



Convolutional Code와 Turbo Code의 Decoder State 수 변경을 이용한 비 균등 오류 정정 부호의 설계

박진수, 남미영, 송민규, 전석민, 송홍엽

Coding and Crypto Lab.
Yonsei University, Seoul, Korea

2012 / 2 / 9

한국통신학회 동계종합학술발표회



I. 서 론



- Unequal Error Protection(UEP)
 - 서로 다른 중요도를 갖는 부분을 서로 다른 강도로 보호
 - More significant bits(MSB)
 - Less significant bits(LSB)
 - Equal error protection(EEP)와 상반되는 개념
- Convolutional Code와 Turbo Code의 UEP
 - 각 message bit에 서로 다른 수의 parity bit들을 대응시키는 방법이 일반적
 - Puncturing
 - Path pruning
- 새로운 UEP Convolutional Code의 설계 방법
 - LSB보다 MSB가 더 많은 state를 갖도록 구성
 - EEP와 같은 decoding complexity를 갖도록 파라미터를 결정



II. 본 론



- BCJR Decoding를 기준으로
 - Viterbi decoding algorithm도 마찬가지로 적용 가능
 - 결과적으로 같은 parameter를 갖게 됨
- EEP Convolutional Codes
 - k message bit수
 - n codeword 길이
 - ν encoder에서 사용하는 메모리 수
 - $S=2^\nu$ decoder의 한 stage에 필요한 state 수
- Decoding Complexity of EEP Convolutional Codes
 - BCJR algorithm의 α, β, γ 연산은 각 state당 S 번 필요
 - 각 state node 당 각 연산의 complexity를 $C(\alpha), C(\beta), C(\gamma)$ 라 하고
 - 처음과 마지막 각각 ν 개의 stage에 대한 오차를 감수하여
 - 총 decoding complexity를 다음과 같이 근사화 가능

$$C(\nu) \approx k \cdot S(C(\alpha) + C(\beta) + C(\gamma)) \quad (1)$$



II. 본 론



- UEP Convolutional Codes

- EEP convolutional code와 같은 k 를 갖는다고 가정
- 전체 message bit 중 MSB의 비율 $0 < t < 1$
- MSB와 LSB가 각각 서로 다른 decoding state 수를 갖도록 하자
 - MSB와 LSB가 서로 다른 수의 memory를 사용
 - Decoding state 수의 증가를 통한 부분적 성능 향상 도모
- ν_m input이 MSB일 때 사용하는 메모리 수
- ν_l input이 LSB일 때 사용하는 메모리 수
- $S_m = 2^{\nu_m}$ MSB에 해당하는 한 decoding stage의 state 수
- $S_l = 2^{\nu_l}$ LSB에 해당하는 한 decoding stage의 state 수

- Decoding Complexity of UEP Convolutional Codes

- 각 α, β, γ 연산에 대해 총 decoding complexity를 다음과 같이 근사화 가능

$$C(\nu_m, \nu_l) \approx (tk \cdot S_m + (1-t)k \cdot S_l) \cdot (C(\alpha) + C(\beta) + C(\gamma)) \quad (2)$$



II. 본 론



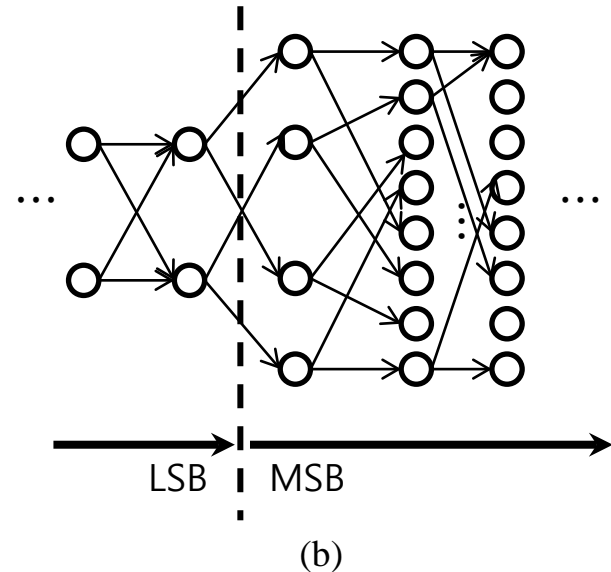
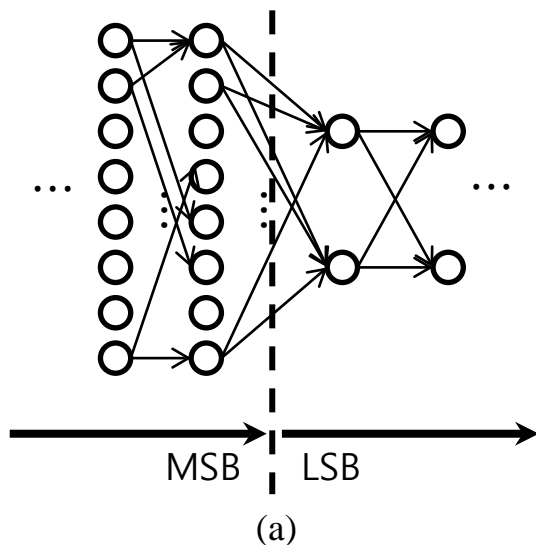
- EEP Convolutional Codes vs UEP Convolutional Codes
 - (1) 식과 (2) 식을 같게하는 파라미터들은 EEP convolutional code와 UEP convolutional code의 decoding complexity를 같게 한다.

