

MIL인터리버를 이용한 W-CDMA시스템에서 터보부호의 성능

김경선*, 안도랑, 이동욱
동국대학교 전기공학과

Performance of Turbo Code Using Multi-Stage Interleaver in W-CDMA System

Kyung-Sun Kim*, Do-Rang Ahn, Dong-Wook Lee,
Dept. of Electrical Engineering, Dongguk Univ.

Abstract -Turbo Code is one of the most useful and powerful channel coding technique. It is adopted in radio transmission of IMT-2000, both CDMA2000 and W-CDMA system. The performance of this code depends on many parameters such as encoder memory size, free distance of codewords, interleaver size, number of decoding iteration, and so on. This paper describes a Multi-Stage Interleaver which is used in Turbo Codes for W-CDMA channel coding. We compare the performance of a Multi-Stage Interleaver with that of Mother Interleaver. The simulation results show that Multi-Stage Interleaver performs better than Mother Interleaver in bit error rate from 10^{-3} to 10^{-6} .

1. 서 론

현대 디지털 통신 시스템에서 여러 정정 기술은 송신전력의 감소와 신뢰성 측면에서 상당히 중요한 역할을 한다. 터보 부호는 1993년 Berrou[1]에 의해 처음 발표되었다. 이후 AWGN 채널에서 Shannon의 한계에 가장 근접한 성능을 내는 터보부호는 초기에는 우주통신 분야에 적용되었고, IMT-2000 시스템의 무선 데이터 전송 규격[2]으로 설정되는 등 미래 통신 분야뿐만 아니라 방송분야 등의 많은 분야에서 적용되고 있다. 본 논문에서는 터보부호의 성능에 많은 영향을 미치는 터보부호의 부호기, 복호기, 인터리버, 복호방식을 설명하고 W-CDMA시스템에서 요구하는 표준을 바탕으로 Log-MAP 복호방식과 MIL 인터리버를 사용하여 모의 실험하였다. 반복 횟수는 자연과 요구되는 BER을 고려하여 4번까지 반복 복호하였고 각 stage에서의 형태를 단순화시켜 인터리버 크기변화에 따른 성능변화와 각 stage에서의 인터리버 형태변화에 따른 성능변화, 그리고 Mother 인터리버와의 성능을 비교하여 설명할 것이다.

2. 본 론

2.1 터보부호의 구조

터보부호의 구조는 크게 부호기, 복호기, 인터리버로 구분할 수 있다. 본문에서는 일반적으로 사용되는 터보부호의 구조를 설명한다.

2.1.1 터보부호의 부호기

터보부호의 구조는 그림과 같이 RSC부호기를 병렬로 연결하여 사용한다. 부호기의 출력 X_k, Y_{1k}, Y_{2k} 는 생성 다항식(G)에 의해 결정되고 부호율은 각각의 출력이 평처링되어 결정된다. W-CDMA시스템에서 생성다항식 $G(D)$ 는 다음과 같이 delay 함수를 사용하여 나타낸다.

$$G(D) = [1, \frac{n(D)}{d(D)}] \quad (2.1.1)$$

$$d(D) = 1 + D^2 + D^3 \quad (2.1.2)$$

$$n(D) = 1 + D + D^3 \quad (2.1.3)$$

여기서, $d(D)$ 는 부호어를 생성하기 위한 항이 되고 $n(D)$ 는 회귀되는 항을 8진수 형태로 표현한 식이다. 위식 (2.1.2)와 (2.1.3)을 생성다항식으로 갖는 터보부호기가 그림 1에 나타나 있다.

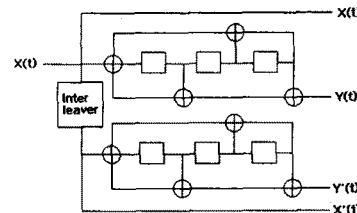


그림 1 터보부호기의 구조

2.1.2 터보부호의 복호기

그림 2는 두 개의 RSC부호기에 해당되는 복호기를 나타내고 있다. 첫 번째 복호기 DEC1에 의해서 복호된 d_k 에 관한 $A_1(d_k)$, 즉 LLR은 두 번째 복호기 DEC2에 대한 정보로 사용되고, 두 번째 복호기에서 나오는 출력은 전송된 채널에서 오류가 없을 경우 원래의 전송비트와 동일한 값이 되지만 전송채널에서 오류가 발생된다면 출력값이 원래의 전송 비트와 다르므로 반복적으로 복호를 수행하여 오류를 개선하는 방법을 사용한다.

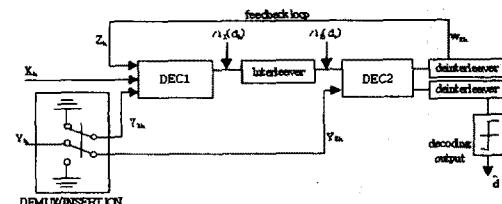


그림 2 터보부호의 복호기 구조.

