일반투고

Turbo Codes의 이론과 구조 및 성능

李 門 浩, 崔 勝 倍
全北大學校 情報通信工學科

I. 서 론

최근 관심이 고조되고 있는 채널 부호(Channel code)의 새로운 Turbo Codes에 대하여 소개한다. 부호이론(Coding Theory)이나 정보이론(Information Theory)에 대한 연구가 시작된 것은 역사적으로 오래전의 일이다. 이미 여러 가지 기본적인 생각들이 1940년대 이전에 정리되기 시작했으며, 그후 1948년에 C. E. Shannon에 의해 "통신의 수학적 이론(A Mathematical Theory of Communication)"에 관한 두편의 논문이 BSTJ (Bell System Technical Journal)에 발표되면서부터 정보에 관한 이론이 처음으로 확립되었다. 이 두편의 논문이 곧바로 정보이론 분야에 대단한 주목을 받기 시작하면서 이 분야의 여러 우수한 논문들이 BSTJ에 발표되기 시작하였다.

Shannon의 정보이론은 정보에 확률의 개념을 도입함으로써 정보를 비트로 표현, 계산된 정보량 및 정보 전송 속도의 통신 용량과의 관계, 그리고 이들을 이용하여 통신 용량의 한계 내에서의 오류없는 정보의 전송을 가능하게 하는 통신로에 관한 이론으로 압축될 수 있다. 그러나 이 당시의 그의 이론에서는 부호화 및 복호화에 관한 구체적인 방법들은 제시되지 않고 있었다.

한편, 그와 비슷한 시기에 부호이론이 소개되었는데, 기본적인 논문들은 특허권에 관한 문제 때문에 1950년 4월쯤에서야 BSTJ에 발표되었으며 Group 이론, Galois 이론, 사영기하학 등의 여러 가지 수학적 이론과 함께 선형 프로그래밍 기법들이 개발, 적용되기에 이르렀다. 이 시기의 대표적인 논문으로는 R. W. Hamming의 "Error Detecting and Error Correcting Codes"(BSTJ 1950)과 D. H. Huffman의 "A Method for the Construction of Minimum Redundancy Codes" (Proc. IRE, 1952)를 들 수 있다. 이 두편의 논문은 각각 채널 부호화(Channel Coding)와 소스 부호화(Source Coding)에 관한 것이며, 일반적으로 채널 부호화는 전송 정보에 중복도(Redundancy)를 증가시켜 전송 효율을 향상시키고자 하는 의도

이고, 이와 반대로 소스 부호화는 중복도를 최대한 감소시켜 전송 효율을 높이고자 하는 개념에서 출발한 것이다.

여기에서 채널 부호는 크게 블럭 부호(Block Codes)와 길쌈 부호(Convolutional Codes)로 나누어지며, 이 두 부호를 나누는 현저한 기준은 "기억(memory)의 존재 유무"에 있다고 할 수 있다. 즉, 개념적으로 볼 때 블럭 부호는 부호화된 부호어들이 상호 독립인 무기억 장치인 반면에 길쌈 부호는 그 출력 계열이 현재 뿐만 아니라 과거의 입력 계열에 영향을 받아 결정되기 때문이다.

또 블럭 부호는 선형 부호(Linear Codes)와 순회 부호(Cyclic Codes)로 나눌 수 있는데, n, k, n-k, R=k/n 및 dmin 등의 매개 변수를 사용하여 n은 부호 길이로 부호기 출력 계열의 블럭당 비트수를 의미하고 정보길이(Information bits) k는 정보 비트의 수를 뜻하는데, 그 실용치는 3에서부터 수백 비트까지에 이르고 n-k비트의 검사 길이는

〈표 1〉 대표적 부호의 비교

부 호	특성 및 장단점
Convolutional Codes – Viterbi Algorithm (1955, Elias & 1967, Viterbi)	1. 구속장 K의 증가에 따른 복잡성의 지수적 증가 (특히 K > 8 인 경우, 복호기의 구현이 어렵다.) 2. 긴 지연(delay)으로 인한 저속 전송(수십 K ~ 수십 Mbps) 3. 연집에러(burst error)에 약함(가우시안 채널에서 이상적) 4. 격자도 진행의 한계 (더이상 에러율이 향상되지 않는 임계점이 설정됨) 5. 10^{-4} 이상의 에러율에서 좋은 성능 6. 이동통신에 사용(CDMA)
Block Codes – Reed-Solomon Algorithm (1961, Reed & Solomon)	1. 에러 정정 능력 t에 따른 복잡도의 선형적 또는 지수적 증가 2. 정정 능력 t 이상의 에러는 수정 불가 3. 고속 전송이 가능(수백 Mbps) 4. 연집에러(burst error)에 강함(페이딩 채널에서 좋은 성능) 5. Erasure의 검출 및 정정 6. 10^{-4}이하의 에러율에서 좋은 성능 7. CD, DAT(Digital Audio Tape), ATM 망에서 사용
Concatenated Codes – Viterbi/Reed-Solomon (1967, Forney)	1. 짧은 구속장과 낮은 에러 정정 능력 t로 좋은 성능 2. 긴 지연(delay)으로 인한 저속 전송(수십 K ~ 수십 Mbps) 3. 랜덤(random), 연집에러에 모두 강함 4. 구현의 어려움(인터페이스와 전력문제 등등) 5. 위성통신에 사용
Turbo Codes (1993, Berrou et al.)	1. 재귀적(recursive) 구조의 연산 가능(매우 높은 연산량) 2. 재귀를 통한 반복이 많아질수록 좋은 성능 (반면, 긴 지연으로 인한 저속 동작) 3. 복호기의 복잡성이 매우 높음(인터리버와 디인터리버 포함)

※ 특징

① 길쌈 부호는 블럭 부호에 비해 낮은 전력에서 오류 정정 능력이 우수.

② 임계 복호법은 순회 부호의 복호 방법 중 하나의 다수결 논리 복호법과 같고 간단하며 대수적으로 취 급할 수 있으며 블럭 부호의 경우와 유사하다.

③ 최우 복호법은 이론적으로 가장 능률적이나 복잡도가 부호 길이와 더불어 지수 함수적으로 증가하는 단점이 있다.

④ 축차 복호법은 복호에 소요되는 계산 횟수가 확률 변수라는 확률 과정 복호법으로 최우 복호법과 유사 하다.

전송로 상의 잡음으로부터 정보를 보호하기 위해서 삽입한 중복도(Redundancy)이다. 그리고 부호화율(Code rate) R=k/n은 대개 $1/4 \leq R \leq 7/8$의 한계 내에 있으며 정보 전달 속도를 뜻하고 dmin은 부호의 오류 검출 및 정정 가능성을 알려주는 부호어 간의 최소거리(Minimum distance)를 말한다.

