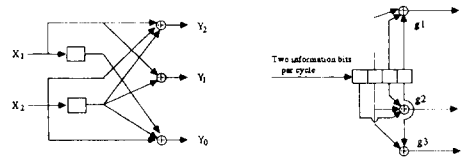


그림 2. 길쌈부호기의 두가지 형태
(a) (3, 2, 1) 병렬형
(b) $r=2/3, L=4$ 직렬형
Fig. 2. Two types of convolutional encoder.
(a) (3, 2, 1) parallel type,
(b) $r=2/3, L=4$ serial type.

벡터 U는 k비트를 심볼단위로 하여 시작과 끝의 심볼이 영이 아니고 m_{min} 심볼 이상의 연속적인 영을 포함하지 않으며, j개 심볼로 구성되어 있다고 하자. U에 m_{min} 개의 영 심볼이 이어져 부호율 k/n 비조직 feedforward 길쌈부호기에 입력된다면 이에 대응하는 출력벡터 C는 격자상도에서 해밍거리 d, 길이 $l=j+m_{min}$ 인 제 1류 오류경로가 된다. O(all zero vector)가 이산 무기억통신로(discrete memoryless channel)를 통하여 전송되어 O 대신 C가 선택되었을 경우, U상에서 b비트 심볼이 (b-k)비트씩 겹쳐 심볼오류가 발생할 경우의 수는 $(l-m_{min}+b/k-1)$ 이다. m_{min} 은 오류경로가 발생되기 위하여 맨 마지막에 연속으로 입력되어야할 최소한의 영 심볼갯수를 의미한다. 비조직 feedforward 부호기에서 모든 오류경로들은 맨 마지막에 최소한 m_{min} 가지길이 만큼 0로 되어 있으며 이들은 오류심볼에 기여하지 않으므로 m_{min} 만큼 빼어 준다. [6]에서는 병렬형 부호기만 고려하였으며 이 경우에는 가장 짧은 쉬프트레지스터의 길이에 해당한다. 직렬형에서는 $m_{min} = \lfloor \frac{L-k}{k} \rfloor$ 과 같이 정의되어야 한다. 여기서 $\lfloor x \rfloor$ 는 x 보다 적은 가장 큰 정수를 나타낸다. 조직 feedback 부호기에서는 영아닌 정보비트들이 입력되어야 부호기가 영상태에 도달하여 정확한 경로에 합류한다. 따라서 A(d, l)을 해밍거리 d, 길이 l인 오류경로의 수로 정의하면 b비트 심볼오율의 상한 SER_b 는

$$SER_b \leq \sum_{d=1}^{\infty} \sum_{l=m_{min}+1}^{\infty} \left(l - E \cdot m_{min} + \frac{b}{k} - 1 \right) A(d, l) P_d \quad (2)$$

과 같이 표현된다.^[5,6] 여기서 P_d 는 해밍거리가 d인

오류경로의 발생확률이다. 그리고 E^* 는

$$E^* = \begin{cases} 1 & \text{feedforward encoder} \\ 0 & \text{feedback encoder} \end{cases} \quad (3)$$

로 표시되는 함수이다. 식 (2)를 식 (1)에 대입하면 RS/길쌈 연쇄부호의 비트오율을 얻게된다. 그림 3에 $r=1/2, L=4$ 부호에서 $t=10$ 인 오류경로에 대한 2비트 오류심볼 계산의 예를 보였다. 예에서 알 수 있듯이 영 심볼도 오류심볼로 계산되므로 상한식 계산에 영 심볼이 산입되지 않도록 하면 상한은 더욱 정확해 질 것이다.

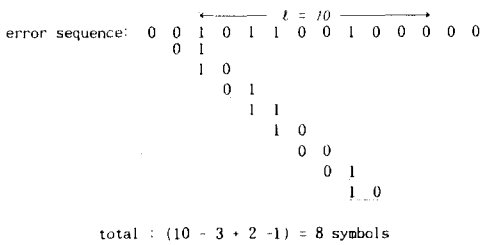


그림 3. $r=1/2, L=4$ 길쌈부호에서 2비트 오류심볼 계산의 예

Fig. 3. 2-bit error symbol count example for $r=1/2, L=4$ convolutional code.

