

무직류 격자형부호를 구성부호기로 가지는 터보부호

이병길^o, 정창기, 배상재, 주연경
 경북대학교 전자공학과
 E-mail : bglee@dcl4.kyungpook.ac.kr

Turbo Codes with DC-Free Trellis Codes as Constituent Codes

Byoung Gil Lee^o, Chang Ki Jeong, Sang Jae Bae, Eon Kyeong Joo
 Dept. of Electronics, Kyungpook National University

Abstract - In this paper, turbo codes with the DC-free trellis codes based on partition chain as constituent codes are presented. And efficient methods to design the DC-free turbo codes are introduced. An iterative decoding with the MAP algorithm is used for the decoding of the turbo codes designed by various methods. As results of simulations, the presented DC-free turbo codes show better error performances than the DC-free trellis codes.

1. 서론

무직류 부호(DC-free codes)는 디지털 저장 및 전송시스템에서 전송신호의 스펙트럼을 주어진 채널의 주파수 전달특성에 맞추기 위하여 널리 사용된다. 무직류 특성을 만족하기 위한 필요충분조건은 그 부호가 항상 유한한 축적디지털합(running digital sum; RDS)을 가지는 것이다[1]. 이러한 무직류 부호는 잡음에 대한 변역성을 향상시켜 주기 위하여 오류정정부호와 결합될 수 있다.

터보부호(turbo codes)는 최근 Berrou등에 의하여 발표된 오류정정부호로서 비교적 간단한 구조를 가지면서도 성능이 매우 우수한 것으로 알려져 있다[2]. 터보부호의 부호기는 2개 이상의 구성부호(constituent codes)가 인터리버(interleaver)를 사이에 두고 병렬 연결된 형태로 되어 있다. 일반적인 터보부호에서는 구성부호로 기억소자의 개수가 2~5인 순환체계적결합부호(recursive systematic convolutional codes; RSC codes)가 사용되지만 블록부호(block codes)나 격자형부호(trellis codes) 등도 사용될 수 있다. 복호기는 각 구성 부호기에 대응되는 연판정 출력이 가능한 MAP(maximum a posteriori probability) 복호기가 직렬로 연결된 형태로 구현된다. 따라서 터보부호의 경우, 반복복호(iterative decoding)가 가능하며 이로 인해 기존의 오류정정부호에 비해 아주 우수한 성능을 얻을 수 있다. 그러나 이러한 터보부호도 그 자체로는 무직류 특성을 만족하지 못

한다.

따라서 본 논문에서는 Reed-Muller 부호의 부호분할고리물 이용하여 설계된 무직류 격자형부호를 구성부호로 가지는 터보부호를 구성하고 그 성능을 구한다. 그 결과, 구성된 무직류 터보부호는 무직류 격자형부호에 비해 더 우수한 오류성능을 가진다.

2. 무직류 격자형부호

길이가 2^n 인 r 차 Reed-Muller 부호 $RM(r,n)$ 은 차원이 $\sum_{i=0}^r \binom{n}{i}$ 이고 최소 Hamming 거리가 2^{n-r} 인 투명한 선형블록 부호이다. 편의상 길이가 2^n 인 모두 0인 벡터로만 구성된 부호를 Reed-Muller 부호로 간주하여 $RM(-1,n)$ 이라 표시하면 동일한 길이를 가지는 Reed-Muller 부호들은 $(n+1)$ -레벨 부호분할고리 $RM(n,n)/RM(n-1,n)/\dots/RM(-1,n)$ 으로 표현될 수 있다.

일반적으로 무직류 격자형부호 $C(RM(r,n)/RM(r-j,n);E)$ 는 그림 1에서와 같이 부호분할고리의 일부분인 $(r+1)$ -레벨 분할고리 $RM(r,n)/RM(r-1,n)/\dots/RM(-1,n)$ 을 기반으로 설계될 수 있다[3]-[5].

