

〈主 題〉

IMT-2000을 위한 채널 코딩 기술

이 문 호

(전북대학교 정보통신공학과)

□ 차 례 □

- I. 서 론
- II. IMT-2000 채널 코딩의 표준화
- III. 연결 부호기(Concatenated Code)
- IV. 터보 코드(Turbo Code)의 성능 분석
- V. 결 론

I. 서 론

IMT-2000(International Mobile Telecommunications-2000)/FPLMST(Future Public Land Mobile Telecommunication Systems)은 차세대 이동 멀티미디어 통신의 고품질 서비스를 목표로 전세계 어느곳에서나 하나의 단말기로 통신이 가능하도록 하자는데 있다. 현재의 이동 통신 방식에서는 한국에서 동작하는 이동 전화기를 가지고 일본이나 기타지역으로 이동하였을 경우 서비스를 받을수 없다. 이러한 1,2세대 이동 통신의 불편함을 제거하기 위하여 ITU에서는 전 세계 어느 지역에서나 동일 단말기로 통신 가능한 차세대 이동 통신에 대한 표준화를 추진하기로 결정하고 이를 위한 작업을 진행중이다. 이러한 작업은 이동 통신 시스템은 점점더 발전하고 있지만 각기 나름대로의 다른 방식을 채택한다면 전세계적인 로밍(Roaming)과는 점점더 멀어지게 되므로 조기에 표준화를 진행하여 이를 방지하고자 함이다. 이러한 배경에서 출발한 IMT-2000이 이동 통신의 최종 목표라는 의미에서 현재 이동 통신에서 제공하는 모든 서비스는 물론이고 유선에서 제공되는 서비스 및 앞으로 도래할수 있는 모든 서비스를 수용할수 있어야 한다. 또한 2세대 이동 통신 시스템에서는 음성과 저속 데이터를 이동 차량과 보행자 환경에서 제공하고 있지만 실내 환경이나 음영 지역에서는 서비스가 잘되지 않고 있는데 반하여 IMT-2000에서는 실내·외 환경 및 사용자 밀집 지역에서도

서비스가 가능해야 하며, 육·해·공 모든 상황에서 서비스가 이루어 져야 한다. 결국 있을수 있는 모든 환경에서 가능한 모든 서비스를 제공 해야 할 뿐 아니라 현재 논의 되고 있는 Universal Personal Telecommunication(UPT)에 의한 개인 이동성(Personal Mobility)까지도 보장하여야 한다.

따라서 IMT-2000의 목표를 요약하자면, 전세계 어느 지역에서나 하나의 단말기로 통신 가능한 단말 이동성(Terminal Mobility)과 개인이 어느 상황에 있던 자신의 고유번호로 통신 가능한 개인 이동성(Personal Mobility)을 제공하여야 하고 가능한 모든 서비스를 모든 환경에서 제공하자는 것이다.

IMT-2000의 기술동향을 살펴보면 현재 국내에서는 개인 휴대통신 및 무선 가입자망 기술은 관련업체들이 차세대 이동 통신 분야인 IMT-2000을 개발중에 있다. IMT-2000이란 2Mbps까지의 전송 속도를 가지고 언제, 어디서, 누구에게나 음성, 데이터, 영상 등의 멀티미디어 통신을 가능케하고 국제 로밍 서비스를 제공하게 하는것이며 시스템 차원에서 볼 때 ATM 등 초고속 정보통신망을 기반으로 하는 유·무선 통합 시스템이다. 이러한 IMT-2000의 선진국 동향을 살펴보면 미국, 일본, 유럽등 선진국은 차세대 이동 통신 분야에서도 지속적으로 기술적인 우위를 확보하기 위해 업체별로 협력과 경쟁을 통하여 IMT-2000시스템의 기술 개발을 적극 추진중이며 이를 국제 표준화에 반영하기 위해 ITU 표준화 활동에 적극 참여하

고 있다. 미국은 민간 기업의 자유경쟁 개발체제로서 기존의 PCS 시스템을 보완, 발전시켜 IMT-2000으로 진화시키고자 하며 이에따라 최근 모토로라, 퀄컴, 노던텔레콤 등이 광대역 CDMA로 진화시켜 나가려고 합의했다. 일본은 IMT-2000 시스템을 구체화시키고 이에 필요한 기술적 과제 등을 검토하기 위하여 우정성이 관련 분야 전문가로 IMT-2000에 관한 조사연구회를 구성하여 운영중이다. 또한 NTT 도코모, KDD, NEC등이 독자 모델을 개발중이나 '92년부터 NTT 도코모가 연구개발을 주도해 나가고 있다. 그리고 무선 접속방식은 다양한 방식이 시도 되고 있으나 광대역 CDMA 방식이 주류를 이루고 있다. 유럽은 UMTS(Universal Personal Telecommunication System)란 이름으로 IMT-2000을 추진하며 분야별로 업체가 공동 연구를 수행하고 있다. 한편 RACE 프로젝트를 통하여 '95년 까지 무선 접속 방식, 망 구조 등에 대한 연구개발을 수행했으며 현재는 ACTS 프로젝트의 일환으로 가능 검증을 위한 Test-bed를

구현중에 있다. 또한 UMTS의 무선접속규격 연구는 에릭슨 중심의 Advanced-TDMA 접속방식을 병행 추진 하여 왔고 현재는 ETRI에서 광대역 TDMA 및 광대역 CDMA 를 연구중이다. 따라서 세계적인 추세로 볼 때 광대역 CDMA방식이 IMT-2000기술의 주류를 이룰것으로 보인다.

국제 표준화 기구인 ITU의 표준화 추진 일정을 살펴보면 '92년에 WARC-92(World Administrative Radio Commission)에서 230MHz 대역폭으로 IMT-2000용 주파수 대역이 결정되었다. 2000년대초 서비스를 목표로 '98년 ITU-T에서 IMT-2000의 서비스, 망 구조, 프로토콜 표준을, '99년 ITU-R에서 IMT-2000의 무선 접속 규격 표준을 제정할 예정이다.

본 논문에서는 IMT-2000의 무선접속에 관련된 채널 코딩에 관하여 표준화 동향을 살펴 보고 표준화 경향을 벌이고 있는 연접부호기 및 터보 코드에 대하여 검토한다.

표 1. IMT-2000을 위한 채널 코딩 기법

Nation	IIC
Europe	Convolution + Reed-Solomon
U.S.A	Conv.(K=9, R=1/2)(forward link) Conv.(K=9,R=1/2)(Return) <i>Hughes network system : Turbo(R=1/2,1/3)</i>
Japan	TDMA : Conv.(K=7, R=1/2) (RC-ARQ for data, Equal. FRS (Recursive-least-squares)-DF (Decision Feedback Equalizer)(7,2) CDMA-A : Conv.(K=7, R=1/3) + RS (data only)
Korea	IS-95 : Conv.(forward K=9, R=1/2) (Reverse K=9, R=1/3) SKT : Conv. (K=9, R=1/2) + RS FHRL : Conv. (K=9, R=1/3) + RS(17,11) KT : Conv. (K=9, R=1/2) + RS DACOM : <i>Turbo</i> SAMSUNG : <i>Turbo</i>

II. IMT-2000의 채널 코딩의 표준화 동향

IMT-2000에서는 다양한 서비스를 제공한다. 따라서 IMT-2000용 단말기는 각 서비스마다 요구되는 전송율 및 최대 BER과 최대 지연을 만족하도록 설계되어야 한다. IMT-2000에서 제공할 서비스들을 살펴보면 크게 음성 서비스와 데이터 서비스로 구별할 수 있다. 음성 서비스는 8kbps, 32kbps와 같은 저속 전송율을 사용하고 지연은 40ms이내가 되어야 하고 BER < 10⁻⁴을 요구한다. 데이터 서비스는 144kbps, 138kbps, 2048kbps와 같은 고속 전송율을 사용하고 지연은 200ms정도로 허용되며 BER < 10⁻⁶을 요구한다.

프레임 단위로 처리하는 시스템에 있어서 지연은 프레임의 크기와 직접적인 연관이 있다. 따라서 IMT-2000의 요구조건을 다시 표현하면 IMT-2000시스템은 프레임이 작은 경우에는 높은 BER을 허용하고 프레임이 큰 경우에는 낮은 BER을 요구한다.

표 1은 유럽, 미국, 한국의 IMT-2000을 위한 채널 코딩기법을 제시하고 있다. 여기서 기존의 길쌈 부호(Convolutional Code)와 1993년에 발표된 터보 코드(Turbo Code)를 채택하려는 것을 알 수 있다.

한편, 채널 부호는 크게 블록 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 블록 부호는 부호화된 부호어들이 상호 독립인 무기억 장치인 반면에 길쌈 부호는 그 출력 계열이 현재 뿐만 아니라 과거의 입력 계열에 영향을 받아 결정되기 때문이다.

