

멀티미디어 데이터를 위한 빠른 처리 속도를 가지는 새로운 채널코드, MLC(Multi-Level Code)의 설계

정희원 공형윤*, 학생회원 이창희*

Design of New Channel Codes, MLC(Multi-Level Code), with Fast Coding Time for Processing of Multimedia Data

Hyung-Yun Kong* Regular Member, Chang-Hee Lee* Student Member

요 약

본 논문에서는 차세대 무선 통신 시스템에 적용이 가능한 새로운 FEC(Forward Error Correction) 부호화 방법으로 MLC(Multi-Level Convolutional) 부호화 방식을 제안한다. 차세대 무선통신서비스는 음성, 데이터, 영상 등 많은 종류의 서비스를 함으로써 데이터의 처리속도가 빠른 시스템이 요구된다. 데이터 처리시간을 단축시키기 위한 방법으로 다중 레벨을 이용하여 부호어를 만들어 내는 방식의 부호화 시스템을 설계하였다. MLC는 부호 처리 시간을 단축시킬 뿐만 아니라 다양한 알고리즘을 이용해 부호어를 만들어 낼 수 있다는 특징을 가지게 된다. 모의 실험은 MLC 코드의 두 가지 방법, Modulo-operation 방식과 Galois Field-Operation 방식을 이용하여 수행하였다. 또한 모의실험을 통하여 $(s=2, T=2)$ 인 경우가 MLC 부호기의 최적 연결다항식임을 알았다.

ABSTRACT

In this paper, we propose new forward error correction (FEC) coding method named MLC(Multi-Level Convolutional) code for next generation wireless communication systems. Next generation wireless data service should have large user capacity and provide many kinds of services(voice, data, video, etc), so required coding system with fast coding procedure. To decrease coding procedure time, we design the MLC code, which is designed by using a non-binary operation algorithm and also can be designed by various advanced algorithms. This paper adapt two MLC coding techniques; MLC code based on modulo operation and Galois Field(GF) operation for simulations. To verify and compare the performance of our proposed coding technique, the computer simulations have been performed under non-gaussian noise environments. From the simulation results, we also know that the optimum connection polynomial is $(s=2, T=2)$ in case of MLC.

I. 서론

최근 정보통신분야뿐만 아니라 제어, 의료, 산업 등 각 분야에서 무선을 이용한 정보 데이터 전송에 많은 관심을 가지고 이에 대한 연구가 꾸준히 진행되고 있으며, 정보의 전송에 있어 가장 많은 부분을

차지하고 있는 분야로는 이동통신분야(PCS, IMT-2000 ..) 및 가입자망 서비스 분야(MMDS, LMDS ...), 위성통신분야 등으로 볼 수 있다. 무선을 이용한 데이터의 전송은 유선 채널을 이용한 통신 선로의 구축보다 그 소요비용 뿐만 아니라 소요시간 등에 장점을 가지고, 또한 유지, 보수 및 그 서비스의

* 울산대학교 전기전자 및 자동화공학부(hkong@uou.ulsan.ac.kr),

논문번호 : 00296-0729, 접수일자 : 2000년 7월 29일

* 본 연구는 SK 텔레텍 지원 및 울산대학교 전기전자 통신연구소 관리로 수행되었습니다.