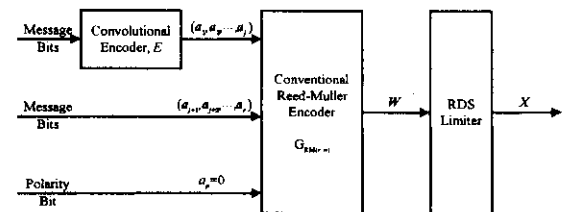


그림 1. 무직류 격자형부호의 부호기

그림 1에서 $a_i (1 \leq i \leq r)$ 는 분할고리 $RM(r,n)/RM(r-1,n)/\dots$

$/RM(-1,n)$ 에서 i 번째 분할 $RM(r-i+1,n)/RM(r-i,n)$ 에 대한 라벨로서 $\log_2 q_i$ 비트로서 표현된다. 여기서 q_i 는 i 번째 분할의 위수(order)이다. 따라서 레벨 $r+2$ 에 위치한 부분집합들은 $(a_1, a_2, \dots, a_{r-1})$ 에 의해 표시될 수 있고 벡터 $W = [a_1, a_2, \dots, a_r, 0]$ $\cdot G_{RM(r,n)}$ 라 두면 전송벡터 X 는 다음과 같이 구해진다.

$$X = \begin{cases} W, & \text{for } RDS_{t-1} = 0, DS(W) \leq 0 \\ & \text{or } RDS_{t-1} \neq 0, RDS_{t-1} \cdot DS(W) \leq 0 \\ \bar{W}, & \text{otherwise.} \end{cases} \quad (1)$$

$$= [a_1, a_2, \dots, a_r, a_p] \cdot G_{RM(r,n)}$$

여기서 RDS_{t-1} 는 시간 $t-1$ 에서 전송된 벡터 끝에서 축적디지털합이고 $DS(W)$ 는 벡터 W 의 디지털합(digital sum)이다. 그리고 식(1)에서 a_p 가 0일때는 W 가 전송되며 1일때는 \bar{W} 가 전송됨을 의미한다. 무직류 격자형부호기에서 처음에는 a_p 가 0으로 고정되어 있으므로 $RM(r,n)$ 의 벡터의 반만이 선택되지만 RDS 제한기에 의하여 나머지 반도 선택되어 전송될 수 있다.

3. 무직류 터보부호의 설계

3.1 기존의 터보부호

터보부호의 구성부호로는 RSC 부호가 사용되지만 블록부호나 격자형부호 등도 사용될 수 있다. 그림 2는 2개의 구성부호기를 가지는 일반적인 터보부호의 부호기를 개괄적으로 나타낸 것이다.

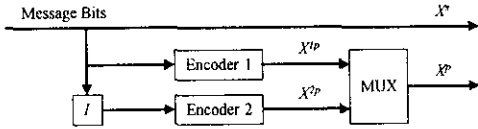


그림 2. 터보부호의 부호기

터보부호의 복호기는 각각의 구성부호기에 대응되는 구성복호기가 인터리버와 디인터리버를 사이에 두고 직렬로 연결된 형태로 구현된다. 구성복호기로 MAP 복호기가 사용되는 경우의 터보복호기를 그림 3에 나타내었다.

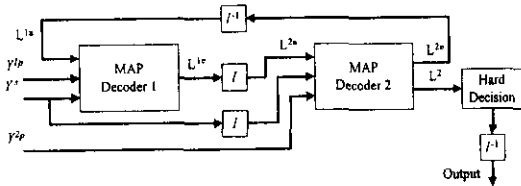


그림 3. 터보부호의 복호기

그림 3에서 각 구성복호기는 잉여정보(extrinsic information) L^e 를 출력시키며 이 잉여정보는 다음단의 구성복호기에 사전정보(priori information) L^p 로 입력되어 반복복호를 가능하게 한

다.

