

Iteration 부호의 가변반복복호 제어기법

백승재* · 이성우* · 박진수*

*청주대학교

Variable Iteration Decoding Control Method for Iteration Codes

Seung-jae Baek* · Sung-woo Lee* · Jin-soo Park*

*Chongju University

E-mail : bsj3386@empal.com

요 약

본 논문에서는 반복복호 부호의 복호과정시 채널 적응적으로 반복복호수를 가변할 수 있는 효율적인 제어기법을 제안한다. 반복복호 부호는 반복구조를 가지며 그 특성상 반복복호수가 증가할 수록 성능이 우수하게 향상된다. 그러나 반복복호수가 증가할 수록 복호과정시 적용된 알고리즘의 복잡도에 따라 다소 차이는 있지만 공통적으로 계산량의 증가를 가지게 되며 이는 복호지연시간 증가로 나타난다. 또한 일정 반복복호수 이상에 도달하게 되면 그 성능 변화가 거의 없는 오류마루(floor) 현상이 나타난다. 즉 성능변화가 없는 적절한 반복복호수 종료점을 찾아야 한다. 따라서 본 논문에서는 프레임 주기로 수신된 정보를 프레임 에러체크 지시자를 이용하여 채널의 변화를 감시하며 반복복호 부호의 반복복호수를 채널 적응적으로 증가, 감소할 수 있도록 제어하는 기법을 제안하였으며 결과적으로 반복구조를 가지는 부호의 방대한 계산량 감소와 이로인한 복호지연 시간을 성능저하없이 효율적으로 단축시킬 수 있음을 보였다.

ABSTRACT

In this Paper, We propose an efficient iteration decoding control method with variable iteration decoding for iteration codes decoding. As the number of iterations increases, the bit error rate and frame error rate of the decoder decrease and the incremental improvement gradually diminishes. However, as the iteration decoding number is increase, it require much delay and amount of processing for decoding. Thus we propose variable iteration control method to adapt variation of channel using Frame Error-Check indicator. Therefore, the CRC method requires the fewest iterations and less computation than the CE method and the SCR methods.

키워드

터보코드, Iteration codes, 프레임에러체크, CRC

1. 서 론

터보코드는 1993년에 Berrou et.al에 의하여 발표되었다[가]. 그 성능의 우수함에 의문을 제기하기도 하였으나 그 우수한 성능은 많은 연구가들에 의하여 입증되었다[나][다]. 그 결과 차세대 이동통신인 IMT2000의 표준화를 주도하고 있는 3GPP (3rd Generation Partnership Project)와 3GPP2의 표준으로 채택되었다. 터보코드는 RSC (recursive systematic convolution)부호를 병렬로 연결하여 부호화하며 반복복호를 통하여 복호동작을 수행하게 된다. 터보코드는 인터리버의 크기가 충분히 크고, 반복복호수가 증가함에 따라 사

논한계에 근접하는 성능을 보인다. 그러나 터보코드는 사용되어지는 복호알고리즘에 따라 다소 차이는 있지만 반복복호의 증가로 인하여 야기되는 방대한 계산량을 수행해야 하는 문제점을 가지고 있다.

본 논문에서는 반복복호에 따른 연산량 증가와 복호지연 관점에서 이를 해결하기 위한 방법으로 CRC(cyclic Redundancy Check)부호를 이용하여 적절한 반복복호 종료시점을 제시하여 복호가 멈추는 정지기준 구조를 제안한다.

먼저 II장에서는 터보부호의 구조를 간단하게

설명하고, 기존에 제안된 적절한 반복복호 정지기준 구조를 분석한다. III장에서는 제안한 구조의 개선된 성능을 컴퓨터 시뮬레이션을 통하여 기존의 구조들과 비교 분석한다. 마지막으로 IV장에서는 본 논문의 결론을 맺는다.

II. Iteration 부·복호기

Turbo code의 부호기는 두 개의 RSC와 인터리버의 결합으로 구성되어있다. 그림1에서 부호율이 1/2, 구성장의 길이가 K인 길쌈 부호기의 k번째 부호기 입력이 u_k 비트 일 때, 출력 $c_k^{(1)}$, $c_k^{(2)}$, $\hat{c}_k^{(2)}$ 를 출력한다.

