

# 코딩 속도향상을 위한 채널 코드의 설계

공형윤, 이창희

울산대학교 전기, 전자 및 자동화공학부

전화: (052) 259-2194 / 팩스 : (052) 259-1685

## Design of New Channel Codes for Speed Up Coding Procedure

Hyung-Yun Kon, Chang-Hee Lee

School of Electronics and Electrical, Electronics & Automations, Ulsan University

E-mail : hkong@uou.ulsan.ac.kr, leechee@kmail.com

### Abstract

*In this paper, we present a new channel coding method, so called MLC (Multi-Level Codes), for error detection and correction in digital wireless communications. MLC coding method use the same coding procedure used in the convolutional coding but it is distinguished from the existing convolutional coding in point of generating the code word by using multi-level information data (M-ary signal) and in point of speed of coding procedure. Through computer simulation, we analyze the performance of the coding method suggested here compared to convolutional coding method in case of modulo-operation and in case of non-binary coding procedure respectively under various channel environments.*

### I. 서론

최근 정보통신분야뿐만 아니라 제어, 의료, 산업등 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 유연성에서도 우수성을 가지고 있다. 하지만 무선을 이용한 데이터의 전송은 유선에 비해서 전송상에서 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. 이러한 이유로 최근 들어 데이터의 전송 중 발생할 수 있는 에러에 대하여 효과적으로 대처할 수 있는 채널코딩에 대하여 중요성을 인식하고 많은 관심과 연구가 진행되고 있다 채널코딩방식으로 많은 무선통신시스템에 적용하고 있는 방식으로 콘볼루션코드/RS code의 연결부호방식(concatenated codes) 및 Turbo code등을 들 수 있다. 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 채널 코딩으로써의 적용이 활발하게 이루어지게 되었다. 본 논문에서는 무선 통신 시스템에서의 채널코딩방식으로 사용되어질 수 있는 새로운 콘볼루션코드로 MLC (Multi-Level Codes)방법을 제시하고, 이에 대한 특징, 동작 및 성능을 분석하였다. MLC는 다양한 알고리즘을 가질 수 있는 코딩 방식으로 앞으로 많은 연구가 필요한 부분이다. 본 논문에서는 두 가지 방식의 MLC에 대하여 논하고 있다. 첫 번째 방식으로는 modulo-operation을 이용한 방식으로 단순 병렬 콘볼루션코드의 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 콘볼루션코드의 병렬 구성의 형태를 가지게 됨으로써 코딩 처리 속도를 향상시킨 코딩 기법이다. 또한 본 논문에서는 컴퓨터 시뮬레이션을 통하여 기존의 콘볼루션코드와 다양한 채널 환경에서 성능을 비교 분석함으로써, MLC는 기존의 채널 코딩기법을 대신할 수 있는 개선된 코딩 방법임을 보여주고 있다. 본 논문의 구성은 다음과 같다. 2절에서는 기존의 콘볼루션코드와 본 논문에서 제시하고 있는 MLC의 두 가지 방법에 대한 동작, 특성과 차이점에 대하여 기술한다. 3절에서는 가정된 채널 환경에서의 시뮬레이션을 다루고 있으며, 4절에서는 3절의 시뮬레이션 결과를 토대로 MLC 기법의 적용가능성 및 그 타당성에 대하여 기술하였다.

### II. Multi-Level Code

#### 1. 콘볼루션코드

그림.1은 K 개의 shift register를 가지고, data rate가 1/2인 기존의 콘볼루션코드의 부호기를 나타낸 것이다. 그림에서 보는 바와 같이 콘볼루션코드는 연속된 shift register의 저장 값들이 천이 되면서 입력 값과 register의 저장 값과의 콘볼루션에 의하여 부호어를 만들어 내는 방식이다. 이러한 코딩 방식은 이전 입력 값과 현재의 입력 값을 가지고 부호어를 만들어 내기 때문에 복호시 현재 입력 값이 오류를 가지고 있더라도 이전의 값이 올바르다는 가정 하에서 바르게 복구할 수도 있다는 개념의 채널 코딩 방식이다.