$$S = tS_m + (1-t)S_l$$

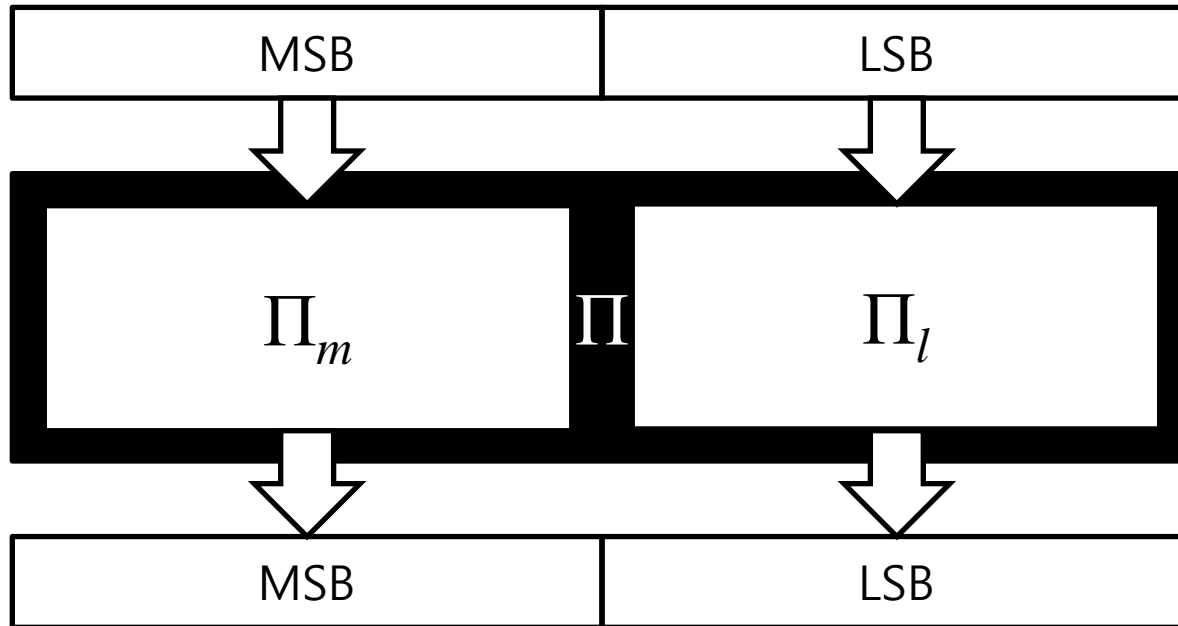
$$2^\nu = t2^{\nu_m} + (1-t)2^{\nu_l} \quad (3)$$

- 더 많은 state를 사용할 때 더 좋은 성능을 갖을 것으로 기대
- 따라서, $\nu_m > \nu > \nu_l$ 을 가정
- 각 ν, ν_m, ν_l 을 결정하면, (3)에 의해 t 가 결정됨
- EEP, UEP convolutional code간 같은 decoding complexity를 갖는 파라미터가 결정됨

- Input Bits의 Encoding 순서
 - $v_m > v_l$ 일 때,
 - MSB가 LSB보다 더 먼저 encoding 되어야 함
 - LSB가 MSB보다 나중에 encoding 되면(b),
 - LSB에 해당하는 stage에 이어서 MSB가 들어오는 부분에서
 - Decoding trellis상의 state 수가 2배씩 증가됨
 - $v_m - v_l > 2$ 일 경우, MSB가 S_m 개의 state를 갖기 위해 2회 이상의 stage가 필요
 - 이는 MSB의 몇 bit들의 decoding 성능이 다른 MSB보다 낮아질 수 있음
 - MSB 다음 LSB가 encoding되면(a) 1 stage만에 state 수를 줄일 수 있음