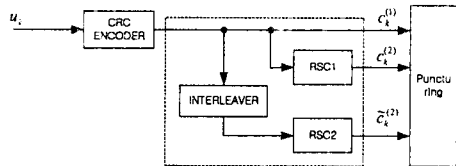


그림 1. 터보코드 부호기

그림 1에서의 구성부호인 두 개의 RSC부호는 조직 길쌈부호의 궤환이 이루어진 형태라고 할 수 있는데 그 실제적인 예로 그림 2와 같은 형태로 구성할 수 있으며 생성다항식의 변형으로 다양하게 구성할 수 있다.

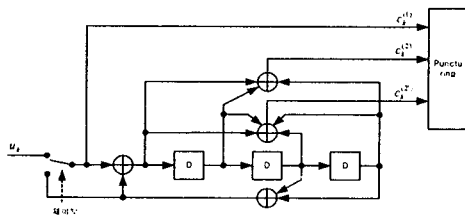


그림 2. RSC(Recursive Systematic Code)부호기

본 논문에서는 그림 1에서 보는 바와 같이 입력비트 u_k 가 CRC부호기를 통과하여 터보코드에 입력되도록 하였다. CRC 부호는 실제적으로 짧은 사이클릭 부호로 일반적으로는 사이클릭이 아니다. 그러나 사이클릭 부호로 분류되었기에 그 명칭을 사용하게 된 것이다. 간단한 부호기와 복호기 구조를 가지고 연접성 오류검출 능력을 가지는 CRC 부호는 매우 보편화된 에러검출 부호이다. 원래의 사이클릭 부호의 쉬프트 레지스터 신드롬 계산회로는 CRC부호의 부호기 및 복호기로 사용되어 진다.

예로 생성다항식 $g(x) = x^{16} + x^{15} + x^2 + 1$ 을

가지며 조직적 부호기의 쉬프트 레지스터 회로는 다음 그림 3과 같이 표현된다.

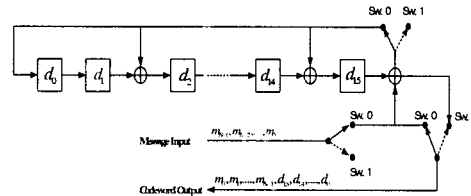


그림 3. $g(x) = x^{16} + x^{15} + x^2 + 1$ CRC 부/복호기

CRC 부호기를 통과한 정보비트와 CRC 용장은 프레임의 구성하게 되고 이 프레임은 터보코드의 복호과정에서 반복복호 정지기준을 제공하는 기술적 방법을 이루는데 이용된다. 즉 반복복호 과정에서 CRC의 프레임 체크 비트의 값이 '0'이 되는 시점에서 복호를 멈추게 하는 것이다.

그림 4는 터보코드의 복호기 구조로서 복호기의 결정단에 CRC복호기를 추가하여 CRC 프레임을 체크하여 반복복호를 제어하게 된다. 프레임 체크 값이 '0'이 되는 시점에서 복호가 종료되고 제어신호에 의하여 추정된 복호출력을 제공하게 된다.

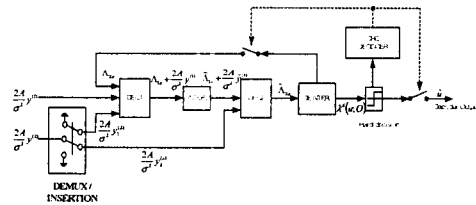


그림 4. 터보코드의 복호기

III. 제안한 반복복호수 제어기법

반복복호의 수가 증가할수록 복호기는 점차적으로 비트에러율(BER)과 프레임에러율(FER)을 감소시키며 개선된 성능을 나타낸다. 각 프레임들은 선정된 고정 프레임수 T_max동안 반복복호 되어 복호된다. 대개 T_max는 임의로 가장 훼손된 프레임의 기준으로 하여 설정한다. 대부분의 프레임들은 적은 수의 반복복호로 복호된다. 따라서 터보복호하는데 요구되는 처리량은 각 프레임들이 설정된 반복복호 내에서 비트들이 정확하게 추정되어 즉각적으로 종료된다면 성능저하 없이 감소될 수 있다. 하지만 무선통신에서는 전송된 정보가 알 수 없는 확률통계적 문제이기 때문에 고정 반복복호를 수행한다는 것은 현실성이 없다. 반복복호를 종료하는 시점을 조절하는 몇 가지 구조들이 제안되어졌다[라][마]. 비교를 목적으로 기존

의 두 가지 구조들과 본 논문에서 제안한 구조를 살펴보자. 시뮬레이션 결과는 기존의 구조들과 비교하였다. 이미 알려진 두 가지 정지구조를 검토해보자. 최대 T_max의 반복복호를 수행하는 동적인 정지구조일지라도 그 T_max 반복복호 내에서조차도 그 성능이 만족스럽지 않다.