또 길쌈 부호에서는 n, k, m의 매개 변수가 사용되는데, 블럭 부호와는 달리 n은 부호기의 출력 단자수를, k는 부호기 입력의 단자수를 뜻하며, 일반적으로 k와 n은 작은 정수이고 n>k의 관계이다. 여기서 m은 부호기를 구성하는 데 필요한 기억 소자의 단수를 의미한다.

표 1에서는 현재 많이 쓰이고 있는 좋은 효율의 부호들과 Turbo Codes의 특성 및 장단점을 간략하게 보여주고 있다. Viterbi 알고리즘은 이동 통신 분야에서, Reed-Solomon 알고리즘은 CD(Compact Disk)나 DAT(Digital Audio Tape) 등의 저장 매체에서, 그리고 연접 부호(Concatenated Codes)는 위성 통신 분야에서 각각 그 부호들의 특성을 살려 사용되고 있다.

본 고에서는 앞에서 기술한 것들 중에서 특히, 길쌈 부호에 기반하여 새롭게 제시되고 있는 길쌈 부호의 병렬 연접 방법에 대해 소개하고자 한다. 이것은 현재 "Turbo Codes"로 명명되어져 있고, 상당한 주목을 받으며 서서히 이에 대한 연구가 세계적으로 고조되고 있는 단계에 이르고 있다.

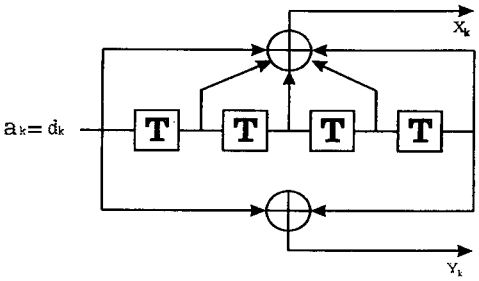
먼저, 같은 길쌈 부호에 대한 복호 방식으로는 임계 복호법(1963, Massey), 축차 복호법(1961, Fano), 그리고 Turbo Codes의 기반을 이루며 현재에는 CDMA 이동 통신에서 많이 쓰이고 있는 최우 복호 방식인 Viterbi 알고리즘에 대하여 간단하게 살펴보자. 이 알고리즘은 1967년에 Viterbi에 의해 제안되었으며, 그동안 거의 사용되지 않던 길쌈 부호를 사용하는 계기가 되었고 이의 복호 방식으로 최우 복호(最尤復號 : Maximum Likelihood Decodeing) 기법을 채용하고 있다. Viterbi 알고리즘은 최우 복호 기법을 사용하여 위상학적 구조를 갖는 격자도(Trellis Diagram)의 진행에 따른 최단 경로를 결정하는 방법으로, 수신 계열에 Hamming 거리나 Euclid 거리를 이용하여 최단 경로를 탐색하고 이를 역추적하여 복호를 수행한다. 하지만 오류 확률에 따른 구속장 K의 증가와 더불어 지수적인 복잡성(Complexity)을 가지기 때문에 K>8인 경우에 대해서는 구현이 어렵고 비실용적이라 할 수 있다.

따라서, 구속장이 짧은 길쌈 부호의 성능을 증가시키기 위해 새로운 방식이 필요하게 되었다. 그 결과, 최근 길쌈 부호의 병렬 연접을 이용한 부호화 방식으로 Turbo Codes의 관심이 고조되고 있다.

왜 Turbo Codes인가?

Turbo Codes는 C. Berrou, A. Glavieux, P. Thitiamjshima("Near Shannon Limit Error-Correcting Coding and Decoding : Turbo Codes", ICC'93)의 3인에 의해서 1993년에 처음으로 발표되면서 불란서, 유럽, 미국에 특허로 등록되어 있다. 이 부호는 길쌈 부호를 응용, 새로운 구조를 제시한 것이며, BER(Bit Error Rate)의 관점에서 Shannon Limit에 근접하는 성능을 보인다.

Turbo Code의 개념은 길쌈 부호기의 최대 단점을 보완하는 관점에서부터 출발한다고 볼 수 있다. 즉, 길쌈 부호의 복호 방법으로 사용되고 있는 Viterbi 알고리즘은 격자도(Trellis diagram)의 진행과정에서, 최악의 경우 잘못된 경로를 선택하여 복호 동작을 완료했을 때, 에러 수정을 제대로 하지 못할 뿐만 아니라 더 많은 에러를 발생시킬 수도 있다. 이것은 부호화 과정에서 Systematic 하지 않은 데이터를 부호기의 출력으로 하기 때문이다. 여기에서 Systematic 이란 데이터를 부호화 했을 경우, 정보 데이터가 부가 비트(Redundancy or parity check bit)와 분리되어 나타나는 현상인데, 일반적으로 블럭 부호 계열에서 주로 Systematic 한 데이터를 부호기의 출력으로한다. 이런 방식으로 부호화를 수행하여 전송하면 비록 복호기가 에러 정정을 하지 못하고, (즉, 블럭 부호에서는 에러 정정 능력을 초과하는 경우이고 길쌈 부호에서는 잘못된 경로를 선택하여 복호하는 것) 복호기



〈그림 1〉 일반적인 길쌈 부호기(Nonsystematic)

는 복호 실패를 선언하여 실패하는 경우가 발생한다고 해도 원래의 정보 데이터만을 출력하여 사용할 수 있게 된다. 이 때 물론, 정보 데이터에는 에러가 발생된 상태 그대로지만, 적어도 에러를 확산시키는 일은 없기 때문에 시스템의 성능에 향상을 기대할 수 있게 된다.

결국, 이를 길쌈 부호기에 적용하면 다음 그림 1과 같은 간단한 시스템을 생각할 수 있게 된다. 이 그림에서 보여지는 길쌈 부호기의 부호화된 출력 X_k와 Y_k는 시간축 상에서 데이터들이 4번의 지연을 가지고 혼합된 형태로 나타나기 때문에 정보 d_k가 가지고 있던 고유의 데이터 특성을 잃어버리게 된다.

부호율 1/2, 기억 차수 M=K-1, 구속장 K인 길쌈 부호기는 그림 1과 같이 구성된다. 이것은 하나의 정보 데이터 입력에 두 개의 부호화된 출력을 나타내는데, 이를 각각 d_k, X_k, Y_k라 하면 다음과 같이 표현할 수 있다.

$$X_k = \sum_{i=0}^{K-1} g_{1i} d_{K-i} \bmod 2 \quad (1)$$

$$Y_k = \sum_{i=0}^{K-1} g_{2i} d_{K-i} \bmod 2 \quad (2)$$

여기에서 g_{1i}, g_{2i}는 각각 0 또는 1의 값을 가지며, $G1:\{g_{1i}\}$와 $G2:\{g_{2i}\}$는 부호기의 생성 다항식의 일반적인 형식이라 할 수 있다.

