일 반 투 고

차세대 통신을 위한 Turbo Codes 반복 복호기의 기본 개념

이 문 호
전북대학교 정보통신공학과

I. 최근 무선 멀티미디어 통신의 채널 부호의 표준화 동향

차세대 무선 멀티미디어 통신에서는 실시간 음성 서비스나 혹은 화상 서비스 같은 낮은 지연을 필요로 하는 applications은 일반적으로 높은 BER(저속 전송률 : 8kbps, 32kbps, 지연 : 40ms이내, BER < 10^{-4}), 컴퓨터의 데이터처럼 낮은 BER을 필요로 하는 응용은 긴 지연을 가질 수 있다 (고속 전송율 : 144kbps, 384kbps, 2048kbps, 지연 : 200ms정도, BER < 10^{-6}). 따라서 이러한 문제점을 해결하고 다양한 서비스를 만족하기 위해서는 채널 에러에 대한 복원 능력이 우수한 부호화 기법이 요구되고 있다. 현재 국내에서는 차세대 무선 멀티미디어 통신을 위한 규격으로 FEC 부분에 관련하여 음성뿐만 아니라 데이터까지의 서비스를 위해 길쌈부호/Viterbi Codec을 사용하고 데이터 서비스를 위해 길쌈부호/Viterbi Codec 과 Reed-Solomon Codec을 채용하려 하고 있다. 이것은 데이터 서비스 측면에서 기존 CDMA기법에서 사용하던 길쌈부호/Viterbi Codec이 요구하는 BER을 만족시키지 못하기 때문에 실용적인 측면에서 현재까지 알려진 가장 강력한 에러 정정 기술로서 두 부호의 연접 기법을 채용할 수밖에 없다는 입장이다. 그러나 이들의 사용으로 인한 복잡성 문제나 내부 지연의 문제들도 심각하게 우려되고 있기 때문에 좀더 효율적인 기법이 필요하게 되었다. 이것은 CDMA나 광대역 CDMA 그리고 WLL 등 무선을 전송매체로 하는 모든 시스템에서 가지게 되는 고질적인 문제라고 할 수 있다. 표 1은 유럽, 미국, 한국의 차세대 무선 멀티미디어 통신을 위한 채널 부호 기법을 제시하고 있다. 일반적으로 데이터 속도가 낮은 음성 쪽에서는 길쌈 부호를 사용하고, Data, 영상 등의 전송 속도가 높은쪽에서는 Turbo Codes를 사용하려는 경향을 알 수 있다.

한편, 채널 부호는 크게 블럭 부호(Block Codes)와 길쌈 부호로 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 블럭 당 비트 수를 의미하고 정보길이(Information bits) k는 정보 비트의 수를 뜻하는데, 그 실용치는 3에서부터 수백 비트까지에 이르고 n-k비트의 검사 길이는 전송로 상의 잡음으로부터 정보를 보호하기 위해서 삽입한 중복도(Redundancy)이다. 그리고 부호화율(Code rate) R=k/n은 대개 $1/4 \le R \le 7/8$의 한계 내에 있으며 정보 전달 속도를 뜻하고 $dmin$은 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 부호

〈표 1〉 차세대 무선 멀티미디어 통신을 위한 채널 코딩 기법 ('98.2월 기준)

	Traffic Channel	Control Channel
cdma2000	◎ Voice/Low Rate Data Service : *Convolutional Code*(K = 9; R = 1/2, 1/3, 1/4) ◎ High Rate Data Service(14.4Kbps 이상) : *Turbo Codes* (K=4; R = 1/2, 1/3, 1/4)	*Convolutional Code* (K=9; R = 1/4 for RL, R = 1/2 for FL)
ETSI WCDMA	*Concatenated Code* : Convolutional Code(K=9; R = 1/2, 1/3) & RS Code	*Convolutional Code* (parameters : TBD)
ARIB	*Concatenated Code* : Convolutional Code(K= 9; R = 1/3) & RS Code	*Convolutional Code* (K = 9; R = 1/2)
ETRI	◎ *Concatenated Code* : Convolutional Code(K=9; R = 1/2, 1/3, 1/4) & RS Code ◎ *Turbo Codes* : (K = 4; R = 1/2, 1/3, 1/4)	

* cdma2000 : IMT-2000 미국 표준화 그룹

* ETSI(European Telecommunication Standards Institute) : 유럽 표준화 기구

* ARIB(Associate of Radio Industries and Business) : 일본 표준화 기구

※ TIA(미국)에서는 HNS(Hughes Network Systems)가 cdma2000에 제안하여 채택된 Turbo Codes를 ARIB 및 ETSI에서도 Data Service용으로 채택하도록 추진하기로 결정함('98. 4. 3).

기의 출력 단자수를, k는 부호기 입력의 단자수를 뜻하며, 일반적으로 k와 n은 작은 정수이고 $n>k$의 관계이다. 여기서 m은 부호기를 구성하는 데 필요한 기억 소자의 단수를 의미한다.

길쌈 부호에 대한 복호 방식으로는 임계 복호법 *(1963, Massey)*, 축차 복호법*(1961, Fano)*, 그리고 *CDMA* 이동 통신에서 많이 쓰이고 있는 최우 복호 방식인 *Viterbi* 알고리즘이 있다. *Viterbi* 알고리즘은 *1967*년에 *Viterbi*에 의해 제안되었으며, 그동안 거의 사용되지 않던 길쌈 부호를 사용하는 계기가 되었고 이의 복호 방식으로 최우 복호*(Maximum Likelihood Decoding)* 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. 그러나 오류 확률에 따른 구속장 K의 증가와 더불어 지수적인 복잡성*(Complexity)*을 가지기 때문에 $K>9$인 경우에 대해서는 구현이 어렵고 비실용적이다.

터보 코드는 다른 부분의 변형없이 인터리버의 크기만을 변화시킴으로써 앞에서 언급한 차세대 무선 멀티미디어 통신의 요구 조건을 만족할수 있기 때문에 차세대 무선 멀티미디어 통신용으로 거론되고 있다. 터보 코드는 반복 회수를 증가시킴으로써 더 낮은 *BER*을 얻을수 있다. 이것은 사용자로 하여금 큰 매력을 느끼게 하는 사항인데 서비스 제공자가 설정한 것 보다 더 좋은 품질의 통화를 원한다면 사용자는 단지 자신의 단말기의 반복 복호 회수를 증가시키면 된다. 반대로 통화 품질은 좀 떨어지더라도 단말기를 오래 사용하기를 원한다면 반복 복호의 회수를 감소시키면 된다. 본 논문에서는 *Turbo Codes*의 반복 복호 알고리즘과 *soft-output* 복호기의 원리에 대한 수학적인 개념을 *Bayes* 정리와 *Tanner* 그래프를 이용하여 설명한다.

