심볼 단위로 복호화가 이루어지는 터보부호에 관한 연구

김태국, 심용걸
단국대학교 전자공학과
전화 : 041-550-3540

A Study on Symbol-based Decoding for Turbo Codes

Tae-Kook Kim, Yong-Geol Shim
Dept, of Electronics Engineering, Dankook University
E-mail : taekooks@hanmail.net

Abstract

In this thesis we propose symbol-based decoding for turbo codes which is used SOVA as a decoding algorithm. The proposed turbo codes that is interleaved on symbol-by-symbol basis and inputted as n-bit symbols to the decoder. This method makes that the stages of the original trellis are merged together and the trellis depth is reduced by 1/n. We research turbo codes with symbol size n=2 in this paper and its performance.

Ⅰ. 서론

터보부호(Turbo Codes)는 Berrou 등이 발표한 "Near Shannon Limit Error Correcting Coding and Decoding"에서 처음 제안된 채널 코딩 기법으로, 인터리버의 크기가 충분히 클수록 Shannon의 채널용량 한계에 근접하는 향상된 오류정정능력을 가지고 있다[1][2]. 최대 2Mbps까지의 전송률을 요구하는 제 3세대 무선 이동 통신인 IMT-2000 시스템에서는 고속 멀티미디어 데이터 전송이 요구되는 부분에서 대부분 터보부호의 사용이 채택되어 있지만, 터보부호의 단점인 반복복호에 따른 지연시간을 줄이는 것이 대단히 중요하다. 이러한 문제를 해결하기 위하여 본 논문에서는 복호 지연을 줄일 수 있도록 심볼 단위로 복호가 이루어지는 터보부호를 제안한다.

기존의 터보부호가 비트 단위로 정보가 입력되는데 비해서 심볼 단위로 복호가 이루어지는 터보부호는 n비트의 심볼로 입력되는 블록을 처리하게 된다. IMT-2000에서 제공하는 동영상 등의 적은 지연을 필요로 하는 서비스에서 본 논문에서 제안하는 터보부호를 적용할 수 있다. 입력 데이터 블록을 n 비트의 심볼로 처리함에 의해서 부호기 트렐리스의 n개 stage를 하나로 결합하게 되므로 복잡도는 커지지만 메모리의 요구는 적어진다. 따라서, 적은 수의 반복복호횟수로도 원하는 성능을 얻을 수 있으므로 지연을 줄일 수 있다. 만약, 심볼단위의 복호과정에 MAP알고리즘을 적용한다면 MAP 알고리즘이 갖는 복잡도의 증가라는 단점에 트렐리스의 결합에 의한 경로 개수의 증가 효과가 겹쳐져서 복잡도가 더욱 늘어나게 된다. 그러므로 MAP알고리즘보다 성능면에서는 약간의 열화를 보이지만, 더 적은 복잡도를 갖는 SOVA를 사용함으로써 기존의 터보부호에 비하여 크게 증가되지 않는 복잡도를 가지면서도 성능이 우수한 복호방법을 제안하고자 한다.

Ⅱ. 터보부호

터보 부호기는 기본적으로 2개 이상의 구성부호기와 터보 인터리버로 구성된다. 일반적으로 구성부호기로는 생성다항식이 같은 길쌈부호기를 사용하는 경우가 많다.

그림 1은 기본적인 터보부호의 부호기인 PCCC(Parallel Concatenated Convolutional Code) 부호기를 나타낸 것이다.

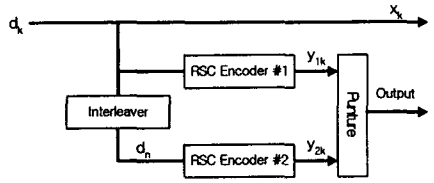

그림 1. PCCC 부호기의 구조

터보부호의 출력은 길쌈부호기의 출력뿐만 아니라 인터리버에 의해 변형된 출력인 두 개의 패리티 정보를 얻게 되지만, 인터리버에 의해서 입력이 지연된다. 부호기의 출력단에서 출력 비트인 Y_{1k}와 Y_{2k}를 모두 전송하면 부호율이 1/3이 된다. 부호기의 출력 비트들을 천공행렬을 사용해서 천공하여 변형된 부호율 1/2로 전송할때 천공은 정보 비트가 아닌 잉여비트(redundancy bit)에 대해서만 적용한다.

터보부호 복호기는 두 개의 복호기 사이에 궤환루프(feedback loop)를 통하여 부가(extrinsic)정보의 교환이 이루어지므로 반복적으로 복호를 수행할 때 반복횟수가 증가함에 따라서 BER성능이 향상된다. 다른 복호기로부터 얻은 정보를 입력으로 받는데, 이 정보는 사전 확률값으로 이용한다.