일반적인 Nonsystematic 길쌈 부호기는 큰 SNR을 가지는 부분에서 systematic 길쌈 부호기보다 성능이 떨어진다는 것은 앞에서 약간 언급한 것으로도 알 수 있으며, Viterbi 복호기의 복잡성을 고려하면 구속장 K를 증가시키는 것에 의해 성능의 증가를 가져올 수 있다. 하지만 구속장을 증가시키는 것에는 항상 복호기의 복잡성을 고려해야 된다는 문제점을 안고 있으며, Shannon의 이론에 비추어볼 때, 구속장 K와 격자도의 진행에 한계가 설정된다는 것을 알 수 있다. 여기서 간단하게 Shannon의 이론과 Random coding bound에 대해서 영문으로 그대로 옮기면 다음과 같다.

Shannon's Channel Coding Theorem (1948) ;

There exist channel codes that makes it possible to achieve reliable communication, with as small an error probability as desired, if the transmission rate $R<C$, where C is the channel capacity.

Random Coding Bound ;

$P_e \le 2^{-nE_r(R)}$, *where* $E_r(R)$

$$= \frac{\max}{0 \le s \le 1} \frac{\max}{p}$$

$$[-sR - \log_2 \sum (\sum p_j Q_{k/j}^{1/(1+s)})^{1+s}] \quad (3)$$

$$\overline{P_e} \le 2^{-n(R_0 - R_c)} \quad (4)$$

, *where* R_0 *is cutoff rate and* R_c *is code rate.*

먼저 Shannon의 이론에서는 채널 용량 C를 넘어서지 않는 범위에서 부호화를 하여 전송율 R로 데이터를 전송할 경우, 만족할만큼 적은 에러율을 가지고 신뢰성 있는 통신을 제공하는 채널 부호가 존재한다는 것이다. 즉, 채널 용량의 범위 내에서 정보 데이터에 부가 정보를 삽입하여 전송했을 때, (채널 용량을 초과할 만큼)많은 부가 정보를 삽입하지 않고도 적은 에러율을 가지게 하는 부가 정보 외의 다른 요소가 존재한다는 것이다. 이에 대한 증명으로 Random coding bound의 개념을 요약하면 다음과 같은 식을 도출할 수 있다.

$$R < R_0 < R' < C \quad (5)$$

, 여기에서 R'는 임의의 부호율.

즉, 식 (3)의 P_e에 대한 식을 근사적인 방법으로 식 (4)와 같이 대치할 수 있다. 또 식 (4)에서 cutoff rate R_0와 부호율 R의 관계를 통해서 식 (5)를 얻을 수 있는데, 이것은 cutoff rate을 넘어

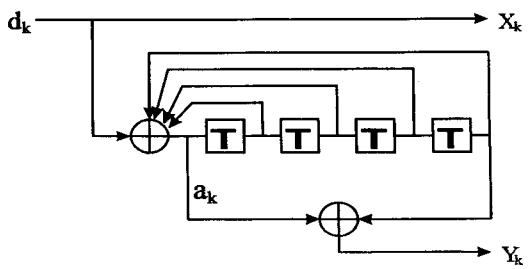
서지 않는 범위 내에서 부호화가 가능하다는 것과 R의 크기가 C를 넘어서지 않는 범위 내에서 증가시킬 수 있는 R'의 부호율을 갖는 임의의 부호가 존재할 수 있다는 것을 암시하고 있다. 또한 이론적으로 채널 용량과 부가 정보를 무한히 증가시키면 에러 없이 전송이 가능하지만 신호 전력 한계 등의 문제가 발생하므로, 일정한 채널 용량을 넘어서지 않는 부가 정보만 가지고도 에러 없이 전송할 수 있게 하는 어떤 요소(factor)가 존재한다는 것을 암시하기도 한다. 예를 들면, 길쌈 부호기의 구속장 K나 격자도의 진행 상태수 같이 부가 정보에 영향을 주지 않는 것이 이에 해당한다.

여기까지의 기술에 대한 간단한 결론은 부호기는 systematic할 필요가 있다는 것과 부가 정보와 정보 데이터가 어떤 관계를 가져야 하며, 적은 부가 정보를 추가하면서 효율적인 복호화를 수행할 수 있도록 도움을 주는 임의의 요소를 찾는 것이다.

결국, 앞의 세가지 조건을 상당히 많은 부분 만족하도록 하는 것이 부호화의 가장 큰 관건이라 할 수 있다. 그림 2의 Recursive systematic code의 경우는 systematic 조건과 어느 정도의 부가 정보, 그리고 복호화에 도움을 주는 임의의 요소를 포함하고 있긴 하지만, 임의의 요소가 복호화를 위해 크게 기여하지 못하기 때문에 이의 개선을 위하여 같은 조건 하에서 피드백이 있는 부분을 병렬 처리하면서 다른 종류의 임의의 요소를 포함하도록 인터리버를 사용하여 얻어진 것이 Turbo codes라 할 수 있다.

II. Recursive Systematic Codes(RSC)의 병렬 연접 부호화

그림 2에서 보이는 구조는 그림 1의 구조를 systematic하게 변형한 것인데, 이 구조에 대한 수식적인 전개는 다음과 같다.



〈그림 2〉 재귀적 시스템 부호

$$a_k = d_k + \sum_{i=0}^{K-1} r_i a_{k-1} \bmod 2 \quad (6)$$

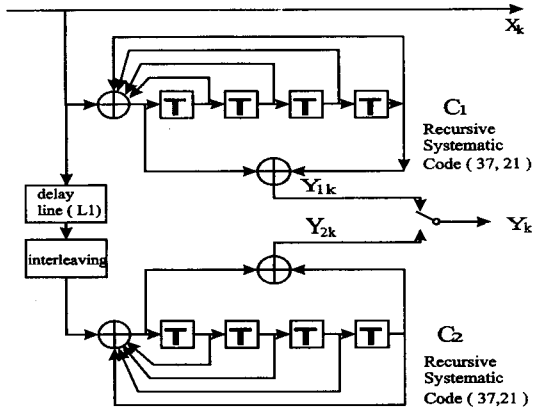
$$d_k = \sum_{i=0}^{K-1} r_i a_{k-1} \bmod 2 \quad (7)$$

, 여기에서 $a_k = g_{1i}$ if $X_k = d_k$

여기에서 a_k는 정보 d_k와 피드백 값의 합(modulo-2)으로 표현되며 X_k는 정보 d_k가 그대로 출력에 나타난다는 것을 의미한다.