III. 길쌈부호의 심볼오율 상한

1 길쌈부호의 부등심볼오율

일반적으로 부호율 $r=k/n, n \geq k$ 가 2이면 길이가 k 인 벡터 (i_1, i_2, \dots, i_k) 가 매 시점마다 i_1, i_2, \dots, i_k 의 순서로 부호기에 입력 된다고 볼 수 있다. 여기서 $i_j \in \{0, 1\}$ 이고 $j=1, 2, \dots, k$ 이다. 격자상도에서 임의의 오류경로를 출력시키기 위하여 입력한 정보심볼들의 구성수는 서로 다를 수 있다. 이를 다시 얘기하면, 각각의 정보심볼이 갖는 오율값이 다름을 의미한다. 한 예로써 생성행렬 $G=(6, 7, 15)$ 인 $r=2/3, L=4$ 인 길쌈부호기의 격자상도를 그림 4에 나타내었다. 상태 (00)에서 상태 (01)로 천이하는 가지의 100/10은 입력벡터(01)와 출력벡터 (100)을 의미하며, 입력벡터 (01)은 부호기에 입력될 때 0부터 먼저 입력된다. 여기서 최소자유거리는 $d=3$ 이고, 두개의 경로가 있다. 모두 영인 경로를 기준으로 해서, 두개의 최소 자유거리를 갖는 오류경로에 대한 심볼열은 (10)(00), (10)(11)(00)이다. 이들 중 (10) 심볼의 오류 발생 빈도가 다른심볼에 비해서 높음을 알 수

있다. (01), (10), (11)의 심볼 오류 형태를 각각 $h=1, 2, 3$ 으로 표시하기로 한다. 임의의 오류경로에 대한 입력심볼열에서 어떤 오류 형태 $h \in \{1, 2, 3\}$ 의 구성수를 t 로 표시한다. 그리고 $A(d, t; h)$ 는 해밍 거리가 d 이며, 해당 입력 심볼열에서 h 형태의 심볼수가 t 인 오류경로 수를 나타낸다고 하자. 여기서 $d \geq d_f, t \geq 1$ 이다. 그리고 부호율이 $r=b/n$ 인 일반적인 경우에는 $1 \leq h \leq 2^k - 1$ 이다.

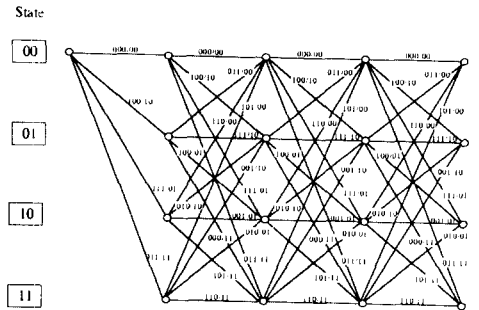


그림 4. $r=2/3, L=4, G=(6, 7, 15)$ 인 길쌈부호의 격자상도

Fig. 4. Trellis diagram for the convolutional code of $r=2/3, L=4$, and $G=(6, 7, 15)$.

만약 임의의 길쌈부호에서 모든 $A(d, t; h)$ 를 얻을 수 있으면 심볼 h 에 대한 오율은

$$P(h) \leq \sum_{d=d_f}^{\infty} \sum_{t=1}^{\infty} t A(d, t; h) P_d \quad (4)$$

이다. [7] 여기서

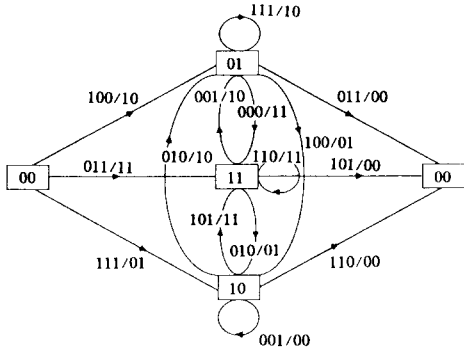
$$P_d = Q \left(\sqrt{\frac{2dE_b}{N_0}} \right) \quad (5)$$

이다. 그리고 N_0 는 잡음의 양주파수대 전력밀도이며, E_b 는 신호에너지이다.