- UEP Convolutional Codes를 이용한 UEP Turbo Codes
 - Interleaver의 사용
 - 각 bit들이 자신과 같은 중요도를 갖는 bit의 위치로 이동해야 함
 - 그렇지 못하면 LSB 이후 MSB가 encoding될 위험성이 있음



< UEP Turbo Codes를 위한 Interleaver의 구성 예시 >

III. 결 론

- EEP Convolutional Code

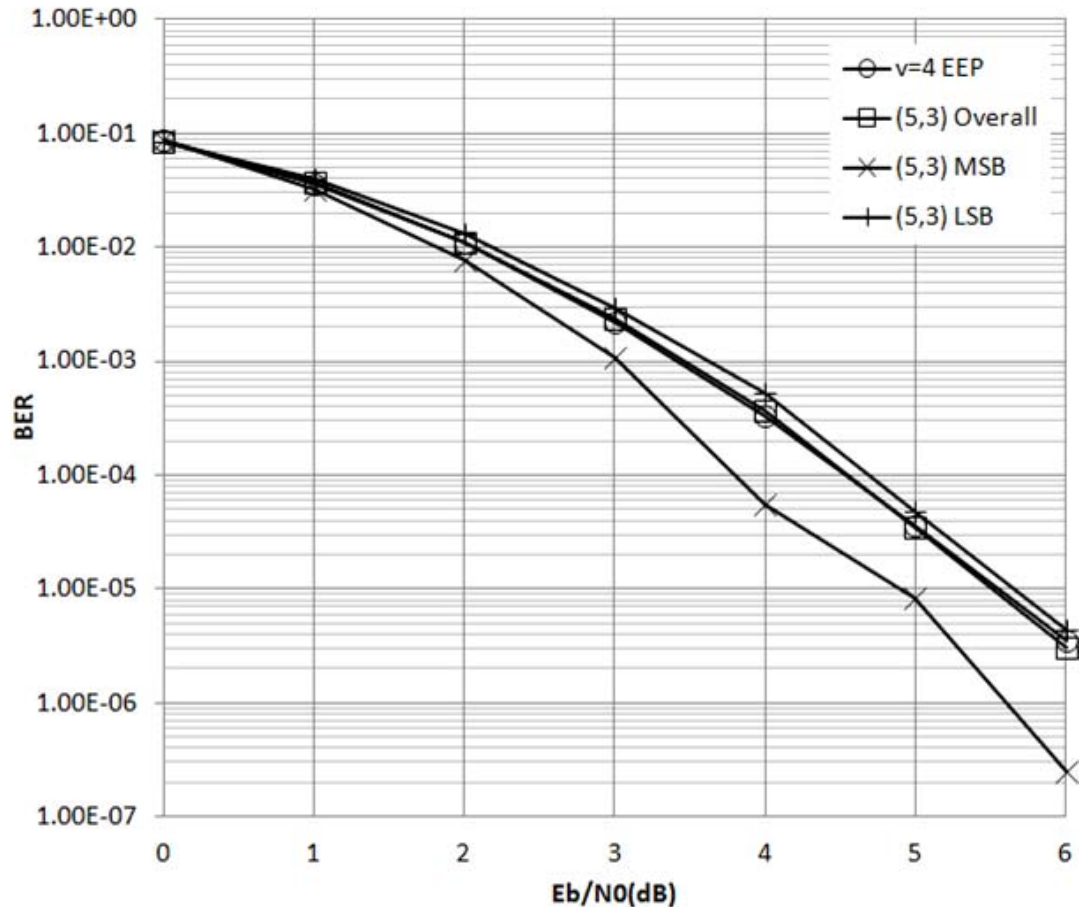
- $k = 1000$
- Code rate = $1/2$
- $v = 4$
- Recursive & Systematic
- BCJR decoding

- UEP Convolutional Code

- $k = 1000$
- Code rate = $1/2$
- $v_m = 5$
- $v_l = 3$
- $t = 1/3$
- Recursive & Systematic
- BCJR decoding