3.2 무직류 터보부호

무직류 격자형부호를 구성부호로 가지는 터보부호는 터보 격자형부호변조(turbo trellis coded modulation; TTCM)[6]-[8]와 비슷한 형태로 설계될 수 있다. 그러나 제안된 무직류 터보부호기는 터보격자형부호변조와는 달리 조건표(look-up table)가 아닌 Reed-Muller 부호의 생성행렬에 의하여 벡터 W 가 구해진다. 그리고 무직류 특성을 얻기 위하여 MUX 뒤에 RDS 제한기가 사용된다. 그림 4는 이러한 무직류 터보부호 중 $C(RM(2,3)/RM(1,3);E)$ 를 구성부호기로 가지는 터보부호의 부호기를 개괄적으로 나타낸 것이다.

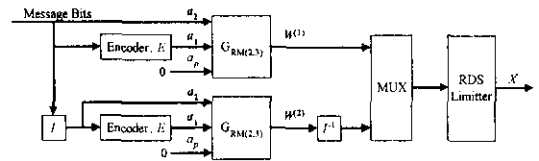


그림 4. 무직류 터보부호의 부호기

여기서 I 는 5비트로 구성된 짝수 번째 심벌을 짝수 번째 위치로 치환하고 홀수 번째 심벌을 홀수 번째 위치로 치환하는 인터리버이고 I^1 은 역치환을 하는 디인터리버이다. 그리고 심벌 $W^{(1)}$ 과 $W^{(2)}$ 는 각각 첫 번째와 두 번째 무직류 격자형부호기에서 출력되는 벡터이다. $W^{(1)}$ 은 그대로 MUX에 입력되지만 $W^{(2)}$ 는 I^1 에 의하여 치환된 후 MUX에 입력된다. MUX는 홀수 번째 시간에는 $W^{(1)}$ 만을, 짝수 번째 시간에는 $W^{(2)}$ 만을 출력한다. 따라서 전체적으로 이 부호는 체계적이다. 마지막으로 RDS 제한기는 식 (1)에 의하여 전송 벡터 X 를 구한다.

무직류 터보부호에서 구성부호로 병렬천이를 가지거나 또는 가지지 않는 무직류 격자형부호가 사용될 수 있다. 일반적으로 메모리가 적은 구성부호기는 병렬천이를 가지는 것이, 메모리가 많은 부호기는 병렬천이를 가지지 않는 것이 유효성능면에서 더 유리한 것으로 알려져 있다. 그림 5는 무직류 격자형부호 $C(RM(2,3)/RM(1,3);E)$ 에서 병렬천이를 만드는 메모리 수가 3인 길쌈부호기 E 를 나타낸 것이다.

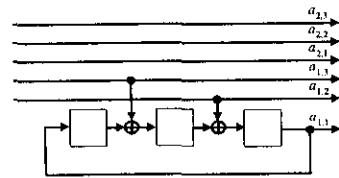


그림 5. 병렬천이를 발생시키는 경우의 길쌈부호기

여기서 부호화된 3비트 $(a_{1,1}, a_{1,2}, a_{1,3})$ 는 분할 $RM(2,3)/RM(1,3)$ 에 대한 라벨 a_1 으로 사용되고 부호화되지 않은 3비트 $(a_{2,1}, a_{2,2}, a_{2,3})$ 는 분할 $RM(1,3)/RM(0,3)$ 에 대한 라벨 a_2 로 사용된다. 그런데 만약 5비트 정보심벌이 I 에 의하여 단순히 치환되면 첫 번째 격자형부호에서 부호화되지 않은 비트들은 두 번째 격자형부호에서도 부호화되지 않는다. 이것은 무직류 터보부호의 취

