

감소된 복호지연을 갖는 Turbo Code

正會員 김 준 범**, 문 태 현*, 임 승 주***, 주 판 유*,
홍 대 식*, 강 창 언*

A Turbo Code with Reduced Decoding Delay

Joon-Beom Kim**, Tae-Hyun Moon*, Seung-Ju Lim***, Pan-You Jou*,
Dae-Sik Hong*, Chang-Eon Kang* *Regular Members*

요 약

Turbo code는 반복적인 복호알고리즘을 사용하므로써 백색 가우시안 잡음(AWGN) 채널 환경에서 채널 용량에 가까운 성능을 보이는 오류정정 방식으로서 제안되었다[1]. 그러나 Turbo code는 큰 인터리버와 반복적인 복호 알고리즘을 사용하기 때문에 수신과정에서 큰 복호지연을 요구하며, 이로 인해 음성전송을 위한 이동통신 시스템에는 적합하지 않다. 본 논문에서는 기존에 사용되었던 Turbo code가 갖는 성능은 저하시키지않고 전송 및 복호에 필요한 시간 지연을 줄일 수 있는 방법을 제안하였다. 실험 결과 가우시안 채널 환경에서 인터리버의 크기를 원래의 제곱근으로 줄여도 기존의 Turbo code의 성능을 유지할 수 있었다. 또한 최근에 이동 통신 기반 기술로 자리를 잡고있는 Multi Carrier CDMA(MC-CDMA) 시스템에 Turbo code를 적용하여 레일리 페이딩 채널 환경에서 그 성능을 분석하였다. 실험 결과 레일리 페이딩 채널 환경에서는 페이딩 현상에 의한 진폭의 왜곡으로 인하여 백색 가우시안 잡음 채널에서 필요한 인터리버 크기의 2배 또는 4배가 필요하였다.

이러한 실험결과를 통해 볼 때, 기존의 Turbo code에 의한 전송 및 복호에 필요한 시간 지연을 인터리버 크기로 한다면, 제안된 Turbo code는 성능은 유지하면서 시간 지연은 원래의 인터리버 크기의 제곱근의 정수배로 감소하는 장점을 갖는다.

ABSTRACT

Turbo codes, decoded through an iterative decoding algorithm, have recently been shown to yield remarkable coding gains close to theoretical limits in the Gaussian channel environment. This thesis presents the performance of Turbo code through the computer simulation. The performance of modified Turbo code is compared to that of

*연세대학교 전자공학과

**한국통신 연구개발본부 무선통신연구소

***중앙일보 전문기자

論文番號:97056-0212

接受日字:1997年 2月 12日

the conventional Turbo codes. The modified Turbo code reduces the time delay in decoding with the minimal effect to the performance for voice transmission systems. To achieve the same performance, random interleaver the size of which is no less than the square root of the original one should be used. Also, the modified Turbo code is applied to MC-CDMA system, and its performance is analyzed under the Rayleigh Fading channel environment. In Rayleigh fading channel environment, due to the amplitude distortion caused by fading, the interleaver of the size twice no less than that in the Gaussian channel environment was required. In overall, the modified Turbo code maintained the performance of the conventional Turbo code while the time delay in transmission and decoding was reduced at the rate of multiples of two times the squared root of the interleaver size.

I. 서 론

Turbo code는 반복적인 복호화 알고리즘을 이용하여 Shannon의 한계에 거의 도달하는 오류 정정 코드로서 1993년 C. Berrou 등에 의하여 제안되었다[1]. Berrou는 인터리버의 크기가 65536이고 복호를 18회 반복했을 때 BER이 10^{-5} 가 되는 지점에서 필요한 비트 에너지 대 잡음의 전력비가 0.7dB임을 보였다. 그러나, [1]에서는 실제로 구현하는데 필요한 많은 사항들이 언급되어 있지 않았고, 그 이후에 여러 논문에서 언급되지 않은 사항들에 대하여 많은 이론들을 전개해 놓았다[4, 5, 6, 13, 14]. 이러한 연구들은 주로 원거리 우주 통신(deep-space communication)에 적용한 것들이며, 많은 성능 분석이 이 관점에서 이루어졌다고 볼 수 있다. 또한 최근에는 이 코드를 TCM이나 Multi Level Coding 등과 같은 다른 시스템에 적용하는 연구들이 진행되고 있다. 하지만, 아직도 정확한 코드의 한계는 유도되어 있지 않고, 이 코드의 복호 알고리즘 단순화에 대한 연구들이 진행되고 있는 실정이다. 하지만 Turbo code는 좋은 성능을 얻기 위하여 큰 인터리버를 사용하고 반복적인 알고리즘을 사용함으로써 복호에 의한 지연이 커지게 되고, 이는 음성통신을 위한 이동통신에 적용하는 것을 불가능하게 한다.

본 논문에서는 Turbo code가 지니는 가장 큰 단점인 복호에 필요한 시간 지연을 줄이도록 변형하고, 모의 실험을 통하여 그 성능을 분석한다. 이는 성능을 거의 변화시키지 않고 음성 전송을 하는 시스템에 적용하기에 적합하게 하는 것이 목적이다. 그리고, Turbo code를 차세대 멀티미디어 통신의 기반기술로 대두되고 있는 Multi Carrier CDMA(MC-CDMA) 시스템에 적용하여 보고, 그 성능을 분석한다.