또 블록 부호는 선형 부호(Linear Codes)와 순회 부호(Cyclic Codes)로 나눌 수 있는데, n, k, n-k, R=k/n 및 d_{min} 등의 매개 변수를 사용하여 n은 부호 길이로 부호기 출력 계열의 블록당 비트수를 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 ≤ R ≤ 7/8의 한계 내에 있으며 정보 전달 속도를 뜻하고 d_{min}은 부호의 오류 검출 및 정정 가능성을 알려주는 부호어 간의 최소거리(Minimum distance)를 말한다.

또 길쌈 부호에서는 n, k, m의 매개 변수가 사용되는데, 블록 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 2에서는 현재 많이 쓰이고 있는 좋은 효율의 부호들과 터보 코드의 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 연결 부호(Concatenated Codes)는 위성 통신 분야에서 각각 그 부호들의 특성을 살려 사용되고 있다.

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 1967년에 Viterbi에 의해 제안되었으며, 그동안 거의 사용되지 않던 길쌈 부호를 사용하는 계기가 되었고 이의 복호 방식으로 최우 복호(最尤復號: 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. 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 연결을 이용한 부호화 방식으로 터보 코드의 관심이 고조되고 있다. 예를 들어 터보 코드는 사용하는 인터리버의 크기를 증가시킴으로써 점점 낮은 BER을 얻을 수 있음은 잘 알려진 사실이다. 따라서 터보 코드는 다른 부분의 변형없이 인터리버의 크기만을 변화시킴으로써 앞에서 언급한 IMT-2000의 요구 조건을 만족할 수 있기 때문에 IMT-2000용으로 거론되고 있다. 터보 코드는 반복 회수를 증가시킴으로써 더 낮은 BER을 얻을 수 있다. 이것은 사용자 하여금 큰 매력력을 느끼게 하는 사항인데 서비스 제공자가 설정한 것 보다 더 좋은 품질의 통화를 원한다면 사용자는 단지 자신의 단말기의 반복 복호 회수를 증가시키면 된다. 반대로 통화 품질

표 2. 대표적 채널 부호의 특성 및 장단점 비교

부 호	특 성 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. 적자도 진행의 한계 (더이상 에러율이 향상되지 않는 임계점이 설정됨) 5. 10^{-7} 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^{-7} 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 취급할 수 있으며 블럭 부호의 경우와 유사하다.
- ③ 최우 복호법은 이론적으로 가장 능률적이나 복잡도가 부호 길이와 더불어 지수 함수적으로 증가하는 단점이 있다.
- ④ 추차 복호법은 복호에 소요되는 계산 횟수가 확률 변수라는 확률 과정 복호법으로 최우 복호법과 유사하다.

질은 좀 떨어지더라도 단말기를 오래 사용하기를 원한다면 반복 복호의 회수를 감소시키면 된다.

으로 도입되었다. 이러한 연접 부호의 특성은 비교적 복잡도가 낮은 두 부호를 연이어 사용하여 연접 에러와 랜덤 에러 모두에 강력한 정정 능력을 갖도록 하는 것이다. 연접 부호의 기본 개념도는 다음과 같다.

Ⅲ. 연접 부호기(Concatenated Code)

연접 부호는 각 에러 정정 부호들이 갖는 특성이 다르게 나타나기 때문에 다양한 에러에 대하여 적절하게 대처하기 위하여 1973년 Forney에 의해서 처음

지금까지의 연접 부호들중에 가장 좋은 성능을 보이는 것은 내부 부호로 Convolutional code/Viterbi decoder를 채용하고 외부 부호로서 Reed-Solomon Code를 사용하는 연접 시스템이다. 이것은 각각 길쌈

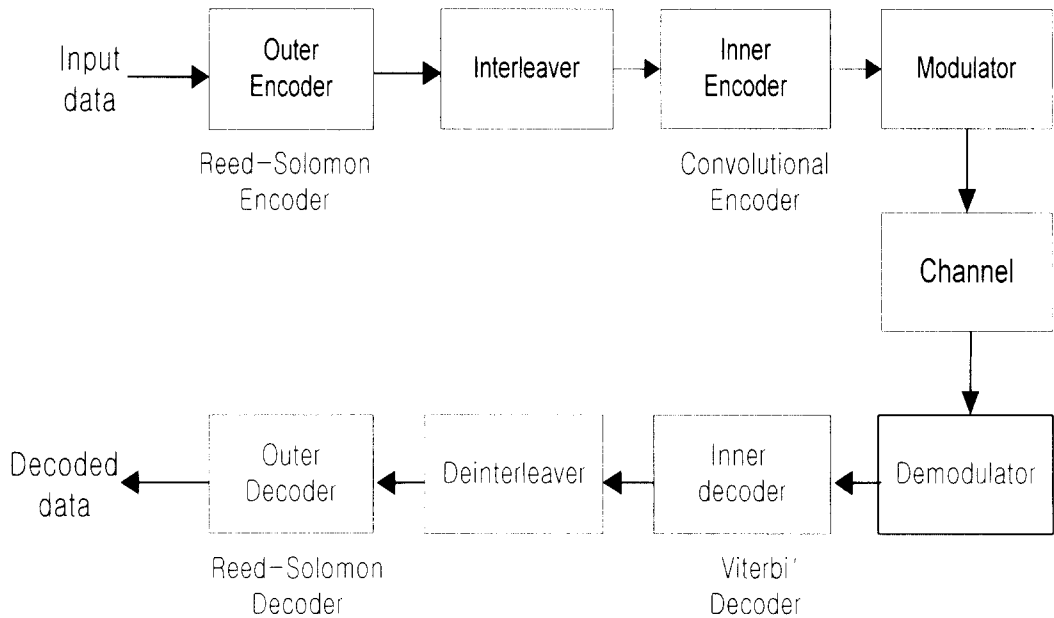


그림 3.1 연접 부호의 블록도

부호에 대한 가장 강력한 복호 방식인 Viterbi 복호 방식과 블록 부호 계열중 가장 강력한 Reed-Solomon code를 사용하기 때문이다. 이러한 연접 시스템이 사용되는 이유를 Eb/No의 관점에서 살펴보면 다음과 같다.

낮은 Eb/No에서는 길쌈 부호(Convolutional code)보다 블록 부호의 BER성능이 떨어지기 때문에 내부 부호로 길쌈 부호가 사용되는데 이를 BER의 관점에서 다시 정리하여 보면 BER이 낮은 데이터가 입력되는 경우 정정 능력이 길쌈 부호쪽에서 우수하고 상대적으로 BER이 높은 데이터가 입력되는 경우 RS부호의 출력이 현저한 성능의 향상을 보이기 때문이다.

이것은 블록 부호 계열들은 soft decision 하기 어려운 단점(complexity의 현저한 증가)을 가지고 있기 때문이며, 실제로 길쌈 부호 역시 자체적인 성능이라기 보다는 soft decision의 고유한 이득이라고 보는 것이 타당할 것이다.

또한, 채널 특성과 에러 형태, 그리고 복호기 통과 후의 데이터 열에 나타나는 에러 특성의 관점에서 고려해보면 다음과 같다. 즉, 블록 부호 중 RS 부호는 nonbinary 계열로서 심볼 단위의 에러 정정 기능을

가지고 있기 때문에 연집에러(burst error)에 강한 특성을 나타낸다. 그리고 주로 내부 부호로 사용되는 길쌈 부호의 경우, 복호 후의 출력이 연집 에러를 유발하기 때문에 다음 단계인 외부부호에서는 블록 복호화하여 이에 적용할 수 있는 RS 부호를 사용하는 것이 바람직하다.

3.1 Viterbi 복호기

(n, k) 길쌈 부호기와 연관정 Viterbi 복호기는 우수한 오류 정정 능력 때문에 위성 통신과 디지털 셀룰라 시스템, CEPT GSM 시스템, CDMA 시스템 등에 널리 사용되고 있다.

이 부호의 특징은 AWGN(Additive White Gaussian Noise) 채널에서 오류 정정 능력이 특히 우수하며 연관정 복호를 채택할 경우, 경관정보보다 약 2dB의 부호 이득(coding gain)을 추가로 얻을 수 있는 장점을 가지고 있다. 또 잡음이 연집 특성을 갖는 채널에서도 인터리버/디인터리버(Interleaver/ Deinterleaver)를 이용하여 연집 에러(burst error)에도 적용이 가능하다.

길쌈부호의 복호화 방법으로는 축차 복호화

(Sequential decoding), 임계 복호화(Threshold decoding), Maximum Likelihood 방식의 Viterbi 복호화 등이 있는데, 여러 종류 중 연판정 복호가 용이하여 높은 부호 이득을 얻을 수 있으나, 하드웨어 복잡성 때문에 사용되지 않다가 최근에 VLSI 기술의 발달에 의해 Maximum Likelihood 방식의 Viterbi 복호화 방식이 많이 사용되고 있다. Maximum Likelihood 방식의 Viterbi 복호화 알고리즘은 수신된 부호 심볼을 격자도(trellis) 상에서 가능한 여러 경로에 대해 확률을 계산하여 가장 큰 확률을 갖는 경로를 선택하는 방법이다. 즉, 각 가지(Branch)에서 계산되는 d_{free} (Euclidean free Distance)에 의해서 경로에 할당되는 값을 계산하는 방법이다.

