

CRC부호를 이용한 반복복호부호의 반복복호 제어기법

백 승 재[†] · 박 진 수^{††}

요 약

본 논문에서는 반복복호부호의 복호과정시 CRC(Cyclic Redundancy Check)검사를 이용하여 반복복호수가 가변하는 효율적인 제어기법을 제안한다. 반복복호 부호는 반복구조를 가지며 그 특성상 반복복호수가 증가할수록 BER/FER성능이 우수하게 향상된다. 그러나 반복복호수가 증가할수록 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(error floor)현상이 나타난다. 즉 성능변화가 없는 적절한 반복복호수 종료점을 찾아야 한다. 따라서 본 논문에서는 프레임 주기로 수신된 정보를 프레임 오류검사 지시자(FCS : Frame Check Sequence Indicator)를 이용하여 채널의 변화를 감시하며 반복복호 부호의 반복복호 횟수를 채널 적응적으로 증가, 감소할 수 있도록 제어하는 기법을 제안하여 결과적으로 반복구조를 가지는 부호의 방대한 계산량 감소와 이로 인한 복호지연 시간을 성능저하 없이 효율적으로 단축시킬 수 있음을 확인하였다.

Variable Iteration Decoding Control Method of Iteration Codes using CRC-code

Seung-Jae Baek[†] · Jin-Soo Park^{††}

ABSTRACT

In this Paper, We propose an efficient iteration decoding control method with variable iteration decoding of iteration codes decoding using Cyclic Redundancy Check. 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, when the iteration decoding number is increased, it require much delay and amount of processing time for decoding. Also, It can be observed the error floor that the performance cannot be improved even though increasing of the number of iterations and SNR. So, Suitable number of iterations for stopping criterion is required. we propose variable iteration control method to adapt variation of channel using Frame Error-Check indicator. Therefore, the amount of computation and the number of iterations required for iteration decoding with CRC method can be reduced without sacrificing performance.

키워드 : 부호이론, 반복복호코드(Iteration Codes), 터보코드(Turbo Code), 프레임오류검사(Frame Check Sequence), CRC

1. 서 론

최근 채널부호기들은 다양한 구조들로 발표되고 있으며 그 중에서도 수신된 심볼을 반복적으로 복호하여 반복복호 횟수가 증가할수록 전송데이터에 근접하는 심볼로 복호하는 반복복호(Iteration)구조가 큰 줄기를 이루고 있다. 이런 케환(feedback)구조를 가지는 부호로는 병렬연접컨볼루션부호(PCCC : Parallel Concatenated Convolutional Code)가 대표적이다. PCCC는 발표 당시 '터보코드(Turbo Code)'라고 명명되었으며 그 이론은 1993년에 Berrou et.al에 의하여 발표되었다[1]. 그 성능의 우수함에 의문을 제기하기도 하

였으나 그 우수한 성능은 많은 연구가들의 노력으로 입증되었다[2,3]. 그 결과 차세대 이동통신인 IMT2000의 표준화를 주도하고 있는 단체인 3GPP(3rd Generation Partnership Project)와 3GPP2(3rd Generation Partnership Project 2)의 표준으로 채택되었다.

터보코드는 RSC(recursive systematic convolution) 부호를 병렬로 연접하여 입력된 정보를 부호화하며 반복복호를 통하여 복호동작을 수행할 뿐만 아니라 인터리버의 크기가 충분히 크고, 반복복호수가 증가함에 따라 샤논한계(shannon limit)에 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

* 본 연구는 과학기술부·한국과학재단 지정 청주대학교 정보통신연구센터의 지원에 의한 것입니다.

† 준 회 원 : 청주기능대학 정보통신시스템과 IT교수

†† 종 신 회 원 : 청주대학교 정보통신공학부 교수

논문접수 : 2004년 2월 18일, 심사완료 : 2004년 4월 27일

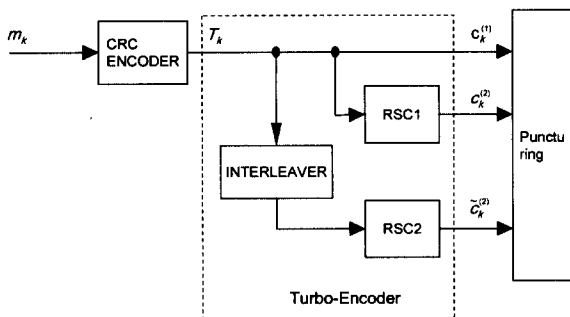
관점에서 이를 해결하기 위한 방법으로 CRC(cyclic Redundancy Check)부호를 이용하여 적절한 반복복호 종료시점을 결정하여 복호가 멈추는 정지기준(stop criterion) 구조를 제안한다. 기존에 제시된 구조는 터보코드의 복호기 입력전에 미리 정해진 반복 복호수에 따라 복호를 수행한다. 실제적으로 채널구조는 가변적이며 반복 복호수를 어느 정도 진행하면 그 이상의 반복 복호수 이상에서는 부호이득이 매우 적게 되며, 임의의 반복 복호수에서는 이미 요구하는 성능을 얻었다 하더라도 고정으로 정해진 반복복호 수만큼 복호를 수행함으로써 계산량과 지연이 증가하게 되는 문제점이 야기된다. 이러한 고정 반복 복호수가 아닌 가변적으로 반복 복호를 멈추는 정지기준 구조가 발표되었다[5]. 기존의 가변적 정지 기준구조는 복호기의 출력단에서 외부정보나 LLR값과 같은 기준값을 사용하여 반복복호를 멈추는 기준으로 이용하고 있다. 그러나 이러한 구조는 한 프레임에서 반복복호 때마다 기준값을 측정해야 하는 계산상의 복잡도를 가진다. 따라서 본 논문에서는 이러한 문제점을 해결하는 효율적인 기법을 제시하고 기존 구조와 비교, 분석한다.