본 연구는 SK Teletech. 연구비지원의 결과물입니다.

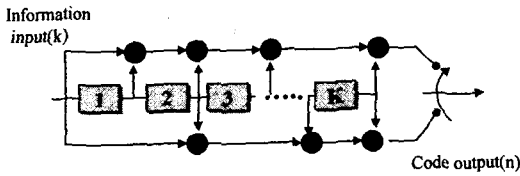


그림 1. 콘볼루션 부호기  
Fig.1. Convolutional Encoder

콘볼루션코드는 현재의 무선 이동 통신 시스템(PCS, IMT-2000..) 및 가입자망 서비스(WLL, LMDS..)의 채널 코딩 방식으로 사용되어지고 있다.

### 2. MLC Code

MLC 부호기는 그림.2에서 보는 바와 같이 s 개의 register를 갖는 K개의 Tuple로 구성되어 진다. Register는 1 bit 저장장치이고, Tuple은 s 개의 register의 묶음으로 정의할 수 있다. Register가 0과 1의 두 가지 상태에 대한 값을 저장한다면, Tuple은 2<sup>s</sup>개의 상태를 가질 수 있는 multi-level 값을 저장할 수 있다.

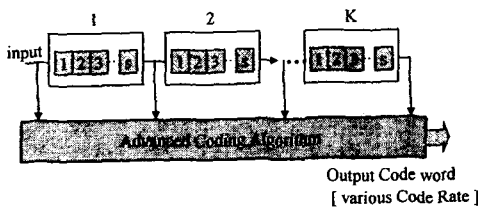


그림 2. MLC 부호기  
Fig. 2. MLC Encoder

ACA ( Advanced Coding Algorithm ) 블록은 부호어를 만들어내기 위한 가장 중요한 부분으로 각각의 입력 원소들에 대한 정해진 생성식에 의해 연산을 하는 부분이다. NBC code의 경우, 입력되는 원소는 multi-level의 값을 가지며, 이들의 연산에 의해 multi-level의 부호어를 만들어내게 된다. 이러한 코딩 기법은 multi-level의 값을 하나의 단위로 코딩하기 때문에 입력 데이터들에 대한 처리 시간을 절약할 수 있다는 장점을 가지는 요인이기도 하다. 이러한 MLC 부호기를 구성하기 위해서 다양한 방법의 코딩 알고리즘을 구상할 수 있는데, 본 논문에서는 두 가지 방법을 제시한다. 한 방법은 기존의 콘볼루션코드의 병렬 연결 개념을 가진 modulo-operation 방식이다. Modulo-operation 방식을 이용한 MLC는 코딩 처리 작업이 기존의 콘볼루션코드와 거의 유사하여 쉽게 구현이 가능하고 구조가 간단하다. 다른 한 가지는 Galois Field를 이용한 코딩방식으로 multi-level 연산을 이용하여 코딩처리를 수행함으로써 처리 시간을 보다 단축시킬 수 있고, 코딩처리를 위한 multi-level 연산이 정의된다면, 기존의 콘볼루션코드보다 소프트웨어 구현이 간단하다. 이러한 multi-level 값을 처리하기 위한 좋은 방법으로 Galois Field의 원소를 이용하면 코딩 처리 연산을 간단하게 할 수 있어 보다 효율적이다. 여기서 Galois Field는 유한한 공간을 가지는 집합으로 정의되며, 집합의 모든 원소들은 multi-

level 값을 표현한다. 다음은 두가지 방식 (modulo-operation과 Galois Field 연산에 의한 코딩 방식)에 대하여 예를 보여주고 있다.

**Example) Modulo-Operation**을 이용한 MLC. Modulo-operation을 이용한 코딩방법은 기존의 콘볼루션코드의 방법을 그대로 이용하면서 그 입력 데이터를 병렬로 바꾸어 부호어를 만들어내는 방식이다. 이러한 방식은 기존의 콘볼루션코딩을 그대로 이용할 수 있어 그 구현이 쉽다는 점에서 장점을 가질 수 있다. 이러한 Modulo-operation을 이용한 방식의 예를 간단히 설명하면 다음과 같다.