Maximum Likelihood 방법은 격자도에서 하나의 상태(state)에서 나가는 가지 수는 $2k$ 개가 되고, 또한 하나의 상태로 들어오는 가지 수 역시 $2k$ 개가 되는데 수신된 부호 심볼과 비교하여 각 가지 수의 확률을 계산한 후, 그 중 가장 큰 확률을 갖는 가지가 남게 되고 나머지는 제거된다.

3.1.1 부호화 및 복호화

가. 부호화(Encoding)

(n, k) 길쌈 부호기는 m 개의 메모리와 k 개의 입력, n 개의 출력을 갖는 선형 시퀀셜 회로(linear sequential circuit)로 정의된다. 실제의 경우 $n=2, k=1$ 즉, 부호율 $1/2$ 이 주로 사용된다. 그림 3.2는 부호율 $1/2$, 구속장 $(K) = 7$ 인 길쌈 부호기이다. 높은 전송율이 요구되는 무선 데이터 이동통신 시스템이 길쌈 부호를 사용할 때 높은 부호율을 갖는 길쌈 부호가 필요하다. 단순히 K 를 증가시키면, 하나의 상태에서 다음

상태로 천이할 때 2개의 경로가 존재하게 되어 복호기의 복잡성은 지수적으로 증가하게 된다. 이러한 문제를 피하기 위해서 (2,1) 길쌈 부호기의 출력 중에서 특정한 위치의 출력값을 전송하지 않음으로써 부호율이 $(n-1)/n$ 인 길쌈 부호를 얻을 수 있는데, 이를 punctured 길쌈 부호라 한다. 하지만, puncturing을 함으로써 부호율과 함께 정보 데이터 전송율은 증가하지만 부호 이득이 감소하고 전송 전력을 증가시켜야 하는 단점도 발생한다.

채널에서 에러정정부호를 사용하는 경우에 에러정정부호의 성능을 평가하는 척도로 부호이득(coding gain)이 있다. 즉, 에러정정부호를 사용했을 경우와 사용하지 않았을 경우 같은 $P(E)$ performance를 달성시키기 위하여 요구되는 E_b/N_0 사이의 차이를 말한다. 채널을 통하여 수신된 신호가 복조기에서 복조되었을 때 복조된 데이터를 양자화하는 방법에는 2-레벨로 판정하는 경판정(hard decision)과 Q-레벨로 판정하는 연판정(soft decision)의 방법이 있다. 일반적으로 부호이득을 증가 시키기위해 수신된 데이터를 경판정(hard decision)하는 대신에 A/D변환기를 이용하여 연판정(soft decision)하여 데이터를 복호하면 부호이득을 추가로 약 2dB정도 향상시킬 수 있다.

나. 복호화(Decoding)

Viterbi 복호알고리즘은 입력값에 대한 가장 유사한 복호 시퀀스를 찾는 것이다. 확률값의 누적적인 상태값을 발생 가능한 모든 경로에 대해 계산하고 두 개의 경로가 만날 때 큰 경로 값을 갖는 경로를 선택하여 가장 유사한 시퀀스를 통해 traceback과정을 통해 출력값을 선택하는 알고리즘이다. 이 과정의 주요

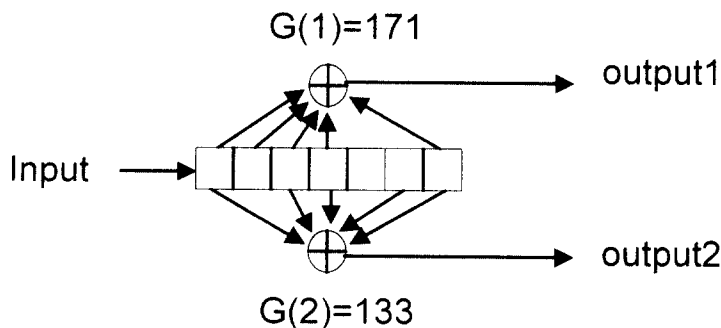


그림 3.2 길쌈 부호기

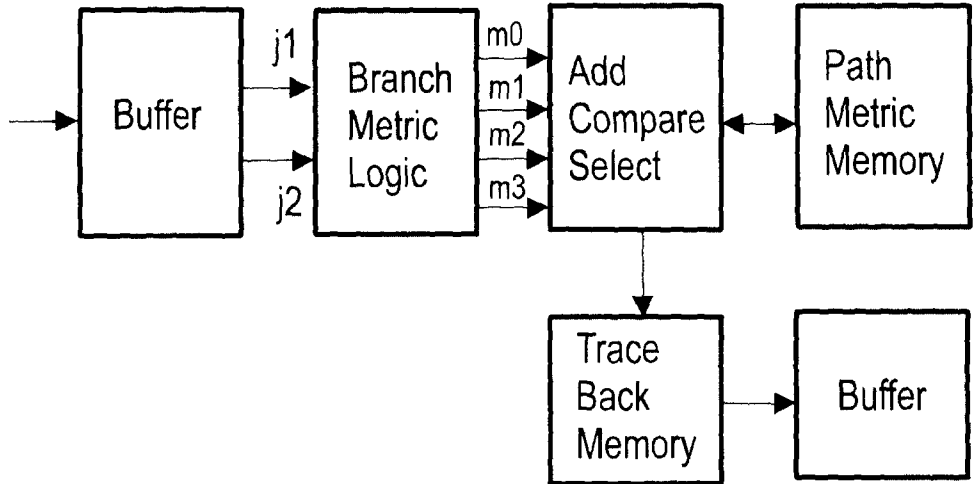


그림 3.3 Viterbi 복호기 블록도

단계는 BML (Branch metric logic), ACS (Add-compare-select), PMM(Path metric memory), 그리고 traceback이다. 여기서 branch metric이란 수신되는 부호어 값과 각 상태 천이에 따른 부호어 값의 거리이고 path metric은 현 상태에 이르는 전 경로의 누적치이다.

그림 3.3은 Viterbi 복호기의 블록도를 보여주고 있다. 그러나 길쌈 부호화 방식이긴 하지만 복호의 관점에서 보면 부호화 격자도를 복호기에서 재사용하여 경로를 추정하므로 부호화의 의도와는 다르게 비트들의 상호 연관성 이용 문제의 관점에서 convolution의 개념이 다소 떨어지는 것을 알 수 있다.

3.2 Reed-Solomon 복호기

RS 부호는 CD, DAT, 전문적인 비디오 레코더(recoder)와 같은 디지털 오디오/비디오(audio/vidio) 장비에 응용되고 있으며, 그 이유는 연립오류정정(burst error correcting)을 위한 능력이 우수하며 쉽게 구현할 수 있기 때문이다. 특히 VCR(vidio cassette recoder)이나 TV 같은 비디오 장비에 있어서 디지털 화가 시작되고 있으며 SDTV와 HDTV의 압축 비디오 정보를 위해 높은 속도의 RS 복호기가 요구되고

있다. 또 디지털 VCR의 경우, 고화질과 신뢰성을 특징으로 original sampling frequency, quantization step, 압축 기법 등이 화질을 좌우하며 신뢰성은 오류 정정 능력과 정보 재생시의 에러율에 의존한다.

3.1.2 부호화와 복호화(Encoding & Decoding)

가. Reed-Solomon 부호기

그림 3.4는 Reed-Solomon 부호기의 간단한 개념도와 내부구조를 나타낸다. 이들은 부호기로서 일반적인 cyclic 부호기를 채택하고 있으며 이것은 심볼 단위의 부호화를 가능하게 하며 수신단에서 이의 역동작을 수행하여 에러가 발생했을 경우에는 신드롬이 발생하게 된다.

즉, 채널을 통과하여 수신단에서 받은 수신다항식에 생성다항식의 근을 대입하여 앞의 조건들을 만족하면 에러가 없는 상태이며 그렇지 않을 경우에는 구해진 신드롬을 이용하여 에러를 정정한다.