먼저 2장에서는 터보부호의 구조를 간단하게 설명하고, 기존에 제안된 적절한 반복복호 정지기준 구조를 분석한다. 3장에서는 제안한 구조의 개선된 성능을 컴퓨터 모의실험을 통하여 기존의 구조들과 비교 분석한다. 마지막으로 4장에서는 본 논문의 결론을 맺는다.

2. 반복복호부호의 부·복호기

2.1 터보코드 부호기

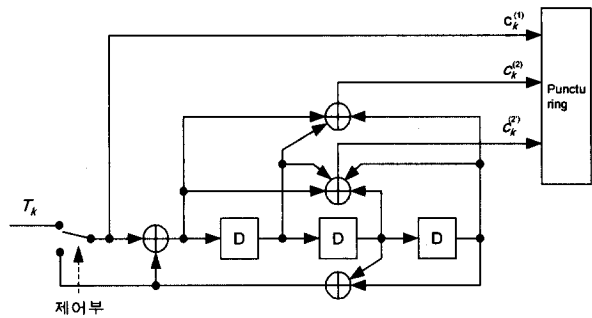
터보코드의 부호기는 두 개의 RSC와 인터리버의 결합으로 구성된다. (그림 1)은 부호율이 1/2, 구속장의 길이가 K인 길쌈 부호기로서 k번째 부호기 입력이 u_k 비트 일 때, 출력부호계열 $c_k^{(1)}, c_k^{(2)}, \tilde{c}_k^{(2)}$ 를 출력한다. (그림 1)은 터보코드 부호기구조로서 구성부호로 사용되는 두 개의 RSC부호는 조직 길쌈부호(systematic convolutional code)의 계



(그림 1) 터보코드 부호기

환(recursive)이 이루어진 형태이며 정보신호인 입력계열(input sequence) m_k 가 CRC부호기를 통과하여 터보부호기의 입력계열이 된다.

(그림 2)는 RSC의 실제적인 내부구조이며 부호어(code-word)를 구성하는 생성다항식(Generate Polynomial)의 변형으로 다양하게 구성할 수 있다.



(그림 2) RSC(Recursive Systematic Code) 부호기

2.2 CRC 부호기

CRC부호는 시리얼 전송에서 데이터의 신뢰성을 검증하기 위한 오류 검출 방법의 일종이다. 간단한 오류 검출 방법으로는 잉여(parity)비트에 의한 방법과 체크-섬(check-sum)에 의한 오류 검출 방법이 있지만 잉여비트에 의한 방법은 한 바이트에서 +1, 다른 바이트에서는 -1로 오류가 생기는 경우만 해도 오류는 검출되지 않는다. 즉 이들 방법으로는 오류를 검출해 낼 수 있는 확률이 대단히 낮다. CRC에 의한 방법은 높은 신뢰도를 확보하며 오류 검출을 위한 오버헤드가 적고, 랜덤오류나 연접오류를 포함한 오류 검출에 매우 좋은 성능을 갖는 것을 특징으로 한다. CRC 부호는 실제로 짧은 사이클릭 부호로 분류되어지고 있으나 일반적인 관점으로는 사이클릭 부호로 볼 수 없다. 그러나 사이클릭 부호로 분류되었기에 그 명칭을 사용하게 된 것이다. 간단한 부호기와 복호기 구조를 가지고 연접성 오류 검출 능력을 가지는 CRC 부호는 매우 보편화된 오류검출 부호이다. 원래의 사이클릭 부호의 쉬프트 레지스터 신드롬 계산회로는 CRC부호의 부호기 및 복호기로 사용되어 진다. <표 1>은 보편적으로 사용되어지고 있는 CRC부호들을 나타내었으며 다항식 차수(order)가 높을수록 정확한 오류검출이 필요한 경우에 사용된다.

CRC의 기본이론은 n비트 길이의 주어진 정보가 있을때, 이를 k비트 길이만큼 자리올림하고 미리 약속한 k비트의 CRC생성다항식으로 나누면 r비트의 나머지가 남게 된다. 송신측에서는 원래의 정보비트를 k비트 자리 올린 것에 r비트의 나머지를 더해 n + r비트의 데이터를 만들어 보낸다. 수신측에서는 수신된 n + r비트의 데이터를 미리 약속된 CRC생성다항식으로 나누어서 나머지가 정확히 '0'이

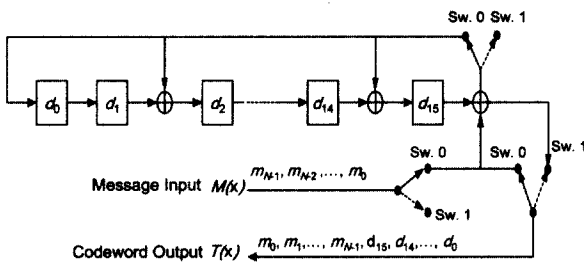
되는지를 검사하면 된다. 이 때 '0'이 아닌 수가 되면, 전송식에 에러가 발생한 것이다.