그림 1에서 보인 Nonsystematic 부호와 그림 2의 Systematic 부호는 같은 격자도의 진행을 따르며, 또한 같은 자유거리 d_{free}를 가진다. 하지만 이들의 출력은 X_k에 대해 서로 다르며 두 부호의 가장 큰 차이라고 할 수 있다.

결국, RSC 구조를 병렬 연접(Parallel concatenation)하여 새로운 구조가 제안되었는데, 그림 3에서와 같이 두 개의 동일한 RSC 부호를 사용한다. 하지만 이 두 구조에 입력은 전혀 다른 값이 된다. 즉, 하나는 기존의 피드백 구조에 입력 데이터를 그대로 적용한 것이고, 다른 하나는 입력



〈그림 3〉 병렬 연접 재귀 시스템 부호

데이터를 인터리빙(Interleaving)하여 변형된 입력 데이터를 적용한다.

그림 3에서 입력 데이터 열 $\{d_k\}$의 경우, 임의의 시간 k에서의 부호기 출력 $\{X_k, Y_k\}$는 각각 d_k와 부호기 C1의 출력, Y_{1k}와 부호기 C2의 출력, Y_{1k}와 같다. 만약 두 부호기의 출력이 각각 n_1횟수와 n_2 횟수만큼 사용되었다면 부호가 C1의 부효율 R_1과 C2의 부효율 R_2의 관계는 다음 식 (8), (9)와 같다.

$$R_1=\frac{n_1+n_2}{2n_1+n_2} \tag{8}$$

$$R_2=\frac{n_1+n_2}{n_1+2n_2} \tag{9}$$

또한, 부호기의 입력과 출력은 다음과 같은 관계를 가지고 있다.

$$d_k=X_k \tag{10}$$

$$Y_{1k}=d_k+\sum_{i=0}^{K-1} r_i a_{1k-i} \bmod 2 \tag{11}$$

$$Y_{2k}=\overline{d_k}+\sum_{i=0}^{K-1} r_i a_{2k-i} \bmod 2 \tag{12}$$

, 여기에서 $\overline{d_k}$는 d_k의 인터리빙 형식.

일반적인 RSC 구조에 병렬 연접의 개념을 도입함으로써 두가지 잇점을 얻을 수 있다. 이것은 앞부분에서 언급했던 효율적인 복호화를 수행하기 위해 같은 부가 정보를 가지고 더 많은 임의의 정보를 부호기의 출력에 추가하는, 즉 기존의 구조로 부호화한 출력과 입력을 인터리빙하여 변형된 출력을 동시에 전송함으로써 이중의 부가 정보를 준 것과 같은 효과를 가져온다. 그리고 또 하나는 복호기에서의 동작을 회전적으로 수행하게 만드는 것이다. 이런 회전적인 연산이 시스템의 에러율 향상에 큰 도움을 줄 수 있는데 반하여 낮은 속도의 전송율을 초래하여 응용하고자 하는 시스템의 속도나 에러율, 그리고 복잡성의 문제를 신중히 고려해야 한다.

따라서, Shannon의 이론에서 제기된 같은 부가 정보를 할당한 상태에서 복호화를 위해 도움이 되는 임의의 요소를 기존의 구조보다 더 많이 포함했다고 볼 수 있다. 이것은 물론 복호화 알고리즘에 복잡성을 증가시키는 의미도 함축적으로 내포하고 있지만, 부호기의 입장에서 보면 좋은 부호로서의 Shannon 이론의 관점을 많은 부분 만족시키고 있다고 볼 수 있다.

1. Turbo 부호기의 블럭 부호 형식

Turbo 부호기 구조에서 생성 다항식을 다음과 같이 표현할 수 있으며 이와 같은 표현은 이 부호기가 systematic하기 때문에 가능하다.

$$G=[I \mid P] \tag{13}$$

그리고 임의의 인터리빙 구조는 $M=N\times N$인 행렬로서 표현될 수 있다. 만약 이런 행렬이 주어졌다면, 인터리버를 통과한 후의 입력은 다음 식 (14)와 같다.

$$d^I=dM \tag{14}$$

또한 그림 3의 두 부호기가 동일하다면, 인터리버 후의 부호기의 출력은 다음과 같은 형식을 가지게 될 것이다.

$$\begin{aligned} Y_{2k}&=d^IP \\ &=dMP \end{aligned} \tag{15}$$

여기에서 MP를 정보 d에 대해 검사 행렬로 볼 수 있으므로 Turbo Codes 생성 행렬을 다음과 같이 얻을 수 있게 된다.

$$G_T=[I|P|MP] \tag{16}$$

따라서 그림 3의 인터리버 후의 부호기 출력을 구하기 위해 다음과 같은 행렬을 고려해보자.
여기에서 행렬 P는 블럭 길이 12인 것을 예로 사용한다.

$$P=\begin{pmatrix} 1&1&1&0&1&1&0&1&1&0&1&1 \\ 0&1&1&1&0&1&1&0&1&1&0&1 \\ 0&0&1&1&1&0&1&1&0&1&1&0 \\ \cdot&\cdot&\cdot&\cdot&\cdot&\cdot&\cdot&\cdot&\cdot&\cdot&\cdot&\cdot \\ \cdot&\cdot&\cdot&\cdot&\cdot&\cdot&\cdot&\cdot&\cdot&\cdot&\cdot&\cdot \\ \cdot&\cdot&\cdot&\cdot&\cdot&\cdot&\cdot&\cdot&\cdot&\cdot&\cdot&\cdot \\ \cdot&\cdot&\cdot&\cdot&\cdot&\cdot&\cdot&\cdot&\cdot&\cdot&\cdot&\cdot \\ 0&0&0&0&0&0&0&0&0&1&1&1 \\ 0&0&0&0&0&0&0&0&0&0&1&1 \\ 0&0&0&0&0&0&0&0&0&0&0&1 \end{pmatrix} \tag{17}$$

위의 행렬 (17)은 간단하게 N−1개의 연속적인 0과 1에 의해 부호화되는 동작을 보이기 위한 것으로 인접한 행들 사이의 관계는 0의 삽입에 의한 1비트 천이(shift)이다. 먼저 nonsystematic 출력

(인터리버를 거치지 않은 길쌈 부호화된 데이터)은 P 행렬의 첫 번째 행(111011011011)이 되고, 임시(transient) 동작을 위한 벡터를 g^t와 동작의 반복 형태(cyclic portion)를 위한 벡터를 결정할 수 있다.

$$g^t=(111)$$
$$g^c=(011) \quad (18)$$

이때 행렬 P의 첫 번째 행을 두 벡터로 표시하면 또한 다음 식 (19)와 같다.

$$g=(g^t,\ g^c,\ g^c,\cdots) \quad (19)$$

만약, 입력 데이터가 $1+D^9$이었다면 위의 P 행렬을 이용하여 인터리버 후의 출력을 다음과 같이 구할 수 있다. 즉, 행렬 P의 첫 번째와 9번째 행을 modulo-2 한 것과 같다.