2.2 터보부호의 인터리버

터보부호의 인터리버는 연집오류(Burst error)를 랜덤하게 변환시키는 역할을 한다. 대표적인 인터리버로는 블록, 랜덤, MIL, Mother, GF인터리버 등이 있으며 MIL, Mother, GF 인터리버가 우수한 성능을 내는 것으로 알려져 있다[3]. 본문에서는 본 논문에서 사용된 MIL 인터리버에 대해서만 설명하기로 한다.

MIL 인터리버는 ARIB에서 IMT-2000시스템의 무선 전송 기술 중 터보부호의 인터리버 규격으로 제안한 방식으로 인터리빙 된 이전 stage의 출력이 다음 stage에서 다시 인터리빙

되는 다중 단계 블록 인터리버를 말한다[4]. 처음 stage에서 입력 데이터는 row-by-row 순서로 인터리버에 쓰여지고 column-by-column 순서로 읽혀진다. 처음 stage의 인터리빙 과정은 $R \times C$ 크기의 길쌈부호의 블록 인터리빙과 동일하다. 두번째 stage에서 각 열 데이터 sequence들은 두 번째 stage의 블록인터리버에 row-by-row 순서로 입력된다. 두 번째 stage 인터리버에서의 출력 sequence는 다시 column-by-column 순서로 읽혀져 세 번째 stage에 입력된다. 이러한 과정을 거치게되면 블록 인터리버에서 입력정보의 인접 비트간 간격을 분리시키는 것보다 더 크게 분리시킬 수 있다.

2.3 복호방식

터보부호의 복호방식은 크게 Bahl 알고리즘을 변형시킨 MAP와 Viterbi 알고리즘을 변형시킨 SOVA로 분류할 수 있다. 성능은 MAP알고리즘이 SOVA방식에 비해 0.7dB정도의 좋다고 알려져 있지만[5] 복잡도가 SOVA방식에 비해 크다. 그리고 본 논문에서 사용한 Log-MAP알고리즘은 MAP알고리즘의 연산량과 사용되는 메모리양을 개선한 복호방식으로 성능면에서 MAP알고리즘과 큰 차이가 없다. 본문에서는 MAP알고리즘을 중심으로 설명한다.

2.3.1 MAP알고리즘

MAP 알고리즘은 1974년 Bahl 등에 의해 제시된 알고리즘을 변형시킨 형태이다. 시간 k 에서 부호를 $1/2$ 인 RSC 부호기의 상태가 S_k 일 때 정보 비트 d_k 가 입력된다고 하면 부호화 되지 않은 원래의 입력 정보 비트 d_k 와 부가 비트 y_k 가 출력되고 부호기의 상태는 S_{k+1} 로 변환된다. 이 출력이 변조되어 AWGN 채널로 보내지고 수신단에서의 수신 시퀀스는

$$R_1^N = \{R_1, R_2, \dots, R_k, \dots, R_N\} \quad (2.3.1)$$

이때, $R_k = (x_k, y_k)$ 로 시간 k 에서의 수신 심볼이다. x_k 와 y_k 는 다음과 같이 쓸 수 있다.

$$\begin{aligned} x_k &= (2d_k - 1) + p_k \\ y_k &= (2Y_k - 1) + q_k \end{aligned} \quad (2.3.2)$$

p_k 와 q_k 는 분산 σ^2 을 가지는 독립변수이다

이 채널의 끝에서 로그상관율(LLR), $L(d_k)$ 는 다음과 같이 정의할 수 있다.

$$L(d_k) = \log \frac{P(d_k=1 | \text{observation})}{P(d_k=0 | \text{observation})} \quad (2.3.3)$$

또한 d_k 의 APP는 다음과 같이 정의되는 λ 의 결합확률식을 이용하여 변형시킬 수 있다.

$$\lambda_k^i(m) = P(d_k=i, S_k=m | R_1^N) \quad (2.3.4)$$

$$L(d_k) = \log \frac{\sum_m \lambda_k^1(m)}{\sum_m \lambda_k^0(m)} \quad (2.3.5)$$

또한, 다음과 같이 정의되는 α 와 β 를 사용하여 결합확률식을 변형하여 다음과 같이 나타낼 수 있다.

$$\begin{aligned} \alpha_k^i(m) &= P(d_k=i, S_k=m, R_1^k) \\ &= \delta(R_k, m) \sum_{j=0}^1 \alpha_{k-1}^j(S_j^i(m)) \end{aligned} \quad (2.3.6)$$

$$\begin{aligned} \beta_k^i(m) &= P(R_{k+1}^N | d_k=i, S_k=m) \\ &= \sum_{j=0}^1 \beta_{k+1}^j(S_f^i(m)) \delta(R_{k+1}, S_f^i(m)) \end{aligned} \quad (2.3.7)$$

$$L(d_k) = \log \frac{\sum_m \alpha_k^1(m) \beta_k^1(m)}{\sum_m \alpha_k^0(m) \beta_k^0(m)} \quad (2.3.8)$$