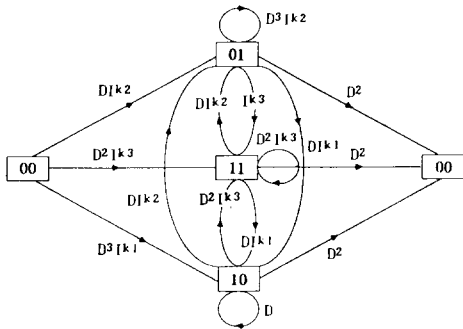
식 (4)의 계산은 격자상도상에서 $A(d, t; h)$ 를 컴퓨터 프로그램으로 구해서 할 수 있으며, 또한 다음의 예와 같이 전달함수를 사용할 수도 있다. 그림 5는 그림 4의 격자상도로부터 얻은 상태도이다. 변수 D, I , 그리고 $k_i, i \in \{1, 2, 3\}$ 이 도입되었다. 역역수의 기호로 k_1, k_2 , 그리고 k_3 가 사용되었는데, 해당 입력심볼의 형태를 나타낸다. 상태도에서 전달함수는 $T(D, I; k_1, k_2, k_3)$ 로 표시할 수 있으며

$$T(D, I; k_1, k_2, k_3) = D^3 I^{k_1} + 2D^3 I^{k_1} D^4 I^{k_2} + 2D^4 I^{k_3} \quad (6)$$

로 표현된다.



(a)



(b)

그림 5. $r=2/3, L=4, G=(6, 7, 15)$ 인 길쌈부호의 상태도

(a) 상태도 (b) 그림 5(a)의 지수형 표현

Fig. 5. State diagram for the convolutional code of $r=2/3, L=4,$ and $G=(6, 7, 15)$.

(a) State diagram.

(b) Exponential representation of Fig. 5(a).

일반적으로는 $T(D, I; k_1, k_2, \dots, k_{L-1})$ 로 표시할 수 있으며, 오류형태 h 의 변수인 k_h 이외에 모두 영으로 두어 심볼오출을 구하면

$$P(h) = \frac{\partial}{\partial I} T(D, I; 0, \dots, k_h, 0, \dots, 0) \Big|_{k_h = 1, D^d = P_d} \quad (7)$$

이다. 여기서 $h=1, \dots, 2^k-1$ 이고 P_d 는 식 (5)에 정의되어 있다. 그림 6은 식 (7)을 사용하여 구한 $r=2/3, L=4, G=(6, 7, 15)$ 에 대한 심볼오출이다. 여기서 심

볼오출이 서로 다른 부등심볼오출 특성을 볼 수 있다.

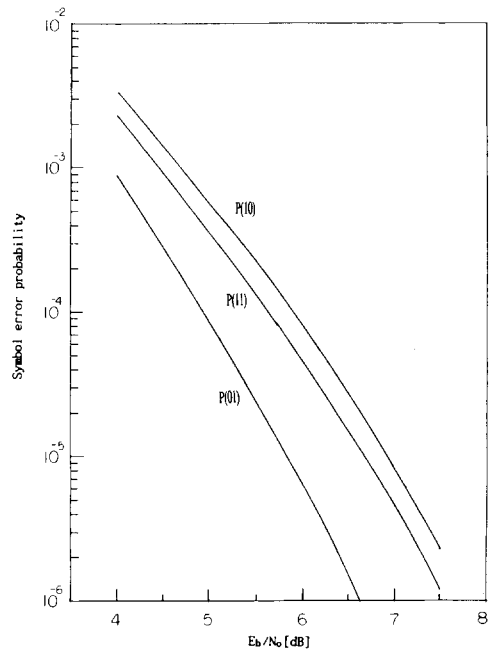


그림 6. $r=2/3, L=4, G=(6, 7, 15)$ 인 길쌈부호의 부등심볼오출 특성

Fig. 6. Unequal symbol error probability for the convolutional code of $r=2/3, L=4,$ and $G=(6, 7, 15)$.

2 부등심볼오출식을 이용한 심볼오출 상한 식 (4) 또는 식 (7)에서 오류 심볼형태 변수 h 에 대하여 합하면 k 비트 심볼오출이 되며

$$P_{sk} = \sum_{h=1}^{2^k-1} P(h) \quad (8)$$

와 같이 주어진다. 심볼크기가 b 비트인 RS/길쌈 연쇄부호의 비트오출 해석에 필요한 길쌈부호의 b 비트 (k 의 정수배) 심볼오출은 식 (8)을 확장 적용하면 구할 수 있다. $b=ck$ (c 는 상수)인 b 비트 심볼오출을 해석하기 위하여 후호출 k/n 길쌈부호 격자상도의 연이은 c 개 가지를 병합하여 가지당 b 비트로 구성된 초격자상도(super-trellis diagram)를 구성한다. 그림 7에 $r=1/2, L=3, G=(7, 5)$ 인 격자상도에서 $c=2$ 인 경우의 병합과정을 나타내었다. 병합과정에서 초격자상도의 무효한 초가지(super-branch)는 제거되어야 한다. 영이 아닌 상태에서 영 상태에 도달하거나 영