〈표 1〉 보편적인 CRC부호 종류

명 칭	생 성 다 항 식
CRC-8	$x^8 + x^5 + x^4 + 1$
CRC-12	$x^{12} + x^{11} + x^3 + x^2 + x + 1$
CRC-16	$x^{16} + x^{15} + x^2 + 1$
CRC-ITU-T	$x^{16} + x^{12} + x^5 + 1$
CRC-32	$x^{32} + x^{26} + x^{23} + x^{22} + x^{16} + x^{12} + x^{11} + x^{10} + x^8 + x^7 + x^5 + x^4 + x^2 + x + 1$

이 때 k가 16비트이면 CRC-16, 32비트이면 CRC-32가 되고 키 값으로는 수학적으로 정해진 값을 사용한다. CRC-16에서는 0x8005, CRC-32에서는 0x04c11db7이 주로 사용된다. 그리고 r비트의 나머지를 CRC에서 가장 중요한 Frame Check Sequence(FCS)라고 한다.

예로 (그림 3)은 생성다항식 $g(x) = x^{16} + x^{15} + x^2 + 1$ 을 가지는 CRC부호이며 조직적 부호기의 쉬프트 레지스터 회로 형태로 표현된다.



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

식 (1)은 n비트의 입력데이터(Message Input)열에 대한 정보다항식의 표현이다.

$$M(x) = m_n x^{n-1} + \dots + m_2 x^1 + m_1 \quad (1)$$

식 (2)는 CRC-16 생성을 위한 CRC생성다항식의 표현이다.

$$g(x) = x^{16} + x^{15} + x^2 + 1 \quad (2)$$

몫은 $Q(x)$, 나머지는 $R(x)$, 전송데이터는 $T(x)$ 로 표기한다.

입력데이터 $M(x)$ 를 k비트 만큼 자리를 올리고 $g(x)$ 로 나누면 식 (3)과 같이 쓸 수 있다.

$$x^k M(x) = Q(x) * g(x) + R(x) \quad (3)$$

여기서

$$R(x) = d_k x^{k-1} + \dots + d_2 x^1 + d_1 \quad (4)$$

($R(x)$ 의 최고 차수는 k-1)이고 비트열 $d_k \dots d_2 d_1$ 는 나머지로써 FCS가 된다.

식 (3)에서 $x^k M(x) + R(x)$ 는 $g(x)$ 의 배수이므로 $g(x)$ 로 나누면 '0'임을 알 수 있다. 식 (5)와 같이 쓸 수 있다.

$$x^k M(x) + R(x) = Q(x) * g(x) \quad (5)$$

결과적으로 전송되는 데이터 비트열은 $m_n \dots m_3 m_2 m_1 d_k \dots d_2 d_1$ 가 된다.

따라서 전송다항식은 식 (6)과 같이 쓸 수 있다.

$$T(x) = x^k M(x) + R(x) \quad (6)$$

식 (7)은 수신측에서 수신된 전송다항식을 생성다항식 $g(x)$ 로 나누어 '0'을 만족하면 오류가 발생하지 않았음을 나타낸다.

$$T(x)/g(x) = x^k M(x)/g(x) + R(x)/g(x) = 0 \quad (7)$$

CRC 부호기를 통과한 정보비트와 CRC 용장(Redundancy)은 종합적으로 프레임을 구성하게 되고 이 프레임은 터보코드의 복호과정에서 반복복호 정지기준을 제공하는 기술적 방법을 이루는데 이용된다. 즉 반복복호 과정에서 CRC의 FCS 값이 정수 '0'이 되면 복호과정을 정지시키는 제어신호를 발생하여 정지시점을 제공하게 되는 것이다.

2.3 터보코드 복호기

(그림 4)는 수신된 심볼을 복호하는 터보코드의 복호기 구조로서 두 개의 이상적인 RSC로 구성된 터보코드 부호기의 정보비트는 u_k 로 기호화하고 BPSK변조하여 잡음평균이 0이고, 분산이 σ^2 인 AWGN채널로 전송하면 수신기에서는 정보비트 u_k 에 대응하는 $y_{1k}^{(1)}, y_{1k}^{(2)}, y_{2k}^{(2)}$ 신호들이 입력된다.

$y_{1k}^{(1)}$ 는 조직적신호이고, $y_{1k}^{(2)}, y_{2k}^{(2)}$ 는 RSC1과 RSC2가 각각 산출한 잉여신호이다. 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 추정된 정보의 LLR(Log Likelihood Ratio)표기이다. τ 번째 반복복호에서 $\Lambda_{1a}^{\tau}(u_k)$ 와 $\Lambda_{1e}^{\tau}(u_k)$ 는 DEC1에서의 사전정보와 외부정보이다. 그리고 $\Lambda_{2a}^{\tau}(u_k)$ 와 $\Lambda_{2e}^{\tau}(u_k)$ 는 각각 DEC2의 사전정보와 외부정보이다.