3-1 Cross Entropy(CE)

반복복호후에 CE기술은 구성복호기간의 근사 Cross Entropy 를 계산한다. 여기서

$$T(\tau) = \sum_k \frac{|\Lambda_{2e}^{\tau}(u_k) - \Lambda_{2e}^{\tau-1}(u_k)|^2}{\exp(|\lambda_{1k}^{A(\tau)}(u; O)|)}$$

$\Lambda_{2e}^{\tau}(u_k)$ 는 반복복호 τ 시점에서 두 번째 구성 복호기가 산출하는 정보비트 u_k 의 외부정보이다. 그리고 $\lambda_{1k}^{A(\tau)}(u; O)$ 는 첫 번째 구성 복호기에 의해 산출되는 완전한 정보비트 u_k 이다. 반복 복호 동작의 정지시점은 $T(\tau) < (10^{-2} \sim 10^{-4}) * T(1)$ 이다.

3-2 Sign Change Ratio (SCR)

SCR정지기준 구조는 CE방법과 관련이 있다. 이 방법은 반복복호 ($\tau-1$)시점 과 τ 시점간에 외부정보의 부호전환의 의 횟수 $C(\tau)$ 를 계산하는 것이다. 복호는 $C(\tau) \leq q_c N$ 을 만족할 때 정지한다.

여기서 q_c 는 상수로서 대개 $0.005 \leq q_c \leq 0.03$ 이고 N은 프레임 크기이다. 두 개의 이상적인 리커시브 시스템에 킨블루션 부호기로 구성된 터보코드를 고려해보자. 정보비트는 u_k 로 기호화하고 BPSK변조 한다. 잡음평균이 0이고, 분산이 σ^2 인 AWGN채널로 전송한다. 수신기에서는 정보비트 u_k 에 대응하는 $y_{1k}^{(1)}, y_{1k}^{(2)}, y_{2k}^{(2)}$ 신호들이 입력된다. $y_{1k}^{(1)}$ 는 시스템에 킨 신호이고, $y_{1k}^{(2)}, y_{2k}^{(2)}$ 는 RSC1과 RSC2가 각각 산출한 패리티 신호이다. 이 신호들은 추정치 u_k 를 산출하기 위해 복호기 모듈인 DEC1과 DEC2로 각각 입력으로 들어간다. 복호기의 입력 $\lambda_{1k}(c^{(1)}; I) = (2A/\sigma^2) y_{1k}^{(1)}$ 는 u_k 에 대응되는 측정된 정보의 LLR표기이다. τ 번째 반복복호에서 $\Lambda_{1a}^{\tau}(u_k)$ 와 $\Lambda_{1e}^{\tau}(u_k)$ 는 DEC1에서 각각 사전정보와 외부정보이다. 그리고 $\Lambda_{2a}^{\tau}(u_k)$ 와 $\Lambda_{2e}^{\tau}(u_k)$ 는 DEC2의 사전정보와 외부정보이다. DEC1과 DEC2에서 완성된 정보는 $\lambda_{1k}^{A(\tau)}(u; O)$ 와 $\lambda_{2k}^{A(\tau)}(u; O)$ 으로 각각 출력된다.

$$\lambda_{1k}^{A(\tau)}(u; O) = \lambda_{1k}(c^{(1)}; I) + \Lambda_{1a}^{\tau}(u_k) + \Lambda_{1e}^{\tau}(u_k)$$

$$\lambda_{2k}^{A(\tau)}(u; O) = \lambda_{1a(\alpha)}(c^{(1)}; I) + \Lambda_{2a}^{\tau}(u_k) + \Lambda_{2e}^{\tau}(u_k) \dots \dots \dots (1)$$

반복처리는 다음과 같이 할당한다.