*(*Turbo Codes*는 블란서 *C. Berrou* 교수와 *A. Glavieux* 교수가 *Viterbi* 복호기를 연구하던 중, 독일 뮌헨 공대 *J. Hagenauer* 교수가 발표한 *soft-output viterbi* 알고리즘의 S/N비 증폭에 대한 논문을 보고 이를 전자회로의 *feedback* 회로에 응용하여 *Convolutional* 부호기를 병렬

로 처리한 것이 *Shannon limit*에 접근되었는데, 이를 *Turbo Codes*라 명명하고 *1993*년 *ICC* 학술회의 및 특허로 발표하면서 세상에 알려졌다.)

II. 기존의 Bayes 정리와 반복 복호 알고리즘

이 장에서는 *Turbo Codes*의 반복 복호 알고리즘의 기반이 되는 *Bayes* 정리에 대해 기술하며 반복 복호에 따른 *LLR(Log-likelihood Ratio)* 개념에 대하여 고찰한다. *Turbo Codes*의 복호 동작에 대한 기본적인 개념의 수학적인 배경은 *Bayes* 정리에 근거하고 있으며, 다음과 같은 사건 A 와 B의 조건부 확률*(conditional)*과 결합 확률*(joint probability)*사이의 관계를 가지고 얻어질 수 있다.

$$P(A|B)P(B)=P(B|A)P(A)=P(A,B) \qquad (1)$$

여기에서 $P(A)$와 $P(B)$를 사전 확률(a priori probability)이라 하며 또한 $P(A \mid B)$를 사후 확률(*a posteriori* probability : APP)라 한다. 확률 $P(A \mid B)$는 식 (1)로부터 얻을 수 있다.

$$P(A|B)=\frac{P(B|A)P(A)}{P(B)} \qquad (2)$$

가우시안 채널에서 연속적인 값을 가지는 랜덤 변수 x에 대한 APP를 계산하기 위하여 Bayes 정리을 적용하면 다음과 같다.

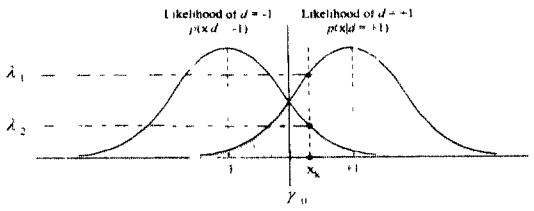
$$P(d=i|x)=\frac{p(x|d=i)P(d=i)}{p(x)},\ i=1,\cdots,M \qquad (3)$$

그리고

$$p(x)=\sum_{i=1}^{M}p(x|d=i)P(d=i) \qquad (4)$$

여기에서 $d=i$가 의미하는 것은 M개의 신호 집합 가운데 i번째 신호에 속하는 데이터 d를 나타낸다. 또한 $p(x \mid d=i)$는 $d=i$가 주어졌을 때, 잡음이 더해져 수신된 신호 x의 확률 밀도 함수(pdf)를 나타내며 $p(x)$는 전체 신호집합에 대하여 수신된 신호 x의 pdf 이다. 식 (3)에서 p는 연속적인 값을 가지는 신호의 pdf를 나타내며 P는 a priori와 APP값을 나타내기 위하여 사용하였다. 식 (3)은 수신된 신호와 그 신호가 속해 있는 어떠한 신호 집합에 대한 정보를 통하여 얻어진 결과라고 할 수 있다. i번째 신호의 발생 확률, 즉 $P(d=i)$는 사전 확률(*a priori* probability)이며 이를 통하여 APP, 즉 $P(d=i \mid x)$를 얻을 수 있다.

일반적으로 이진 논리 성분 1 과 0은 전기적 신호 +1과 -1로 나타낼 수 있으며 이 같은 한 쌍의 전기적 신호는 변수 d에 할당되며 따라서 $d=+1$과 $d=-1$값을 가지게 된다. 가우시안 채널에 대하여 그림 1은 likelihood 함수으로서 나타난 조건부 확률 밀도 함수(conditional pdf)를 보여주고 있다.



〈그림 1) Likelihood 함수

확률 $p(x \mid d=+1)$는 $d=+1$이 전송되었다는 조건하에 주어진 랜덤 변수 x의 pdf를 보여주고 있으며 또한 함수 $p(x \mid d=-1)$은 $d=-1$이 전송되었다는 조건하에 주어진 랜덤 변수 x의 pdf를 보여주고 있다. 그림 1은 임의의 검출된 값 x_k를 보여주고 있다. x_k에서 보면 두 개의 likelihood function값 λ_1과 λ_2를 가지게 된다. *maximum likelihood*로서 잘 알려진 경판정 기법(hard decision)은 두 값 λ_1과 λ_2 가운데 큰 값과 관련하여 $d=+1$ 또는 $d=-1$을 선택하는 것이다. 이것은 γ_0를 기준으로 하여 만약 x_k가 오른쪽에 위치하

게 되면 $d=+1$로 결정하고 그렇지 않다면 $d=-1$로 선택하는 것과 같은 이치이다. 이러한 *maximum likelihood* 결정 방법과 유사한 방법으로 아래의 식 (6)에서 보이는 사전 확률(*a priori* probability)를 고려한 *maximum a posteriori* (MAP) 또는 *minimum error rule* 방식이 있다. MAP 규칙은 다음과 같은 *APP*로서 표현된다.

$$P(d=+1|x) \underset{H_2}{\overset{H_1}{\gtrless}} P(d=-1|x) \tag{5}$$

식 (5)는 만약 APP $P(d=+1|x)$이 APP $P(d=-1|x)$보다 크다면 $H_1(d=+1)$을 선택하고 그렇지 않다면 $H_2(d=-1)$를 선택하는 것을 나타내고 있다. 식 (3)의 Bayes 정리를 이용하여 식 (5)를 다시 나타내면 식 (6)과 같다.

$$p(x|d=+1)P(d=+1) \underset{H_2}{\overset{H_1}{\gtrless}} p(x|d=-1)P(d=-1) \tag{6}$$

식 (6)은 일반적으로 *likelihood ratio*라 하며 다음과 같이 나타낼 수 있다.