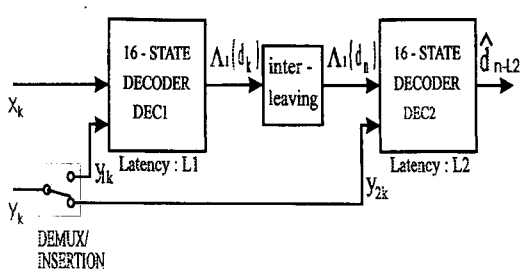
$$(111011011011)\oplus(000000000111)$$
$$=(111011011100)$$

III. Turbo Codes의 복호화

복호화의 경우는 부호화 방식에서 직관적으로 암시했던 이중의 부가 정보를 어떻게 하면 좀더 효율적으로 적용할 수 있는 지에 대한 관점으로 접근을 좁힐 수가 있다. 복호화라는 것은 부호화의 역과정이므로 그림 4에서 보는 바와 같이 간단한 구조를 고려해볼 수 있다.

부호기에서 인터리빙을 사용하여 부호화하는 데이터의 특성을 다르게 설정하였기 때문에 복호 과정에서도 같은 인터리빙을 사용하며, 특성이 다른 두 길쌈 부호화된 입력을 분리하여 복호화에 적용시킨다. 따라서 전체적인 구조는 순차적(serial) 연접 구조를 가지며 그 구조의 중심에 인터리빙을 사용하여 부호기와의 구조적 동일성을 얻을 수 있다.

먼저, DEC1에서는 채널을 통과한 후의 수신 데이터를 입력으로 삼아서 동작한다. 즉 부호화 하지 않은 데이터 X_k와 부호화된 데이터 Y_k중의 하나인 Y_{1k}를 입력으로 받아서 일차적인 복호 동작을 수행한다. 이 때, DEC1에서의 출력은 인터리버에



〈그림 4〉 직렬 연접 복호기

의해서 연집된 특성의 에러를 분산시키며, DEC2의 입력으로 사용된다.

이산적인 가우시안 채널과 이진 변조를 적용했을 경우를 고려해보면, 복호기의 입력은 임의의 시간 k에서 두 개의 랜덤 변수 x_k와 y_k로 볼 수 있으며 다음과 같이 정의할 수 있다.

$$x_k=(2d_k-1)+i_k \quad (20)$$
$$y_k=(2Y_k-1)+q_k \quad (21)$$

, 여기에서 i_k & q_k는 같은 분산 σ^2을 가진 독립적인 잡음.

부가 정보 y_k는 디멀티플렉서에 의해 두 개로 나누어지며 각각 $Y_k=Y_{1k}$일 경우, DEC1에 $Y_k=Y_{2k}$인 경우, DEC2의 입력으로 결정된다. 또한 주어진 부호기에 대한 부가 정보가 방출되지 않을 때, 관계되는 복호기의 입력은 모두 'zero'가 되며 이의 기능은 디멀티플렉서에서 제공한다. DEC1은 DEC2에 연판정(Soft decision : weighted) 정보를 전달하며, DEC1에 의해서 복호된 정보 d_k와 관계되는 LLR(Logarithm Likelihood Ratio), $\Lambda_1(d_k)$는 DEC2를 위한 정보가 된다. 여기서 $\Lambda_1(d_k)$는 다음과 같이 표현할 수 있다.

$$\Lambda_1(d_k)=Log\frac{P_r\{d_k=1/observation\}}{P_r\{d_k=0/observation\}} \quad (22)$$

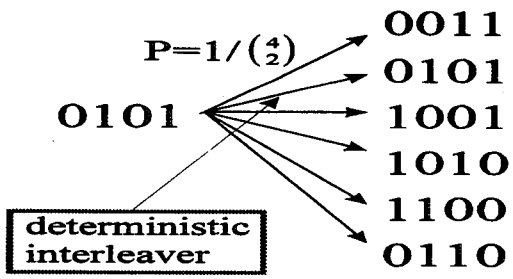
, 여기에서 $P_r\{d_k=i/observation\}$. $i=0$, 1 데이터 비트 d_k의 *a posteriori probability* (APP).

현재까지 Viterbi 알고리즘은 길쌈 부호의 복호 방법으로, 특히 연판정을 사용했을 경우, 길쌈 부호에 대한 이론적인 에러 한계에 근접한 성능을 보이고 있다. 하지만 이를 위해 필요한 구속장이나 SNR에 대하여 상당한 복잡성을 감안해야 하며, 에러율에서도 어느 경계점을 극복할 수 없게 되는

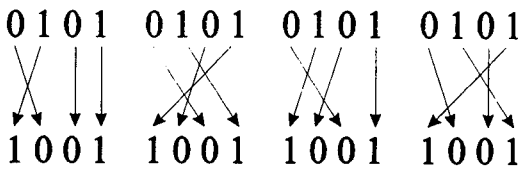
단점과 단지 가우시안 채널에 대한 길쌈 부호의 최적 복호법으로 알려져 있다. 특히 페이딩 채널에서는 인터리버를 사용하지 않을 수 없으며 연판정 기법을 부수적으로 채용해야만 하는데, 이 경우 인터리버의 크기를 결정하는 문제나 정확한 연판정 데이터를 생성하는 문제 등은 상당히 난해하다고 할 수 있다.

또한, Viterbi 알고리즘은 복호된 데이터에 대한 APP(A Posteriori Probability)를 산출할 수 없기 때문에 BAHL et al.에 의해 제안된 알고리즘을 적용하여 비트 에러율을 최소화하고 각 복호된 데이터에 따른 APP를 산출한다. 그리고 BAHL et al. 알고리즘은 재귀적(recursive) 특성을 위해 변경되어 적용되어야 한다.

1. 균일 인터리버(Uniform Interleaver)



〈그림 5〉 (a) 균일 인터리버



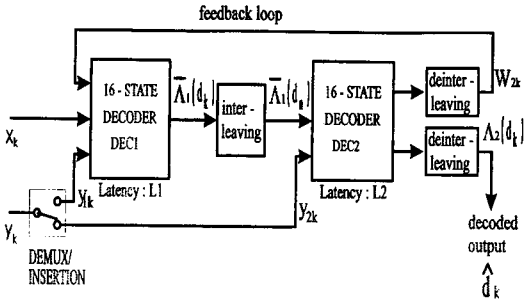
〈그림 5〉 (b) 사상 규칙(mapping rules)

그림 5(a)는 길의 N의 deterministic 비트 인터리버의 동작을 동일한 확률상에서 표현한 것이다. 이것은 weight W와 길이 N의 비트열에서 같은 확률을 가진 모든 가능한 사상(mapping)을 보여주며 확률$P=1/(\frac{N}{W})$로 주어진다. 그림 5(a)는 "0101"에서 "1001"로 항상 같은 deterministic을 가지는 예이다. 또 그림 5(b)는 "0101"에서 "1001"로의 사상 규칙을 나타내는 것으로 인터리버가 어떤 규칙을 따를 것인지를 표현하고 있다.