여기서 $S_f^i(m)$ 은 $S_k = m$ 이고, $d_k = i$ 일 때 다음 상태를 말한다. α_k^i 는 시간 k 에서 α_k^i 의 과거값을 참조하는 순방향 반복(Forward Recursive) 함수처럼 표현되고 β_k^i 는 시간 k 에서 β_k^i 의 미래값을 참조하는 역방향 반복(Backward Recursive) 함수처럼 표현된다.

2.3.1 Log-MAP알고리즘

MAP알고리즘은 비터비 알고리즘에 비해 약 4배정도의 복잡도를 갖는다고 알려져 있다. Log-MAP알고리즘은 이러한 복잡도를 줄이기 위해 곱셈연산을 로그를 이용하여 덧셈으로 연산할 수 있도록 한 알고리즘이다[6]. Log-MAP복호기는 각각의 상태에서 다음 상태로의 전이에 따른 branch metric을 계산하는 BMC(branch metric calculator), 역방향 metric β 에 따른 상태를 계산하는 RSMC(reverse state metric calculator), 순방향 metric α 에 따른 상태를 계산하는 FSMC(forward state metric calculator), 로그상관률을 계산하는 LLRC(log likelihood ratio calculator)로 구성된다. 여기에 많은 시간이 사용되는 계산상의 부분이 E함수 테이블로 추가된다. E함수는

$$\begin{aligned} xEy &= -\ln(e^x + e^y) \\ &= x - \ln(1 + e^{x-y}) \\ &= y - \ln(e^{y-x} + 1) \\ &= \min(x, y) - \ln(1 + e^{-|x-y|}) \end{aligned} \quad (2.3.9)$$

로 정의 할 수 있다. E함수를 이용하여 (2.3.6)과 (2.3.7)의 로그값을 계산하면 다음과 같고

$$\begin{aligned} \ln \alpha_k^i(m) &= \ln[\delta(R_k, m) \sum_{j=0}^1 \alpha_{k-1}^j(S_j^i(m))] \\ &= D_i(R_k, m) + \ln[\sum_{j=0}^1 \alpha_{k-1}^j(S_j^i(m))] \\ &= D_i(R_k, m) + E_{j=0}^1 A_{k-1}^j(S_j^i(m)) \\ &= A_k^i(m) \end{aligned} \quad (2.3.10)$$

$$\begin{aligned} \ln \beta_k^i(m) &= \ln[\sum_{j=0}^1 \beta_{k+1}^j(S_f^i(m)) \delta(R_{k+1}, S_f^i(m))] \\ &= E_{j=0}^1 B_{k+1}^j(S_f^i(m)) \delta(R_{k+1}, S_f^i(m)) \\ &= B_k^i(m) \end{aligned} \quad (2.3.11)$$

$$\begin{aligned} \ln \delta(R_k, m) &= D(R_k, m) \\ &= (\frac{2}{\sigma^2} (x_k d_k + y_k Y_k)) \end{aligned} \quad (2.3.12)$$

위 식들을 식(2.3.8)에 대입하면 다음과 같이 로그상관율 $L(d_k)$ 를 정의 할 수 있다.

$$L(d_k) = E_{m=0}^{2^{m-1}} [A_k^1(m) + B_k^1(m)] - E_{m=0}^{2^{m-1}} [A_k^0(m) + B_k^0(m)] \quad (2.3.12)$$

2.4 모의실험 및 결과

W-CDMA시스템에서는 데이터 전송의 경우 예러 정정

성능이 우수한 터보부호를 사용하고 음성서비스의 경우 길쌈부호를 사용한다. 모의실험에서는 W-CDMA시스템에서 요구하는 $10^{-3} \sim 10^{-6}$ 에서의 BER변화에 따른 SNR의 성능을 인터리버 크기를 다르게 한 경우와 각 stage에서의 인터리버 형태를 달리한 경우에 대해서 모의 실험하고 Mother 인터리버를 사용한 경우와 비교한다.

2.4.1 인터리버 크기에 따른 성능 비교

그림 3과 4는 인터리버의 크기에 따른 터보부호의 성능을 나타낸다. 랜덤 인터리버와 마찬가지로 인터리버의 크기가 커질수록 성능도 좋게 나온다. 하지만 성능개선 정도는 랜덤 인터리버에 비해 크지 않게 나타난다.