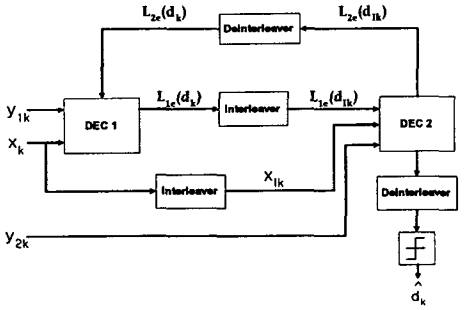

그림 2. 터보 복호기의 구조

각 구성 복호기에서 연성출력을 구하기 위한 복호 알고리즘으로 SOVA 알고리즘을 사용하며 기존의 비터비 알고리즘에 부가정보를 갱신하는 과정을 첨가함으로써 복호를 수행한다[3]. SOVA 복호기는 LLR 입력 $L(u)$와 $L_c y$을 처리하며 추정된 정보 시퀀스인 $\hat{u}$와 관련된 LLR 시퀀스인 $L(\hat{u})$를 출력한다. 조직적 부호에 대한 SOVA 메트릭은 다음과 같다.

$$M_t^{(m)} = M_{t-1}^{(m)} + u_t^{(m)} L_{cy_{t,1}} + \sum_{j=2}^{N} x_{t,j}^{(m)} L_{ct,j} y_{t,j} + u_t^{(m)} L(u_t) \quad (1)$$

이상과 같이 SOVA 메트릭은 지난 메트릭과 채널 신뢰값, 신뢰값(사전 정보)을 결합한다.

시간 t에서 트렐리스 내의 Node에 할당된 기억 단계 MEM에서 신뢰값은 다음과 같다.

$$\Delta_t^{MEM} = \frac{1}{2} \mid M_t^{(1)} - M_t^{(2)} \mid \quad (2)$$

이 노드에 대해, 만일 $MEM = k$ 에서의 생존경로상의 비트가 경쟁경로상의 비트와 동일하다면, 경쟁경로가 선택되더라도 오류가 없을 것이므로 신뢰값은 변하지 않고 남아있게 된다. 반대로 경로상의 비트가 다르면 비트 오류가 발생한 것으로 이 비트 오류 위치에서의 신뢰값은 갱신과정을 통하여 갱신되어야 한다. 신뢰값 갱신은 "soft" 또는 L값을 향상시키기 위해 수행된다. 즉,

$$L(\hat{u}_{t-MEM}) = \hat{u}_{t-MEM} \sum_{k=0}^{MEM} \Delta_t^k \approx \hat{u}_{t-MEM} \min_{k=0,\cdots,MEM} \{\Delta_t^k\} \quad (3)$$

SOVA 복호기의 순방향 수행은 모든 생존경로들을 갱신해야하기 때문에 크기가 큰 프레임과 구속장이 클 경우 많은 계산량을 필요로 한다. 갱신과정은 단지 ML 경로에 대해서만 의미가 있으므로, SOVA 복호기의 수행은 단지 ML 경로에 대해서만 갱신과정을 수행한다.

SOVA 복호기는 추정된 비트 $\hat{u}_t$에 대한 L값 $L(\hat{u})$을 생성한다. $L(\hat{u})$는 세 개의 거리 조건으로 재구성될 수 있는데, 복호기들 사이에 교환되는 정보는 외부정보이므로 다음과 같이 놓을 수 있다[4].

$$L_e(\hat{u}_t) = L(\hat{u}_t) - L(u_t) - L_c y_{t,1} \quad (4)$$

사전 정보 $L(u_t)$는 L값 정보가 복호기로 되돌아가는 것을 막기 위해 $L(\hat{u}_t)$에서 제거된다. 또한, 수신된 조직적 채널 정보 $L_c y_{t,1}$가 복호기들에서 공통된 정보를 없애기 위해 제거된다. 폐쇄 루프 복호 구조에서 각각의 복호기는 다른 무게의 패리티 검사 열을 사용하여 정보 시퀀스를 추정한다. 복호기는 복호 성능을 향상시키기 위해, 다른 무게의 패리티 검사 열로부터 사전 추정값과 더 믿을 수 있는 신뢰값을 공급하기 위해 반복 복호를 수행한다.