본 논문의 구성은 2장에서 Turbo code를 구성하기 위한 기본적인 사항들과 부호화에 대한 사항을 먼저 언급하고, 인터리버가 Turbo code의 성능에 미치는 영향을 기술한다. 3장에서는 성능이 우수한 코드를 설계하기 위하여 고려해야 할 사항들을 살펴보고, 일반적인 Turbo code를 변형한 모델을 제안한다. 4장에서는 일반적인 Turbo code와 변형 제안된 Turbo code와 비교하여 그 성능을 분석한다. 그리고, 기존의 Turbo code와 변형된 Turbo code를 MC-CDMA 이동 통신 시스템에 적용하여 그 시스템의 성능을 체계적인 컨벌루션 코드로 비교한다.

II. Turbo code의 구조

2.1 Turbo code의 기본구조

Turbo code의 기본 구조는 두 개의 RSC(Recursive Systematic Convolutional) 코드와 두 부호화기를 분리해 주기 위하여 사이에 첨가해 주는 인터리버로 구성이 된다. 이 인터리버에 입력 시퀀스가 입력되기 전에 입력을 지연 시키는 요소가 첨가되기도 한다 [1, 12, 13].

Turbo code를 구성할 때 부호화기는 반드시 RSC 코드일 필요는 없다. 하지만, 기존의 컨벌루션 코드를 사용하여 Turbo code를 구성한다면, Turbo code를 다단계의 반복적인 복호 알고리즘으로 복호할 때 매트릭의 구성이 RSC 코드보다 복잡하게 된다. 또 체계적인 컨벌루션 코드는 낮은 비트에너지 대 잡음비에서 기존의 컨벌루션 코드보다 약간 좋은 성능을 보이는 장점을 갖는다. Turbo code는 주로 낮은 신호 대 잡음비에서 성능을 분석하는 원거리 우주 통신(deep space communication)에 사용되는 코드와 성능을 비교하기 때문에 낮은 비트에너지 대 잡음비에서

서 성능이 좋은 RSC 코드를 Turbo code를 구성하는 요소로 사용하는 것이 좋다.

Turbo code는 두 부호화기가 같은 입력을 사용하는 하지만, 시간 지연과 인터리버가 첨가됨으로써, 실제적인 입력 열의 순서는 바뀌게 된다. 부호화기의 출력단에서 출력 비트를 모두 전송하게 되면, 이 Turbo code의 코드율은 1/3이 된다. 하지만, 대체로 부호화기의 두 출력을 평치링하므로 코드율을 1/2로 변형하여 전송을 하게된다. 이때, 평치링은 정보비트에 대해서는 적용되지 않고, 잉여비트에 대해서만 적용된다. 그림 1은 Turbo code의 기본 구조를 나타낸 블럭도이다.

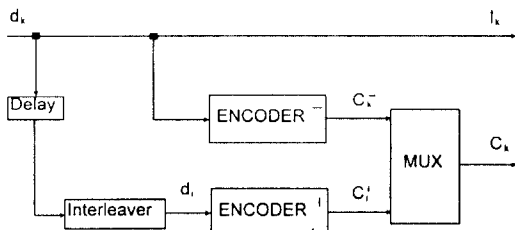


그림 1. Turbo code의 부호화기 구조
Fig. 1. The Structure of the Turbo Code Encoder

한 블록의 정보가 모두 전송된 후, 트렐리스를 종료하기 위해서는 기존의 컨벌루션 코드와는 달리, 두 개의 부호화기 모두를 종료하여야 한다[5]. 이때, 기본적인 부호화기가 순환적이기 때문에, 모든 정보 비트들을 부호화한 후 천이 레지스터의 상태를 0으로 하기 위해서 단순히 0만을 입력시키는 것은 충분하지 않다. 즉, 프레임의 모두 부호화한 후에, 부호화기의 상태에 따라 메모리의 수만개의 비트값을 결정해 주어야 한다. 그리고, 두 번째 부호화기가 종료되기 전에 첫 번째 부호화기의 트렐리스가 먼저 종료되므로 그 상황의 처리를 고려해 보아야 한다. 일반적으로 첫 번째 부호화기의 트렐리스가 종료되면, 두 번째 부호화기는 입력부분을 개방된 상태로 처리한다.

2.2 인터리버에 의한 영향

사용되는 인터리버의 종류에는 대표적으로 블록, Helical, 의사 랜덤 인터리버가 있으며, 이 중에서 의사 랜덤 인터리버가 성능면에서 가장 큰 효과를 얻을 수

있다[12, 14, 15]. 그러므로, 변형된 Turbo code에서는 의사 랜덤 인터리버의 위주로 사용하였다.

Turbo code는 생성 다항식이 다른 두 개의 부호화기를 사용할 수도 있으나, 일반적으로 같은 생성 다항식을 사용하는 경우가 많으며, 이러한 경우에 입력의 순서를 재배열하는 것이 필요하게 된다. 이 역할을 수행할 수 있는 것이 인터리버이다. 이 인터리버는 시간을 지연시켜 주는 요소와 더불어 페이딩 채널에서 시간 다이버시티를 구현해 주는 작용을 하게되고, 백색 가우시안 잡음 채널에서는 연접 오류를 랜덤 오류로 전환하여 주는 역할을 하게 된다. 그러므로, 시간 지연의 요소와 인터리버는 Turbo code에 직접적으로 영향을 준다는 것을 의미한다. 즉, 인터리버가 오율 성능의 변화를 초래하므로 인터리버가 직접적으로 Turbo code의 최소 거리에 영향을 준다는 것을 의미한다. 그래서, 인터리버의 크기가 증가할수록 인터리빙한 효과는 커지게 되어 더 좋은 오율 성능을 보이게 된다. 그러나, 인터리버는 일반적인 오율 성능을 구하기 어렵게 만드는 요소가 되기도 한다.