$$\frac{p(x|d=+1)}{p(x|d=-1)} \underset{H_2}{\overset{H_1}{\gtrless}} \frac{P(d=-1)}{P(d=+1)}, \quad \frac{p(x|d=+1)}{p(x|d=-1)} \frac{P(d=+1)}{P(d=-1)} \underset{H_2}{\overset{H_1}{\gtrless}} 1 \tag{7}$$

식 (7)에 나타난 Likelihood Ratio에 Logarithm을 취하여 Log-Likelihood Ratio(LLR)을 얻을 수 있으며 이것은 검출기(detector) 출력에서의 연판정(soft decision)값을 나타낸다.

$$L(d|x)=\log\left[\frac{P(d=+1|x)}{P(d=-1|x)}\right] = \log\left[\frac{P(x|d=+1)P(d=+1)}{P(x|d=-1)P(d=-1)}\right] \tag{8}$$

$$L(d|x)=\log\left[\frac{p(x|d=+1)}{p(x|d=-1)}\right] + \log\left[\frac{P(d=+1)}{P(d=-1)}\right] \tag{9}$$

$$L(d|x)=L(x|d)+L(d) \tag{10}$$

여기에서 $L(x|d)$는 $d=+1$ 또는 $d=-1$이 전송되었다는 조건하에 x에 대한 채널 추정값(channel measurements)에 대한 LLR값이다. 그리고 $L(d)$는 데이터 비트 d에 대한 *a priori* LLR 값이다. 표현상의 간략화를 위하여 식 (10)을 다음과 같이 나타낼 수 있다.

$$L'(d)=L_c(x)+L(d) \tag{11}$$

systematic 부호에 대하여 복호기 출력의 LLR 값 $L(\hat{d})$는 다음과 같다.

$$L(\hat{d})=L'(\hat{d})+L_e(\hat{d}) \tag{12}$$

여기에서 $L'(\hat{d})$는 검출기 출력 비트(복호기 입력)의 LLR 값이며 $L_e(\hat{d})$는 extrinsic LLR이며 복호 과정에서 얻어지는 정보이다. 식 (11)을 이용하여 식 (12)를 나타내면 다음과 같다.

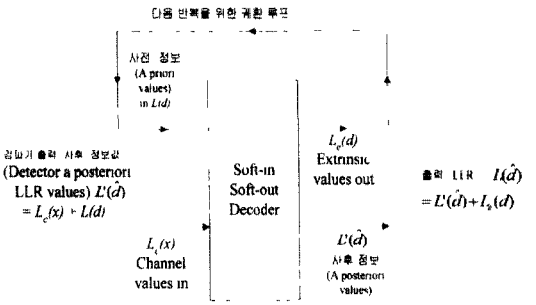
$$L(\hat{d})=L_c(x)+L(d)+L_e(\hat{d}) \tag{13}$$

연판정 출력값(soft decision) $L(\hat{d})$는 복호 비트 결정을 위한 신뢰성(reliability)뿐 아니라 경판정 출력(hard decision)을 제공하는 실수값이다. $L(\hat{d})$의 부호를 통하여 만약 $L(\hat{d})$의 부호가 양수라면 +1로 결정하고 그렇지 않다면 -1로 결정하게 된다.

반복 복호의 원리

그림 2에서는 첫번째 복호 과정에서의 soft-in soft-out 복호기의 동작을 보이고 있다.

일반적으로 이진 데이터가 동일하게 발생하는 것을 가정하며 식 (9)의 두번째 항인 a priori

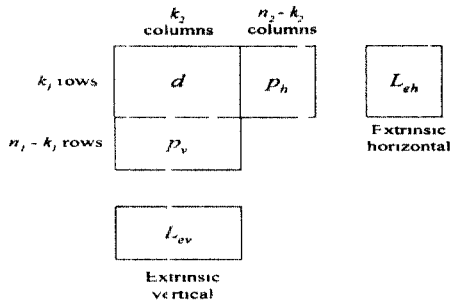


〈그림 2〉 *Soft input/Soft output decoder(for a systematic code)*

LLR 값인 $L(d)=0$값으로 초기화된다. 채널 추정(channel measurements) LLR $L_c(x)$는 그림 1에서 보인 λ_1과 λ_2 사이의 비(ratio)의 logarithm을 통하여 얻을 수 있으며 식 (9)의 첫번째 항으로 나타난다. 그림 2의 복호기 출력 $L(\hat{d})$는 검출기(detector) 출력의 LLR $L'(\hat{d})$와 복호 과정에서 얻지는 정보인 *extrinsic* LLR $L_e(d)$ 값으로 이루어진다. 그림 2에서 보인 것처럼 반복 복호 과정에서 이러한 *extrinsic likelihood*는 다음 복호 단계에서 a priori 정보로 대체되기 위하여 복호기의 입력으로 전환된다.

그림 3에 보인 이차원 부호(product code)를 생각하여 보자. 그림에서 알 수 있듯이 데이터 열은 k_1의 열과 k_2의 행으로 구성되어 있다. k_1 각각의 열들은 k_2의 데이터 비트와 n_2-k_2의 패리티 비트로 이루어진 부호 벡터(code vector)이다. 마찬가지로 k_2 각각의 행들 역시 k_1의 데이터 비트와 n_1-k_1의 패리티 비트로 이루어진 부호 벡터(code vector)이다. 그림에서 L_{eh}와 L_{ev}는 각각 수평, 수직 복호 단계에서 얻은 *extrinsic* LLR 값이다. 이러한 이차원 부호는 연접 부호(concatenated code)의 한 예이다. 이러한 부호의 부호기 구조는 수평, 수직의 두 가지 단계로 분리할 수 있으며 이차원 부호(product code)를 위한 반복 복호 알고리즘의 과정은 다음과 같다.