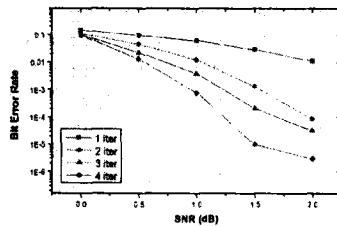


그림 3 인터리버크기 N=1024

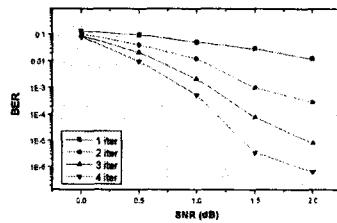


그림 4 인터리버크기 N=5120

2.4.2 인터리버 형태 변화에 따른 성능

그림 5는 각각의 stage에서 인터리버 형태변화에 따른 성능을 나타내는 그림이다. 모의 실험결과 다중 stage 인터리버의 형태 변화에 따른 성능은 크게 차이가 나지 않는 것으로 나타난다.

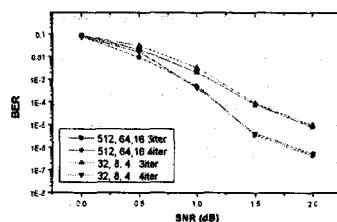


그림 5 Stage크기에 따른 성능비교

2.4.3 Mother 인터리버와 비교

그림 6과 7은 인터리버 크기가 1024와 5120인 Mother인터리버와 MIL인터리버의 성능을 비교한 것이다. 그림에서 볼 수 있듯이 인터리버 크기가 1024일 때 BER 10^{-5} 에서 MIL인터리버가 약 0.35dB정도 5120일 경우 0.3 dB 우수한 성능을 내는 것으로 나타났다. 또한 10^{-3} 일 경우는 이 보다 더 큰 성능 차이가 나타난다.

다.

$10^{-3} \sim 10^{-6}$ 의 BER을 요구하는 W-CDMA시스템에서의 요구조건에는 MIL인터리버가 더 낮은 SNR로 만족시킬 수 있다.

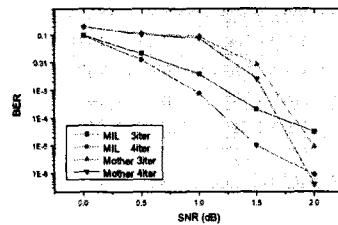


그림 6 터보부호의 성능(N=1024)

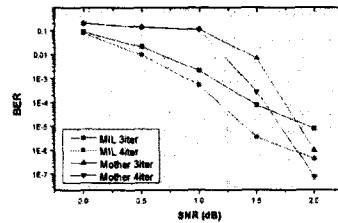


그림 7 터보부호의 성능 (N=5120)

3. 결 론

본 논문에서는 3stage MIL 인터리버와 Log-MAP복호방식을 사용하여 W-CDMA 시스템 환경에서의 성능을 모의 실험하였고, 통신 채널은 W-CDMA 시스템 특성상 적응적 Fast TPC와 Rake 수신기의 사용으로 다중 경로 페이딩을 무시하고 AWGN 채널만을 고려하였다. 현재 W-CDMA시스템에서는 Mother 인터리버에 비중을 두고 있는데 모의 실험한 결과 W-CDMA시스템의 요구조건을 만족시키는 구간에서 MIL 인터리버가 Mother 인터리버 보다 0.3 dB이상 우수한 성능을 내는 것으로 나타났다. 또한 모의 실험에서는 MIL 인터리버의 stage별 인터리버 형태를 같게 단순화 시켜 실험하였기 때문에 각각 다른 형태의 인터리버를 사용할 경우와 채널 인터리버로 사용되는 블록 인터리버를 MIL 인터리버로 대체하여 사용하는 것도 우수한 성능을 낼 수 있을 것이다.

(참 고 문 헌)

- [1] C. Berrou, A. Glavieux, and P. Thitimajshima, "Near Shannon limit error-correcting coding and decoding: Turbo-codes", in Proc. IEEE Int. Conf. on Communication, pp. 1640-1670, May 1993.
- [2] 3GPP technical specification, <http://www.3gpp.org>.
- [3] 이문호, "IMT-2000에서 채널 코드의 터보 인터리버", 전자전총, 9권, 6호, 1999.
- [4] Akira Shibusaki, Hirohito Ssuda, and Fumiayuki Adachi, "Multi-Stage Interleaver for Turbo Codes in DS-CDMA mobile radio", APCC/ICCS '98, pp. 391-394, 1998.
- [5] L. Papke, P. Robertson, and E. Villebrun, "Improved decoding with the SOVA in a parallel concatenated(turbo-code) scheme", Proc. of ICC '96, pp.102-106, June 1996.
- [6] S. S. Pietrobon and A. S. Barbulessu, "A simplification of the modified Bahl decoding algorithm for systematic convolutional codes", Int. Symp. Inform. Theory & its Applic., pp.1073-1077, Nov. 1994.