소 Hamming 거리를 감소시키는 것으로 알려져 있다. 따라서 첫 번째 구성부호기에서 부호화되지 않은 비트들은 두 번째 구성부호기에서 부호화 될 필요가 있으며 이러한 목적으로 사용되는 것이 사상기(mapper)이다[8]. 사상기는 동일 심벌내의 비트를 교환하는 단순한 구조로 구현될 수 있다. 그러나 $C(RM(2,3)/RM(1,3);E)$ 를 구성부호로 가지는 무직류 터보부호기에서 구성부호기를 2개만 사용할 경우에는 어떤 사상기를 사용하더라도 여전히 1비트가 각 구성부호기에서 부호화되지 않는다. 이러한 문제를 해결하는 방법으로 우선 3개 이상의 구성부호기를 사용하는 경우를 들 수 있다. 그러나 이 경우 부호율이 저하되고 부호의 복잡도가 증가하는 단점이 있다. 따라서 메모리 수가 3개인 2개의 구성부호기만을 사용하더라도 모든 정보 비트들이 최소한 한 번 이상 부호화되도록 할 필요가 있다. 그림 6은 본 논문에서 이러한 목적으로 사용될 수 있는 길쌈부호기와 사상기의 한 예를 나타낸 것이다.

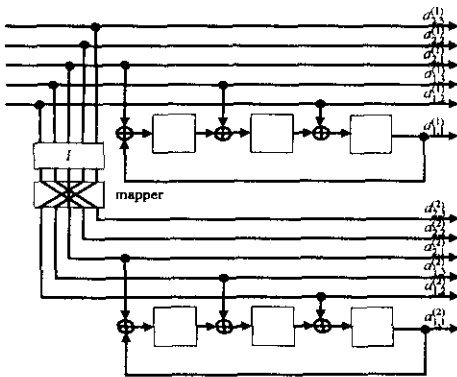


그림 6. 사상기를 사용할 경우 적합하게 변형된 길쌈부호기들

한편, 메모리가 많은 경우에는 병렬천이를 가지지 않는 무직류 격자형부호를 구성부호로 사용하는 것이 오류성능면에서 더 유리하다. 그림 7은 병렬천이를 발생시키지 않는 길쌈부호기와 $C(RM(2,3)/RM(1,3);E)$ 를 구성부호기로 가지는 무직류 터보부호의 부호기를 나타낸 것이다.

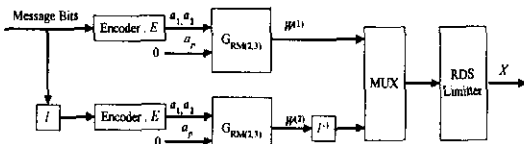


그림 7. 병렬천이가 없는 경우의 무직류 터보부호의 부호기

그리고 그림 8은 그림 7에서 사용 가능한 길쌈부호기의 한 예를 나타낸 것이다.

이러한 무직류 터보부호들의 복호기는 부호기의 설계방법에 따라 여러 가지로 구현될 수 있다. 그림 4, 5의 경우에는 부호화되는 정보비트에 대해서만 MAP 복호 알고리즘을 사용하여 반복복호를 하고 마지막 복호단에서 그 결과를 이용하여 그렇지 않은 비트와 함께 MAP 알고리즘을 사용하는 부분집합복호

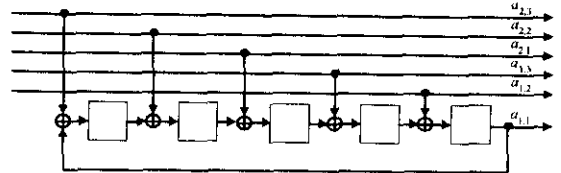


그림 8. 병렬천이가 발생하지 않는 경우의 길쌈부호기

(subset decoding) 방법[7]이 복호 복잡도 면에서 더 유리하다. 그러나 그림 6의 경우에는 사상기에 의하여 모든 정보비트들이 부호화되므로 모든 입력비트에 대하여 MAP 복호가 수행되어야 한다. 물론 그림 7, 8의 경우에도 모든 입력비트에 대하여 MAP 복호가 수행되어야 한다.