상태에서 영이 아닌 입력에 의하여 영 상태에 도달한 후, 다시 영이 아닌 상태로 천이하거나 영이 아닌 입력에 의하여 영 상태에 도달하는 가치를 포함하는 초가치는 무효이다. 예로써 그림 7(b)의 상태 (10)에서 시작하여 상태 (01)로 끝나는 정보심볼열 (01)의 초가치는 무효이다. 무효 초가치를 포함하는 경로는 기초 격자상도상에서 영 상태에 도달한 후 다시 새로운 오류경로를 시작하게 되므로 단일오류경로로 볼 수 없다. 모든 가능한 무효 초가치의 형태를 그림 8에 나타내었다.

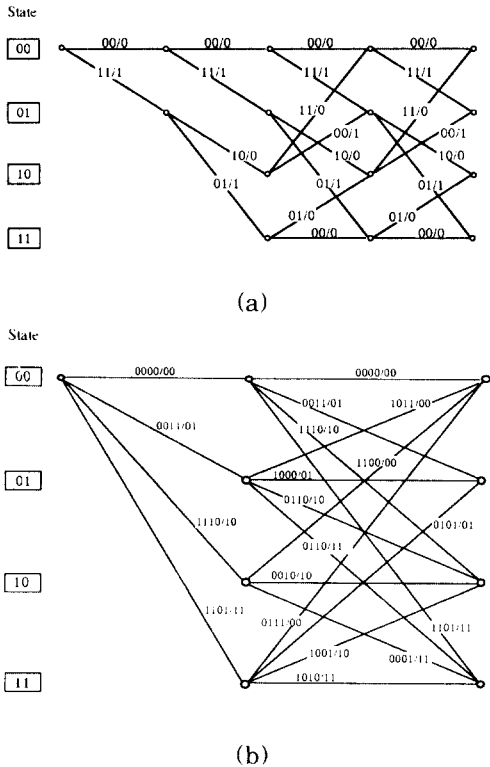


그림 7. 격자상도의 병합과정

- (a) $r=1/2$, $L=3$, $G=(7, 5)$ 길쌈부호의 격자상도
- (b) 그림 7(a)의 초격자상도, $c=2$

Fig. 7. Trellis diagram merging procedure.

- (a) Trellis diagram of $r=1/2$, $L=3$, $G=(7, 5)$ convolutional code.
- (b) Super-trellis diagram of Fig. 7(a), $c=2$.

그림 7의 초격자상도에서 2비트 심볼오율을 구할 수 있다. 그림 7에서 최소 해밍거리 5인 오류경로의

해당 입력 열들은 (0 1 0 0), (1 0 0 0) 이므로 (01), (10) 2개의 심볼오류가 발생하였다.

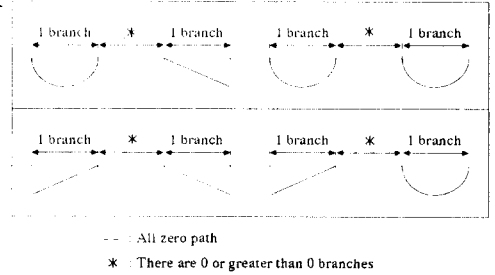


그림 8. 모든 가능한 무효 초가치의 형태들
Fig. 8. All possible patterns of invalid super-branches.

마찬가지로 각 해밍거리별 오류경로로부터 2비트 심볼오류 발생수 $A(d, t; h)$ 를 구하면 2비트로 구성된 심볼별 오율을 구할 수 있으므로 이를 모두 합하여 심볼오율을 계산한다. 심볼오율 계산과정을 일반화하면 다음과 같다.

단계 1) 부호율 k/n 길쌈부호의 격자상도에서 c 개의 연이은 가지들을 병합하여 b 비트 단위까지의 초격자상도를 구성한다. 그리고 무효한 초가치는 제거한다.