또, Turbo code의 내부를 구성하는 RSC 코드들은 최적의 1/n 컨벌루션 코드일 필요는 없다는 것이다. 왜냐하면, Turbo code의 성능은 코드의 부호화된 비트들의 최소 거리(minimum distance)에 의존하기 보다는 인터리버에 의존하기 때문이다. 그러나, 크기가 큰 인터리버를 사용하는 것은 복호시간지연이 커지므로 음성 전송에 부적합하다.

Turbo code에서는 매우 높은 비트에너지 대 잡음비에서 최소 거리가 중요한 요소가 되지 못한다. 그러므로, 적당한 비트에너지 대 잡음비에서는 무게 분포(weight distribution)이 성능을 계산하는데 중요하다고 볼 수 있다. 성능이 좋은 Turbo code를 만들어 내기 위해서는 특정한 인터리버에 따라서 각각의 부호화기로부터 부호어(codeword)의 쌍을 찾아내는 것이다. 직관적으로 한 부호화기의 낮은 무게 워드(weight word)가 다른 부호화기와 서로 쌍을 이루는 것을 피해야 한다는 것을 알 수 있다[14, 15].

또, 가우시안 채널에 대하여 원거리 우주 통신에서는 오류가 연접성을 지니고 있다. 그러므로, 반복적인 복호기에 입력이 연접 오류를 지닌 정보라면, 복호기의 출력인 최대 근사화 비도 서로 상관 관계를 지니게 된다. 이렇게 상관 관계를 지닌 정보가 다음 단의

반복적인 복호기의 입력으로 사용된다면, 만족할 만한 성능을 얻을 수 없게 된다. 그러므로, 상관 관계가 있는 정보를 효과적으로 상관 관계가 없는 정보로 전환하기 위해서는 인터리빙 기법이 필요하게 된다.

Ⅲ. 변형 제안된 Turbo code의 구조

3.1 Turbo code의 성능 향상을 위한 구성 요소

2장에서 보았던 것처럼 Turbo code의 성능은 인터리버에 의존하게 된다. 그러므로, 인터리버의 구성이 Turbo code의 성능을 좌우한다고 할 수 있다[13, 14, 15]. 랜덤한 인터리버를 구성하기 위해서 무엇보다도 중요한 것은 인터리버의 크기가 된다. 인터리버의 크기가 중요한 것은 Turbo code가 연관정을 한다는 것으로부터 유도해 낼 수 있다. 외부 정보는 수신된 한 블록의 데이터를 가지고 MAP 복호된 순방향과 역방향 메트릭을 통하여 계산될 수 있다. 그래서 시간 k 에서 외부 정보는 이전의 $k-N+1$ 번째 수신된 정보와 이후의 $k+N+1$ 번째 수신된 정보사이의 $2N$ 개의 데이터에 의하여 영향을 받는다고 할 수 있다. 이때 N 은 구속장 길이의 약 10배 정도가 된다. 그러므로, 인터리버의 크기가 커지면 커질수록 k 시간에 수신된 정보가 속한 $2N$ 개의 영역에서 원래의 정보를 수신한 것과 인터리빙된 정보를 수신한 것사이에는 서로의 상관 관계가 감소하게 된다. 이러한 이유로 인해서 한 복호기에서 한 비트에 대한 연관정 값은 다음에 직렬로 연결된 복호기에 전달할때는 순서가 바뀐 그 정보 비트에 대한 독립적인 외부 정보를 얻을 수 있게 된다.

위의 첫 번째 조건을 만족하게 되면, 두 번째로 고려해야 할 사항은 선택된 인터리버가 입력되는 데이터를 효과적으로 랜덤하게 배열할 수 있는가이다. 컨벌루션 코드들은 일반적으로 연접 오류에 약한 단점을 지니고 있다. 그러므로, 인터리버가 랜덤하게 정보를 다시 배열한다는 것은 첫 번째 복호기가 정정하지 못했던 연접 오류를 랜덤하게 다시 배열하여 랜덤 오류로 전환하기 때문에 두 번째 복호기가 랜덤하게 변환된 첫 번째 복호기의 오류를 정정할 수 있게 된다.

위의 두 조건을 만족하게 되면, 세 번째로 고려해야 할 사항은 Turbo code의 성능에 영향을 주는 것은 Turbo code를 구성하는 각각의 RSC 코드라고 할 수

있다. 왜냐하면, 반복적인 복호가 어떤 한계 횡수에 도달하기 전에는 빠르게 성능이 좋아지지만, 그 이상의 횡수에서는 성능이 향상되지 않는 것을 보면 알 수 있다.