유연성에서도 우수성을 가지고 있다. 반면 무선을 이용한 데이터의 전송은 유선에 비해서 여러 가지 장애요소를 가지고 있으며, 이로 인하여 무선 채널을 통한 데이터의 전송 중 에러가 발생할 수 있는 확률은 더욱 높아져 전송 데이터에 대한 신뢰도를 저하시키는 요인이 되고 있다^{[1][2]}. 이러한 이유로 최근 들어 데이터의 전송 중 발생할 수 있는 에러에 대하여 효과적으로 대처할 수 있는 채널부호화 방식에 대하여 중요성을 인식하고 많은 관심과 연구가 진행되고 있다 현재 많이 사용되고 있는 채널 부호화 방식으로는 콘볼루션 코드/RS 코드의 직렬 연결부호화 방식(serial concatenated codes) 및 터보 코드의 병렬 연결부호화 방식(parallel concatenated codes)을 들 수 있다^{[3][4]}. 이중 콘볼루션코드는 예전부터 현재까지 무선 및 유선 통신시스템에서 가장 많이 적용되어지고 있는 채널 코딩 방법 중의 하나로, 1955년 Elias에 의해서 개발되고, 이후 1967년 Viterbi에 의해서 Viterbi 복호 알고리즘의 개발로 채널부호화 방식으로써의 적용이 활발하게 이루어지게 되었다. 본 논문에서는 무선 통신 시스템에서의 채널부호화 방식으로 사용되어질 수 있는 새로운 콘볼루션 코드로 MLC방법을 제시하고, 이에 대한 특징, 동작 설명 및 성능을 분석하였다. MLC코드는 기존의 채널 부호화방식과 비교하여 데이터 처리속도를 향상시킨 부호화 방식으로, 다양한 부호화 알고리즘을 이용하여 시스템의 성능을 개선시킬 수 있는 부호화 방식이다. 본 논문에서는 두 가지 방식의 MLC 코드에 대하여 논하고 있는데, 첫 번째 방식으로는 Modulo-Operation을 이용한 방식으로 단순 병렬 콘볼루션 코드의 형태를 가지는 방식이며 두 번째는 Galois Field의 원소 값들을 이용하고, 각 원소들간의 코딩 연산을 위한 연산 규칙을 메모리 화하여 코딩 처리하는 방식이다. 두 가지 방식 모두 개념적으로는 기존의 콘볼루션 코드의 병렬 구성의 형태를 가지게 됨으로써 코딩 처리 속도를 향상시킨 코딩 기법이다. 컴퓨터 모의실험을 통하여 기존의 콘볼루션 코드와 성능을 비교하기 위해 다양한 채널에서 비트 오차율(Bit Error Ratio: BER)을 비교 분석하였다. 또한, MLC를 구성하는데 있어 최적의 연결다항식을 찾는 것은 MLC를 응용한 부호화시스템을 구성할 때 보다 효과적인 방법을 제시하게 될 것이다. 본 논문의 구성은 다음과 같다. 2절에서는 기존의 콘볼루션 코드와 본 논문에서 제시하고 있는 MLC의 두 가지 방법에 대한 동작, 특성과 차이점등에 대하여 기술한다.

3절에서는 가정된 채널 환경에서의 시뮬레이션을 다루고 있으며, 4절에서는 3절의 모의실험 결과를 토대로 MLC 기법의 적용가능성 및 그 타당성에 대하여 기술하였다.

II. MLC (Multi-Level Code)

1. 콘볼루션코드

그림 1은 K 개의 shift register를 가지고, code rate가 1/2인 기존의 콘볼루션 코드의 부호기를 나타낸 것이다. 그림에서 보는 바와 같이 콘볼루션 코드는 연속된 shift register의 저장 값들이 천이 되면서 입력 값과 register의 저장 값과의 콘볼루션에 의하여 부호어를 만들어 내는 방식이다. 이러한 부호화 방식은 이전 입력 값과 현재의 입력 값을 가지고 부호어를 만들어 내기 때문에 복호 시 현재 입력 값이 오류를 가지고 있더라도 이전의 값이 올라간다는 가정 하에서 빠르게 복구할 수도 있다는 개념의 채널 부호화 방식이다.

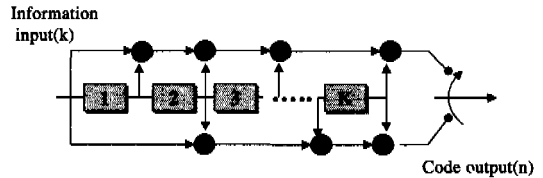


그림 1. 콘볼루션 부호기

2. MLC Code

비이진 콘볼루션 부호화 방식은 William E. Ryan, Jinn-Ja Chang 등에 의해 이전에도 연구가 이루어진 부분이다^{[5][6]}. 그러나 본 논문에서 제시하고 있는 MLC 방식은 기존의 비이진 콘볼루션 코딩방식과 다음과 같은 점에 있어 구분되어진다. 기존의 비이진 부호화 방식은 부호어를 생성하는데 있어 각각의 register에 연결되어 추출되어지는 값에 비이진 값(contents)을 곱하는 형식으로 부호어를 만들어 내게 된다. 즉, 이진 코드의 생성다항식 $(1+x^2+x^3+\dots+x^m)$ 을 따르지 않고 비이진의 생성다항식 $(a^r+a^r x+a^r x^2+\dots+a^r x^m)$ 을 가지게 된다. 여기서, x는 생성다항식에서 메모리의 연결되는 위치를 나타내고 있으며, a^r 는 임의의 비이진 값(예 Galois Field 원소)을 나타낸다. 반면, MLC방식은 생성다항식이 이진 코드의 형식을 따르게 된다. 즉, register 자체를 Tuple이라는 묶음 단위로 구성하여