DEC1과 DEC2에서 완성된 정보는 $\lambda_{1k}^{A(\tau)}(u; O)$ 와 $\lambda_{2k}^{A(\tau)}(u; O)$ 으로 각각 출력되며 다음 식 (8)로 표현된다.

$$\begin{aligned} \lambda_{1k}^{A(i)}(u; O) &= \lambda_{1k}(c^{(1)}; I) + A_{1a}^i(u_k) + A_{1e}^i(u_k) \\ \lambda_{2k}^{A(i)}(u; O) &= \lambda_{1\alpha(k)}(c^{(1)}; I) + A_{2a}^i(u_k) + A_{2e}^i(u_k) \end{aligned} \quad (8)$$

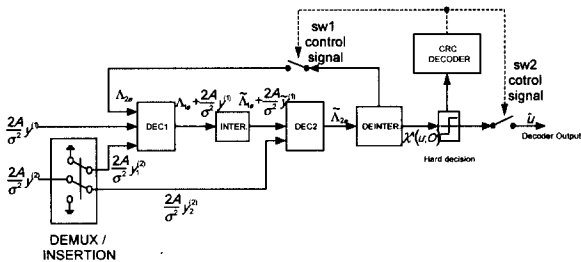
반복처리는 다음 식 (9)와 같이 할당되어 사전정보값으로 재입력된다.

$$\begin{aligned} A_{1a}^i(u_{\alpha(k)}) &= A_{1e}^{i-1}(u_k) \\ A_{2a}^i(u_k) &= A_{1e}^i(u_{\alpha(k)}) \end{aligned} \quad (9)$$

여기서 $\alpha(k)$ 는 인터리버 매핑을 의미한다. 식 (8)에서 두 개의 완전한 정보를 표현하기 위해 $j=1,2$ 조건으로 놓는다. j 번째 복호기에서 산출되는 완전한 정보는 세 개의 추정치들로 구성된다. 채널로부터 직접 전달된 $\lambda_{1k}(c^{(1)}; I)$, 사전값 $A_{1a}^i(u_k)$ 와 트렐리스 구조를 기초로 하여 얻어진 $A_{1e}^i(u_k)$ 들로 구성된다.

세 가지의 추정치 중 $\lambda_{1k}(c^{(1)}; I)$ 값은 매 반복복호마다 고정값이며 $A_{1a}^i(u_k)$ 값과 $A_{1e}^i(u_k)$ 값은 반복복호를 할 때마다 계속적으로 갱신된다. (그림 4)에서 터보복호기의 구조에서 외부정보가 다음 복호기의 사전정보 값으로 사용된다면, 즉 $\tau \geq 2$ 면 $A_{1a}^i(u_k)$ 와 $A_{1e}^i(u_k)$ 는 식 (9)에 명시된 것처럼 상호관계를 가지도록 할당한다.

복호기의 경판정(Hard decision) 결정단에 CRC복호기를 부가하여 CRC프레임을 검사하여 반복복호를 제어하게 된다. CRC복호기는 FCS 값이 '0'이 된다면 SW1을 OFF하고 SW2를 ON하는 제어신호를 발생한다. 반대로 프레임 검사한 나머지 값이 '0'이 아닌 임의의 값을 가지게 된다면 SW2를 OFF시키고 SW1이 ON되도록 제어신호를 발생하여 반복복호를 진행하게 되는 것이다. FCS 값이 '0'이 되는 시점에서 복호가 종료되고 제어신호에 의하여 추정된 복호출력을 제공하게 된다.

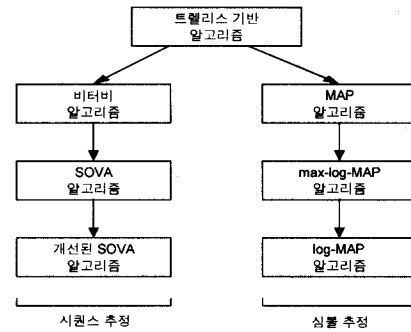


(그림 4) 터보코드의 복호기

터보코드는 트렐리스를 기반으로 하는 복호알고리즘을 이용하여 복호가 수행되는데 트렐리스를 기반으로 하는 복호알고리즘은 크게 두 종류로 대별된다. 첫 번째로 시퀀스(Sequence) 오류를 최소화하기 위한 비터비알고리즘(Viterbi algorithm : VA)류와 심볼(Symbol)오류를 최소화하기 위한

MAP(Maximum a posteriori probability)알고리즘류이다[7, 10, 11].

(그림 5)는 트렐리스 기반에서 터보코드를 복호하기 위한 대표적 알고리즘들 관점에서 분류한 것이다. 일반적으로 그림 하단에 위치한 알고리즘일수록 상단의 알고리즘보다 복잡도는 높아지나 복호성능은 더 우수하게 나타난다. 또한 MAP을 기반으로 한 알고리즘이 비터비 알고리즘을 기반으로 한 알고리즘보다 복잡도는 높으나 성능은 더욱더 우수하게 나타난다.



(그림 5) 트렐리스 기반 복호알고리즘의 구분

반복복호구조를 가지는 부호기는 SISO(Soft Input Soft Output) 복호 알고리즘을 수행하기 위해 실수 연산이 필수적이다. <표 2>는 SISO 복호를 수행하는 알고리즘들을 계산복잡도에 따라 나타낸 것이다. 대표적으로 예를 들어 MAP알고리즘이 수행하는 연산은 N비트 프레임 복호시 실수덧셈, 실수곱셈, 지수연산을 각각 $N \times (2 \times 2^k \times 2^v \times 6)$, $N \times (5 \times 2^k \times 2^v + 8)$, $N \times (2 \times 2^k \times 2^v)$ 회를 수행하며 CRC연산은 $3 \times N$ 회의 이진덧셈을 수행한다. 여기서 k는 부호기의 입력수이며, v는 부호기의 메모리 차수이다. <표 2>에서 CRC 연산량은 알고리즘별 SISO복호기의 실수연산량 관점에서 본다면 매우 작은 연산량을 수행한다. CRC를 채용하여 반복복호횟수를 평균 1회 줄일 수 있다면 CRC의 비트연산을 감수할 수 있을 정도의 trade off를 가진다고 할 수 있다.