나. Reed-Solomon 복호기 구조

그림 3.5와 3.6은 하드웨어 구현을 위한 Reed-

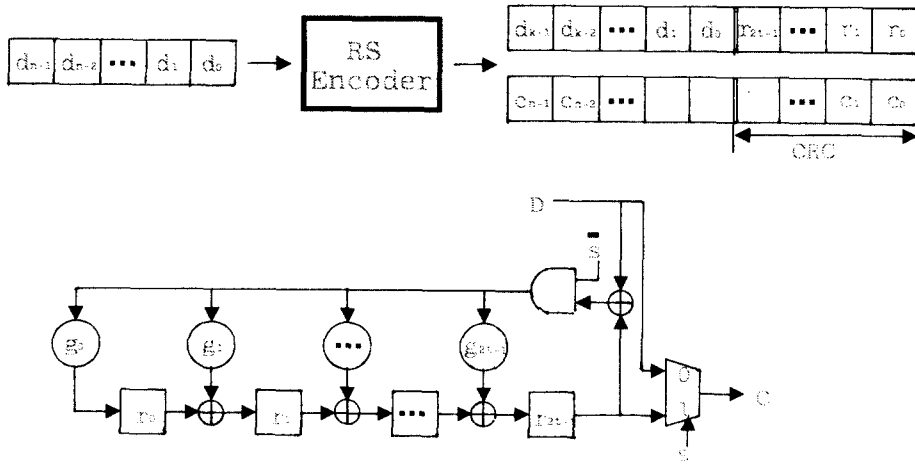


그림 3.4 Reed-Solomon 부호기

Solomon 부호기의 구조이다. 여기에서 그림 3.5는 유통성을 허용하지 않지만 좀더 빠른 연산과 구조적인 동작을 위하여 Systolic과 병렬 처리 (Parallel processing)를 채택한 경우이고, 그림 3.6은 4단계 파이프라인 처리 과정을 보이는데, 이것은 그림 3.5의 3단계 처리 구조 중 두번째 단계를 분할하여 각각 다

항식 생성 단계와 다항식 추정 단계로 나누어 계산하는 과정이다. 4단계 처리의 필요성은 유통성을 고려할 경우에 생기며, 이때는 Galois Field를 계산하는데 많은 단계가 요구된다. 따라서 유통성을 가지는 Micro-programmed ROM을 이용하여 구현되며, 한 단계에서 4개의 Instruction Set을 수행하므로

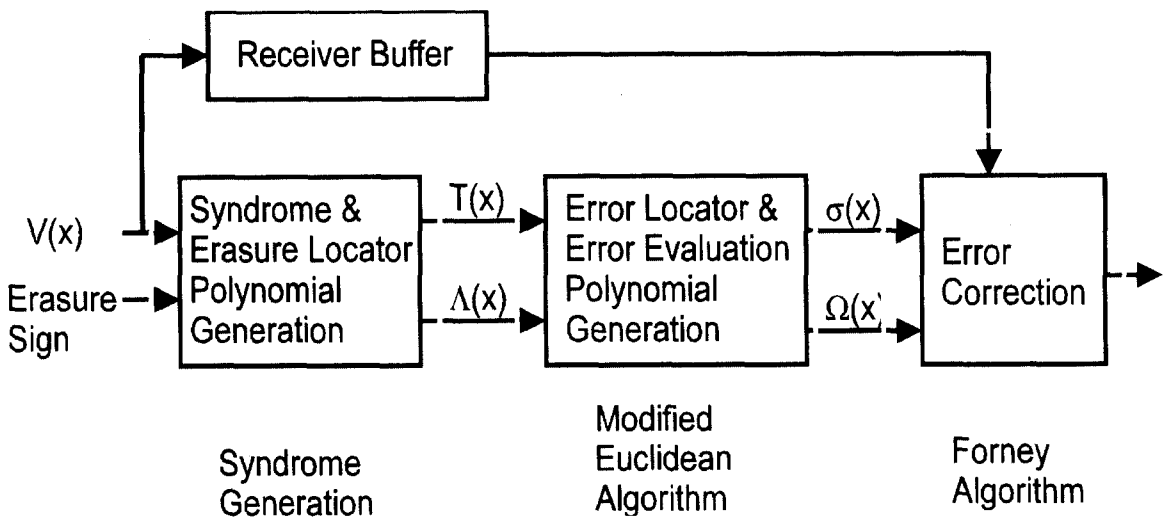


그림 3.5 Reed-Solomon 부호기 3-stage 구조

Superscalar 방식을 적용하여 오중 계산 단계의 최대 소요 단계 수와 일치하도록 한다. 결국 Reed-Solomon 복호기는 3단계 또는 4단계 파이프라인 처리 방식을 적용할 수 있다.

차세대 이동통신 시스템에 RS 부호를 적용할 경우에 가장 문제가 되는 것은 복잡성의 문제와 데이터 전송율이다. Viterbi 복호기만을 사용할 경우에도 tail 비트 때문에 발생하는 정보 전송율의 저하를 가져오며, 일반적으로 8 비트의 tail 비트를 사용할 경우에 2.5% 정도의 데이터 증가를 가져오게 되며 채널 부호화와 심볼 반복(전송율을 일정하게 유지시키기 위해), MUX와 변조등을 거치고 나면 8 비트의 추가가 수백 비트로 늘어나게 된다. 따라서 이를 위해 Puncturing 기법을 통해 데이터량을 줄이는 기법 또한 연구되고 있으며, 이를 토대로 다시 시스템의 구성을 생각해 보면 두 구조 모두 ATM 망에서 쓰이는 시스템과는 달리 이레이저(Erasur:Cell Loss에 의해 발생하며 실제 전송되지 않는 데이터)에 대한 능력을 충분히 발휘할 수 없는 단점이 발생한다. 실제로 이레이저에 대한 복호 능력이 에러에 대한 정정 능력보다 2배가 되므로 이를 적절히 이용해야만 하는 데, 한가지 방법은 여기에서 다시 puncturing의 장점을 최대한 살려서 임의의 이레이저를 발생시킴으로 부가

정보가 아닌 정보 데이터의 실제 전송율을 높일 수 있을 것이다.

IV. 터보 코드(Turbo Code)의 성능 분석

무선 통신 시스템에서는 일반적으로 짧은 프레임(일반적으로 300 비트 이하)에서 정보를 전송하게 되는데 가령, GSM(Global System for Mobile communications)과 DCS 1800(Digital Cellular Systems)의 경우 음성 프레임은 189비트이며 최근 소개된 JD-CDMA(Joint Detection Code Division Multiple Access)의 경우 uplink에서 192비트의 프레임 길이를 사용한다. 이것은 오류 정정 부호 선택에 제한을 주는 요소가 된다. 일반적으로 무선 통신 시스템에서 흔히 사용되는 부호는 길쌈 부호(Convolutional Code)로서 이러한 부호의 장점은 구속장(K)의 길이보다 정보 프레임의 길이가 충분히 크다고 한다면 그들의 성능은 프레임 길이와는 상관없이 독립적이라는 것이다.

최근 Recursive Systematic Convolutional codes로 부터 나온 Parallel Concatenated Convolutional Codes(PCCC or Turbo Codes)가 제안되었는데 이러한 계

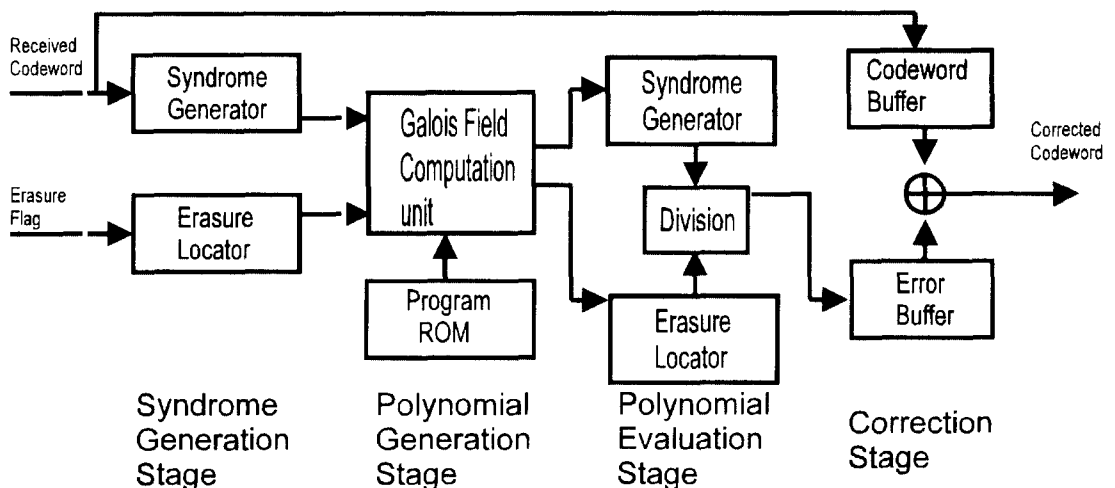


그림 3.6 Reed-Solomon 복호기의 4-stage 구조

열의 부호는 긴 블록 크기를 위한 채널 용량에 근접하는 성능을 지닌 것으로 보여지고 있으며 비교적 간단한 component codes와 큰 인터리버를 가지고 Shannon Limit에 근접하는 오류 정정 능력(1/2 turbo codes에 대해서 약 0.7dB에서 BER = 10⁻⁵)을 가지는 것으로 알려져 있다. 사실, 이러한 부호의 성능은 인코딩된 프레임 크기가 증가할수록 지수적으로 향상되는데 이것은 PCCCs에서 얻어지는 "인터리버 이득(interleaving gain)"이 프레임의 크기에 따라 지수적으로 증가하기 때문이다. 그림 4.1은 터보 코드의 부호화 과정을 나타낸 것이다.