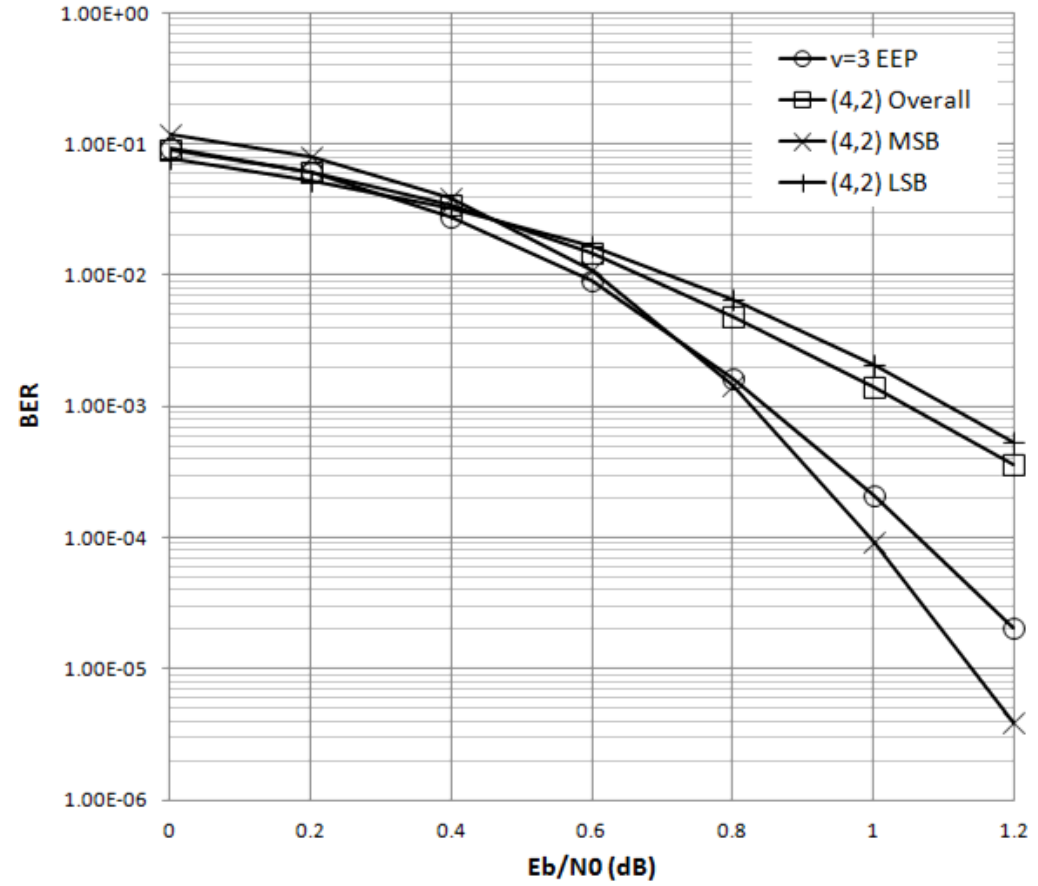
- 실험결과

- MSB의 성능이 향상됨
- UEP의 평균 성능이 EEP에 비해 크게 뒤쳐지지 않음



<EEP & UEP Convolutional Code의 성능 비교>

- EEP Turbo Code
 - EEP convolutional component codes
 - $k = 10000$
 - Code rate = $1/3$
 - $v = 3$
- UEP Turbo Code
 - UEP convolutional component codes
 - $k = 10000$
 - Code rate = $1/3$
 - $v_m = 4$
 - $v_l = 2$
 - $t = 1/3$
 - 3회의 iteration
 - MSB와 LSB에 대해 각각 random interleaver 적용



<EEP & UEP Turbo Code의 성능 비교>

- 실험결과
 - Convolutional code의 경우와 마찬가지로, MSB의 성능이 향상됨
 - Convolutional code의 경우 보다 평균 성능의 손해율이 더 높음



III. 결 론



- 본 논문에서는 convolutional code 기반의 새로운 UEP 방법을 제안
- 더 중요한 message가 더 많은 수의 decoding state를 사용하도록 구성
- EEP convolutional code와 같은 decoding complexity를 갖는 UEP convolutional code를 제안
- UEP convolutional code를 turbo code의 component code로 사용하여 UEP turbo code를 구성
- 제안된 방법으로 구성된 convolutional code와 turbo code의 UEP 효과를 컴퓨터 시뮬레이션으로 확인



참고문헌



- [1] G. Cair and G. Lechner, "Turbo codes with unequal error protection," *Electronics Letters*, vol. 32, no. 7, March 1996.
- [2] Zude Zhou, and Chao Xu, "An improved unequal error protection turbo codes," 2005 International Conference on Wireless Communications, Networking, and Mobile Computing, Wuhan, China, September 23-26, 2005.
- [3] Guisepppe Caire, and Ezio Biglieri, "Parallel Concatenated Codes with Unequal Error Protection," *IEEE Transactions on Communications*, vol. 46, no. 5, May 1998.
- [4] Werner Henkel, and Neele von Deetzen, "Path pruning for unequal error protection turbo codes," International Zurich Seminar on Communications (IZS), February 22-24, 2006.
- [5] L. R. Bahl, J. Cocke, F. Jelinek, and J. Raviv, "Optimal Decoding of Linear Codes for Minimizing Symbol Error Rate," *IEEE Transactions on Information Theory*, March 1974.