<표 2> N비트 프레임 1회 복호시 알고리즘에 따른 계산복잡도와 CRC 계산복잡도 비교

연산종류	MAP	Log-MAP	Max-Log-MAP	SOVA	CRC
실수덧셈	$N \times (2 \times 2^k \times 2^v + 6)$	$N \times (6 \times 2^k \times 2^v + 6)$	$N \times (4 \times 2^k \times 2^v + 8)$	$N \times (2 \times 2^k \times 2^v + 9)$	-
실수곱셈	$N \times (5 \times 2^k \times 2^v + 8)$	$N \times (2^k \times 2^v)$	$N \times (2^k \times 2^v)$	$N \times (2^k \times 2^v)$	-
이진덧셈	-	-	-	-	3N이상
실수최대값 비교	-	$N \times (4 \times 2^v - 2)$	$N \times (4 \times 2^v - 2)$	$N \times (2 \times 2^v - 1)$	-
Look-up 테이블	-	$N \times (4 \times 2^v - 2)$	-	-	-
지수연산	$N \times (2 \times 2^k \times 2^v)$	-	-	-	-

3. 제안한 반복복호수 제어기법

반복복호의 횟수가 증가할수록 복호기는 점차적으로 비트오류율(BER)과 프레임오류율(FER)을 감소시키며 개선된 성능을 나타낸다. 각 프레임들은 선정된 고정 프레임 횟수 T_{max} 동안 반복복호되어 복호된다. 대개 T_{max} 는 임의로 가장 훼손된 프레임의 기준으로하여 설정한다. 실험에 의하면 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[4,5]. 비교를 목적으로 기존의 두 가지 구조들과 본 논문에서 제안한 구조를 제시하였다. 모의실험 결과는 기존의 구조들과 비교하였으며 이미 알려진 두 가지 정지구조를 검토해 보자. 비교대상인 정지구조들은 최대 T_{max} 의 반복복호를 수행하는 동적인 정지구조일지라도 그 T_{max} 반복복호 내에서조차도 그 성능이 만족스럽지 않다.

3.1 교차 엔트로피(CE : Cross Entropy)

m 번째 복호기의 출력에서 두 개의 연속적인 복호 동작의 사후확률(a posteriori) 분포가 있다고 가정하자. $P(\hat{u})$ 는 랜덤변수로서 DEC2의 사후확률분포이고, $Q(\hat{u})$ 는 DEC1의 사후확률분포라고 정의하여 다음 식 (10)과 같이 두 분포의 차의 측정치로서 정의할 수 있다[4,6].

$$E_p \left\{ \log \frac{P(\hat{u})}{Q(\hat{u})} \right\} \quad (10)$$

여기서 E_p 는 분포 $P(\hat{u})$ 상에서 기댓값 연산자(expectation operator)를 나타낸다. 통계적으로 독립을 가정하면 다음 식 (11)을 얻을 수 있다.

$$\log \frac{P(\hat{u})}{Q(\hat{u})} = \sum_k \log \frac{P(\hat{u}_k)}{Q(\hat{u}_k)} \quad (11)$$

DEC1과 DEC2로 구성되어 있는 첫 번째 반복에서 두 개의 연속적인 반복 복호수를 $(\tau-1)$ 과 (τ) 라 하고 복호기의 연관정(Soft decision)출력을 다음과 같이 정의하자.

$$\begin{aligned} \Lambda_Q^{(\tau)}(\hat{u}_k) &= L_c \cdot y_k + \Lambda_{2e}^{(\tau-1)}(\hat{u}_k) + \Lambda_{1e}^{(\tau)}(\hat{u}_k) \\ \Lambda_P^{(\tau)}(\hat{u}_k) &= L_c \cdot y_k + \Lambda_{1e}^{(\tau)}(\hat{u}_k) + \Lambda_{2e}^{(\tau)}(\hat{u}_k) \end{aligned} \quad (12)$$

식 (11)에서 (\hat{u}_k) 는 추정된 k 번째 정보 비트이며, y_k 는

k 번째 수신 비트, $\Lambda_Q^{(\tau)}(\hat{u}_k)$ 는 τ 번째 반복 복호중 첫 번째 복호기 DEC1에서 얻은 k 번째 정보 비트에 대한 외부 정보(extrinsic information)이며 다음 복호기 DEC2의 사전 확률(a priori) 값으로 사용된다. 또한 L_c 는 채널 신뢰도를 나타낸다. 식 (12)에서 두식의 차는 식 (13)와 같이 되며

$$\Lambda_P^{(\tau)}(\hat{u}_k) - \Lambda_Q^{(\tau)}(\hat{u}_k) = \Lambda_{2e}^{(\tau)}(\hat{u}_k) - \Lambda_{2e}^{(\tau-1)}(\hat{u}_k) = \Delta \Lambda_{2e}^{(\tau)}(\hat{u}_k) \quad (13)$$

결정이 더 이상 변하지 않을 때 예를 들어 부호를 가지고 크기가 1보다 작을 때 식 (10)은 다음과 같이 간략화 될 수 있다.

$$E_p \left\{ \log \frac{P(\hat{u})}{Q(\hat{u})} \right\} = T(\tau) = \sum_k \frac{|\Delta \Lambda_{2e}^{(\tau)}(\hat{u}_k)|^2}{\exp(|\Lambda_Q^{(\tau)}(\hat{u}_k)|)} \quad (14)$$