터보 코드의 복호 과정은 MAP(Maximum a Posteriori) 혹은 SOVA(Soft-output Viterbi Algorithm) 알고리즘을 이용한 반복 복호 방법을 사용하는데 이것은 convolutional code의 복호 방법인 ML(Maximum Likelihood) 복호 방법 보다 매우 복잡한 것으로 나타나고 있지만 상대적으로 우수한 성

능을 보이는 것으로 알려져 있다.

4.1. MAP 알고리즘

MAP 알고리즘은 잡음이 섞인 수신신호로부터 APP(a posteriori probability)를 계산하는 알고리즘이며 APP값을 산출하는 원리는 다음과 같다. MAP 디코더의 입력이 다음과 같이 BPSK 변조된 수신신호 $R_{1,N} = (R_1, R_2, \dots, R_N)$ 라 하자.

여기에서 $R_k = (a_k, b_k)$ 이며 $a_k = (2d_k - 1) + n_k$, $b_k = (2p_k - 1) + m_k$ 이고 d_k, p_k 는 인코더 출력이며 n_k, m_k 는 variance가 σ^2 인 가우시안 잡음이다. MAP 알고리즘의 목적은 주어진 $R_{1,N}$ 에서, most likely한 d_k 를 발견하는 것인데 이것은 정의되는 log-likelihood ratio(LLR) A_k 값에 대응하는 각각의 d_k 를 계산함으로써 얻어질 수 있다.

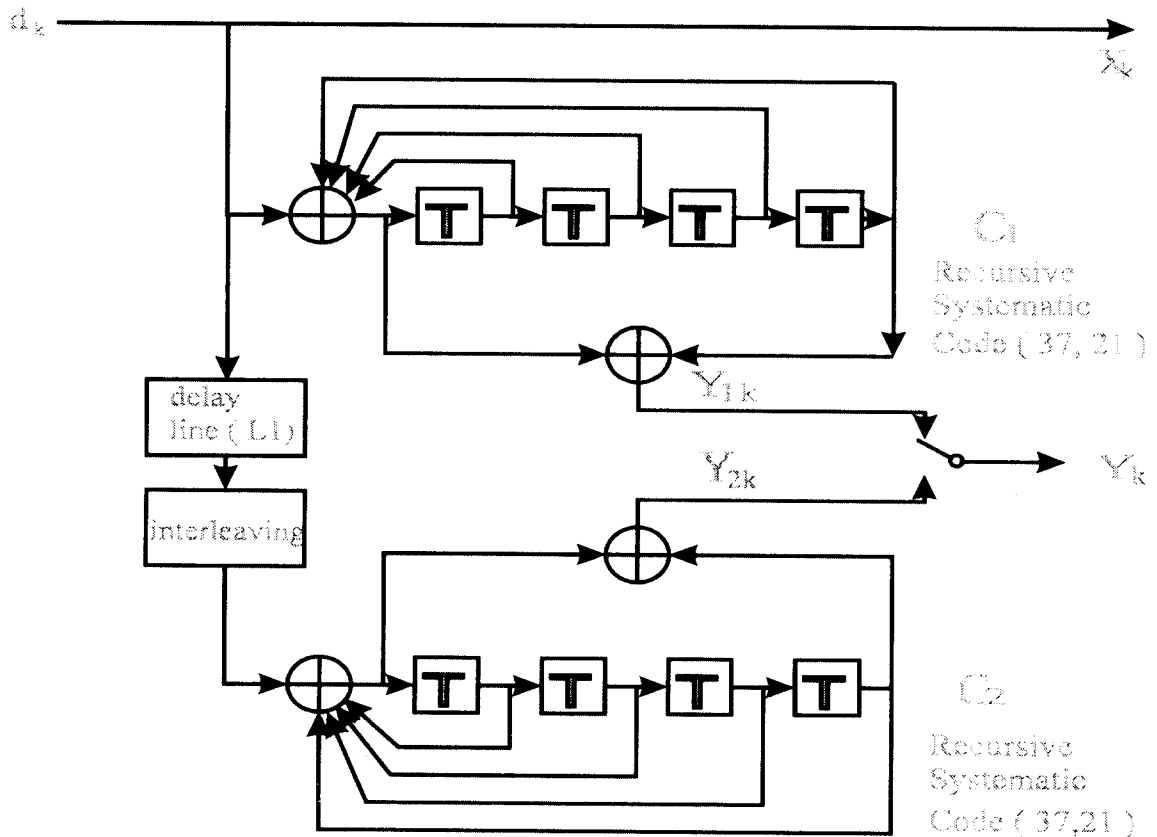


그림 4.1 병렬 연결 재귀 시스템 부호

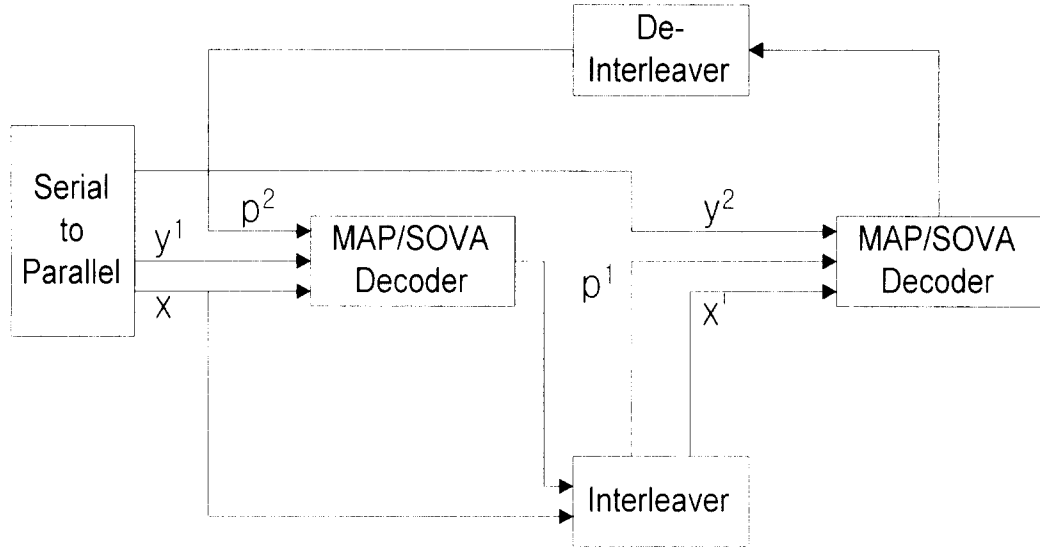


그림 4.2 터보 코드의 피드백 복호기

이러한 A_k 를 계산하기 위하여 MAP 알고리즘에서는 다음과 같은 3개의 parameter를 정의하고 있으며 이들은 다음과 같다.

$$\Lambda_k(d_k) = \log \left(\frac{P_r(d_k = 1 | R_{1,N})}{P_r(d_k = 0 | R_{1,N})} \right)$$

만약 S_k 가 시간 k 에서의 인코더 상태라고 한다면, 다음과 같은 식을 얻을수 있다.

$$P_r(d_k = 1 | R_{1,N}) = \sum_{m=0}^{2^i-1} P_r(d_k = 1, S_k = m | R_{1,N})$$

또한, $\lambda_{k,i}(m) = P_k(d_k = i, S_k = m | R_{1,N})$ 라 하고 이를 인코더의 모든 가능한 상태들에 대한 합으로 나타낸다고 하면 A_k 는 다음과 같다.

$$\Lambda_k = \log \left(\frac{\sum_{m=0}^{2^i-1} \lambda_{k,1}(m)}{\sum_{m=0}^{2^i-1} \lambda_{k,0}(m)} \right)$$

$$\alpha_{k,i}(m) = P_r(d_k = i, S_k = m, R_{1,N})$$

$$\beta_{k,i}(m) = P_r(R_{k+1,N} | d_k = i, S_k = m)$$

$$\delta_i(R_k, m) = \exp \left(\frac{2}{\sigma^2} (a_k i + b_k, p_{k,i}(m)) \right)$$

where $i, p_{k,i}$ correspond to encoder outputs.

위의 $\alpha_{k,i}(m), \beta_{k,i}(m), \delta(R_k, m)$ 를 살펴보면 다음과 같다.

첫째, $\delta(R_k, m)$ 을 위한 표현

$\delta(R_k, m)$ 는 branch metric이라 하며 전송 채널 (transmission channel)에 의존하게 된다. trellis상에서 $\delta(R_k, m)$ 는 초기 인코더의 상태 m 과 정보 비트 i 를 가지고 시간 k 에서 $k + 1$ 로 천이하는 branch에 대

응하는 branch metric값이다.

다음의 그림에서 0 0, 1 1인 branch를 $\delta(R_k, m)$ 를 가지고 표현하면 각각 $\delta(R_2, 0 1)$ 와 $\delta(R_3, 0 0)$ 이다. 이러한 branch들은 모두 자신의 $\delta(R_k, m)$ 를 가지며 이것은 위 식 $\delta_i(R_k, m) = \exp\left(\frac{2}{\sigma^2}(a_k i + b_k, p_{k,i}(m))\right)$

으로부터 계산할 수 있다.