연판정(soft decisions) 결과 값에 대한 반복 순환(iterative feedback)에 대하여 좀 더 자세히 알아보기 위하여 log-likelihood algebra 개념을 도



〈그림 3〉 이차원 product code

1. 사전 정보(*a priori* information) $L(d)=0$으로 초기화한다.
2. 수평적으로 복호 동작을 수행하며 식 (13)을 이용하여 외부 정보(extrinsic information)를 얻어낸다.
$$L_{eh}(\hat{d}) = L(\hat{d}) - L_c(x) - L(d).$$
3. $L(d) = L_{eh}(\hat{d})$.
4. 수직적으로 복호 동작을 수행하며 식 (13)을 이용하여 외부 정보(*extrinsic* information)를 얻어낸다.
$$L_{ev}(\hat{d}) = L(\hat{d}) - L_c(x) - L(d)$$
5. $L(d) = L_{ev}(\hat{d})$.
6. 만약, 충분한 반복을 통하여 신뢰할 수 있는 결정값을 얻는다면 단계 7로 이동하고 그렇지 않다면 단계 2로 이동한다.
7. 연판정 출력값(soft output) 결정
$$L(\hat{d}) = L_c(x) + L_{eh}(\hat{d}) + L_{ev}(\hat{d})$$

입해서 통계적으로 독립인 d에 대하여 다음과 같은 두 개의 log-likelihood ratios(LLRs)의 합을 정의하도록 한다.

$$L(d_1) \boxplus L(d_2) \triangleq L(d_1 \oplus d_2)$$
$$= \log_e \left[\frac{e^{L(d_1)} + e^{L(d_2)}}{1 + e^{L(d_1)} e^{L(d_2)}} \right] \qquad (14)$$

$$\approx (-1) \times sign[L(d_1)] \times sign[L(d_2)] \times \min(|L(d_1)|, |L(d_2)|) \qquad (15)$$

식 (14)에서는 세 개의 덧셈 연산이 사용되고

있다. +는 일반적인 덧셈 연산이며, ⊕는 modulo -2 연산을 의미한다. 마지막으로 ⊞는 log-likelihood 덧셈 연산을 위하여 사용되었다. 식 (14)의 유도 과정은 부록 A에 제시되고 있다. 식 (14)와 식 (15)에 나타난 두 개의 LLR 값의 합은 LLR 값이 아주 크거나 작을 때 다음과 같은 아주 흥미 있는 결과를 나타내고 있다.

$$L(d) \boxplus \infty = -L(d)$$
$$L(d) \boxplus 0 = 0$$

2차원 부호에 대하여 식 (13)을 이용하여 임의의 데이터 d_l과 관계되어 수신된 심볼의 soft output $L(\hat{d}_l)$을 나타내면 다음과 같다.

$$L(\hat{d}_1) = L_c(x_1) + L(d_1) = ([L_c(x_2) + L(d_2)] \boxplus L_c(x_{12})) \quad (16)$$

여기에서 $([L_c(x_2)+L(d_2)] \boxplus L_c(x_{12}))$는 부호에 의해서 나타나는 *extrinsic* LLR이다. (즉, 데이터 d_2와 관계되는 수신 심볼과 그것의 *a priori* probability, 그리고 패리티 비트 p_{12}와 관계되는 신호와 관련되어 나타난 *extrinsic* LLR 값이다.) 일반적으로 데이터 d_l와 관계된 수신 심볼의 soft output $L(\hat{d}_l)$는 다음과 같다.

$$L(\hat{d}_l) = L_c(x_l) + L(d_l) + ([L_c(x_j) + L(d_j)] \boxplus L_c(x_{lj})) \quad (17)$$

여기에서 $L_c(x_l)$, $L_c(x_j)$, $L_c(x_{lj})$는 각각 d_l, d_j, p_{lj}과 관계된 수신 심볼에 대한 채널 추정값(channel measurements)의 LLR 값이다. $L(d_l)$, $L(d_j)$는 *a priori* probability이며 $([L_c(x_j)+L(d_j)] \boxplus L_c(x_{lj}))$는 *extrinsic* LLR 값이다. 식 (17)를 이용하여 반복 복호 과정중 LLR값의 변화를 계산할 수 있으며 실제 반복 복호를 통하여 복호 비트 결정을 위한 어느 정도의 LLR 값의 향상을 가져오는 것을 알 수 있으며 또한 복호 하고자 하는 비트 사이의 값이 균형을 이루는 것을 알 수 있다.[18]

Soft-output 복호의 원리

랜덤 변수 X ∈ {+1, -1}을 고려하자. 이 때, LLR $L_x(x)$를 다음과 같이 정의한다.

$$L_x(x) = \ln \frac{P_x(x=+1)}{P_x(x=-1)}, \quad L_x(x) \in IR \quad (18)$$

이 때 랜덤 변수(bit) X ∈ {+1, -1}에 대해서 GF(2)상에서의 연산은 다음과 같은 덧셈(Exclusive-OR)을 만족한다. 그리고 '+1'은 이 연산에서 'null'값을 가지게 된다.

⊕	+1	−1
+1	+1	−1
−1	−1	+1

⟶

⊕	0	1
0	0	1
1	1	0

그리고 정수에서의 연산과 같이, GF(2)상에서의 곱셈은 다음과 같으며 위의 덧셈의 결과와 일치한다.

•	+1	−1
+1	+1	−1
−1	−1	+1

AWGN 채널을 통하여 bit x를 전송하였을 때, *a posteriori* LLR $L(x \mid y)$를 계산하면 다음과 같다.

$$\begin{aligned} L(x|y) &= \ln \frac{P(x=+1|y)}{P(x=-1|y)} \\ &= \ln \left(\frac{f(y|x=+1)}{f(y|x=-1)} \cdot \frac{P(x=+1)}{P(x=-1)} \right) \\ &= \ln \frac{f(y|x=+1)}{f(y|x=-1)} + L(x) \\ &= L_c \cdot y + L(x) \end{aligned} \quad (19)$$

여기에서 L_c는 CSI(channel state information)로 나타내며 $4\frac{E_s}{N_0}(=\frac{2}{\sigma^2})$로 주어지며 $L(x)$는 *a priori* information이다. 일반적으로 $L_x(x)$를 부

호화한 값으로 분리하여 $\bar{x}=sign(L_\lambda(x))$으로 hard 값을 결정하고 $|L_\lambda(x)|$는 이러한 결정의 신뢰도를 나타낸다[13].