3.2 질 변형 제안된 Turbo code

처음으로 제안된 Turbo code의 구조는 부호화율이 1/3인 코드를 평처링하여 코드율을 1/2로 만들어 전송하는 구조로 크기가 65536인 인터리버를 사용하였고, 여기에 MAP을 사용한 반복적인 복호 알고리즘을 첨가해서 성능면에서 탁월하다는 것을 알 수 있었다[1]. 하지만, 큰 크기의 인터리버와 반복적인 복호 알고리즘으로 인하여 복호를 해서 정보를 복구하는데 필요한 시간이 길어지게 된다. 이러한 시스템은 이 코드가 음성 정보를 전송하기 위한 시스템에는 시간 지연이 크기 때문에 적합하지 않다는 것을 보여준다. 즉, 인터리버로 인한 시간 지연은 병렬로 RSC 코드를 더 덧붙여 주고 인터리버의 크기를 원래의 인터리버의 크기보다 작게 한다면, 시간 지연 측면에서 효과를 얻을 수 있게 된다. 병렬로 연결된 코드는 각각 부호화율이 1/2의 형태이므로 각각의 RSC에서 출력된 비트들을 평처링 패턴을 변화시켜 전송을 하게 된다면 부호화율은 감소시키지만 성능은 거의 유사한 코드를 찾을 수 있다. 평처링을 할 때 평처링율은 시변적으로 변화시켜 줄 수도 있다. 예를들면, 세 단계의 Turbo code를 사용하였을 때, 부호화된 정보를 평처링하지 않고 Y_{k1}, Y_{k2}, Y_{k3} 를 모두 전송하면 부호화율은 1/4이 되고, 부호화된 세 비트의 정보 중 어느 두 정보만을 전송하게 된다면, 부호화율은 1/3이 된다. 세 단계의 부호화기에서 출력된 정보를 평처링하여 단지 한 부호화기의 정보만을 전송하게 된다면, 부호화율은 기본적인 Turbo code와 변화가 없을 것이다. 하지만, 부호화율이 1/2이라면 평처링된 정보가 작은 크기의 인터리버를 사용한 정보가 된다. 이것은 기존의 Turbo code보다 랜덤성이 작게 되기 때문에 RSC 코드를 하나 더 연결해 준다고 해도 성능은 저하되게 된다. 즉, 부호화율의 변화없이 인터리버의 크기를 줄이는 것은 Turbo code의 성능의 저하를 의미하게 된다. 그러므로, 세 단계의 Turbo code를 사용하여 인터리버의 크기를 최소한의 성능 유지를 위한 요소로만 가정하면 인터리버의 크기는 최소한 기

본적인 Turbo code의 인터리버 크기의 제공된 크기의 형태로 구성해야 하고, 기존의 Turbo code의 성능을 유지하기 위하여 부호화율을 감소시키는 것이 필요하다. 그림 2는 변형 제안한 Turbo code의 구조를 나타낸다.

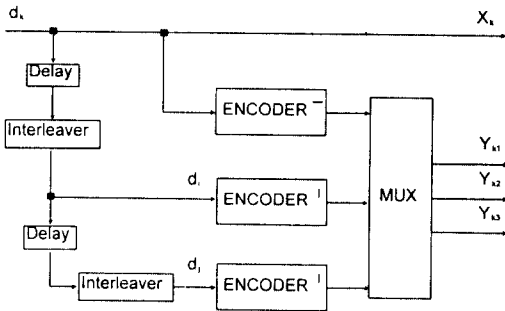


그림 2. 변형 제안된 Turbo code의 블록도
Fig. 2. The block diagram of a Modified Turbo code

인터리버가 Turbo code의 성능에 가장 큰 변수로 작용하기 때문에, 변형된 Turbo code에서 먼저 고려해야 할 사항은 인터리버의 랜덤성이다. 즉, 이 시스템은 인터리버가 원래 크기의 인터리버와 최대한 유사하게 랜덤한 성질을 유지하는 것이 필요하지만, 크기의 감소로 인하여 변형된 Turbo code에서 사용하는 인터리버는 같은 종류라고 할지라도 원래의 것보다 랜덤성이 작게 된다. 예를 들어서 앞에서 언급했던 블록 인터리버나 Helical 인터리버는 정보 자체를 연접 오류에 강하게 인터리빙해 줄 수 없으므로 이 시스템에는 적합하지 않다. 하지만 의사 랜덤 인터리버는 거의 랜덤하게 인터리빙을 해주고 또다시 3차원으로 확장된 부호화에 입력되기 전에 다시 인터리빙을 해주므로 인터리버에서 연속되는 정보의 수 만큼의 연접 오류가 생기지 않는다면 원래의 성능과 유사한 결과를 얻을 수 있을 것이다. 하지만, 세 단계로 나누어 부호화를 하게되고 모두 두번의 인터리빙을 하기 때문에 의사 랜덤 인터리버에서 배열을 결정하는 생성 다항식도 중요한 요소로 작용할 수 있다. 즉, 처음 인터리빙 한 결과를 다시 인터리빙하게 되는데, 다시 인터리빙된 결과가 처음 인터리빙하기 전의 결과와 유사한 패턴을 갖게되면 두 번째 인터리빙한 정보 비트는 원하는 성능을 얻는데 장애 요소가 될 수

있다.

또한, RSC 코드를 구성하는 체계적인 컨벌루션 코드도 고려 대상이 된다. 기존의 Turbo code의 성능은 가장 중요한 요소가 인터리버의 크기, 인터리버의 랜덤 성질 그리고, 체계적인 컨벌루션 코드의 순으로 좌우된다. 가장 중요한 요소인 인터리버의 크기를 줄이고 부호화율을 감소시켰지만, 그만큼의 성능 향상을 인터리버의 랜덤 성질과 체계적인 컨벌루션 코드에서 얻는 것이 필요하다.