전체를 Shift 시켜 부호어를 만들어 내게 된다. 기존의 비이진 콘볼루션 부호화 방식은 데이터를 처리하는데 걸리는 시간은 기존의 이진 콘볼루션 부호화 방식과 비교하여 보다 많은 시간을 요구하게 된다. 즉, 입력 및 출력시간은 동일하나 중간과정으로 부호어를 만드는데 있어 비이진 값을 곱하는 만큼의 시간 지연이 발생하게 된다. 그러나 MLC 방식은 II.4 절에서 설명되듯이 기존의 콘볼루션 코드보다 향상된 처리속도로 부호어를 만들어냄으로써 데이터 처리시간을 줄인다는 특성을 가지고 있다. MLC 부호기는 그림 2에서 보는 바와 같이 s 개의 register를 갖는 K개의 Tuple로 구성되어 있다. Register는 1 bit 저장장치이고, Tuple은 s 개의 register의 묶음으로 정의할 수 있다. Register가 0과 1의 두 가지 상태에 대한 값을 저장한다면, Tuple은 2^s개의 상태를 가질 수 있는 multi-level 값을 저장할 수 있다.

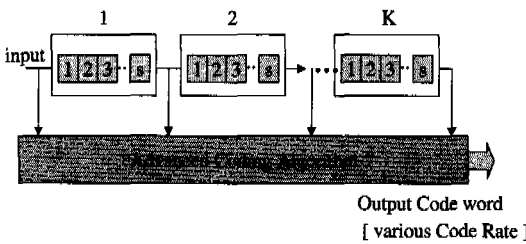


그림 2. MLC 부호기

ACA (Advanced Coding Algorithm) 블록은 부호어를 만들어내기 위한 가장 중요한 부분으로 각각의 입력 원소들에 대한 정해진 생성식에 의해 연산을 하는 부분이다. ACA블록은 이진 콘볼루션 부호기의 생성다항식이 어떻게 구성되는지를 표현하게 된다. MLC의 경우, 입력되는 원소는 multi-level의 값을 가지며, 이들의 연산에 의해 multi-level의 부호어를 만들어내게 된다. 이러한 부호화 기법은 multi-level의 값을 하나의 단위로 부호화하기 때문에 입력 데이터들에 대한 처리 시간을 절약할 수 있다는 장점을 가지는 요인이기도 하다. 이러한 MLC 부호기를 구성하기 위해서 다양한 방법의 부호화 알고리즘을 구상할 수 있는데, 본 논문에서는 두 가지 방법을 제시한다. 한 방법은 기존의 콘볼루션 코드의 병렬 연결 개념을 가진 Modulo-operation 방식이다. Modulo-operation 방식을 이용한 MLC는 부호화 처리 작업이 기존의 콘볼루션 코드와 거의 유사하여 쉽게 구현이 가능하고 구조

가 간단하다. 다른 한 가지는 Galois Field를 이용한 부호화방식으로 multi-level 연산을 이용하여 부호화 처리를 수행함으로써 처리 시간을 보다 단축시킬 수 있고, 부호화 처리를 위한 multi-level 연산이 정의된다면, 기존의 콘볼루션 코드와 같은 구조를 가지고 있으므로 소프트웨어적으로 구현이 간단하다. 여기서 Galois Field는 유한한 공간을 가지는 집합으로 정의되며, 집합의 모든 원소들은 multi-level 값을 표현한다. 다음은 두 가지 방식 (Modulo-operation과 Galois Field 연산에 의한 부호화 방식)에 대하여 예를 보여준다.

Example) Modulo-Operation을 이용한 MLC.

Modulo-operation을 이용한 부호화방법은 기존의 콘볼루션 코드의 방법을 그대로 이용하면서 그 입력 데이터를 병렬로 바꾸어 부호어를 만들어내는 방식이다. 이러한 방식은 기존의 콘볼루션 코드를 그대로 이용할 수 있어 그 구현이 쉽다는 점에서 장점을 가진다. 이러한 Modulo-operation을 이용한 방식의 예를 간단히 설명하면 다음과 같다.

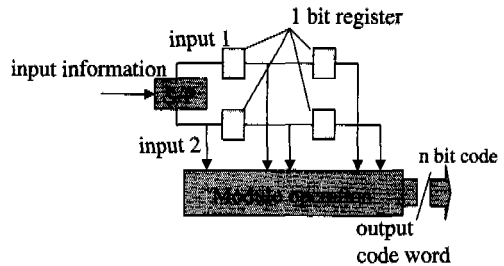


그림 3. (s=2,K=2)MLC와 등가인 콘볼루션 부호기

그림 3 은 Modulo-operation을 이용해서 각각의 부호어를 만드는 방식의 부호기 부분을 나타낸 것이다. 만약, 입력 정보 데이터가 [01], [10], [11], ... 이고, 초기 Tuple의 상태를 [00], [00]라 두면 아래와 같은 부호어가 생성되게 된다. 각각의 입력 데이터가 들어오는 시간을 t로 두어, t₀ ~ t₃에서 생성되어 지는 부호어를 나타낸 것이다. 여기서, ACA블록은 c_{1t}=d₀⊕d_{t-2}, c_{2t}=d_t⊕d_{t-1}⊕d_{t-2} 로 정의되면 다음의 예로 부호어가 생성되게 된다.