비록 최우도(likelihood)값 사이의 통계적 독립의 가정은 약간의 반복 복호 뒤에는 정확하지 않지만 반복 복호를 위한 정지 기준으로 다음 식(15)을 사용할 수 있다.

$$T(\tau) = \sum_k \frac{|\Lambda_{2e}^{(\tau)}(u_k) - \Lambda_{2e}^{(\tau-1)}(u_k)|^2}{\exp(|\lambda_{1k}^{(\tau)}(u_k)|)} < threshold \quad (15)$$

$\Lambda_{2e}^{(\tau)}(u_k)$ 는 반복복호 τ 시점에서 두 번째 구성 복호기가 산출하는 정보비트 u_k 의 외부정보(extrinsic information)이다. 그리고 $\lambda_{1k}^{(\tau)}(u; 0)$ 는 첫 번째 구성 복호기에 의해 산출되는 완전한 정보비트 u_k 이다. 반복 복호 동작은 CE가 조건 $T(\tau) < (10^{-2} \sim 10^{-4}) * T(1)$ 을 만족하면 정지하게 된다.

3.2 부호전환율(SCR : Sign Change Ratio)

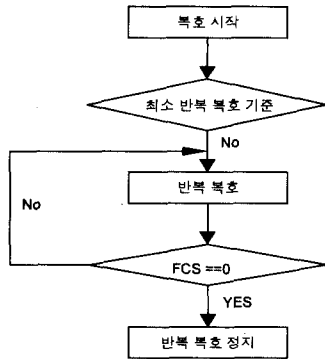
교차엔트로피는 복잡한 계산을 포함하고 있어 구현상의 어려움이 있게 된다. 이를 해결하는 구조로 SCR정지기준 구조가 제안되었는데 CE방법과 유사한 정지기준 구조를 가진다. 이 방법은 반복복호 $(\tau-1)$ 시점과 τ 시점간에 외부정보 $\Lambda_{2e}^{(\tau-1)}(u_k)$ 와 $\Lambda_{2e}^{(\tau)}(u_k)$ 의 부호전환의 횟수 $C(\tau)$ 를 계산하는 것이다. 반복복호는 부호전환 횟수가 조건 $C(\tau) \leq q_c N$ 을 만족할 때 정지하는데 이를 SCR(sign change ratio)정지기준이라 한다.

여기서 q_c 는 상수로서 대개 $0.005 \leq q_c \leq 0.03$ 이고 N 은 프레임 크기이다. CE방식과 SCR방식을 비교해보면 평균 반복복호수에서 CE방식이 $(6N-1)$ 번의 실수연산과 $(N+2)$ 크기의 실수 메모리가 필요하지만, SCR방식은 N 번의 정수연산과 $(N+2)$ 크기의 정수 메모리만 필요하므로 계산량과 하드웨어 복잡도가 감소하게 된다는 장점을 가진다.

3.3 제안한 CRC 정지기준구조

(그림 1)에서는 각 정보계열의 후미에 N_{crc} 비트 길이

를 추가하여 프레임을 생성한다. 이렇게 확장된 프레임은 터보코드 부호기의 입력이 되어 부호어를 생성하게 된다. 수신된 부호어는 복호기에서 각 반복복호마다 경판정을 실시한 후 CRC복호기의 입력이 되어 프레임오류를 검사하기 위해 이용된다. (그림 6)은 CRC부호기가 반복복호를 정지하기 위한 절차를 흐름도로 나타내었다.



(그림 6) 반복복호 정지 절차

CRC검출에 오류가 없을 때 즉 FCS값이 '0'이면 반복복호는 종료하게 되는 것이다. 각 반복복호에서 CRC방법은 LLRs 계산과정을 거치고 $\lambda_{ik}^{(n)}(u; O)$ 의 부호를 기반으로 하는 경판정을 한다. 그리고 마지막으로 경판정을 이용한 CRC복호가 이루어진다. 대개 CRC복호시 적어도 3N번의 이진덧셈과 $\lambda_{ik}^{(n)}(u; O)$ 를 얻어 경판정하기 위해 N번의 실수 덧셈연산이 필요하지만 CE방법은 $\lambda_{ik}^{(n)}(u; O)$ 를 얻기 위한 N번 실수덧셈연산을 포함하여 6N-1번의 실수연산이 수행되며 저장을 위한 (N+2) 크기의 실수 메모리 유닛이 필요하다. SCR방법은 오직 N번의 이진 덧셈이 이루어지며 N보다 큰 카운터는 필요없다.

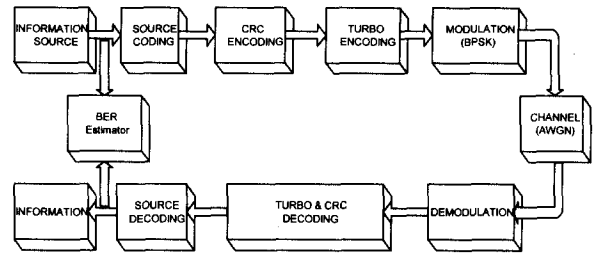
<표 3> 정지기준에 따른 계산 복잡도 비교

정지기준	프레임당 부가비트	실수연산	이진덧셈	메모리 사용
CRC	4~32	N(정수)	3N 이상 (다항식생성)	0
CE	0	6N-1(실수)	0	N+2(실수)
SCR	0	N(정수)	N	N+2(정수)