3.3 절 변형 제안된 Turbo code의 반복적인 복호

일반적인 연쇄 코드는 첫 번째 복호기의 신뢰도의 값을 이용하여, 두 번째 복호기에서 복호를 하게 되면, 복호의 과정은 종료된다. 첫 번째 복호기가 두 번째 복호기에서 출력된 신뢰도값을 이용하게 된다면, 성능의 향상을 기대할 수 있다[7]. 일반적인 Turbo code는 구조가 이러한 형태의 복호 알고리즘을 사용할 수 있도록 되어있다. 그러므로, Turbo code는 반복적인 복호 알고리즘을 사용하여 성능을 향상시킬 수 있게 된다. 이때 첫 번째 복호기로 재귀되는 두 번째 복호기의 신뢰도값을 외부 정보(extrinsic information)라고 한다[1, 4, 6].

부호화된 정보를 복호를 할 때, 수신단에서 부호화된 정보를 이상적으로는 한번에 복호를 하여야 하지만, 실제적으로 불가능하다. 일반적인 연쇄 코드는 두 번째단에서 부호화된 정보를 복호하고 복호된 그 정보를 사용하여 첫번째 단을 복호하게 되면 복호의 과정은 종료된다. 이때 첫번째단의 복호된 정보를 다시 두 번째단의 복호에 사용하는 반복 복호 알고리즘을 사용한다면 그만큼의 성능향상이 있게 된다. 즉, 수평적인 부호화에 입력된 정보를 먼저 복호하여 그 연판정 값을 다음 단에서 사용하여 순서가 재배열된 코드를 다시 연판정한다. 이것은 Turbo code의 수평적인 복호기가 세 개의 입력을 필요로 한다는 것을 의미하며, 그 세 개의 값은 이전의 복호기에서 출력된 연판정 값, 평처링되어 전송된 정보, 추정된 RSC 코드의 정보의 최대 근사화 값으로 구성된다. 이전의 복호기나 이전의 단계의 복호기에서 출력된 연판정 값을 외부 정보(extrinsic information)이라고 한다.

반복적인 복호과정을 확률적으로 표현하면 다음과 같은 형태로 표현될 수 있다. 우선 각 복호화된 정보

를 \hat{d}_k 의 대수 근사화 확률비(Log Likelihood Ratio, LLR)의 조건부 랜덤변수로 정의하면, 이는 식 (1)과 같이 표현된다.

$$\begin{aligned} L(I_k | R_k) &= \log \{ P(I_k = 1 | R_k) / P(I_k = 0 | R_k) \} \\ &= L(R_k | I_k) + L(I_k) \\ &= \log \left[\frac{\exp \{ -E_s(R_k + r)^2 / N_0 \}}{\exp \{ -E_s(R_k - r)^2 / N_0 \}} \right] + \log \left[\frac{P(I_k = 1)}{P(I_k = 0)} \right] \\ &= L_c \cdot R_k + L(I_k) \end{aligned} \quad (1)$$

여기서 $P(I_k = i) = \{ e^{L(I_k)} / [1 + e^{L(I_k)}] \}$, $L_c = 4r \cdot E_s / N_0$ 으로 정의되고, r 은 부가 가우시안 채널인 경우 I_k 가 0, 1에 상관없이 1이다. 또, 이진 대칭 채널에서 L_c 는 잘못된 전송의 확률 P_0 의 대수 근사비이므로 $L_c = \log [(1 - P_0) / P_0]$ 이다. 만약 페이딩 채널이라면, L_c 는 시간에 따라 변하게 된다.

Turbo code에서는 시간 다이버시티를 이용하는 구조를 가지므로, 한 프레임의 정보비트를 순서를 바꾸어서 부호화를 되어 통계적으로 독립적인 전송이 된다. 그러므로, 식 (1)의 확률비는 식 (2)으로 다시 표현될 수 있다.

$$L(I_k | R_{k1}, R_{k2}, R_{k3}) = L_{c1} \cdot R_{k1} + L_{c2} \cdot R_{k2} + L_{c3} \cdot R_{k3} + L(I_k) \quad (2)$$

제한된 Turbo code의 반복적인 복호는 3차원으로 가정하여 복호를 할 수 있다. 이 3차원의 정보를 이상적으로는 한꺼번에 복호를 해야 하지만, 실제로는 불가능하기 때문에 먼저 첫 번째 코드를 복호하고, 다음에 두 번째 코드를 복호하고, 마지막으로 세

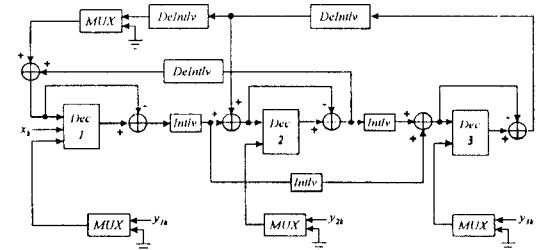


그림 3. Iterative Decoder의 구조
Fig. 3. The Structure of Iterative Decoder

번째 코드를 복호한다. 그림 3은 반복적인 복호 알고리즘을 블럭도로 표현한 것이다. 그림의 한 블럭은 Turbo code의 한 복호기를 구성하는 MAP복호기로 구성된다. Dec 1, Dec 2, Dec 3는 각각 첫 번째, 두 번째, 세 번째 코드의 복호기를 나타낸다.