4. 모의실험 결과 및 고찰

무직류 터보부호에서 복호기의 복잡도는 복호기에서 고려되는 심벌의 개수와 부호기의 메모리의 상태 수에 비례한다. 그림 4, 5의 경우에는 부호화되는 2개의 정보비트에 대해서만 MAP 복호를 수행하게 되므로 복호기에서 고려되는 심벌의 수는 2^2 개이다. 그러나 그림 6과 그림 7, 8의 경우에는 복호기에서 고려되는 심벌의 수는 2^3 개이다. 또한, 그림 4, 5와 6의 경우에는 부호기의 상태 수가 2^3 개이고 그림 7, 8의 경우에는 2^2 개이다. 따라서 그림 4, 5에 해당하는 터보부호의 복호기는 마지막 복호단에서의 MAP 복호에 의한 복잡도를 고려하지 않으면 그림 6에 해당하는 터보부호의 복호기에 비해 약 8배 정도, 그리고 그림 7, 8에 해당하는 터보부호의 복호기에 대해서는 대략 32배정도 감소된 복잡도를 가진다. 그러나 오류성능은 복잡도에 반비례함을 예상할 수 있으며 본 논문에서는 모의 실험을 통하여 그 결과를 확인하였다.

그림 9는 그림 6에 사상기를 사용하지 않는 경우에 해당하는 터보부호에 대하여 부분집합복호를 사용하고 인터리버의 크기가 5000인 경우의 성능을 나타낸 것이다.

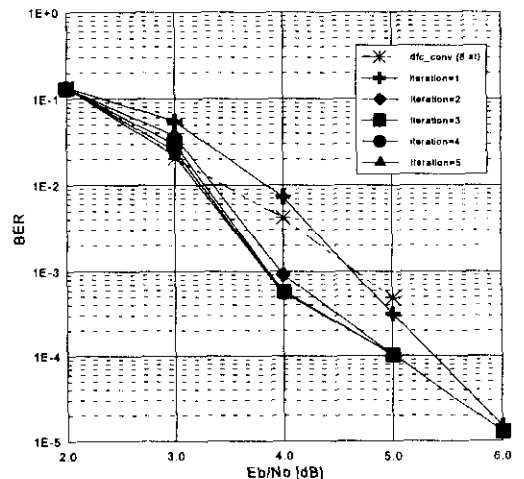


그림 9. 부분집합복호를 하는 경우의 무직류 터보부호의 성능

그림 9에서 점선으로 표시된 무직류 격자형부호 C(RM(2,3)/RM(0,3);E)에 비해 무직류 터보부호가 약 1dB 더 우수한 성능을 보인다. 그러나 부분집합복호 시에는 반복복호횟수를 증가시키더라도 오류성능이 더 이상 향상되지 않음을 알 수 있다. 이는 부호화되지 않은 입력비트들의 성능이 반복횟수에 상관없이 포화되어 전체 입력비트들의 오류성능이 제한되기 때문이다. 그림 10은 그림 6에 해당하는 무직류 터보부호의 성능을 나타낸 것이다.

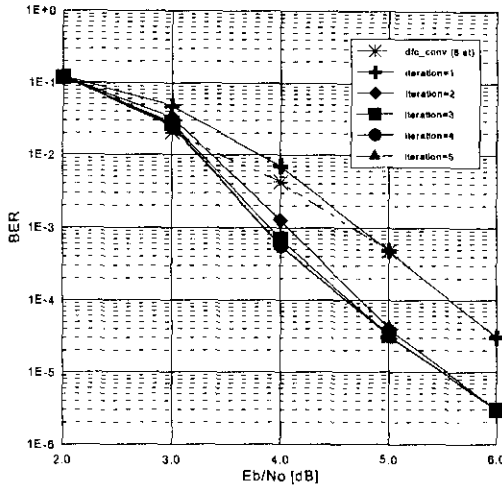


그림 10. 사상기를 사용하는 경우의 무직류 터보부호의 성능

그림 10을 그림 9와 비교하면 사상기를 포함하는 무직류 터보부호가 4dB정도까지는 사상기를 포함하지 않는 무직류 터보부호와 거의 비슷한 성능을 가지지만 5dB이상에서는 좀 더 좋은 성능을 가진다. 그러나 두 경우 모두 반복복호의 효과가 3회 이상부터는 미미함을 알 수 있다. 그림 11은 그림 7,8에 해당하는 무직류 터보부호의 오류성능을 나타낸 것이다.