$$\Lambda_{1a}^{\tau}(u_{\alpha(k)}) = \Lambda_{2e}^{\tau-1}(u_k) \dots \dots \dots (2)$$

$$\Lambda_{2a}^{\tau}(u_k) = \Lambda_{1e}^{\tau}(u_{\alpha(k)})$$

여기서 $\alpha(k)$ 는 인터리버 매핑을 의미한다. (1) 식에서 두 개의 완전한 정보를 표현하기 위해 $j=1,2$ 조건으로 놓는다. j번째 복호기에서 산출되는 완전한 정보는 세 개의 추정치들로 구성된다. 채널로부터 직접 전달된 $\lambda_{1k}(c^{(1)}; I)$, 사전값 $\Lambda_{1a}^{\tau}(u_k)$ 와 트렐리스 구조를 기초로하여 얻어진 $\Lambda_{1e}^{\tau}(u_k)$ 들로 구성된다. 세 가지의 추정치 중 $\lambda_{1k}(c^{(1)}; I)$ 값은 매 반복복호마다 고정값이며 $\Lambda_{1a}^{\tau}(u_k)$ 값 $\Lambda_{1e}^{\tau}(u_k)$ 값은 반복복호를 할 때마다 계속적으로 업데이트 된다. 외부정보가 다음 복호기의 사전정보 값으로 사용된다면 (2)식에 명시된 것처럼 할당하며 $\tau \geq 2$ 면 $\Lambda_{1a}^{\tau}(u_k)$ 와 $\Lambda_{1e}^{\tau}(u_k)$ 는 상호관계가 있다. 그림 4에서 터보복호기의 구조를 고려해보자.

3-3 제안한 CRC 구조

그림 1에서 CRC엔코더는 각 정보프레임의 후미에 N_crc 비트 길이를 부가한다. 이렇게 확장된 프레임은 터보엔코더로 보내진다. 복호기에서는 각 반복복호마다 복호기는 경판정을 실시하고 CRC비트들은 프레임에러를 검사하기 위해 이용된다. CRC검출에 에러가 없을 때 반복복호는 종료하게 되는 것이다. 각 반복복호에서 CRC방법은 LLRs 계산과정을 거치고 $\lambda_{1k}^{A(\tau)}(u; O)$ 의 부호를 기반으로 하는 경판정을 한다. 그리고 마지막으로 경판정을 이용한 CRC복호가 이루어진다. 대개 CRC복호시 적어도 3N번의 이진 덧셈이 이루어진다. CE방법은 6N-1번의 실수연산이 요구된다. ($\lambda_{1k}^{A(\tau)}(u; O)$ 를 얻기위한 N번 덧셈연산을 포함하여 그리고 저장을 위한 (N+2) 크기의 실수 메모리 유닛이 필요하다. SCR방법은 오직 N번의 이진 덧셈이 이루어지며 N보다 큰 카운터는 필요없다.

표 1. 정지기준의 계산 복잡도 비교

Criteria	Extra Bits per frame	Non-binary	Binary Additions	Storage
CRC	4~32	N	>3N	0
CE	0	6N-1	0	N+2 Non-binary numbers
SCR	0	N	N	N bits

IV. 시뮬레이션 결과

시뮬레이션 조건들은 그림 5와 같은 환경 모델에서 실시하였다. 또한 그림 6에서 제시된 프로그램 흐름도는 다양한 결과의 도출을 위하여 입력을 다양하게 설정하였다. 입력조건들은 E_b/N_0 , 구성부호, 인터리버크기, 반복복호 횟수, 평서링의 유무, 유한(finit) 프레임 에러들을 입력하여, 출력 결과는 유한 프레임 에러가 발생하였을 경우 시뮬레이션을 중지하고 비트 에러율을 계산하여 출력한다.