일반적으로 인터리버는 비트열의 자유거리를 되도록 크게 하는 효과를 얻기 위해 사용하며 정방행렬 형태를 가진다. 또한 그림 4처럼 DEC1 다음에 위치하기 때문에 nonuniform하다면 정방형 형태로 에러를 확산시킬 수 있다. 그리고 임의의 비트열이 가지는 SNR(signal to noise ratio)의 각 값에 대해, uniform interleaver를 채용하여 얻을 수 있는 성능은 적어도 임의의 한 deterministic 인터리버에 의해 얻을 수 있는 성능과 같다고 볼 수 있다.

IV. Turbo Codes의 성능

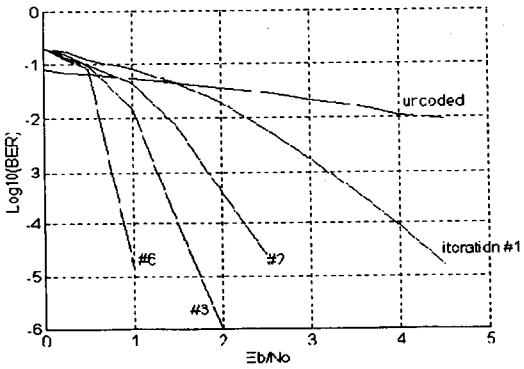
지금까지 앞장에서 설명한 이론과 증명들을 토대로 그림 6의 구조(피드백 동작)에 대해서 그림 7과 같은 성능을 얻을 수 있다. 이것은 구속장 K=5, 생성다항식 $G_1=37$, $G_2=21$, 그리고 병렬 연접($R_1=R_2=2/3$)구조를 이용한 부호율 $R=1/2$의 부호기를 가지고 P_e성능을 평가한 것이다. 여기에서 인터리버는 256×256 행렬로 구성되어 있으며 변형된 BAHL et al. 알고리즘은 데이터 블럭 길이 N=65536 bits를 가지고 적용되었다. 또한 BER 10^{-5}을 생성하기 위해 128 데이터 블럭 즉, 8×10^6 bits의 d_k를 고려하였다.



〈그림 6〉 피드백 복호기

그림에서 보는 것처럼 Turbo Codes의 장점은 반복이 많아질수록 에러율이 Shannon Limit에 근접함을 알 수 있다. 하지만 실제 성능은 E_b/N_o를 1dB나 2~3dB 정도로 낮춰서 전송할 수 있는 것이 아니고 한 번도 반복하지 않은 iteration #1의 성능을 복호기 스스로 개선시켰기 때문에 그 에러율에 맞추어서 전송이 이루어져야 한다는 것이다. 즉 반복이 수행되지 않았을 경우의 에러율은 기존의 연접부호에 비해서 비슷한 복잡도를 감안한다면 상당히 떨어질 것으로 보인다.

또한 인터리버의 크기 결정과 지연 문제도 성능에 적지 않게 영향을 미치기 때문에 이에 대한 연구와 더불어 첫 번째 복호 과정에서의 에러 성능이 심하게 떨어질 경우(즉, Viterbi 복호기처럼 잘못된 경로를 선택했을 때, 더 많은 에러를 발생시킬 수 있는 경우)도 고려해야만 한다.

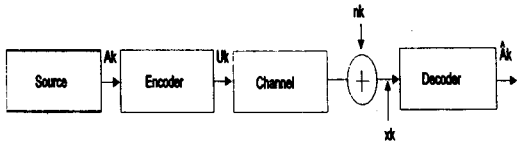


〈그림 7〉 반복적 복호에 의한 Turbo Codes의 BER 성능(p=1, …, 6)

V. Turbo Codes의 응용 : TTCM(Turbo Trellis Coded Modulation

1. TCM의 개념 및 부호화 변조

현재, 변조와 부호화가 독립적인 시스템에서는 부호화를 위하여 부가정보(Redundancy)가 필요하게 된다. 따라서 이런 부가 비트의 사용량에 비례하여 에러 정정 능력은 증가하지만 데이터 전송



A_k : 사용자 데이터
U_k : 부호화된 데이터(전송 데이터)
n_k : 채널 잡음
x_k : 받은 데이터
$\hat{A}_k$: 복화화된 데이터

〈그림 8〉 이산 시간 채널 모델

율의 저하를 초래하게 된다. 즉, bit-rate의 관점에서 탁월한 부호 이득을 얻을 수 있지만, 대역폭의 사용 효율은 떨어지게 된다.

결국, 이런 단점들을 보완하기 위하여 제시된 방식이 TCM이다. 이 방식의 주목적은 대역폭의 증가없이 부호화 하지 않은 경우에 비하여 좀더 나은 부호 이득을 얻기 위한 것이다. 즉, TCM(=부호화 + 변조)은 신호집합(Signal set)에 대한 부호화를 수행함으로써 부호화와 변조를 일체화한 방식이다.

먼저 설계와 구현을 위하여 위하여 다음과 같은 단계가 필요하다.

A. 집합 분할

접근 1:

그림 8은 일반적인 격자 부호화 시스템의 이산 채널 모델이다. 여기서 부호기는 M-ary 소스 데이터를 2M-ary 부호화된 데이터로 사상(mapping)시킨다. 소스(M)를 채널 신호 집합(2M)으로 사상시키기 때문에, 부가정보는 대역폭의 증가와 독립적이 되며, 어떤 신호 공간(Signal space)에서 부가정보가 독립적이라는 것은 신호들 사이의 해밍거리(Hamming distance)를 최대화시키는 이진 부호화 방법을 사용하여 비트 에러율을 향상시킬 수 있다는 것을 암시하고 있다. 이것은 일반적으로 Gray 부호화라 불리우며 성능면에서 상당한 이득을 얻을 수 있다.

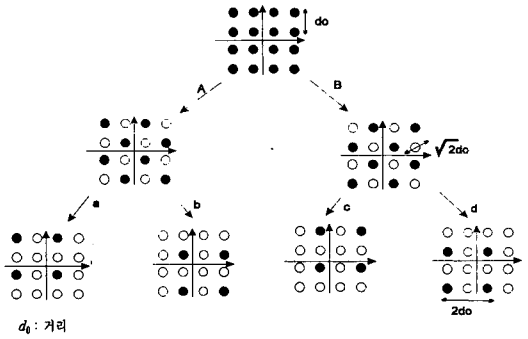
하지만 단순한 신호집합의 확장만으로는 채널 부호화를 통해 얻을 수 있었던 이득에 비하여 너무 적은 이득에 불과하므로 부호화의 가장 중요한

개념을 여기에 도입해야만 한다.