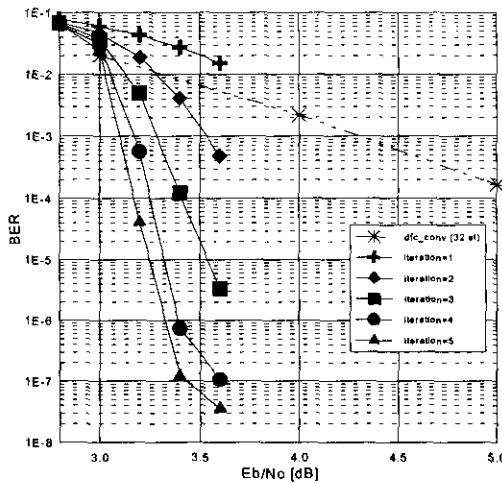


그림 11. 병렬천이 없는 경우의 무직류 터보부호의 성능

여기서 병렬천이를 발생시키지 않는 무직류 격자형부호 C(RM(2,3)/RM(1,3);E)를 구성부호로 가지는 터보부호는 무직류 격자형부호에 비해 매우 큰 부호화 이득을 가짐을 알 수 있다. 또한, 반복회수에 따라 그 차이는 훨씬 더 증가한다.

5. 결론

본 논문에서는 무직류 격자형부호를 구성부호로 가지는 터보부호를 구성하였다. 여기서 구성된 무직류 터보부호의 경우 부호화 과정이 조건표 대신 생성행렬에 의하여 이루어 지므로 부호화가 간단함을 알 수 있었다. 무직류 터보부호는 병렬천이를 발생시키는 구성부호를 가지도록 구성될 수 있으며 또한 충분한 메모리가 사용될 수 있는 경우 병렬천이가 발생하지 않도록 구성될 수도 있다. 각 경우에 대하여 모의 실험을 수행한 결과 모두 무직류 격자형부호에 비해 더 우수한 성능을 보였다. 그리고 후자의 경우는 전자의 경우에 비해 다소 큰 복호 복잡도를 가지지만 이로 인해 매우 우수한 성능을 가짐을 확인할 수 있었다.

참고 문헌

- [1] G.L. Pierobon, "Codes for zero spectral density at zero frequency," *IEEE Trans. Inform. Theory*, vol. IT-30, pp. 435-439, Mar, 1984.
- [2] C. Berrou and A. Glavieux, "Near optimum error correcting coding and decoding: Turbo-codes," *IEEE Trans. Commun.*, vol. 44, no. 10, pp. 1261-1271, Oct. 1996.
- [3] C.K. Jeong and E.K. Joo, "Design of DC-free trellis codes based on the partition chain of Reed-Muller codes," *Proc. APCC'97*, Sydney, vol. 1, pp. 528-532, Dec. 1997.
- [4] 주연경, "디지털 데이터 저장 및 전송을 위한 효율적인 부호기술," 정보통신부 최종 연구개발결과보고서, 1998. 7.
- [5] C.K. Jeong and E.K. Joo, "DC-free trellis codes based on partition chain of Reed-Muller codes," *IEEE Commun. Letters*, vol. 2, no. 8, pp. 232-234, Aug. 1998.
- [6] P. Robertson and T. Woerz, "A novel bandwidth efficient coding scheme employing turbo codes," *Proc. IEEE ICC'96*, Dallas, vol. 2, pp. 962-967, June 1996.
- [7] P. Robertson and T. Woerz, "Extensions of turbo trellis coded modulation to high bandwidth efficiencies," *Proc. IEEE ICC'97*, Montreal, vol. 3, pp. 1251-1255, June 1997.
- [8] W.J. Blackert, "Implementation issues of turbo trellis coded modulation," *MS Thesis*, University of Virginia, May 1996.