III. 제안된 복호 알고리즘

본 논문에서는 심볼 단위로 복호가 이루어지는 SOVA 알고리즘을 제안하여 기존의 터보부호가 비트 단위로 정보가 입력되는데 비해서 제안된 터보부호는

symbol-by-symbol 인터리브되고 n 비트의 심볼로 입력되는 블록을 처리하게 된다. 그림 3과 같이 기존의 트레리스 상에서 한 stage당 한 비트씩 복호되었지만 제안된 터보부호는 n-비트 심볼에 의해 n개의 stage 당 하나의 심볼을 복호하게 된다. 그럼으로써 트렐리스의 길이가 1/n배만큼 줄어드는 효과를 얻는다. 심볼 단위로 복호를 수행하였을 때 기존의 SOVA 복호 수행시의 각 노드에서의 신뢰값의 계산은 트렐리스의 결합으로 인해 1/n배만큼 줄어들게 되고 경로 메트릭은 먼저 상태로부터 다음 상태로 천이 할 때 2개의 메트릭 중에서 생존경로 메트릭과 경쟁경로 메트릭이 각각 하나씩 존재했지만, 이러한 결합으로 인해 2^n개의 메트릭이 생겨나게 된다. 또한 트렐리스의 결합으로 인해 갱신을 할 신뢰값의 개수가 1/n으로 줄어들기 때문에 필요한 메모리는 줄어들게 되지만 계산량은 늘어나는 단점을 갖는다. 경로 메트릭의 계산량 관점에서 제안된 알고리즘은 1.67배 정도의 계산량 증가를 보인다. 심볼크기가 n=2일 때의 경로 메트릭중 최대값이 생존경로인 $M_t^{(1st)}$이며 나머지 메트릭중 최대값이 경쟁경로인 $M_t^{(2nd)}$가 되고 그 다음 최대값을 $M_t^{(3rd)}$, $M_t^{(4th)}$로 각각 놓을 수 있다. 그런데 심볼크기 n이 커지면 $M_t^{(3rd)}$가 경쟁경로 메트릭보다 매우 작다고 볼 수 있다. 즉,

$$M^{(3rd)} \ll M^{(2nd)} \qquad (5)$$

그러므로 시간 t, MEM=k에서 트렐리스내의 Node에 할당된 신뢰값은

$$\triangle_t^k \approx \frac{1}{2} \mid M_t^{(1st)} - M_t^{(2nd)} \mid \qquad (6)$$

따라서, 정확한 생존경로를 선택할 확률은

$$\begin{aligned} P(correct) &\approx \frac{P(path(1st))}{P(path(1st)) + P(path(2nd))} \\ &= \frac{e^{\frac{M_t^{(1st)}}{2}}}{e^{\frac{M_t^{(1st)}}{2}} + e^{\frac{M_t^{(2nd)}}{2}}} \\ &= \frac{e^{\triangle_t^k}}{1 + e^{\triangle_t^k}} \end{aligned} \qquad (7)$$

$$P(incorrect) = 1 - P(correct) \approx \frac{1}{1 + e^{\triangle_t^k}} \qquad (8)$$ 이다.

이런 경로 결정의 신뢰값은 다음과 같다.

$$\log \frac{P(correct)}{P(incorrect)} = \log \frac{\frac{e^{\triangle_t^k}}{1 + e^{\triangle_t^k}}}{\frac{1}{1 + e^{\triangle_t^k}}} = \triangle_t^k \qquad (9)$$

그러므로, 정보심볼의 사후정보값은 다음과 같다.

$$\begin{aligned} L(\hat{u} = a^{1st}) &\approx \hat{u} \cdot \min_{k=0,\cdots,MEM} \left\{ \log \frac{P(correct)}{P(incorrect)} \right\} \\ &= \hat{u} \cdot \min_{k=0,\cdots,MEM} \{ \triangle_t^k \} \\ &= \hat{u} \cdot \triangle_t^k \end{aligned} \qquad (10)$$

여기서 a^{1st}는 생존경로에서의 정보심볼이다.

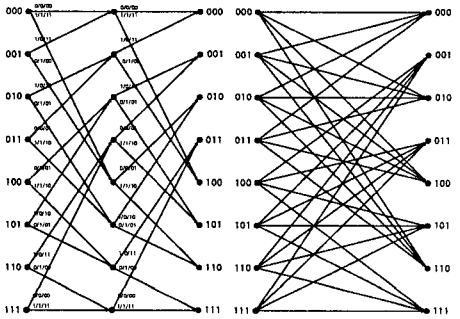

그림 3. n=1과 n=2일 때의 트렐리스 다이어그램

IV. 모의실험 결과 및 분석

이번 절에서는 제안된 알고리즘을 이용한 터보부호와 기존의 터보부호를 모의실험을 통하여 비교하여보았다. 모의실험에 사용된 파라미터들은 표 1과 같다. 그림 4과

부호율	R=1/3
인터리버 크기	N=192, 256, 512
구속장, 생성다항식	K=4 $(15, 13)_8$
인터리버	블록 인터리버
반복복호횟수	가변
변조	BPSK

표 1. 모의실험에 사용된 파라미터

그림 5에서 반복복호를 4회와 8회 실행하여 비교한 결과에서 반복복호횟수가 4일 때의 성능은 1.5dB까지 거의 같지만 그 이상부터는 성능이 개선되어짐이 보여진다. 인터리버 크기 N=256일 때의 터보부호에서 비트 오류확률 10^{-3}에서 봤을 때 제안된 SOVA 알고리즘은 반복복호 4회일 때 0.2dB정도의 성능개선을 가지며, 반복복호가 8회인 기존 SOVA에 비해 0.1dB정도 성능 차이가 보여진다. 그림 6은 N=512일 때 터보부호의 에러확률이며, 비트 오류확률 10^{-3}에서의 제안된 SOVA는 2.5dB에서 3dB로 갈수록 기존의 SOVA와 성능이 비슷해짐을 볼 수 있다.