$$\begin{aligned}
 t_0 : c_{10} &= d_0 \oplus d_{-2} = [01] \oplus [00] = [01] \\
 c_{20} &= d_0 \oplus d_{-1} \oplus d_{-2} = [01] \oplus [00] \oplus [00] = [01] \\
 t_1 : c_{11} &= d_1 \oplus d_{-1} = [10] \oplus [00] = [10] \\
 c_{21} &= d_1 \oplus d_0 \oplus d_{-1} = [10] \oplus [01] \oplus [00] = [11] \\
 t_2 : c_{12} &= d_2 \oplus d_0 = [11] \oplus [01] = [10]
 \end{aligned}$$

$$c_{22} = d_2 \oplus d_1 \oplus d_0 = [11] \oplus [10] \oplus [01] = [00]$$

t_0 에서 첫 번째 부호어 c_{10} 은 입력데이터 d_0 와 d_2 의 Modulo-operation에 의해서 [01] 값이 생성되어지게 된다. 여기서 d_0 와 d_2 의 modulo-operation은 부호기의 생성다항식에 해당한다. 그리고 같은 시간에 생성되어지는 c_{20} 는 d_0 , d_1 과 d_2 의 modulo-Operation에 의해서 [01] 이 생성된다. c_{20} 역시 생성식으로 정의되어진 것이다. 이러한 부호화 방식은 병렬의 콘볼루션 코드로 부호기를 구성할 때 기존의 부호기/복호기를 그대로 이용할 수 있다는 점에서 효과적이다.

Example) Galois Field를 이용한 MLC Code case

Galois Field를 이용한 부호어를 만드는 부호기는 그림.4와 같이 구성되어지고 부호어를 만들기 위한 부호화 연산 block이 따로 구성되어져서 multi-level (Galois Field elements)연산을 하게 된다.

$s=2, K=2$, data rate 1/2 MLC [그림 4]

여기서 ACA block은 식(1)과 (2)로 정의된다.

$$c_{1k} = d_k \oplus d_{k-2} \tag{1}$$

$$c_{2k} = d_k \oplus d_{k-1} \oplus d_{k-2} \tag{2}$$

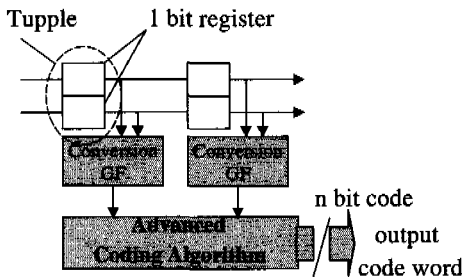


그림 4. $s=2, K=2$ 인 MLC 부호기

여기서, c_{1k} , c_{2k} 는 k 시간에 부호화 되어지는 multi-level 부호어이고, d_k 는 k 시간에 입력되어지는 multi-level의 입력데이터이다. 여기서 d_k 및 c_k 는 표 1의 Galois Field [GF(2²)]의 원소들로 표현된다.

표 1. GF(2²)의 원소

Exponential Rep.	Binary Rep.
0	00
1	01
α	10
α^2	11

Galois Field 원소의 연산 규칙은 다음과 같이 정의되어진다.

$$0 \oplus 0 = 0, \quad 0 \oplus 1 = 1, \quad 0 \oplus \alpha = \alpha \dots\dots\dots$$

$$\alpha \oplus \alpha^2 = 1, \quad \alpha^2 \oplus \alpha^2 = 0.$$

그리고, MLC를 해석하기 위한 방법으로 격자도 (trellis diagram)를 사용할 수 있다. 격자도의 형상을 보면 기존의 콘볼루션 코드와의 차이를 쉽게 발견할 수 있다. 기존의 콘볼루션 코드의 경우에는 하나의 상태에서 다음의 상태로 연결 가능한 가지의 수가 2가지인데 반해, MLC의 경우에는 2^s가지의 경로를 가진다. 그림 5는 $s=2, K=2$ 인 MLC 에 해당하는 격자도이다.