단계 2) 초격자상도의 오류경로수 $A(d, t; h)$ 를 $A_b(d, t; h)$ 라 두자. $A_b(d, t; h)$ 를 구하고 식 (9)로부터 b 비트 심볼오율을 구한다.

$$P_{sb} = \sum_{k=1}^{2^b-1} P(h) = \sum_{k=1}^{2^b-1} \sum_{t=1}^{\infty} t A_b(d, t; h) P_d \tag{9}$$

식 (9)를 식 (1)에 대입하면 RS/길쌈 연쇄부호의 비트오율이 된다. RS부호의 부호율이 K/N 이고 길쌈부호의 부호율이 k/n 이면 정보비트에너지 E_b 와 E_s 사이에는

$$E_s = \left(\frac{K}{N} \times \frac{k}{n} \right) E_b \tag{10}$$

가 성립한다.

IV. 두 심볼오율 상한의 비교, 검토 및 시뮬레이션

본 논문에서 구한 상한을 기존의 상한과 비교한다. 식 (2)와 식 (9)를 비교해 볼때 차이점은

$$\sum_{i=m_{\min}+1}^{\infty} \left(l - E \cdot m_{\min} + \frac{b}{k} - 1 \right) A(d, l) \quad (11)$$

와

$$\sum_{h=1}^{2^b-1} \sum_{t=1}^{\infty} t A_b(d, t; h) \quad (12)$$

이며, 이들은 각 식에서 해당 해밍거리 d에 대한 전체 오류심볼수이다. 몇몇 직렬형 이진 길쌈부호기에 대하여 두 상한을 비교한다. 부호율 1/2, 1/3, 2/3, 3/4, 2/5 그리고 3/5 직렬형 길쌈부호기^[8]를 고려하며 심볼크기 b는 2에서 최대 9까지 고려한다. 각 부호의 격자상도상에서 수식값을 구하는 전산프로그램을 개발하여 계산한 결과들을 비교하였다. 계산된 결과의 일부를 선정하여 표 1에 정리하였다.

표 1. 길쌈부호의 오류심볼수의 비교

상한식 a : 본 논문의 상한

상한식 b : McEliece 또는 Perez의 상한

Table 1. Comparison of error symbol count of convolutional codes.

bound a : this paper's upper bound

bound b : upper bound of McEliece or Perez

(a) r=2/3, L=5, G=(16, 23, 31)

Symbol size	bound \ d	4	5	6	7	8	9	10	11
2 bit	a	1	9	82	320	1417	6053	24583	98158
	b	1	12	98	388	1697	7228	29327	116896
4 bit	a	2	14	115	436	1855	7805	31322	123838
	b	2	17	122	459	1935	8090	32318	127284
6 bit	a	3	19	139	507	2093	8667	34313	134226
	b	3	22	146	530	2173	8952	35309	137672
8 bit	a	4	24	163	578	2331	9529	37304	144616
	b	4	27	170	601	2411	9814	38300	148060

(b) (r=3/2, L=7, G=(177, 55, 112))

Symbol size	bound \ d	6	7	8	9	10	11	12	13
2 bit	a	26	70	434	1759	8507	35781	154216	644208
	b	34	84	565	2257	11032	46303	200414	837445
4 bit	a	40	97	606	2397	11539	48034	206154	856566
	b	43	103	645	2533	12154	50424	215933	895534
6 bit	a	49	116	692	2706	12847	52985	225559	931640
	b	52	122	725	2809	13276	54545	231452	953623
8 bit	a	58	135	772	2982	13969	57106	241078	989729
	b	61	141	805	3085	14398	58666	246971	1011712

(c) r=3/4, L=5, G=(36, 14, 32, 07)

Symbol size	bound \ d	3	4	5	6	7	8	9
3 bit	a	11	66	318	1492	6736	29407	125534
	b	15	81	370	1686	7463	32114	135602
6 bit	a	17	89	398	1788	7845	33537	140896
	b	21	104	450	1982	8572	36244	150964
9 bit	a	23	112	478	2084	8954	37667	156258
	b	27	127	530	2278	9681	40374	166326

(d)r=3/4, L=7, G=(127, 45, 106, 172)

Symbol size	bound \ d	4	5	6	7	8	9	10
3 bit	a	7	84	580	3449	20339	118131	665049
	b	7	88	620	3712	21959	127768	720245
6 bit	a	10	113	732	4229	24502	140328	781675
	b	10	115	747	4294	24795	141816	879016
9 bit	a	13	140	859	4811	27338	161265	850446
	b	13	142	874	4876	27631	155864	857787