또한, 식 (A. 6)와 식 (A. 7)에서 알 수 있는 것처럼 $P(x=i)$의 값은 다음과 같다.

$$P(x=+1)=\frac{e^{L(x)}}{1+e^{L(x)}} \tag{20}$$

$$P(x=-1)=\frac{e^{-L(x)}}{1+e^{-L(x)}} \tag{21}$$

위의 식들을 이용하여 bit x에 대한 기대값을 구해보면 다음과 같다.

$$E\{x\}=(+1)\frac{e^{L(x)}}{1+e^{L(x)}}+(-1)\frac{e^{-L(x)}}{1+e^{-L(x)}}$$
$$=tanh(L(x)/2) \tag{22}$$

또한 다음과 같은 "soft-bit" $\lambda(x)$를 정의할 수 있다.

$$\lambda(x)=E\{x\}=tanh(L(x)/2) \tag{23}$$

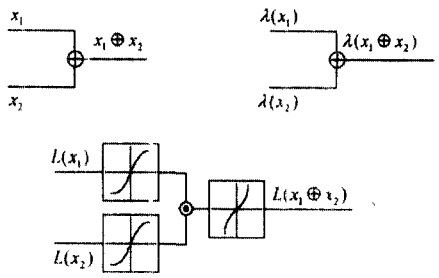
$$L(x)=2arctanh(\lambda(x)) \tag{24}$$

따라서,

$$\lambda(x_3)=\lambda(x_1\oplus x_2)=E\{x_1\cdot x_2\}=E\{x_1\}\cdot E\{x_2\}$$
$$=\lambda(x_1)\cdot\lambda(x_2) \tag{25}$$

이며 LLR은 modulo-2 덧셈($x_3=x_1\oplus x_2$)에 의해서 bit x_1과 x_2로 나누어 계산할 수 있으며 각각 그림 3-1과 같다.

$$L(x_3)=2\arctan h(\lambda(x_3))=2arctanh\{tanh(L(x_1)/2)$$
$$\cdot tanh(L(x_2)/2)\} \tag{26}$$



〈그림 3-1〉 LLR의 modulo 연산 과정

III. TWL 그래프와 순환 반복

반복적인 MAP 복호 알고리즘은 LDPC(Low-Density Parity Check) 부호의 복호를 위하여 Gallager에 의해서 제안되었다. 그리고 MAP 심볼 검출과 hidden Markov 모델을 탐색(tracking)하기 위하여 발전된 BFA(Backward-Forward Algorithm)은 Tanner 혹은 Wiberg 등의 많은 기존의 학자들에 의해서 다양하게 일반화되었다[5, 6]. LDPC 부호가 sum-product 알고리즘의 많은 반복을 통하여 가우시안 채널 상에서 Turbo Codes와 거의 비슷한 성능에 도달할 수 있으며 또한 채널 용량에 도달할 수 있다. Turbo Codes는 무기억 채널 상에서 가장 높은 성능을 보이며 이는 TWL(Tanner Wiberg Loeliger) 그래프의 순환(cycle) 구조에서 기인한다. TWA(Two-way algorithm)의 하나로 LDPC 부호에 대한 Gallager의 APP 복호 알고리즘이 있으며[1, 2] 다른 하나는 ISI 채널상에서 MAP 심볼 검출에 대한 Chang과 Hancock[19]의 MAP 또는 hidden Markov 모델의 탐색(tracking)을 위한 Baum과 Petrie[3]에 의한 MAP 알고리즘이다. 현재 부호 이론 분야에서는 종종 BCJR 알고리즘으로 불리며 음성 인식[3] 등을 위하여 널리 사용된다.

Tanner는 심볼들과 패리티 검사를 표현하는 두 가지 형태의 노드를 가지는 Tanner 그래프 상에서 Gallager의 LDPC 부호를 일반화시켰다[2]. 또한 max-sum과 sum-product라 불리는 두 알고리

즘을 발전시켰으며 유한한 Cycle-free 그래프 상에서 이들이 수렴(converge) 한다는 것을 증명하였다. 또한 Mackay와 Neal은 LDPC 부호가 Turbo Codes와 유사하다는 것을 보였으며 둘 다 샤논의 채널 용량(channel capacity)에 도달할 수 있다는 것을 보였다[10].

1. 그래프 상에서 정의된 부호

유한 필드 F 상의 선형 블록 부호는 패리티 검사들의 집합에 의해 정의될 수 있다. H는 F 상에서 $\gamma \times n$ 행렬이라 하자. 그 때 패리티 검사 행렬에 의해서 정의되는 부호 C는 $\boldsymbol{H}_x=0$이 되는 F상의 모든 x들의 집합이다.

예를 들어 3×7 패리티 검사 행렬

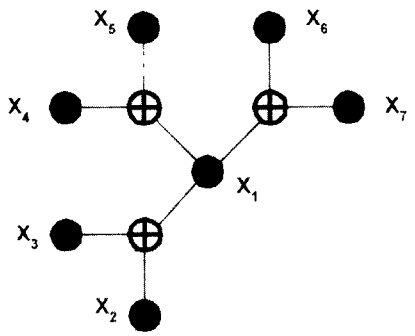
$$[\boldsymbol{H}]=\begin{bmatrix}1110000\\1001100\\1000011\end{bmatrix}$$

는 간단한 이진 선형 블록 부호 C를 정의한다. 즉, $2^4=16$ 이진 7-tuples $\boldsymbol{X}=(x_1, x_2, \cdots, x_7)$은 다음의 검사를 만족한다.

$$\begin{aligned}x_1 \oplus x_2 \oplus x_3 &= 0,\\ x_1 \oplus x_4 \oplus x_5 &= 0,\\ x_1 \oplus x_6 \oplus x_7 &= 0;\end{aligned}$$

이 부호는 길이 7, $(\boldsymbol{F}_2)^7$의 subspace로서 차원 4이며 최소 해밍거리 2인 (7, 4, 2) 이진 선형 부호이다. Tanner 그래프[5]는 부호를 만족하는 패리티 검사들과 일치하는 부호의 도식적인 표현이다. 각 심볼은 심볼 노드에 의해 표현되며 각 패리티 검사는 검사 노드에 의해 표현된다. 그래서 Tanner 그래프는 심볼 노드와 검사 노드만이 존재하는 그래프이다.

예를 들어, 그림 4는 앞의 (7, 4, 2)부호와 일치하는 Tanner 그래프를 보인다. 심볼 노드는 ●으로 표현되며 검사 노드는 ⊕에 의해 표현되었다. 확률적 관점에서의 Tanner 그래프는 Bayesian 망에서 의존성(dependency)을 정의하며[9] 특히, 조건부 의존성(conditional dependency)의 경우에 대해서이다. 예를 들어, 그림 4에서 x_1이 주어졌을 때 (x_2, x_3), (x_4, x_5), (x_6, x_7)은 서로 조건부 독립(conditional independent)이다.