각각의 복호기는 모든 정보 비트의 a priori 값 $L(I_k)$, 부호화 된 비트가 채널을 통과한 후의 값 $L_c \cdot R_k$ 와 이전의 복호 반복에서 출력된 외부 정보 $L_e(I_k)$ 을 이용하여 복호를 하게 된다. 그러므로, $L(I_k)$ 는 다음의 세 항으로 다시 표현이 가능하다.

$$L_e(I_k) = L_c \cdot R_k + L(I_k) + L_e(\hat{I}_k) \quad (3)$$

정보 비트의 발생 확률이 동일하다면, 첫 번째 복호기에서는 어떤 a priori 정보도 없다. 그래서, $L(I_k) = 0$ 이 된다.

$$\begin{aligned} L_e^1(I_k) &= L(I_k) - L_c \cdot R_k - L(I_k) \\ &= L(\hat{I}_k) - L_c \cdot R_k - L^2(\hat{I}_k) \end{aligned} \quad (4)$$

$L_e^1(I_k)$ 를 a priori 값으로 사용하여, 두 번째 부호화에 대한 복호화를 수행한다.

$$\begin{aligned} L_e^2(I_k) &= L^2(\hat{I}_k) - L_c \cdot R_k - L^2(\hat{I}_k) \\ &= L^2(\hat{I}_k) - L_c \cdot R_k - L_e^1(\hat{I}_k) \end{aligned} \quad (5)$$

다시 이 두 번째 복호기의 외부 정보를 이용하여 세 번째 부호화의 복호를 수행한다.

$$\begin{aligned} L_e^3(I_k) &= L^3(\hat{I}_k) - L_c \cdot R_k - L^3(\hat{I}_k) \\ &= L^3(\hat{I}_k) - L_c \cdot R_k - L_e^2(\hat{I}_k) \end{aligned} \quad (6)$$

세 번째 복호기에서 출력된 외부 정보 $L_e^3(I_k)$ 를 다음 반복의 첫 번째 복호기의 외부 정보로 사용한다. 그러나, 작은 인터리버의 사용은 복호의 반복 횟수가 증가함에 따라 최대 확률비가 서로 독립적에서 점차로 상관 관계를 갖는 값으로 변하게 된다. 따라서, 여러 번의 복호 반복이 지난 후에는 더 이상 반복적인 복호의 장점을 얻지 못할 수도 있다. 마지막 수직적

복호 반복이 지난 후에 최대 근사화 비는 식 (7)과 같 이 표현된다.

$$L(\hat{I}_k) = L_c \cdot R_k + L_e^1(\hat{I}_k) + L_e^2(\hat{I}_k) + L_e^3(\hat{I}_k) \quad (7)$$

IV. 실험 및 결과 분석

4.1 질 채널의 실험 환경

부가 가우시안 채널 환경의 경우 두 가지의 Turbo code 모두 변조방식은 BPSK이며, 일반적인 Turbo code의 코드율은 1/2을 사용하였고, 변형한 모델은 코드율을 1/3, 1/4로 하였다. 실험한 코드는 상태의 수가 4, 8인 것을 사용하였고, 인터리버의 종류는 의 사 랜덤 인터리버를 사용하였다. 사용한 생성 다항식은 $G_1 = (g_{1,1}, \dots, g_{1,L_1})$, $G_2 = (g_{2,1}, \dots, g_{2,L_2})$ 의 형태이며 G_2/G_1 의 형태로 표현하고 G_1 은 feedback 다항식, G_2 는 feedforward 다항식을 나타낸다. $G_{02}/G_{01} - G_{12}/G_{11}$ 인 경우는 기존의 Turbo code를 나타내며, $G_{02}/G_{01} - G_{12}/G_{11} - G_{22}/G_{21}$ 는 변형 제안된 Turbo code의 생성다항식을 나타낸다.

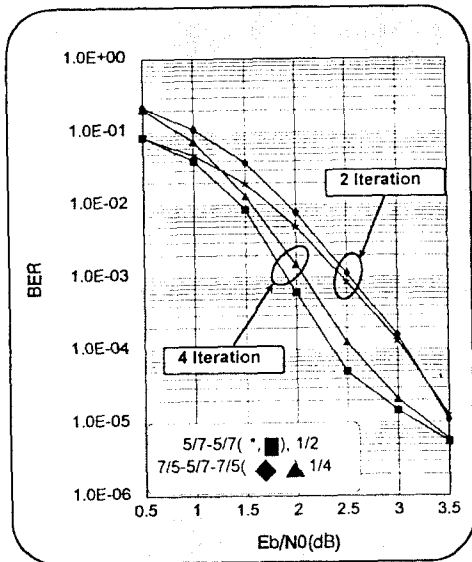


그림 4. Iteration에 따른 성능
Fig. 4. The Performance according to the number of iteration