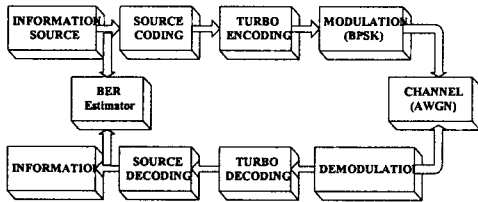


그림 5. 시뮬레이션 환경 모델

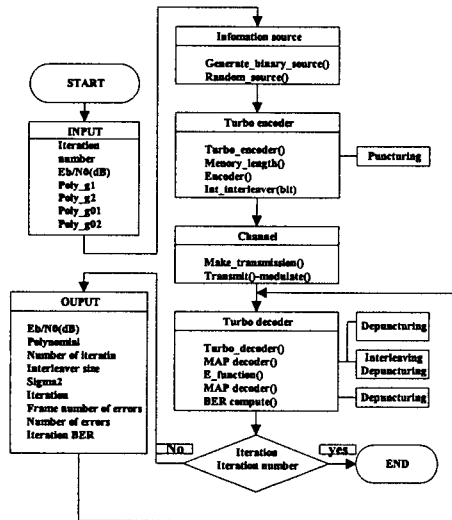


그림 6. 시뮬레이션 프로그램 흐름도

시뮬레이션은 구성부호 $g(D) = [1, 1 + D + D^2 / 1 + D^2 + D^3]$ 인 경우로 수행하였다. 시뮬레이션 비교는 5가지 종류구조로 Fixed, CRC, CE, SCR, Limit로 명명하여 실험하였다. Limit의 경우는 정보비트를 알고 있고 프레임이 올바르게 복호되었을 때 즉시 종료된다. 즉 모든 구조의 제한이라고 볼 수 있다. CRC발생기의 다항식은 $g(x) = x^{16} + x^{15} + x^2 + 1$ 을 이용하였다. 그림 7은 E_b/N_0 대 BER과 FER을 비교하였다. CRC방법이 모든 구조의 제한이라고 할 수 있는 Limit와 비슷한 성능

을 보임을 알 수 있다. 그림 8은 E_b/N_0 대 평균 반복복호수를 도기한 것으로 CRC정지기준 구조가 가장 적은 반복복호수로 복호되어짐을 관찰할 수 있다.

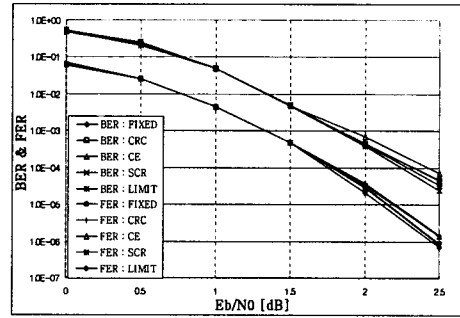


그림 7. E_b/N_0 대 BER과FER 비교

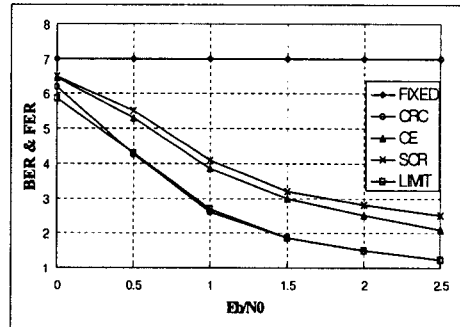


그림 8. E_b/N_0 대 평균반복복수 비교

V. 결 론

반복부호구조를 가지는 부호는 BER관점에서 본다면 반복복호가 증가할 수록 그 성능은 점차적으로 증가한다. 하지만 그 복호수 증가는 방대한 연산량 증가와 실시간 설계에 부적합하다. 이러한 문제점을 해결하기 위해 본 논문에서는 CRC부호를 이용하여 효과적인 반복복호 정지시점을 제시할 수 있는 구조를 제안하였다. 결과적으로 기존의 다른 구조에 비하여 BER 성능 감소 없이 평균반복복호 횟수가 다른 정지기준보다 적은 횟수로 반복복호 되었다.

결론적으로 본 논문에서 제시한 구조는 반복복호를 가지는 부호에서 그 반복복호 정지기준을 제공하기 위한 구조로 효과적임을 증명하였다.

참고문헌

- [1] Berrou, A. Glavieux, P. Thitimajshima. "Near Shannon Limit Error-Correcting Coding and Decoding : Turbo Codes," in *ICC*, pp. 1064-1070, 1993.
- [2] S. Dolinar and D. Divsalar, "Weight Distributions for Turbo Codes Using Random and Nonrandom Permutations" , *The JPL TDA Progress Report 42-122*, Aug. 15, 1995.
- [3] Benedetto, S. and Montorsi, G., "Unveiling turbo codes: some results on parallel concatenated coding schemes," *IEEE Trans. on Info. Theory*, vol. 42, no. 2, pp. 409-429, Mar. 1996.
- [4] J. Hagenauer, E. Offer and Lutz Papke, " Iterative Decoding of Binary Block and Convolutional Codes," *IEEE Transactions on Information Theory*, vol.42. no.2, March 1996.
- [5] Rose Y. Shao, Shu Lin, and Marc P.C Fossorier, "Two Simple Stopping Criteria For TurboDecoding," *IEEE Transaction on Comm. Vol. 47, No. 8*, pp 1117-1120, August 1999.