접근 2:

부호화를 통해 높은 이득을 얻기 위해서 신호점들의 최소 유클리드 거리(Minimum Euclidean distance)라 불리우는 자유거리(Free distance)를 증가시켜야 한다. 즉 Ungerboeck이 제안한 '집합 분할에 의한 사상'(mapping by set partitioning)이라 불리는 사상 규칙을 적용하여 높은 성능의 시스템을 구현할 수 있다. 이 방법은 각 신호점들 자체의 거리를 증가시키는 것이 아니라 신호점들의 열(sequence of signal points) 사이의 거리를 증가시키는 방법이다. 즉, 신호열 사이의 거리를 증가시키는 것에 의하여 자유거리가 증가하므로 1.에서의 방법인 신호점 자체의 단순한 해밍거리를 증가시키는 것보다 부호 이득의 측면에서 보다 효율적이라 할 수 있다.

또한 방법적인 면에서는 부가정보가 큰 신호 집합을 작은 부집합으로 변환, 신호점들 사이의 자유 유클리드 거리를 증가시킨다. 결국, 좋은 특성의 거리(nice distance property)를 가지는 부집합의 신호열들을 얻을 수 있다. 그림 2.는 16 개의 신호점을 가지는 QAM에서 집합 분할을 보이고 있으며, 집합 분할을 통해 신호점들 사이의 자유 유클리드 거리를 증가시키는 것을 나타내고 있다.



〈그림 9〉 16 점의 성상도 Partitioning of 16−points constellation

그림 9에서 표현된 것처럼 분할을 통해서 얻어지는 신호점들 사이의 거리는 다음과 같이 증가한다.

$$d=(\sqrt{2})^k d_0,\ k=0,\ 1,\ \cdots,\ n,$$
$$where\ k\ is\ the\ number\ of\ partitioning \quad (23)$$

따라서 8−point uncoded 시스템(한번 분할했을 경우)이나 4−point uncoded 시스템(두번 분할했을 경우)과 같은 성능을 나타낼 수 있으며, 이 경우 각각 3dB, 6dB의 부가적인 부호 이득을 얻을 수 있다.

B. 격자 부호기

먼저, 집합 분할을 통한 신호들과 부호기의 입장에서 본 관계에 대해 기술하려면 부호기의 목적을 고려해야만 한다. 이의 목적은 좋은 거리(nice distance)를 가지는 신호열(신호점들의 허용 또는 생성 가능한 열)을 생성하기 위한 확장된 신호공간을 사용하도록 하기 위한 것이다. 다시 말하면, 확장된 신호공간을 사용하여 좋은 거리를 가지는 신호열을 생성하기 위하여 격자 부호기를 사용한다.

다음과 같은 예를 들어보기로 하자.

만약 AWGN(Additive White Gaussian Noise)이 전송하는 데이터 즉, 송신측의 입장에서 보면 부호화된 데이터에 영향을 주었다고 하자. 그때 부호기는 신호열들의 부호화된 데이터 사이의 거리가 최대가 되도록 하는 것이 가장 이상적이다. 즉, 식(24)를 만족하는 데이터로 부호화되어야 한다.

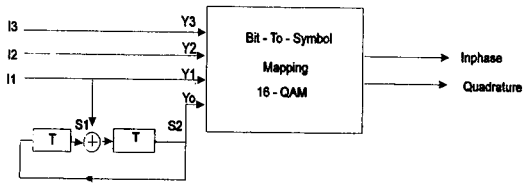
$$d_{free}=\min_{U_k \neq U_{k'}} \sum d^2(U_k,\ U_{k'}),$$
$$-\infty<K<+\infty, \quad (24)$$

U_k & $U_{k'}$: k t번째 시간 간격의 채널 신호

$d(U_k,\ U_{k'})$: U_k & $U_{k'}$ 사이의 듀클리디안 거리.

또한 부호기는 그림 3.처럼 binary convolutional encoder의 형태로 설계할 수 있으며, convolutional encoder는 입력선상에 기억소자를 가지고 있다.

그림 10에서 입력 비트는 사용자 데이터들 중에 한 부집합일 수도 있고 전체일 수도 있다. 또한 임의의 이진 입력 I_1에 의해서 얻어진 Y_1 & Y_0는 분할에 의해서 얻어진 4개의 부집합 중에 하나를 결정하는 선택 신호로 사용된다. 또한 Y_3 & Y_2는



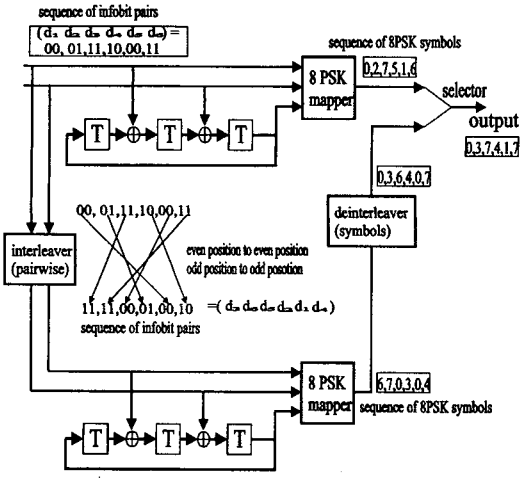
II_1 : 길쌈 부호기의 이진 입력⇨k.
Y_1 & Y_0 : 길쌈 부호기의 출력⇨k+1

〈그림 10〉 4-상태 피드백 부호기(16-QAM)

부호기의 출력(Y_1 & Y_0)에 의해서 선택된 부집합 안에서 4개의 가능한 신호점 중에 하나를 선택하는 신호로 사용된다.

따라서 k+1 분할 후에 2^{k+1}개의 부집합이 얻어지며, 부호기의 출력이 이 부집합 중에 하나를 결정, 선택하게 된다. 또한 각 부집합에서의 신호점 수는 2^{n-k}가 되며, 부호화 되지 않은 비트(Y_3 & Y_2)가 이런 신호점들 중에 하나를 선택하게 된다.

물론, 피드백 형태의 부호기의 피드백 처리 과정에서 에러가 발생했을 때, 상당히 큰 타격을 주는 것은 사실이지만 실제 격자 부호화 시스템의 응용 분야에서는 일반적으로 많이 사용되고 있다. 이것은 부호기의 문제가 아니라, 복호화의 관점에서 복호기가 실패할 경우에, 전송데이타 내에 정보비트가 그대로 존재하기 때문에(*systematic*) 부호화



〈그림 11〉 Encoder of 8PSK TTCM with interleaver length N=6.