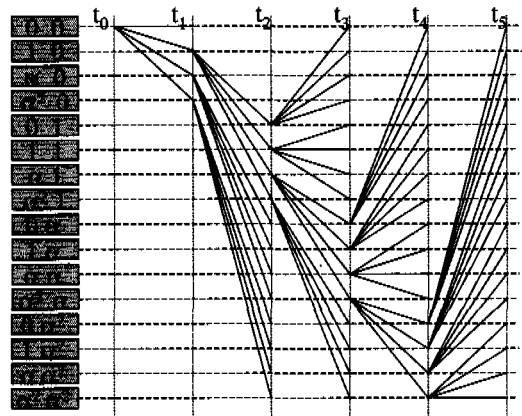


그림 5. $s=2, T=2$ MLC 코드의 격자도

간단히 예를 들면 input information data 가 1, $\alpha, \alpha^2 \dots$ 라고 가정할 때,

$$t_0 : c_{10} = d_0 \oplus d_{-2} = 1 \oplus 0 = 1$$

$$: c_{20} = d_0 \oplus d_{-1} \oplus d_{-2} = 1 \oplus 0 \oplus 0 = 1$$

$$t_1 : c_{11} = d_1 \oplus d_{-1} = \alpha \oplus 0 = \alpha$$

$$: c_{21} = d_1 \oplus d_0 \oplus d_{-1} = \alpha \oplus 1 \oplus 0 = \alpha^2$$

$$t_2 : c_{12} = d_2 \oplus d_0 = \alpha^2 \oplus 1 = \alpha$$

$$: c_{22} = d_2 \oplus d_0 \oplus d_{-1} = \alpha^2 \oplus \alpha \oplus 1 = 0$$

이러한 과정으로 부호어가 생성되어지게 된다. 즉, 생성되어지는 부호어는 1, 1, $\alpha, \alpha^2, \alpha, 0 \dots$ 이 되고, 전송은 0 1 0 1 1 0 1 1 1 0 0 0 ... 의 binary stream으로 전송되어진다.

3. MLC에서의 복호 방법

MLC 복호방법은 전통적인 Viterbi 복호 알고리즘과 Viterbi 복호 방법에서 향상된 성능을 보이는 SOVA(Soft Output Viterbi Algorithm)을 이용할 수

있다. Viterbi 복호 알고리즘은 여러 경로 중 최적의 경로(생존경로)를 선택하는 최우선 복호(Maximum Likelihood) 방식을 이용하는 복호하는 방법으로, 본 논문에서 제시한 MLC 복호의 두 가지 방법 모두에 적용하였다. Modulo- Operation의 경우에는 각각의 원소에 대하여 그 치환과정을 거친 후 각각의 원소들의 거리를 구하게 되고, Galois Field의 원소를 이용한 경우에는 정의된 연산 규칙에 의해서 거리를 구하게 된다. 기존의 콘볼루션 코드와 다른 점은 복호의 복잡도를 줄이기 위해서 t 시간에 어느 한 상태에 모아지는 가지가 모두 4 개가되므로 4 가지 중 최소의 거리를 갖는 가지만을 택하는 것이다.(그림 5) 즉, 어느 한 시간에 대한 경로의 1/4만을 선택하여 복호 한다. 여기서 콘볼루션 코드와의 차이점을 찾을 수 있다. 정상상태 이후 선택 경로의 수는 기존의 콘볼루션 코드의 경우에는 1/2개가 선택(선택 가지 당 2 개의 가지 경로가 가능)되고, MLC code의 경우에는 K개 선택(선택 가지 당 2^K 개의 경로가 가능) 된다. 이러한 방법으로 모든 가지에 대한 경로를 조사하여 최적의 경로를 찾아내어 생존경로의 값을 복호 하게 된다. 이러한 과정을 프레임의 길이 만큼 하게 되면 복호화 과정은 종료한다. SOVA를 이용한 복호법도 ML 방식을 이용한다는 점에서 유사하나, Viterbi 복호법과 다르게 순환 복호가 가능하고, soft-decision을 이용한다는 측면에서 보다 우수한 성능을 보이는 방식이다. 식(3)은 SOVA를 이용한 복호방식을 나타낸 것이다.

$$\log \frac{\sum_{d_k=+1|x} P(d_k=+1|x)}{\sum_{d_k=-1|x} P(d_k=-1|x)} < 0 \quad (3)$$

여기서, x는 수신된 데이터 값이다. 수신된 데이터의 값의 확률적인 값을 비교하여 결정하는 방식이다.