4. 모의실험과 결과

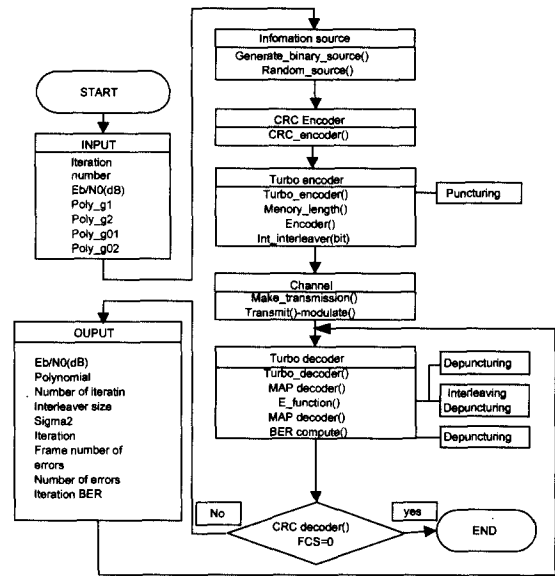
(그림 7)은 기존에 제안된 정지구조와 본 논문에서 새로 제안된 정지구조를 어느 정도의 성능향상을 기대할 수 있는가를 알아보기 위한 모의실험 환경 모델이다. 모의 실험 환경으로는 가장 기본적인 통신 환경을 가정하였다. 채널모델은 백색부가가우시안잡음(AWGN)채널을 가정하였고 변

복조 방법으로는 BPSK(binary phase shift keying)로 가정하였으며 복호알고리즘으로는 MAP(Maximum a posteriori) 알고리즘을 사용하였으며 최대 반복복호수는 7회로 하였다.



(그림 7) 모의실험 환경 모델

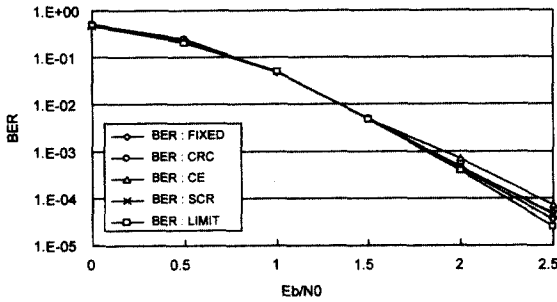
또한 (그림 7)은 모의실험 환경모델을 프로그램화하기 위해 제시한 프로그램 흐름도로 다양한 관점에서의 결과도출을 위하여 여러 가지 입력조건들을 설정하였다. 입력조건들은 E_b/N_0 , 구성부호, 인터리버키, 반복복호 횟수, 평서링의 유무, 유한(finit) 프레임 오류들을 입력으로 설정하였으며 출력결과에는 유한 프레임 오류가 발생하였을 경우 모의 실험을 중지하고 비트 오류율과 프레임 오류율을 계산하여 출력한다.



(그림 8) 모의실험 프로그램 흐름도

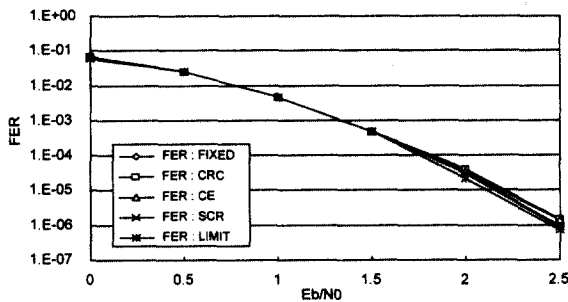
모의실험에 사용된 구성부호의 생성다항식 $g(D)=[1, 1+D+D^3/1+D^2+D^3]$ 으로 수행하였다. 제안된 구조와 비교를 위하여 기존에 제안된 CE, SCR 정지구조와 복호횟수를 사전에 확정하는 Fixed구조, 수신단의 복호데이터와 송신단의 정보데이터를 비교하며 복호하는 Limit구조와 본 논문에서 제안한 CRC구조를 다음과 같이 Fixed, CRC, CE, SCR,

Limit로 명명하여 실험하였다.



(그림 9) SNR에 따른 제안구조의 BER성능

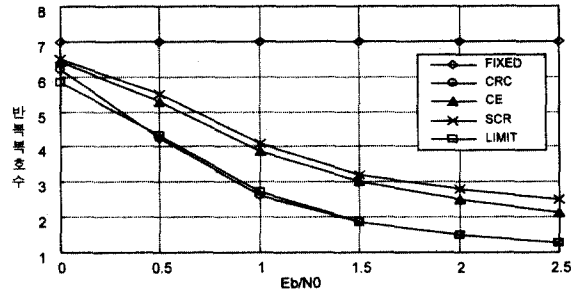
(그림 9)는 SNR에 따른 제안구조들의 BER성능을 나타내었다. 그림에서 보듯이 낮은 신호대 잡음비에서는 거의 동일한 이득을 가지며 높은 신호대 잡음비에서도 약간의 이득차이는 있으나 어느 정지기준도 성능에 큰 우세를 보이지는 않고 있다. Limit 구조의 BER성능이 우수하게 나타나는 것은 송신데이터를 수신기에서 미리 알고 있다는 전제하에 실험하는 구조이기 때문이며 성능이 가장 미미한 CE구조는 반복복호시 교차엔트로피를 계산하여 임계치를 결정하여 복호가 정지하기 때문에 평균복호횟수는 줄어드는 반면 BER성능이 다소 차이가 나타남을 알 수 있다. 제안한 CRC구조는 고정반복복호횟수를 가지는 FIXED구조와 비슷한 성능을 보이는 것을 알 수 있다.