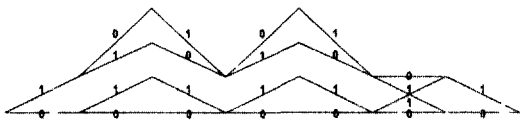
〈그림 4〉 (7,4,2) 이진 부호를 위한 Tanner 그래프

이러한 관점에서 변수 x_1은 시스템 이론의 개념에서 "state"처럼 동작하거나 확률적 추론의 관점에서 "sufficient statistic"처럼 동작한다. 좀 더 일반적으로 보면 연결이 없는 두 부분으로 분할하는 임의의 'cut'은 부호의 "state space"를 정의한다[11].

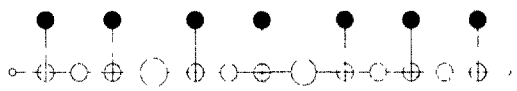
Wiberg 등의 문헌[6,7]에서는 보이지 않는 "state"를 표현하는 노드의 세 번째 유형을 소개한다. 이것은 격자도를 가지는 Tanner 그래프와의 연관을 가지며 부호가 표현되는 그래프의 종류를 확장시킨다. 이것을 Tanner-Wiberg-Loeliger(TWL) 그래프라 부른다.

예를 들어, 그림 5(a)는 앞의 (7, 4, 2) 블록 부호에 대한 최소 격자도를 보이며 크기 {1, 2, 4, 2, 4, 2, 2, 1}의 "state spaces"를 가진다. 그림 5(b)는 이 부호에 일치하는 TWL 그래프를 보인다. 여기에서 모든 선형 부호는 필수적으로 유일한 최소 격자도를 가지며, 이것은 모든 면에서 최소(minimal)를 의미한다. 그리고 격자도의 복잡도는 블록 길이에 대해 지수적으로 증가한다.

윗 그림을 부연 설명하면 ○는 각각 격자도 상에서 분기되는 노드의 수를 나타내며 ●는 부호화된 심볼이다.



〈그림 5(a)〉 (7,4,2) 이진 부호를 위한 최소 격자도 (minimal trellis diagram)



〈그림 5(b)〉 (7,4,2) 이진 부호를 위한 격자도와 일치하는 TWL 그래프

법례 :

: 격자도 상에서 분기되는 노드의 수 (1)

○ : 격자도 상에서 분기되는 노드의 수 (2)

◯ : 격자도 상에서 분기되는 노드의 수 (4)

● : 부호화된 심볼

2. Cycle-Free 그래프에 대해 정의된 부호의 복호화

Two-way 알고리즘은 격자도와 일치하는 그래프를 포함한 유한한 값의 심볼과 state 노드를 가지는 유한한 cycle-free TWL 그래프를 위한 임의의 복호화 문제의 직접적 해이다.

max-sum 알고리즘

각 심볼 노드 x_k의 값 a_k는 weight $w_k(a_k)$로 할당된다. 즉 log-likelihood 값 $w_k(a_k)=\log P(y_k \mid x_k=a_k)$이며 여기서 y_k는 부호 심볼 x_k와 일치하는 수신 심볼이다. 문제는 $x_k=a_k$인 임의의 부호 $X\in C$의 최대 weight인 각 심볼 노드 x_k의 가능한 값 a_k를 계산하는 것이다. 즉, $\max\{w(x)\} \mid X\in C, x_k=a_k\}$이다.

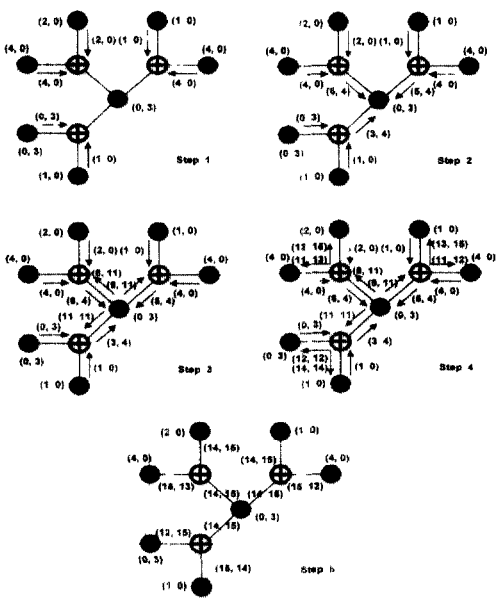
이 때 max-sum 알고리즘은 유한한 순환이 존재하지 않는(cycle-free) 그래프를 이용하여 이 문제를 해결한다. 이를 위하여 "upstream"과 "downstream"이라 부르는 두 개의 분리 부분으로 그래프를 분할한다. 이 때 임의의 부호 X의 *weight* $w(x)$는 $w(x)=w(x^+)+w(x^-)$로 표현할 수 있다. 이 알고리즘은 두 개의 나누어진 부분을 계산하며 "Two-way 알고리즘"이라 부른다. 예를 들어, 그림 6에서 이의 계산과정을 보이며 다음과 같은 'update rule'을 따른다. 여기에서 각 심볼 노드는 그림 6에서와 같이 weight의 쌍 $\{w_k(0), w_k(1)\}$으로 주어진다.

단계 1 : 여섯 개의 잎 노드에 대하여 심볼 weight들은 이웃 edge에 간단하게 전달된다.

단계 2 : 세 개의 parity-check 노드 각각에 대해 두 개의 출력이 존재하며 ACS (Add-Compare-Select) 동작이 최대값을 찾기 위해 수행된다.

단계 3 : 중앙 노드와 외부의 세 edge 각각에서 출력되는 두 값들에 대해서, 두 개의 입력 weight는 출력 weight를 주기 위해 중앙 노드와 합해진다. 이 때 downstream과 upstream의 합이 중앙 노드에서 계산된 결과와 같다.

단계 4 : 단계 2에서의 ACS 연산을 세 개의 parity-check 노드에 대해 수행한다.



〈그림 6〉 max-sum 알고리즘을 이용하여 그림 4의 Tanner 그래프 복호화

단계 5 : upstream과 downstream weight들은 각 edge 값의 마지막 weight를 주기 위해 합해진다.

다시 한번 그림 6의 과정을 살펴보면 step 1에서 각 노드의 weight는 임의의 값이다. step 2에서는 순방향 ACS 값인데 앞에서 설명한 것처럼 최대 weight를 가지는 값을 선택하게 된다. 즉 weight (2,0)과 weight (4,0)에서 최대 weight를 가지는 값은 (2+4=6, 0+4=4)이다. step 3에서는 순방향 ACS에서 계산된 weight에 대하여 중앙 노드는 자신을 포함한 모든 weight를 더한다. 즉, (6,4)+(5,4)+(3,4)+(0,3)=(14,15)이다. 그리고 더해진 weight로부터 순방향 ACS 값을 뺀 후 역방향 ACS 과정을 시작한다.