4. MLC Encoding Time 비교

결론적으로 MLC 방법은 부호화 처리 시간을 감소시킬 수 있다. MLC는 위의 그림 3에서 보는 바와 같이 부호어는 콘볼루션 코드의 병렬 연결에 의해 출력 값이 결정되어지는 개념을 가지고 있다. 즉, MLC 기법은 부호화 시간에 있어서 연산이 병렬로 처리되는 만큼 그 부호화 출력 값을 빠르게 얻을 수 있어 MLC는 Tuple 이 가지고 있는 register의 수가 증가할수록 동일 조건의 콘볼루션 코드보다 부호화 시간을 줄일 수 있다. 콘볼루션 코드와 MLC의 정상상태에 이르는 시간을 비교하면 다음과 같다. 여기서 정상상태라는 것은 모든

register(MLC의 경우 Tuple)의 상태까지 입력 값이 도달한 상태를 말한다. 콘볼루션 코드의 경우에 t = k (Constraint length : register의 수) 시간 이후에 정상상태에서 동작을 시작한다. (예 : k=4 인 경우 : 4), 그리고 MLC의 경우에 (2^s)(2^k) = (2^s)^k로 정의되어지며, 즉, t =K(Tuple의 수) 시간 이후에 정상상태에 이르게 된다. 여기서 알 수 있듯이 MLC의 경우에는 약 두 배의 속도향상을 기대할 수 있다 (K=2, s=2 인 경우 : 정상상태 도달 시간은 2이다). 위와 같은 복잡도를 갖는 콘볼루션 코드와 MLC에서는 코딩 처리 시간이 거의 두 배 가까운 속도향상을 보이고 있다. 또한 위의 경우를 통하여, MLC의 경우에 있어서 하나의 Tuple에 속해 있는 register의 수 s가 증가할수록 부호화 속도는 s배 빠르게 진행됨을 알 수 있다.

III. 모의실험 결과

모의실험의 방식은 컴퓨터를 이용한 Monte-Carlo 모의 실험방식을 이용하였다^{[7]-[8]}. 그림 6은 모의실험에 사용된 부호기들을 간략하게 도식화한 것이다. 그림 6 (a)는 기존의 콘볼루션코드의 경우이며, 그림 6 (b)는 Modulo-operation의 MLC의 경우를 나타낸 것이고, 그림 6 (c)는 Galois Field를 이용한 MLC의 경우를 나타낸 것이다.

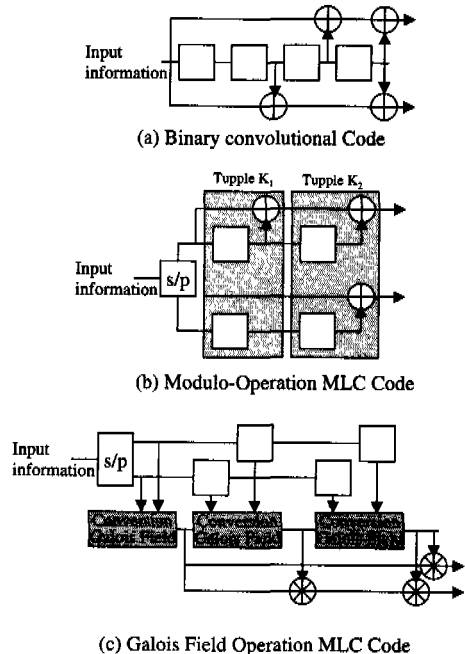


그림 6. 모의실험을 위한 부호기 시스템

일반적으로 무선전송시스템의 성능분석은 fading channel에서 이루어지지만, 본 논문은 기존 시스템과 제안하는 시스템의 단순 성능분석을 위한 모의 실험이기 때문에 가우시안 및 비 가우시안 채널을 설정하여 모의실험하였다. 그림 7은 Modulo-Operation을 적용한 MLC 부호화기법에서 gaussian 채널에서의 성능을 분석한 것이다.

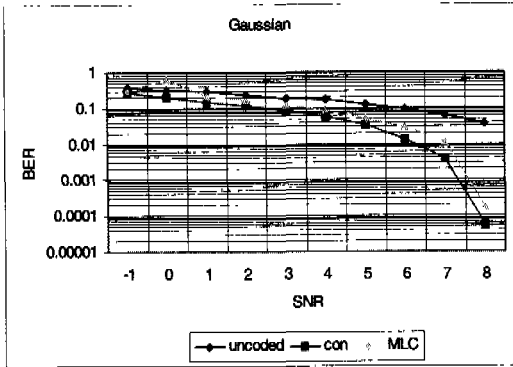


그림 7. 가우시안 환경에서의 MLC의 성능분석

성능에서 다소 차이를 보이고 있지만, 차세대 무선통신시스템의 전송채널을 전형적인 가우시안 채널이 아닌 수직적인 분석이 힘든 비 가우시안 채널이 일반적이다. 이 후 성능 분석 결과는 모두 비 가우시안 채널을 이용하였다. 그림 8은 Modulo-operation을 이용하여 MLC를 구성했을 때 기존의 콘볼루션 코드와의 성능을 비교한 것이다.