V. 결론

심볼 단위로 복호가 이루어지는 터보부호는 복호시의 트렐리스 stage들의 결합으로 인해 트렐리스의 길이가 1/n배로 줄어들게 되지만, 결합에 의해 생긴 메트릭 개수의 증가로 인해 복호 복잡도의 증가를 갖게된다. 그러나 트렐리스 결합으로 인하여 복호시의 신뢰값 계산량이 감소하며 갱신할 신뢰값의 개수도 같은 비율로 줄어들게 되어 복호에 필요한 메모리가 줄어드는 효과를 얻는다. 결국 복잡도의 증가라는 단점이 있지만 기존의 터보부호와 비교해서 더 적은 RAM이 복호기에 필요하다는 것을 알 수 있다. 본 논문에서 사용한 심볼 크기가 n=2일 때의 터보부호는 기존의 터보부호에 비해 같은 반복복호횟수에 대해서 개선된 성능을 갖고, 더 적은 반복복호횟수로도 같은 성능을 얻을 수 있다. 또한 복호기의 메모리 요구량도 감소함을 알 수 있다. 향후에는 실제의 IMT-2000 시스템에 적용하여 더욱 구체적이고 정밀하게 성능을 평가하고, 그 타당성을 입증하는 것이 필요하다.

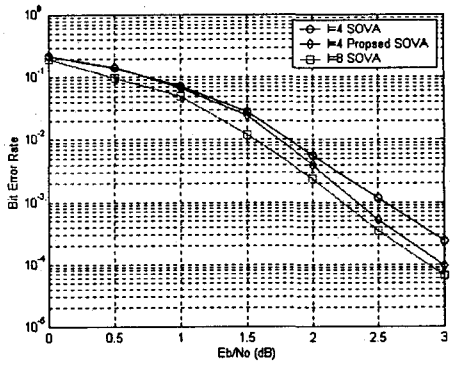

그림 4. N=192에서 제안된 SOVA의 성능비교

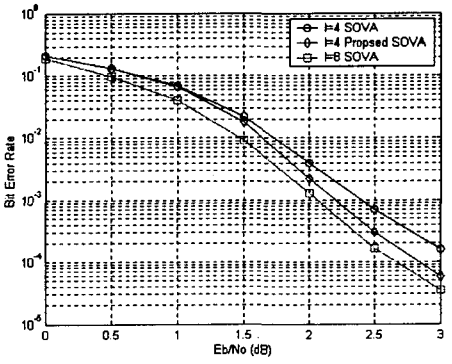

그림 5. N=256에서 제안된 SOVA의 성능비교

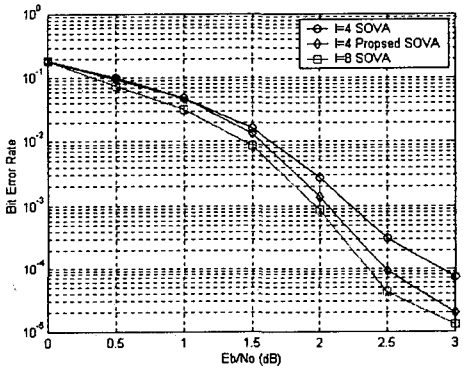

그림 6. N=512에서 제안된 SOVA의 성능비교

References

[1] C. Berrou, A. Glavieux, and P. Thitimajshma, "Near Shannon Limit Error-Correction Coding and Decoding : Turbo-Codes", Proc. ICC, pp. 1064-1070, Geneva, Switzerland, May 1993.

[2] L. Lin and R. Cheng, "Improvements In SOVA-Based Decoding For Turbo Codes", Proc. ICC, pp1473-1478, June 1997

[3] J. Hagenauer and P. Hoeher, "A Viterbi algorithm with soft-decision outputs and its applications", IEEE GLOBECOM '89, pp. 47.1.1-47.1.7, November 1989

[4] Hagenauer, J. and Papke, L.,"Decoding "Turbo" Codes with the Soft Output Viterbi Algorithm(SOVA)," Proceedings of International Symposium on Information Theory, Trondheim, Norway, p.164, June 1994.