(14,15) - (5,4) = (9,13)
(14,15) - (3,4) = (11,11)
(14,15) - (6,4) = (8,11)

step 4에서는 역방향 ACS 과정을 step 2처럼 수행하며 마지막으로 step 5에서는 step 4까지의 weight에 대하여 자신의 노드 weight를 더하여 최종 weight를 계산하게 된다.

(13,15) + (1,0) = (14,15)

3. 순환(Cycle)이 존재하는 그래프 상에서 정의되는 부호의 복호화

그래프에서 순환(cycle)이 존재할 때, 알고리즘을 일반화하는 방법은 명확하지가 않다. 문제는 다음과 같은 것들을 고려하여야 한다.

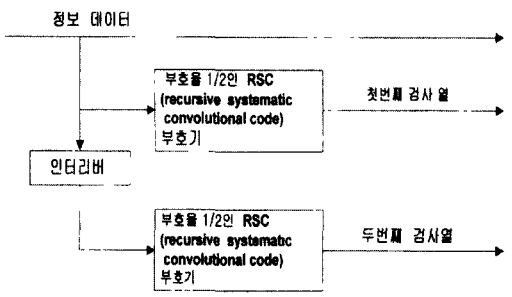
- 초기화(Initialization)
- 값 갱신의 순서(Order of updating)
- 수렴(Convergence)
- 종결 규칙(Stopping rule)

Low-density parity-check 부호

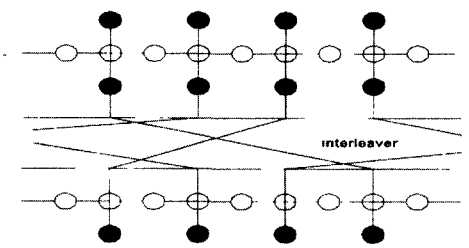
LDPC 부호는 모든 그래프 부호와 복호화 알고리즘의 원조라 볼 수 있으며 Gallager에 의해 소개되었다[2]. (j, k) LDPC 부호에서, 모든 부호 심볼은 정확하게 j개의 패리티 검사에 의해 검사되며, 모든 패리티 검사는 k개의 부호 심볼들을 정확히 검사한다. Gallager는 간단한 "flipping algorithm"뿐만 아니라 sum-product(APP) 알고리즘의 변형과 같은 알고리즘을 제시하였다. 이 알고리즘은 모든 APP 값이 '0' 혹은 '1'에 근접할 때 종료한다. 또한 Mackay와 Neal[8]은 LDPC 부호가 sum-product 알고리즘의 많은 반복을 통해 가우시안 채널 상에서 Turbo Codes와 거의 비슷한 성능에 도달할 수 있으며, 더욱이 원리상으로 LDPC 부호가 채널 용량(channel capacity)에 도달할 수 있다는 것을 보여주었다.

Turbo Codes

Turbo Codes는 가우시안과 같은 무기억 채널 상에서 가장 높은 성능을 가지는 부호이다. 하나의 표준적인 예가 그림 7에서 보이듯이 Berrou 등의 원 논문[17]에 있다. 이러한 부호의 TWL 그래프는 그림 8과 같이 표현하며 Wiberg의 논문[7]에 나타나있다. 그림 8에서 각 구성 부호의 그래프는 순환(cycle)이 없으며 인터리버는 순환(cycle)들을 생성하기 위한 필수적인 상호 연결로 볼 수 있다. 인터리버가 커지면 순환(cycle)이 커지는 것을 볼 수 있으며 Wiberg는 "Turbo Codes의 놀랄만한



〈그림 7〉 Turbo Codes 부호기



〈그림 8〉 Turbo Codes를 위한 TWL 그래프

성능은 TWL 그래프의 순환(cycle) 구조에서 기인한다"고 주장하였다[7].

IV. 결 론

차세대 무선 멀티미디어 통신에서는 음성뿐만 아니라 데이터, 영상까지도 고품질의 다양한 서비스를 제공해야 한다. 그러나 고품질의 서비스를 제공하기 위해서는 이동 통신 채널에서 필연적으로 나타나는 에러를 제어하는 채널코딩에 대한 연구가 무엇보다 중요하다고 본다. 따라서 본 고에서는 차세대 무선 멀티미디어 통신의 무선접속에 관련된 채널코딩에 관하여 표준화 동향을 살펴보았고 Turbo Codes에 대해 기술하였다. 그리고 sum-product 알고리즘의 반복 복호화 과정을 통하여 가우시안 채널 상에서 Turbo Codes와 거의 비슷한 성능에 도달할 수 있으며 또한 채널 용량에 도달할 수 있는 LDPC 부호에 대하여 알아보았다. 또한 Turbo Codes의 반복 복호 알고리즘의 기반이 되는 Bayes' 정리와 반복 복호에 의한 성능 향상을 수학적 모델을 통하여 검증하였다. 심볼들과 패리티 검사를 표현하는 두 가지 형태의 노드를 가지는 Tanner 그래프(TWL 그래프) 상에서 Turbo Codes를 비롯한 여러 부호들의 동작을 알아보고 또한 a priori information을 이용하여 이차원 부호(product code)의 반복 복호 과정과 복호 과정에서의 extrinsic LLR에 대하여 살펴보았다. 한번의 반복 복호(수평 복호와 수직 복호)가 이루어지면 복호 비트 결정을 위한 어느 정도의 LLR 값의 향상을 가져오는 것을 알 수 있었으며 반복의 회수가 증가할수록 최종 복호 비트 결정(hard decision)을 위한 신뢰값이 향상되는 것을 알 수 있다. Turbo Codes의 feedback 알고리즘은 통신의 채널부호 뿐만 아니라 통신 전자 기계 등 제반 공학에 유용하게 쓰일 수 있는 알고리즘으로 주목되고 있다.

(본 논문을 위하여 많은 조언을 해주신 독일 뮌헨 공대의 J. Hagenauer 교수에게 감사드립니다.)

참 고 문 헌

[1] R.G. Gallager, "Low-density parity-check codes," IRE Trans. Inform. Theory, vol. IT-8, pp.21-28, Jan. 1962.

[2] R.G. Gallager, Low-density Parity-check Codes. Cambridge, MA:MIT Press, 1993.