만약 아래그림과 같은 trellis를 가진다면 우리는 그림 4.3과 같이 계산할 수 있다.

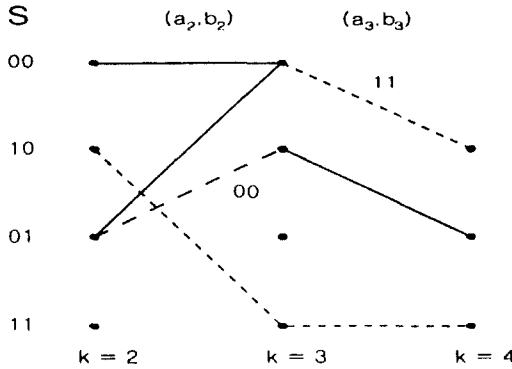


그림 4.3 $\delta_i(R_k, m)$ 의 격자도

$$\delta_0(R_2, 0 1) = (a_2 \times 0 + b_2 \times 0) = 0,$$

$$\delta_1(R_3, 0 0) = (a_3 \times 1 + b_3 \times 1) = a_3 + b_3$$

각각의 δ 에 대해서는 두 개의 같은 값이 존재하게 되는데 이로 인하여 8개의 branch metrics에 대하여 우리는 4개의 state code를 가지고 4개의 서로 다른 값 (d_k, p_k)를 가질 수 있다.

둘째, $a_{k,i}(m)$ 를 위한 표현 :

$a_{k,i}(m)$ 는 forward state metric이라하며 이것은 i 의 상태 천이 비트를 가지고 시간 k 에서 상태 m 으로부터 다음상태로의 천이를 위한 state metric을 나타낸다. $a_{k,i}(m)$ 를 $\delta_i(R_k, m)$ 로 나타내면 다음과 같다.

$$a_{k,i}(m) = \delta_i(R_k, m) \sum_{j=0}^1 a_{k-1,j}(S_{b,j}(m))$$

여기에서 $\delta_i(R_k, m)$ 는 branch metric이며 $S_{b,j}(m)$ 는 이전 상태를 나타낸다.

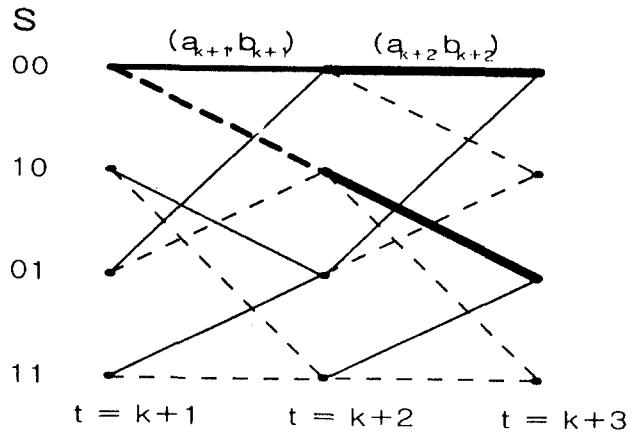


그림 4.4 $S_{b,j}(m)$ 의 격자도

위그림에서 시간 $k+2$ 의 상태 (0 0)에서 $k+3$ 의 상태 (0 0)으로 가는 branch와 $k+2$ 의 상태 (1 0)에서 $k+3$ 의 상태 (0 1)으로 가는 branch에 대응하는 $a_{k,i}(m)$ 값들을 구하면 다음과 같다.

$$a_{k+2,0}(0 0) = [a_{k+1,0}(S_{b,0}(0 0)) + a_{k+1,1}(S_{b,1}(0 0))] \delta_0(R_{k+2}, 0 0)$$

$$a_{k+2,0}(1 0) = [a_{k+1,0}(S_{b,0}(1 0)) + a_{k+1,1}(S_{b,1}(1 0))] \delta_1(R_{k+2}, 1 0)$$

이러한 계산은 초기화가 올바르게 이루어 진다면 모든 a 값에 대하여 타당하다. 일반적으로 초기화값은 다음과 같다.

$$a_{0,0}(0 0) = \delta_0(0 0) \text{ and } a_{0,1}(0 0) = \delta_1(0 0)$$

$$a_{0,0}(m) = a_{0,1}(m) \text{ for all } m \neq (0 0).$$

셋째, $\beta_{k,i}(m)$ 를 위한 표현 :

$\beta_{k,i}(m)$ 는 $a_{k,i}(m)$ 와 유사하며 backward state metrics으로 나타낸다. 또한, $\delta_i(R_k, m)$ 를 이용한 계산식은 다음과 같다.

$$\beta_{k,i}(m) = \sum_{j=0}^1 \beta_{k+1,j}(S_{f,j}(m)) \delta_j(R_{k+1}, S_{f,j}(m))$$

여기에서 $\delta_j(R_k, m)$ 는 branch metrics 이며 $S_{f,j}(m)$ 는 이후 상태를 나타낸다.

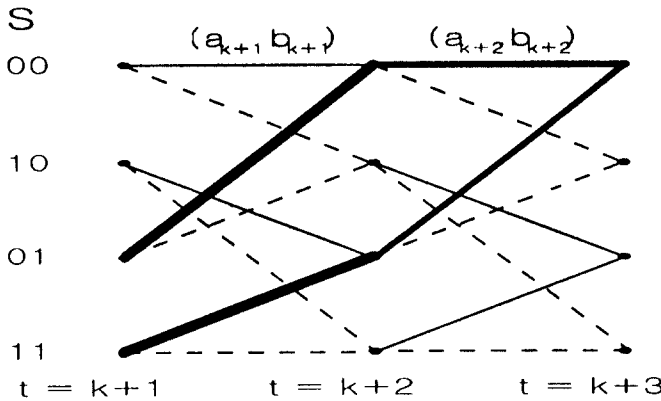


그림 45 $\beta_{k,i}(m)$ 의 격자도

위 그림 45 에서 시간 $k + 1$ 의 상태 (0 1)에서 $k + 2$ 의 상태 (0 0)으로 가는 branch와 $k + 1$ 의 상태 (1 1)에서 $k + 3$ 의 상태 (0 1)으로 가는 branch에 대응하는 $\beta_{k,i}(m)$ 값들을 구하면 다음과 같다.

$$\beta_{k+1,1}(0\ 1) = |\beta_{k+2,0}(S_{f,1}(0\ 1)) \delta_0(R_{k+2}, S_{f,1}(0\ 1)) + \beta_{k+2,1}(S_{f,1}(0\ 1)) \delta_1(R_{k+2}, S_{f,1}(0\ 1))|$$

$$\beta_{k+1,0}(1\ 1) = |\beta_{k+2,0}(S_{f,0}(1\ 1)) \delta_0(R_{k+2}, S_{f,0}(1\ 1)) + \beta_{k+2,1}(S_{f,0}(1\ 1)) \delta_1(R_{k+2}, S_{f,0}(1\ 1))|$$

그리고 이들값의 초기값은 다음과 같다.

$$\beta_{N+r-1,0}(S_{b,0}(0\ 0)) = \beta_{N+r-1,1}(S_{s,1}(0\ 0)) = 1$$

$$\beta_{N+r-1,0}(m) = \beta_{N+r-1,1}(m) \text{ for all } m \neq (0\ 0)$$

따라서 앞에서 설명한 $\alpha_{k,i}(m)$, $\beta_{k,i}(m)$, $\delta_i(R_k, m)$ 의 parameter에 의해서 A_k 는 다음과 같이 변형되며 변형된 계산식으로 부터 복호된 비트에 대한 APP값을 계산할 수 있게 된다.

$$A_k = \log \left(\frac{\sum_{m=0}^1 \alpha_{k,1}(m) \beta_{k,1}(m)}{\sum_{m=0}^1 \alpha_{k,0}(m) \beta_{k,0}(m)} \right)$$

MAP 알고리즘에서 APP값을 계산하는 전체 과정을 요약하면 다음과 같다.

1) 시간 $k = 1$ 에서 시작하여 수신된 모든 sequence에 대해서 $\delta_i(R_k, m)$ 를 계산하고 이것을 2^N 크기의 배열에 저장한다. (여기에서 2^N 은 모든 가능한 coded symbol이다.)