그림 8과 그림 9은 Mixture Noise 환경 하에서 모의실험 결과로 기존의 콘볼루션 코드와 비교하여 모두 BER 10^{-3} 에서 약 0.1 ~0.2 dB의 부호화이득 (Coding Gain) 차이를 보이고 있다. GF Operation

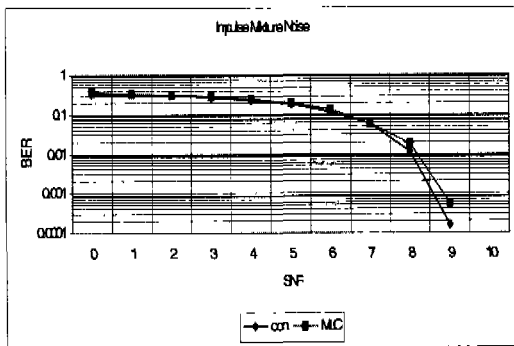


그림 8. Mixture noise의 경우의 모의실험결과 (Modulo-Operation Case)

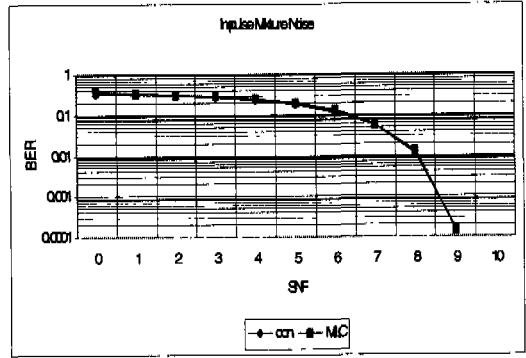


그림 9. Mixture noise Case의 모의실험 결과 (GF Operation Case)

방식이 Modulo Operation 경우보다 약 0.2dB 낮은 부호화 이득을 보이고 있다.

그림 10은 전송환경을 Salt&Pepper Noise로 가정한 상태에서 모의실험 결과로 Salt &Pepper 환경에서는 MLC 부호기의 성능이 기존의 콘볼루션 코드와 비교하여 약 -0.3정도의 부호화 이득을 가지고 있었다. MLC 부호기의 경우에 Mixture Noise의 경우에는 기존의 콘볼루션 코드와 비슷한 성능을 보이지만 Salt&Pepper Noise의 경우에 조금 차이 나는 성능을 보임을 알 수 있었다.

그림 11은 SOVA를 이용하여 복호한 경우에 전송환경을 Salt&Pepper Noise로 가정한 상황에서의 모의실험 결과이다. 전체적인 BER 특성곡선은 Viterbi 복호 방식과 비교하여 BER 10^{-3} 정도에서 약 0.1 dB의 차이가 있음을 알 수 있었다. MLC 코드가 콘볼루션 코드와 비교하여 BER 성능의 차이를 보이는 이유는 다음으로 설명되어 질 수 있다. 격자도에서 콘볼루션 코드는 일정 시간 t에서의 register의 상태 값은 이전의 상태에서 2 가지 경로

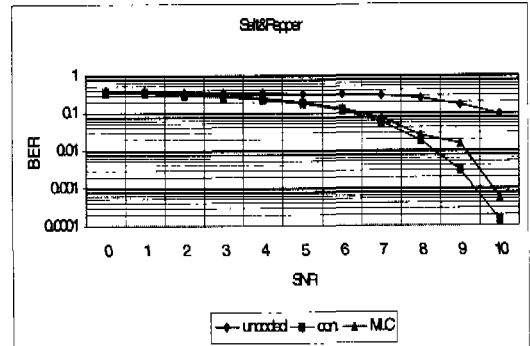


그림 10. Salt&Pepper noise에서의 모의실험결과 (GF Operation Case)