[3] L.E. Baum and T. Petrie, "Statistical inference for probabilistic function of finite-state Markov chains," Ann. Math. Stat., vol. 37, pp.1554-1563, 1966.

[4] L. Rabiner, "A tutorial on hidden Markov models and and selected applications in speech recognition," Proc. IEEE, vol. 77, pp.257-285, Feb. 1989.

[5] R.M. Tanner, "A recursive approach to low-complexity codes," IEEE Trans. Inform. Theory, vol. IT-27, pp.533-547, Sept. 1981.

[6] N. Wiberg, H.A. Loeliger and R. Kotter, "Codes and iterative decoding on general graphs," Euro. Trans. Telelcommun, vol. 6, pp.513-526, Sept. 1995.

[7] N. Wiberg, "Codes and decoding on general graphs," Ph.D. dissertation, Dept. Elec, Engg., U. Linkoping, Sweden, Apr. 1996.

[8] D.J.C. MacKay and R.M. Neal, "Near Shannon limit performance of low-density parity-check codes," Elect. Lett., vol. 32, pp.1645-1646, Aug. 1996.

[9] J. Pearl, Probablilistic Reasoning in Intelligent Systems: Network of Plausible Inference. San Mateo, CA: Morgan

Kaufmann, 1988.
[10] D.J.C. MacKay and R.M. Neal, "Good error-correcting codes based on very sparse matrices," preprint, received Sept. 1996.
[11] G.D. Forney, Jr. and M.D. Trott, "The dynamics of group codes: State spaces, trellis diagrams and canonical encoders," IEEE Trans. Inform. Theory, vol. 39, pp. 1491-1513, Sept. 1993.
[12] G. Solomon and H.C.A. van Tiborg. "A connection between block and convolutional codes," SIAM J. Appl. Math, vol. 37, pp. 358-369, Oct. 1979.
[13] J. Hagenaeur, " The Turbo Principle : Tutorial Introduction and state of the Art", Proc. of the International Symposium on Turbo Codes & Related Topic. Brest, 3-5 Sept. 1997.
[14] D.J. Muder, "Minimal trellises for block codes," IEEE Trans. Inform. Theory, vol. 34, pp.1046-1053, Sept. 1988.
[15] G.D. Forney, "On Iterative Decoding of the Two-way Algorithm," Proc. of the International Symposium on Turbo Codes & Related Topic. Brest, 3-5 Sept. 1997.
[16] C. Berrou, A. Glavieux and P. Thitimajshima, "Near Shannon limit error -correcting coding and decoding: Turbo codes(1)," Proc. ICC '93, Geneva, pp. 1064-1070, June 1993.
[17] B. Sklar," A Primer on Turbo Code Concepts," IEEE Comm. Magazine, pp. 94-102, Dec. 1997.
[18] 이문호, 실용 정보 이론, 복두 출판사, 1998.
[19] 이문호, C·Matlab 실용디지털 통신, 도서출판 영일, 1998.

부록 A.

아래의 계산과정은 식 (22)의 유도 과정이다.

$$L(d_1)\boxplus L(d_2) \triangleq L(d_1 \oplus d_2) = \log_e\left[\frac{e^{L(d_1)}+e^{L(d_2)}}{1+e^{L(d_1)}e^{L(d_2)}}\right] \quad (A1)$$

$$L(d) = \log_e\left[\frac{P(d=+1)}{P(d=-1)}\right] = \log_e\left[\frac{P(d=+1)}{1-P(d=+1)}\right] \quad (A2)$$

$$e^{L(d)} = \left[\frac{P(d=+1)}{1-P(d=+1)}\right] \quad (A3)$$

$$e^{L(d)} - e^{L(d)} \times P(d=+1) = P(d=+1) \quad (A4)$$

$$e^{L(d)} = P(d=+1)\times\left[1+e^{L(d)}\right] \quad (A5)$$

$$P(d=+1) = \frac{e^{L(d)}}{1+e^{L(d)}} \quad (A6)$$

$$P(d=-1) = 1-P(d=+1) = 1-\frac{e^{L(d)}}{1+e^{L(d)}} = \frac{1}{1+e^{L(d)}} \quad (A7)$$

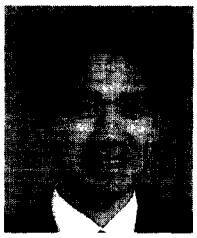
$$L(d_1\oplus d_2) = \log_e \left[\frac{P(d_1=+1)\times P(d_2=-1)+[1-P(d_1=+1)][1-P(d_1=-1)]}{P(d_1=+1)\times P(d_2=+1)+[1-P(d_1=+1)][1-P(d_1=+1)]}\right] \quad (A8)$$

$$= \log_e\left[\frac{\left(\frac{e^{L(d_1)}}{1+e^{L(d_1)}}\right)\left(\frac{1}{1+e^{L(d_2)}}\right)+\left(\frac{1}{1+e^{L(d_1)}}\right)\left(\frac{e^{L(d_2)}}{1+e^{L(d_2)}}\right)}{\left(\frac{e^{L(d_1)}}{1+e^{L(d_1)}}\right)\left(\frac{e^{L(d_2)}}{1+e^{L(d_2)}}\right)+\left(\frac{1}{1+e^{L(d_1)}}\right)\left(\frac{1}{1+e^{L(d_2)}}\right)}\right] \quad (A9)$$

$$= \log_e\left[\frac{\left(\frac{e^{L(d_1)}+e^{L(d_2)}}{[1+e^{L(d_1)}][1+e^{L(d_2)}]}\right)}{\left(\frac{e^{L(d_1)}e^{L(d_2)}+1}{[1+e^{L(d_1)}][1+e^{L(d_2)}]}\right)}\right] \quad (A10)$$

$$= \log_e\left[\frac{e^{L(d_1)}+e^{L(d_2)}}{1+e^{L(d_1)}e^{L(d_2)}}\right] \quad (A11)$$

저 자 소 개



李 門 浩

일본 동경대 전자과 공학 박사(1990), 통신 기술사(1983), 미국 미네소타 주립대 전기과 포스트 닥터(1985), 남양 MBC 송신소장(1970~1980), 독일 하노버 대학(1990 겨울), 아흔 공대(1992 여름, 1995 겨울), 뮌헨 공대(1998 여름) 연구 교수, 1980~현재 전북대학교 정보통신공학과 교수, 1997년~현재 한국 공학 한림원 회원 및 정보통신 정책 심의 위원, 디지털 방송 추진위원