2) $i = 0,1$ 에 대해서 $\beta'_N(S'_b(b)) = 1$ 로, 그 외의 모든 state를 i 에 대해서 $\beta'_N(m) = 0$ 으로 초기화한다. 시간 $k = N - 1$ 에서 시작하여 반복적으로 $\beta'_N(m)$ 값을 다음식을 이용하여 계산하고 2^N 크기의 배열에 저장한다.

$$\beta_{k,i}(m) = \sum_{j=0}^1 \beta_{k+1,j}(S_{f,j}(m)) \delta_j(R_{k+1}, S_{f,j}(m))$$

3) $\alpha'_1(0) = \delta_i(R_1, 0)$ 그리고 0이 아닌 m 에 대하여 $\alpha'_1(m) = 0$ 값으로 초기화 한다. 시간 $k = 2$ 부터 시작하여 반복적으로 다음식을 사용하여 α'_k 를 계산한다.

$$\alpha_{k,i}(m) = \delta_i(R_k, m) \sum_{j=0}^1 \alpha_{k-1,j}(S_{b,j}(m))$$

그리고 각각의 시간 k 에서 $L(d_k)$ 를 다음식을 이용하여 계산한다.

$$L(d_k) = \log \left(\frac{\sum \alpha'_k(m) \beta'_k(m)}{\sum \alpha''_k(m) \beta''_k(m)} \right)$$

4) $\beta'_N(m)$ 와 α'_k 값은 다음과 같은 방식으로 적당하게 normalizing한다.

$$\alpha_{mid} = \exp((\log \alpha_{min} + \log \alpha_{max})/2)$$

4.2 SOVA 알고리즘

SOVA 알고리즘은 a finite state Markov-chain상에서 가장 그럴듯한 path sequence를 찾아내고 이 경로를 따르는 부호화된 정보 비트들에 대해서 reliability value를 전달하는 알고리즘이다. 예를 들어 state s에서 competing paths의 확률이 다음과 같다고 하자.

$$\text{Prob}\{\text{path } m, \text{ state } s\} \approx e^{-\Gamma(m,s)}$$

where, $\Gamma(m,s)$ 는 state s의 끝단에서 m번째 path의 metric의 합이다.

이 때 Rate 1/2 codes에서 $\Gamma(1,s)$ 와 $\Gamma(2,s)$ 에서 $\Gamma(1,s)$ 를 survivor path라 한다면 time k, state s에서 survivor path를 잘못 선택할 확률은 다음과 같다.

$$p_{s,k} = \frac{e^{-\Gamma(2,s)}}{e^{-\Gamma(1,s)} + e^{-\Gamma(2,s)}} = \frac{1}{1 + e^{\Delta s}} \leq \frac{1}{2}$$

where, $\Delta s = \Gamma(2,s) - \Gamma(1,s) \geq 0$

또한 state s의 끝단에서 서로 다른 symbol들에 의해 명명된 competing path의 모든 information symbol들에 대해서 $p_{s,k}$ 의 예러 확률을 만들어 내게 되며 만약 k-j에서 주어진 symbol들이 다르다면 그 symbol이 예러가 될 확률값을 \hat{L}_{k-} 로 update할 수 있다. 이러한 update 과정은 state s와 관계된 모든 survivor path에 대해서 필요하며 \hat{L}_{k-} 값은 다음식으로 얻어질 수 있다.

$$\hat{h}_{s,k-j} := f(\hat{h}_{s,k-j}, \Delta s) := \frac{1}{\alpha} \log \left[\frac{1 + e^{(\alpha \hat{h}_{s,k-j} + \Delta s)}}{e^{\Delta s} + e^{\alpha \hat{h}_{s,k-j}}} \right]$$

SOVA 알고리즘에 관하여 간단히 알아보았는데 MAP 알고리즘이 모든 path에 대한 확률값들을 계산하여 정보 비트에 대한 soft-output값을 출력하는 반면 SOVA 알고리즘은 모든 path에 대해서 soft-output을 만들어내는 것이 아니고 Maximum Likelihood path(혹은 Survivor path)와 ML path에 가장 근접하는 competitor path와의 관계를 통하여 soft-output을 산출해 낸다 이러한 SOVA 알고리즘은 MAP 알고리즘에 비하여 BER = 10⁻⁴, Memory = 4에 대해서 하드웨어 복잡성은 상대적으로 낮으나 0.7dB의 손실이 따른다고 알려지고 있다.

4.3 터보 코드(Turbo Codes)의 성능

지금까지 논의한 내용을 토대로하여 IMT-2000을 위한 Turbo Codes의 성능을 분석하였다. 이를 위하여 구속장 K = 3, 생성다항식 G₁ = 7, G₂ = 5, 그리고 병렬 연결 구조를 이용한 부호율 R=1/2의 부호기를 사용하여 P_e 성능을 평가하였으며 인터리버는 블록 인터리버를 사용하였다. 또한 채널 환경은 AWGN을 가정하여 실험하였으며 복호 방법은 MAP 알고리즘을 사용하였다.

그림 4.6은 인터리버의 크기를 100으로 하여 반복 회수에 따르는 결과를 보여주고 있다. 실험 결과 반복 회수가 증가할수록 성능이 우수한 것을 알 수 있다.

그림 4.7은 인터리버의 크기를 400으로하여 역시 반복 회수에 따르는 성능을 나타내고 있다. 인터리버 크기가 100인 그림4.6과 비교하여 보면 반복 회수가 3일 때 BER = 10⁻³에서 약 1.0dB의 부호 이득을 가져오는 것을 알 수 있으며 이를 통하여 인터리버 크기의 증가에 따라 성능의 향상을 가져오는 것을 확인하였다. 또한 그림 4.7에서 Eb/No = 3에서 2번의 반복만으로 BER = 10⁻⁴을 얻고 있음을 알 수 있다.

그림 4.8은 인터리버의 크기를 400으로 하면서 K = 5이며 생성다항식 G₁ = 37, G₂ = 23인 Turbo Code의 성능을 보여주고 있다. 인터리버 크기가 100인 그림 4.6과 비교하여 보면 반복 회수가 3일 때 BER = 10⁻³에서 약 1.6dB의 부호 이득을 가져오는 것을 알 수 있으며 또한 같은 인터리버 크기에 대해서 구속장이 3인 그림 4.7과 비교해서 반복 회수가 3이고 BER = 10⁻⁴에서 약 1.0dB의 부호 이득을 얻고 있음을 알 수 있다. 그리고 반복 회수가 3을 넘어서게 되면 Eb/No = 3에서 거의 10⁻⁶의 BER를 얻고 있는 것을 알 수 있다.

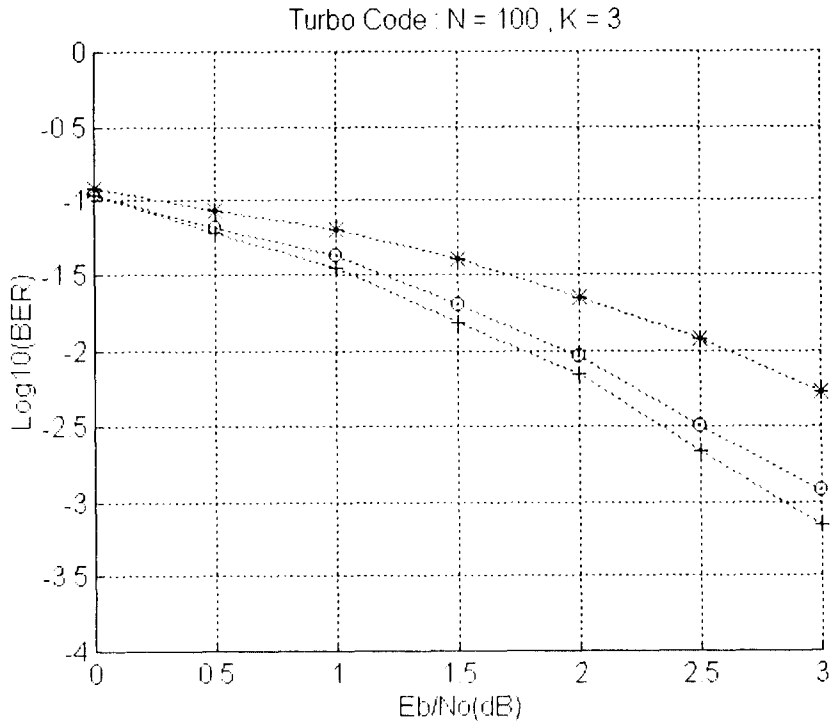


그림 4.6 K = 3 , interleaver_size = 100 , iteration = 1(*), 2(o) , 3(+)