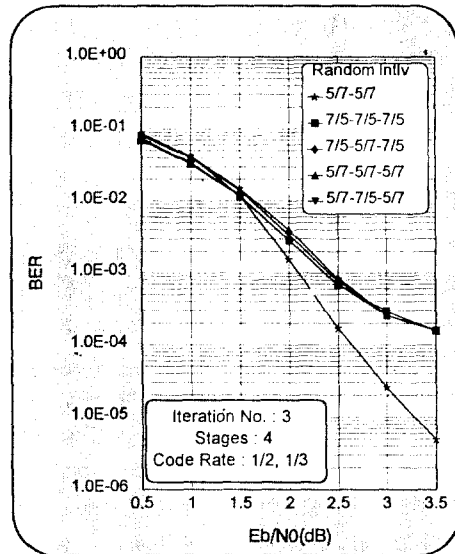


그림 5. 코드율 1/3일 때의 성능
Fig. 5. The Performance with code-rate 1/3

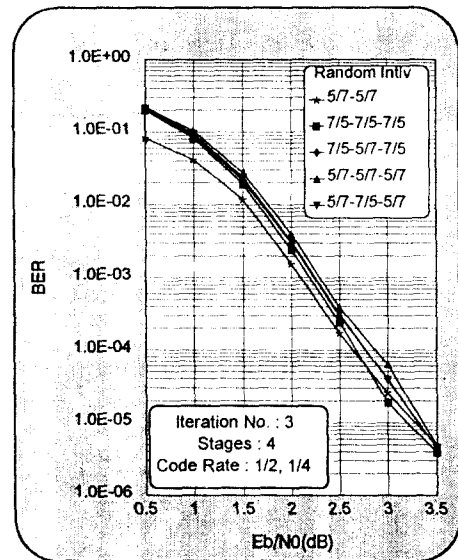


그림 6. 코드율 1/4일 때의 성능
Fig. 6. The Performance with code-rate 1/4

그림 4는 기존의 Turbo code와 변형한 Turbo code의 Iteration에 따른 성능변화를 나타낸다. 비트에너지 대 잡음비가 작을 때에는 성능의 차이가 있었지

만, 높은 값에서는 2번 반복 복호하였을 경우에 비트 오류 확률이 거의 같게 나왔다. 하지만, 4번 반복 복호하였을 경우에는 비트에너지 대 잡음비가 3dB와 3.5dB사이에서 유사한 성능을 얻을 수 있었다. 그림 5, 그림 6은 의사 랜덤 인터리버를 사용할 때, 상태수가 각각 4인 경우의 성능을 비교한 것이다. 그림 5는 상태수가 4이고 코드율이 1/3인 경우이다. 이 때 음성 통신을 위한 비트 에러율을 10^{-3} 으로 볼 때, 약 0.25dB정도의 성능 차이가 있다. 그리고, 그림 6에서 상태수가 4이고 코드율이 1/4인 경우는 변형한 Turbo code가 feedforward 다항식, feedback 다항식의 순서에 관계없이 성능에 차이가 없음을 알 수 있다. 이 경우는 의사 랜덤 인터리버의 변수값, 즉 입력되는 입력 비트의 값, 인터리빙시키기 위한 생성 다항식의 값과 3 단계의 Turbo code에서 코드가 적절히 조합되어 결합되었기 때문이다.

4.2 절 Turbo code를 적용한 Multi-Carrier CDMA의 성능

본 실험에서 기본적인 Turbo code의 코드율은 항상 1/2을 사용하였고, 변형한 모델은 코드율을 1/3로 하였다. 실험 환경으로 사용자는 1명이며, 사용한 도

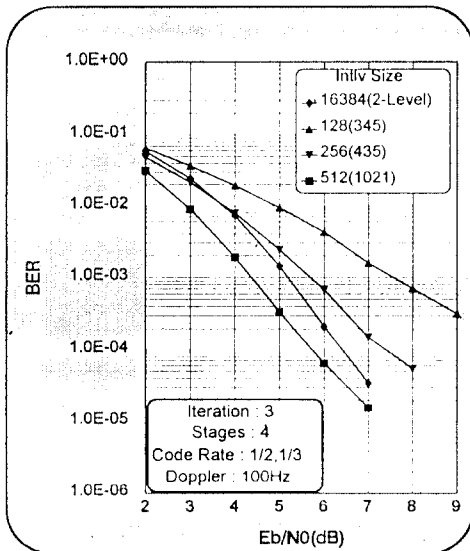


그림 7. 상태수 4에서 성능비교
Fig. 7. The Performance with the number of state = 4

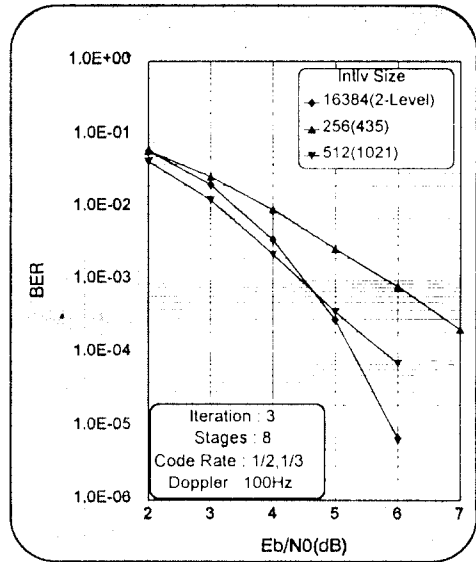


그림 8. 상태수 8에서 성능비교
Fig. 8. The Performance with the number of state = 8

플러 주파수는 100Hz이고, 다이버시티 기법은 MRC (Maximal Ratio Combining)만을 사용하였다. 페이딩 환경은 느린 페이딩이고, 사용한 왈쉬 하다마드 코드는 16개로 제한하였다. 페이딩에 사용한 정보의 비트 전송율은 19.2Kbps이고 변조방식은 부가 가우시안 채널, 페이딩 채널 모두 QPSK이다.