가 없는 시스템처럼 동작하게 되는 잇점이 있다. 또한 수신측에서 요구되는 복호기의 형태가 길쌈 복호화(convolutional decoding)방식이 아니어도 격자도의 정보만 가지고 있으면 복호가 가능하다는 또 하나의 잇점이 있기 때문에 이 방식을 많이 사용한다.

그림 11은 앞에서 설명했던 TCM 방식을 기반으로 하는 TTCM 모델이다. 여기에서는 인터리버의 길이 N=6을 사용하여 블럭당 변조된 심볼의 수를 같게 하고 6 심볼열(d_1, d_2, ⋯, d_6)=(00, 01, 11, 10, 00, 11)이 각 심볼당 두 개의 정보 비트를 가질 때, 시스템을 통과하여 출력되는 값은 (0, 2, 7, 5, 1, 6)이며 정보 비트가 인터리버를 거쳐 다시 부호화 되었을 때 비트 열은 (6, 7, 0, 3, 0, 4)가 된다. 그리고 각 심볼을 지정하는 정보 비트의 순서가 인터리버를 거치지 않은 부호기의 순서와 일치하도록 인터리버를 거친 부호기의 출력 심볼을 디인터리빙한다. 즉, 여기에서 다시 새로운 심볼 열 (0, 3, 6, 4, 0, 7)을 얻을 수 있다. 결국, 인터리버를 거치지 않은 심볼 열과 인터리버와 부호기 그리고 디인터리버를 거친 심볼열을 교대로 선택하여 새로운 심볼 열 (0, 3, 7, 4, 1, 7)을 출력하게 되며, 이 때 각 정보 비트 쌍은 하나의 8PSK 심볼에 포함되고, 부가 비트는 두 부호기로부터 생성된다. 또한 k 번째 정보 비트 쌍은 정확히 k 번째 심볼의 3비트중 2비트를 결정한다.

한편, Turbo Codes의 최근 연구 동향은 1993년 C. Berrou 등에 의해 처음으로 ICC'93 논문에 발표된 후, 이태리, 미국, 독일 등에서 활발히 연구되고 있다. 미국 텍사스 달라스에서 열린 ICC'96('96,6, 23-6,27)에서 Tutorial paper로 발표되었는데, 강사는 이태리의 S. Benedetto 교수였고 Turbo Codes 세션을 별도로 마련해서 13편의 논문이 심도있게 발표되었다. 또한 Turbo Codes의 연구를 주도하고 있는 그룹은 다음과 같고 또한, 재미 한국인 학자로서는 CTA 회사의 이 병관씨가 괄목할

① S, Benedetto 교수	이태리 Politecnico Torino 전자과
② D. Divsalar 박사	미국 Jet Propulsion Lab.
③ J. Hagenauer 교수	독일 Techincal Univ. of Munich

만한 연구를 수행하고 있다.

VI. 결 론

지금까지 Turbo Codes의 구조와 성능 그리고 그 응용 분야의 하나로 TTCM을 간단히 기술하였다. 현재까지 진행된 연구들의 성과에 비추어 볼 때, 아직까지는 연구가 상당히 미비한 실정이며 이에 대한 응용 분야도 명확히 결정되지 않았고 많은 문제점을 안고 있는 것은 자명한 사실이다. 즉, 상당히 높은 복잡도를 가지고 있기 때문에 이에 대한 구현 문제와, 두 개의 복호기(DEC1과 DEC2)와 인터리버, 디인터리버의 채용으로 인한 지연 문제, 그리고 반복적인 처리가 누적될수록 성능의 증가와 함께 더욱더 커지는 지연과의 tradeoff 문제 등을 들 수 있다. 따라서 앞으로 이에 대한 문제 해결과 적합한 응용 분야에서의 많은 연구를 기대해본다.

참 고 문 헌

[1] G. D. Forney Jr. "The Viterbi Argorithm," Proc. IEEE, Vol. 61, pp. 268-279, Mar. 1973.

[2] A. J. Viterbi, "Convolutional Codes and Their Performance in Communication systems," IEEE Trans. on Comm. Vol. COM-19, pp 751-772, 1981.

[3] E. R. Berlekamp, R. E. Peile, S. P. Pope, "The Application of Error Control to Communications," IEEE Comm. Magazine, Vol. 25, No. 4, pp 44-57, Apr. 1987.

[4] C. Berrou, A. Glavieux, and P. Thitimajshima. "Near Shannon Limit Error-Correcting Coding and Decoding : Turbo Codes," in ICC, pp. 1064-1070, 1993.

[5] S.Goff, A. Glavieux, C. Berrou, "Turbo-Codes and High Spectral Efficiency Modulation," in ICC, May 1994.

[6] P. Robertson, "Illuminating the Structure of Code and Decoder of Parallel Concatenated Recursive Systematic (Turbo) Codes," in IEEE Globcom Conference, pp. 1298-1303, 1994.

[7] M. H. Lee, S. B. Choi, J. S. Chang, "A High Speed Reed-Slomon Decoder," IEEE trans. on Consumer Elect. Vol. 41, No. 4, Nov. 1995.

[8] S. Benedetto, and G. Montorsi, "Design of Parallel Concatenated Convolutional Codes," IEEE Trans. on Communications, Vol. 44, No. 5, pp. 591-600, May 1996.

[9] U.-C.G. Fiebig and P. Robertson, "Soft Decision Decoding in Fast Frequency Hopping Systems with Convolutional Codes and Turbo Codes," in ICC, June 1996.

[10] W. Blackert, "An Upper Bound on Turbo Codes Free Distance," in ICC, June 1996.

[11] M. H. Lee, J. S. Chang, S. B. Choi, "A Syndrome Check Error Estimation Algorithm for Convolutional Coding," in Multi-Dimensional Mobile Communication (MDMC), pp 556-560, Jul. 1996.

저 자 소 개

李 門 浩

1945年 1月 15日生
전남대학교 전자공학과 박사,(통신기술사)
일본 동경대학 전자공학 박사

1970年~1980年 : 남양 문화방송(주) 송신소장
1986年~1987年 : 한국통신학회 및 대한전자공학괴 학술논문상
1985年~1986年 : 미국 미네소타 주립대학교 포스트 닥터
1990年 7月, 1992年 11月, 1995年 12月 : 독일 하노버, 아현공대 연구 교수

관심 분야 : 디지털 이동통신, 채널코딩, 영상통신

崔 勝 倍

1952年 1月 23日生
1993年 전북대학교 정보통신공학과 졸업
1996年 전북대학교 대학원 정보통신공학과 졸업(석사)
1996年~현재 전북대학교 정보통신공학과 박사과정