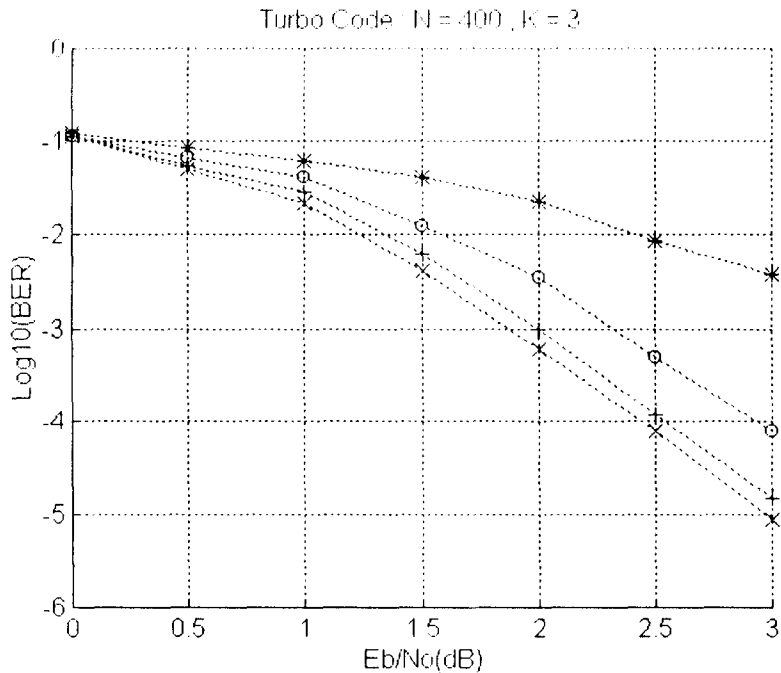


그림 4.7 K = 3 , interleaver_size = 400 , iteration = 1(*), 2(o) , 3(+), 4(x)

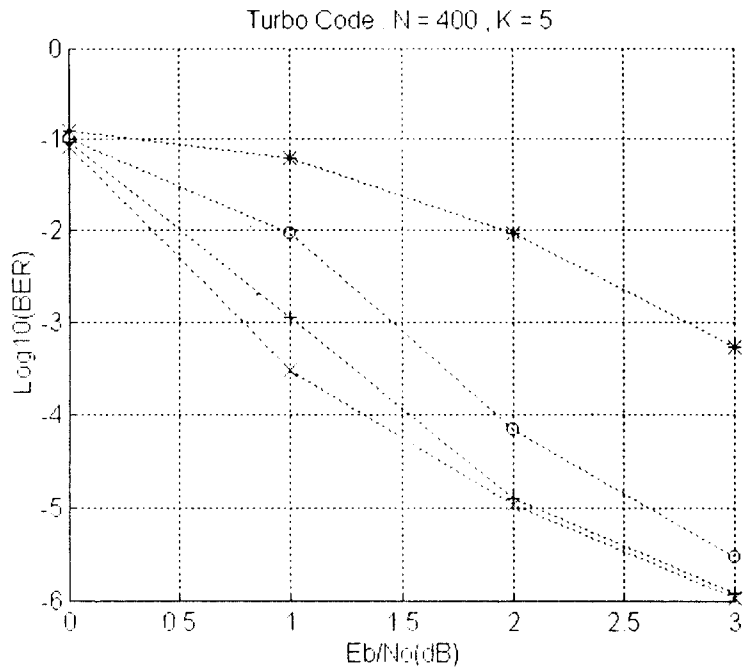


그림 4.8 K = 5 , interleaver_size = 400 , iteration = 1(*), 2(o), 3(+), 4(x)

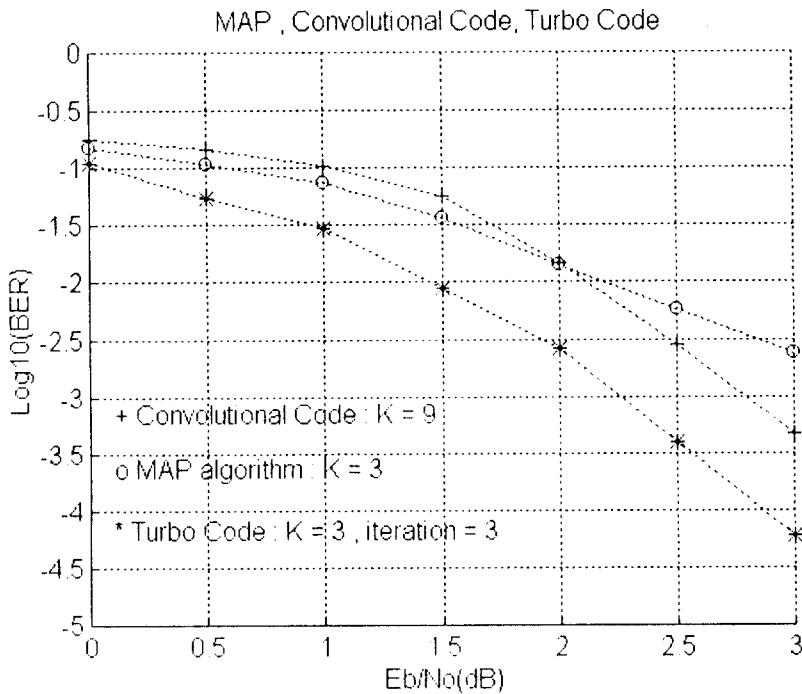


그림 4.9 Convolutional Code, MAP algorithm, Turbo Codes의 비교

그림 4.9는 현재 이동 통신 시스템에서 사용되고 있는 $K = 9$ 인 길쌈 부호(Convolutional code), 구속장 $K = 3$ 인 단일 MAP 알고리즘 그리고 구속장 $K = 3$ 이며 반복 회수가 3인 Turbo Codes의 성능을 비교한 결과이다. 실험 결과 터보 코드가 길쌈 부호에 비하여 $BER = 10^{-3}$ 에서 약 0.8dB정도의 부호 이득을 얻고 있으며 상당히 우수한 성능을 보이고 있음을 알 수 있다. 또한 단일 MAP 알고리즘 역시 하드웨어 복잡성의 관점에서 볼 때 좋은 성능을 보이고 있다.

V. 결 론

IMT-2000에서는 음성뿐만 아니라 데이터, 영상까지도 고품질의 다양한 서비스를 제공해야 한다. 그러나 고품질의 서비스를 제공하기 위해서는 이동통신 채널에서 필연적으로 나타나는 에러를 제어하는 채널 코딩에 대한 연구가 무엇보다 중요하다고 본다. 따라서 본 논문에서는IMT-2000의 무선접속에 관련된 채널코딩에 관하여 표준화 동향을 살펴보고 연접 부호기 및 터보 코드에 대해 검토 하였다. 현재, IMT-2000 채널 코딩의 표준으로 93년 발표된 터보 코드와, Viterbi/RS 연접 부호가 경합을 벌이고 있으며, 국내에서도 삼성, 데이콤, ETRI 등에서 각각의 표준안을 마련하여 이를 통합, 국내의 채널 코딩 기술 표준안을 점점 중에 있다. 비록 터보 코드가 Shannon's limit에 근접하는 최적의 성능을 나타나고 있으나 프레임 길이, interleaver의 지연등 실시간 처리가 불가능하고 구현의 어려움 (MAP 복호기의 구현) 등의 문제로 $K=9$ 인 길쌈 부호/Viterbi 복호기 + RS code의 연접 부호가 채널 코딩의 표준안으로 채택될 가능성이 높으며, 연구과제로 터보 코드를 선정할 것으로 보인다.

참고문헌

[1] G. D. Forney Jr. "The Viterbi Algorithm," 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] 신용섭, "CDMA 개발을 통한 IMT-2000으로의 전환 전략" TTA 저널, 51호 1997
 [5] C. Berrou, A. Glavieux and P. Thitimajshima, "Near Shannon Limit Error-Correcting Coding and Decoding : Turbo Codes(1) ", Proceedings of ICC '93, pp. 1064-1070, Geneva, Switzerland, May 1993
 [6] FPLMTS/IMT-2000, Report of the Tenth Meeting of ITU-R Task Group 8/1, Mainz, April 1996.
 [7] Peter Jung, " Comparision of Turbo-code Decoders Applied to Short Frame Transmission Systems ", IEEE Journal of Selected Areas in Communications, vol. 14, no. 3, pp. 530-537, April 1996
 [8] Steven S. Pietrobon and Adrian S. Barbulescu, "A Simplification of the Modified Bahl Decoding Algorithm for Systematic Convolutional codes ", Proceedings of ISITA' 94, pp. 875-880, Sydney, Australia, November 1994
 [9] S. Dolinar and D. Divsalar, " Weight Distributions for Turbo Codes using Random and Nonrandom Permutation ", TDA Progress Report 42-122, pp. 56-65, August 15, 1995
 [10] 이문호, 최승배, " 터보코드의 이론과 구조 및 성능", 전자공학회지 vol. 23, no 10, 1996
 [11] 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.]
 [12] Seungbae Choi, Jinsu Chang, Juyong Park, Moon Ho Lee, "A New Decoding Algorithm for Convolutional Encoding Using Syndrome Checking Error Estimation," IEEE ICCS/ISPACS '96, Singapore, Nov. 25-29, 1996.

이 문 호

- 1990년 : 일본 동경대 전자과 공학박사
- 1983년 : 통신기술사
- 1984년 : 전남대 전자과 공학박사
- 1985년~86년 : 미국 미네소타 주립대 전기과 연구교수
- 1990년 : 독일 하노버대학 아흔공대 전기과 연구교수
- 1970년~80년 : 남양 MBC 송신소장
- 1997년 : 정보통신부 정책심의위원 및 디지털
방송추진협의회 위원
- 1997년~현재 : 전북대 정보통신 공학과 교수