(e) r=2/3, L=7, G=(71, 166, 112, 55, 177)

Symbol size	bound \ d	11	12	13	14	15	16	17	18
2 bit	a	11	22	25	68	148	343	762	1487
	b	13	27	28	75	175	416	910	1776
4 bit	a	17	31	35	85	196	464	997	1939
	b	18	36	36	91	211	492	1055	2035
6 bit	a	22	40	43	104	232	541	1149	2212
	b	23	45	44	107	247	568	1200	2294
8 bit	a	27	49	51	120	268	617	1294	2471
	b	28	54	52	123	283	644	1345	2553

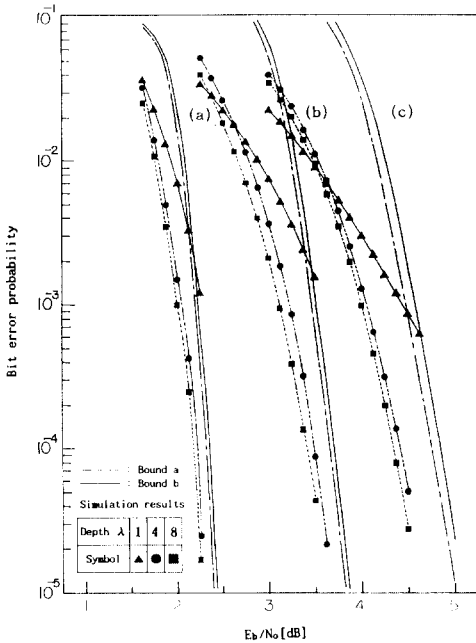
(f) r=3/5, L=7, G=(34, 127, 45, 106, 172)

Symbol size	bound \ d	6	7	8	9	10	11	12	13
2 bit	a	5	49	151	500	1803	6567	22585	73746
	b	5	57	165	544	1963	7116	24441	79524
4 bit	a	9	70	198	634	2228	7990	27059	87051
	b	9	78	212	670	2350	8349	28164	90314
6 bit	a	13	91	245	760	2615	9223	30782	97841
	b	13	99	259	796	2737	9582	31887	101104

(g) r=3/5, L=8, G=(334, 44, 124, 216, 357)

Symbol size	bound \ d	7	8	9	10	11	12	13	14
2 bit	bound a	52	206	791	2942	9773	32449	109820	368340
	bound b	68	254	950	3471	11405	37553	126078	419739
4 bit	bound a	79	250	969	3534	11585	38089	127743	424966
	bound b	87	305	1118	4007	12981	42242	140431	463589
6 bit	bound a	93	303	1136	4072	13166	42760	142072	468802
	bound b	106	356	1286	4543	14557	46931	154784	507439

계산의 비교결과 구속장이 정보어 길이 k 의 정수배 되는 부호를 제외하고는 본 연구의 결과가 기존의 상한보다 정확하며, 구속장이 정보어 길이 k 의 정수배 되는 경우에는 두 결과가 거의 일치한다. 본 논문의 상한이 보다 더 정확해 질 수 있는 이유는 기존의 상한은 오류경로 상에서 발생하는 영 심볼을 전체 심볼 오류수에 포함시키지만, 본 논문의 상한은 영 심볼을 산입하지 않으므로 보다 더 정확하게 된다. 그림 9는 식 (1)에 식 (2) 또는 식 (9)를 대입하여 구한 RS/길쌈 연쇄부호의 이론적 비트오율 성능과 시뮬레이션 결과를 나타낸다. (63, 55) RS부호와 $r=3/4$, $L=5$ 길쌈부호를 사용한 연쇄부호의 경우 $E_b/N_0=4.5$ [dB] 에서 본 논문의 심볼오율 상한을 대입한 비트오율이 기존의 심볼오율 상한을 대입한 것에 비하여 1.6×10^{-3} 에서 8.0×10^{-4} 으로 감소하였다. 시뮬레이션에서는 $N \times \lambda$ 블럭인터리버를 사용하였다.



(a) (255, 223) RS+r=2/5, L=7 C.C
 (b) (63, 55) RS+r=3/5, L=8 C.C
 (c) (63, 55) RS+r=3/4, L=5 C.C

그림 9. RS/길쌈 연쇄부호화의 비트오율과 시뮬레이션 결과

Fig. 9. The bit error probability of RS/convolutional concatenated codes and simulation results.