그림 7은 상태수가 4이고 경로 수가 2인 경우에 도플러 주파수가 100Hz인 경우에 기존의 Turbo code와 변형 제안된 Turbo code의 성능을 비교한 것이다. 그림 8은 상태수가 8인 경우이고, 도플러 주파수는 100Hz일때의 성능을 비교한 것이다. 그림 7에서는 BER이 10^{-3} 에서 인터리버의 크기가 128, 256인 경우에는 각각 2.2, 0.4dB정도의 성능 저하가 있고, 인터리버의 크기가 512인 경우에는 약 0.7dB정도의 성능 향상이 있음을 알 수 있다. 성능 저하의 원인은 실험에 사용한 심볼율이 19.2kbps이므로 도플러 주파수가 100Hz인 경우에 페이딩 채널에 의하여 진폭 왜곡을 겪는 비트수가 한 프레임에 50~100개 정도 생길 수가 있다. 그러므로, 인터리버의 크기가 128인 경우에는 한 프레임 정보의 많은 부분이 페이딩을 겪게 되면, 이 왜곡된 정보를 인터리빙하고, 반복적인 복호를 하여도 성능의 향상을 없게 된다. 즉, 성능을 유지하

기 위해서는 최소한 인터리버의 크기가 256이상은 되어야 한다. 그림 8은 경로 수가 2이고 상태수가 8일 때, 변형 제안된 Turbo code의 부호화율이 1/3인 경우에서의 성능 비교를 나타낸다. 상태수가 4인 경우는 인터리버의 크기가 256정도만 되면, 성능을 유지할 수 있으나, 상태수가 8인 경우는 512가 되어야 성능의 유지가 가능함을 알 수 있다.

V. 결 론

Turbo code는 Shannon의 한계에 가까운 성능을 보이지만, 긴 복호 지연에 의해 음성전송을 위한 이동통신 시스템에의 적용이 불가능하다. 본 논문에서는 이와 같은 Turbo code의 성능은 유지시키면서 음성 전송을 위한 시스템에 적합하도록 세 단계의 Turbo code 방식을 제안하고 그 성능을 모의 실험을 통하여 검증해 보았다. 그리고, Turbo code를 차세대 멀티미디어 통신의 기반 기술로 대두되고 있는 MC-CDMA 시스템에 적용하여 그 성능을 분석해 보았다.

변형하여 제안한 Turbo code는 기존의 Turbo code보다 전송 및 복호 시간에 따른 지연을 기존의 방법에서 사용했던 인터리버 크기의 제곱근의 정수배로 감소시킬 수 있었다. 하지만 trade-off로써 부호화율이 감소하는 단점을 지니고 있다. 실험 결과 변형한 모델이 백색 가우시안 잡음 채널에서 BER이 10^{-3} 에서 작은 복호지연을 갖고도 기존의 Turbo code의 성능을 유지할 수 있었다. 또한 변형된 Turbo code를 MC-CDMA에 적용한 경우, 페이딩 채널에 의한 진폭 왜곡으로 인하여 부가 가우시안 채널에서 사용했던 인터리버의 크기의 2배 정도를 사용하였을 때 성능을 유지할 수 있었다. 향후 연구 과제로는 부호화율을 높이는 방안과 이 알고리즘을 페이딩 채널에 적용하기 위한 일반적인 매트릭 정립이 필요하다.

참 고 문 헌

1. Claude Berrou, Alain Glavieux and Punya Thitimajshima, "Near Shannon Limit Error-Correcting Coding and Decoding: Turbo-Codes(1)", in Proc. IEEE ICC, May. 1993, pp. 1064-1070.
2. L. R. Bahl, K. Cocke, F. Jelinek, and J. Raviv,

- "Optimal Decoding of Linear Code for Minimizing Symbol Error Rate," IEEE Trans. on Inform. Theory, vol IT-20, pp 284-287, Mar. 1974.
3. J. Hagenauer and P. Hoeher, "A Viterbi Algorithm with Soft-Decision Outputs and its Applications," in Proc., IEEE Globecom Conf. (Dallas, TX, Nov. 1989), pp. 1680-1686.
4. Joachim Hagenauer, Elke Offer and Lutz Papke, "Iterative Decoding of Binary Block and Convolutional Code," IEEE Trans. Inform. Theory, vol. IT-42, pp. 429-445, Mar. 1996.
5. P. Robertson, "Illuminating the Structure of Code and Decoder of Parallel Concatenated Recursive Systematic (Turbo) code," in Proc., IEEE Globecom Conf. (San Francisco, CA, Dec. 1994), pp. 1298-1303.
6. Steven S. Pietrobon and Adrian S. Barbulescu, "A Simplification of the Modified Bahl Decoding Algorithm for Systematic Convolutional Code," Int. Symp. Inform. Theory & its Applic., pp. 1073-1077, Sydney, Australia, Nov. 1994.
7. 정호영, 박형진, 홍대식, 강창언, "이동 통신 페이딩 채널에서의 Concatenated 코드 시스템," 한국통신학회 논문지, Vol. 18, No. 1, 1993. 1.
8. K. Fazel and L. Papke, "On the Performance of Convolutionally-Coded CDMA/OFDM for Mobile Communication System," in Proceedings PIMRC '93, Yokohama, Japan, 1993, pp. 468-472.
9. J. Proakis, Digital Communications, New York: McGraw-Hill, 3rd Ed., 1995.
10. Nathan Yee and Jean-Paul Linnartz, "Multi-Carrier CDMA in an Indoor Wireless Radio Channel," Research of the MICRO project 93-101, University of California at Berkeley
11. Stefan Kaiser and Lutz papke, "Optimal Detection when Combining OFDM-CDMA with Convolutional and Turbo Channel Coding," in Proc. IEEE ICC, 1996.
12. Jean Yves Couleaud, "High Gain Coding Schemes for Space Communication," Signal Processing Research Institute, University of South Australia, Sept. 1995.