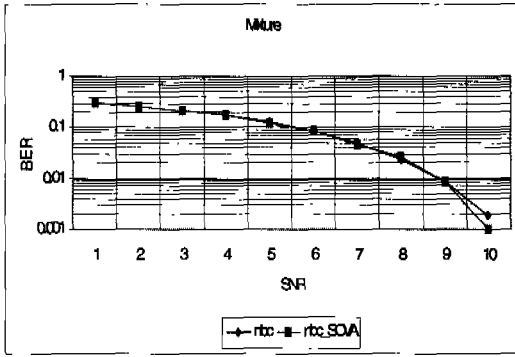


그림 11. SOVA 복호를 이용한 MLC의 모의 실험결과

만을 취하여 해밍거리(Hamming Distance) 값을 계산하고, 여기서 최소의 값을 가지는 경로만을 선택하게 되지만 MLC의 경우에는 s의 값이 증가함에 따라서 한 상태에 모이게 되는 상태는 2^s가지가 되고, 2^s가지에 대해서 해밍거리를 계산하여 최소의 값을 갖는 한 개의 경로만을 선택하여 생존경로로 선택하게 된다. 즉, 본 논문에서 s=2에 해당하는 MLC기법을 사용하였을 경우에는 모두 4(2²)가지의 경로 중에서 최소의 거리를 갖는 경로를 검색하여 복호하게 된다. 콘볼루션 코드의 경우에 정확하게 복호 할 확률이 1/2이지만 MLC의 경우에는 1/4에 해당하게 된다. 이러한 이유로 부호화 이득의 차이를 가지게 된다. 식(4)은 콘볼루션 코드와 MLC의 경우, 올바른 값일 확률을 나타낸다.

$$P_T(c) = \begin{cases} \frac{1}{2} & , \text{ Binary Con. (constant)} \\ \frac{1}{2^s} & , \text{ NBC Code} \end{cases} \quad (4)$$

여기서, P_T(c)는 각각의 코드에 대한 올바른 값일 확률을 나타내고 있다.

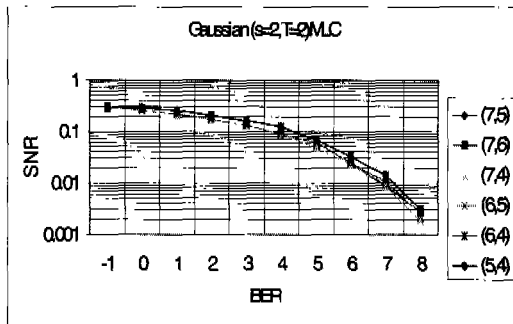


그림 12. 최적의 연결다항식을 찾기 위한 모의실험 결과

그림 12은 최적의 연결다항식을 찾기 위한 모의 실험결과로 (7,5)와 (7,4)가 다른 연결방식에 비해 BER 10⁻³정도에서 약 0.2~0.3dB의 부호화이득의 차이를 보인다는 것을 알 수 있어 MLC를 이용하여 부호기를 구성할 때 이를 생성다항식으로 이용하여 구성하면 보다 효과적이다.

IV. 결론

MLC는 multi-level의 데이터가 처리되는 특성을 가지고 있으며 이러한 이유로 부호화 처리 시간의 단축에 있어서 기존의 콘볼루션코드방식에 비해 우수하다 할 수 있다. 결론적으로 부호화 처리 속도를 향상시키면서, 기존의 콘볼루션코드와 비교하여 크지 않은 성능차이를 보임을 시뮬레이션 결과로 알 수 있었다. 또한 ACA 블록은 Tuple 단위로 multi-level의 값을 이용하여 부호화 하게 되므로 다양한 알고리즘을 적용한 부호화기법을 가질 수 있어 앞으로 보다 많은 연구가 필요할 것이다. 그리고, 한정된 구성장의 길이를 갖는 MLC 코드에서 최적의 연결다항식을 찾는 것은 부호기의 성능을 향상시키는 데 있어 필수적인 요건이 될 수 있다. 본 논문에서는 최적의 연결다항식을 찾기 위해 모의실험방식을 이용하였다. MLC 코드방식의 특성을 요약하면 다음과 같다.

- 부호화 처리 속도를 향상시킨다.
- ACA에 따라서 여러 가지 구조의 MLC를 구성할 수 있다.
- Modulo-operation case : 구조가 간단하고 기존의 콘볼루션코드를 약간의 변형으로 부호기의 설계가 가능하다.
- Galois Field를 이용한 MLC case : 기본적인 연산 규칙을 정의해 놓으면 Modulo-operation case 보다 짧은 시간에 부호화 처리를 할뿐만 아니라 소프트웨어 상으로 구현이 간단하다.
- (s=3, T=2)MLC 코드의 최적의 연결다항식은 (7,5) 또는 (7,4)로 알 수 있었다.

본 연구는 비교적 단순한 ACA block를 이용하여 기존의 콘볼루션코드와 성능을 비교 분석하였다. 차후 연구과제로 ACA block에 기존의 RS 개념을 도입하여 iteration 복호가 가능한 구조를 설계한다면, 속도적인 측면뿐만 아니라 성능에서도 우수한 부호기를 설계하게 될 것이다.