여기서 N 은 RS부호어의 길이, λ 는 인터리빙 심도(depth)이다. b 비트 심볼을 단위로하여 인터리빙

시에는 N 개의 심볼씩 가로로 입력하고 λ 심볼씩 세로로 읽어 출력시킨다. 디인터리빙은 인터리빙의 역과정을 수행한다. λ 를 크게하면 성능은 좋아지나 지연시간이 길어지므로 적절한 trade off가 필요하다. 시뮬레이션에서는 λ 를 1, 4, 8로 하여 수행하였다. $\lambda = 1$ 은 인터리버를 전혀 사용하지 않은 경우이다. 시뮬레이션 결과에서 보듯이 연쇄부호에서는 인터리버의 사용이 요구됨을 알 수 있으며, 비트오율 성능과 인터리버 지연을 고려할때 $\lambda = 4$ 가 적절하다.

V. 결론

본 논문에서는 RS 부호와 연쇄된 길쌈부호 부등심볼오율 상한을 구하여 RS/길쌈 연쇄부호의 비트오율 계산하였다. 연쇄부호에 적합한 부호율을 갖는 길쌈부호들을 대상으로 본 논문의 상한을 기존의 상한과 비교하였다. (63, 55) RS부호와 $r=3/4$, $L=5$ 길쌈부호를 사용한 연쇄부호의 경우 비트에너지대 잡음에너지의 비가 4.5 [dB] 일 때 본 논문의 심볼오율 상한을 대입한 비트오율이 기존의 심볼상한에 의한 것에 대하여 1.6×10^{-3} 에서 8.0×10^{-4} 으로되어 본 논문의 결과가 더 정확하였다. 구속장이 정보어 길이의 정수배가 되는 부호를 제외하고는 본 논문의 상한이 기존의 것보다 정확하고 정수배되는 부호에 대해서는 보다 더 정확하거나 동일하였다.

參考文獻

[1] R. E. Blahut, *Theory and Practice of Error Control Codes*, Addison-Wesley, Massachusetts, chapt. 7, 1983.
 [2] A. J. Viterbi and J. K. Omura, *Principles of Digital Communication and coding*, McGraw-Hill, New-York, chapt. 4, 1976.
 [3] J. P. Odenwalder, "Optimal Decoding of Convolutional Codes." Ph.D *Dissertation, school of Engineering and Applications Sciences, University of California, Los Angeles, U. S. A., 1970.*
 [4] G. Clark, Jr. and J. Cain, *Error-Correction Coding for Digital Communications*, Plenum press, New York, chapt. 8, 1981.
 [5] R. J. McEliece and I. M. Onyszchuk, "A Symbol Error Upper Bound for

- Convolutional Codes," *Proceedings of the 27th Annual Allerton Conference on Communications, Control, and Computing*, Allerton House, Monticello, Illinois, U. S. A., pp. 334-335, Sept. 1989.
- [6] L. C. Perez and D. J. Costello, Jr., "An Upper Bound on the Symbol Error Rate of Convolutional and Trellis Codes," *Proceedings of the 1990 International Symposium on Information Theory and Its Applications*, Hawaii, pp. 27-30, Nov. 1990.
- [7] 임중수, 이상곤, 문상재, "길쌘부호를 사용한 embedded DPCM방식에 관한 연구", 대한전자공학회 논문지, 제 28-a권, 1호, pp. 1-7, 1991년 1월.
- [8] S. Lin and D.J Costello, Jr., *Error Control Coding: Fundamentals and Applications*, Prentice Hall, New Jersey, chap. 10, 1983.
- [9] D. G. Daut, J. W. Modestino, and L. D. Wismer, "New Short Constraint Length Convolutional Code Constructions for Selected Rational Rates," *IEEE Trans. Inform. Theory*, vol. IT-28, pp. 794-800, Sept. 1980.

 著者紹介

文相在(正會員) 第26卷3號 參照
 현재 경북대학교 전자공학과 교수



李相坤(正會員)

1960年 12月 1日生. 1986년 2월 경북대학교 전자공학과(공학사). 1988년 2월 경북대학교 대학원 전자공학과(공학석사). 1993년 2월 경북대학교 대학원 전자공학과(공학박사). 1991년 3월 ~ 현재 창신전문대학 전자통신과 조교수. 주관심분야 부호기술, 이동통신, 위성통신, 암호화 등.