(그림 10) SNR에 따른 제안구조의 FER 성능

(그림 10)은 SNR에 따른 제안구조의 FER 성능을 나타내었다. 그림에서 보듯이 낮은 신호대 잡음비에서는 거의 동일한 FER성능이 나타남을 알 수 있으며 높은 신호대 잡음비에서도 약간의 이득차이는 있으나 거의 비슷한 성능이 나타남을 알 수 있다. FER요구치를 1.E-06이라고 가정했을 때 최대이득을 가지는 구조와 최소이득을 가지는 구조의 이득차이는 0.15dB정도의 미미한 성능차이를 나타내었다. CRC구조와 CE구조는 거의 동일한 성능곡선을 보였으며 SCR구조가 기준이 되는 Limit 다음으로 좋은 성능으로 나타났다. Limit의 경우는 정보비트를 알고 있고 프레임이 올바르게 복호되었을 때 즉시 종료된다. 즉 모든 구조의 임계

치(limit)이라고 볼 수 있다. 실험에 사용된 CRC발생기의 다항식은 $g(x) = x^{16} + x^{15} + x^2 + 1$ 을 이용하였다.



(그림 11) SNR에 따른 평균반복복호수 비교

(그림 11)은 신호대 잡음비와 평균 반복복호수를 BER성능 요구치 1.0E-02 기준에서 도시한 것으로 CRC정지기준 구조가 가장 적은 반복복호수로 복호되어짐을 관찰할 수 있다. BER 및 FER성능곡선에서는 다른구조에 비하여 우수한 이득차이를 보이진 않았지만 평균적으로 다른 구조에 비하여 1회정도의 평균반복복호 횟수를 줄일 수 있었다. <표 4>는 SNR에 따른 평균반복복호수를 보여주고 있는데 제안한 CRC구조가 임계가 되는 Limit구조와 거의 동일한 값을 가짐을 알 수 있다.

<표 4> SNR에 따른 평균반복복호수

SNR	FIXED	CE	SCR	LIMIT	CRC
0.5	7	5.30	5.50	4.30	4.25
1.0	7	3.85	4.08	2.70	2.60
1.5	7	3.00	3.20	1.85	1.85
2.0	7	2.50	2.80	1.50	1.50
2.5	7	2.10	2.50	1.25	1.25

5. 결 론

반복복호구조를 가지는 부호는 BER관점에서 본다면 반복복호수가 증가할수록 그 성능은 점차적으로 향상되지만 반복복호수가 증가할수록 복호를 위한 방대한 연산량과 실시간 설계에 부적합한 문제점을 가지게 된다. 이러한 문제점을 해결하기 위해 본 논문에서는 반복복호구조를 가지는 부호기 전단에 CRC부호를 설정하고 복호단에서는 채널상태에 따른 최소 반복복호를 수행후 프레임 오류 감사단에 입력되어 그 출력값에 따른 ON/OFF 제어신호를 발생하여 효과적인 반복복호 정지시점을 적용적으로 결정할 수 있는 구조를 제안하였다.

기존의 다른 구조에 비하여 BER 성능 감소없이 평균반복복호 횟수가 다른 정지기준보다 적은 횟수로 반복복호되었음을 확인하였으며 BER 요구치 1.0E-02를 기준하여

반복복호한 결과 평균 1회 정도의 반복복호 감소를 증명하였다.

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

참 고 문 헌

[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 Permutation", The JPL TDA Progress Report 42-122, Aug., 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.

[6] Michael Moher, "Decoding via Cross-entropy Minimization," in Proc. IEEE Globecom Conf., Houston, TX, pp.809-813, Dec., 1993,

[7] Branka Vucetic and Jinhong Yuan, "Turbo Codes," Kluwer Academic Publishers, 2000.

[8] Chris Heegard and Stephen B. Wicker, "Turbo Coding," Kluwer Academic Publishers, 1999.

[9] Patrick Robertson, E. Vilebrun and Peter Hoeher, "A Comparison of Optimal and Sub-Optimal MAP Decoding Algorithms Operating in the Log Domain," Proc. ICC'95, Seattle, June, 1995.

[10] A. J. Viterbi, "Error bounds for convolutional codes and an asymptotically optimum decoding algorithm," IEEE Trans. Inform. Theory, Vol.IT-13, pp.260-269, Apr., 1967.

[11] J. Hagenauer and P. Hoerer, "A Viterbi algorithm with soft-decision outputs and its application," in Proc. IEEE GLOBECOM, pp.1680-1686, 1989.

백 승 재

e-mail : bsj3386@empal.com
 1997년 청주대학교 전자공학과(학사)
 1999년 청주대학교 대학원 전자공학과(석사)
 1999년~2002년 청주대학교 대학원 전자공학과 수료(박사수료)

2004년~현재 청주기능대학 정보통신시스템과 IT교수
 관심분야 : 부호이론, 정보이론, 디지털통신, 이동통신

박 진 수

e-mail : parkjs@chongju.ac.kr
 1975년 한양대학교 전자공학과(학사)
 1977년 한양대학교 전자통신과(석사)
 1985년 한양대학교 전자통신과(박사)
 1999년~현재 RRC 정보통신연구센터 소장

2000년~현재 한국정보처리학회 부회장
 1978년~현재 청주대학교 정보통신공학부 교수
 관심분야 : 이동통신, 디지털 통신, 부호이론, 스프레드스펙트럼