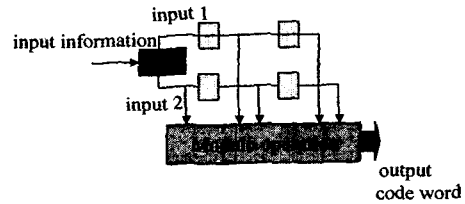


그림 3. (s=2, K=2)MLC와 등가인 콘볼루션코드  
Fig. 3. (s=2, K=2) MLC code

그림.3은 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, t<sub>0</sub> ~ t<sub>3</sub>에서 생성되어 지는 부호어를 나타낸 것이다.

$$t_0 : \begin{aligned} c_{10} &= d_0 \oplus d_{-2} = [01] \oplus [00] = [01] \\ c_{20} &= d_0 \oplus d_{-1} \oplus d_{-2} = [01] \oplus [00] \oplus [00] = [01] \end{aligned}$$

t<sub>0</sub>에서 첫 번째 부호어 c<sub>10</sub>은 입력데이터 d<sub>0</sub>와 d<sub>-2</sub>의 Modulo-operation에 의해서 [01] 값이 생성되어지게 된다. 여기서 d<sub>0</sub>와 d<sub>-2</sub>의 Modulo-operation은 부호기의 생성다항식에 해당한다. 그리고 같은 시간에 생성되어지는 c<sub>20</sub>은 d<sub>0</sub>, d<sub>-1</sub>과 d<sub>-2</sub>의 Modulo-operation에 의해서 [01]이 생성된다. c<sub>20</sub> 역시 생성식으로 정의되어진 것이다. 이와 같은 방식을 t<sub>1</sub>, t<sub>2</sub>에 대하여 부호어를 만들면 다음과 같다.

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

$$t_2 : \begin{aligned} c_{12} &= d_2 \oplus d_0 = [11] \oplus [01] = [10] \\ c_{22} &= d_2 \oplus d_{-1} \oplus d_0 = [11] \oplus [10] \oplus [01] = [00] \end{aligned}$$

이러한 코딩 방식은 병렬의 콘볼루션코드로 그 부호기를 구성할 때 기존의 부호기/복호기를 이용할 수 있다는 점에서 효율적인 방식이다. 이러한 점에서 간단하게 코딩 시스템을 구성할 수 있어 그 적용이 간단하다.

**Example) Galois Field**를 이용한 NBC Code case Galois Field를 이용한 부호어를 만드는 부호기는 그림.4와 같이 구성되어지고 부호어를 만들기 위한 코딩 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 다음과 같이 정의된다.

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

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

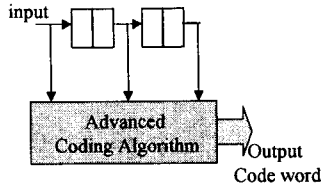


그림 4. s=2, K=2인 MLC  
Fig. 4. (s=2, K=2) MLC

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

표 1.  $GF(2^2)$ 의 원소

Exponential Rep.	Binary Rep.
0	00
1	01
$\alpha$	10
$\alpha^2$	11

Galois Field 원소의 연산 규칙은 다음과 같이 정의되어진다.

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

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

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

### 3. MLC에서의 복호 방법

복호방법은 전통적인 Viterbi 복호알고리즘을 이용할 수 있다. Viterbi 복호알고리즘은 여러 경로 중 최적의 경로(생존경로)를 선택하여 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 콘볼루션코드와 다른 점은 복호의 복잡도를 줄이기 위해서 t 시간에 어느 한 상태에 모아지는 가지가 모두 4 개가 되므로 4 가지 중 최소의 거리를 갖는 가지만을 택하는 것이다.(그림 5) 즉, 어느 한 시간에 대한 경로의 1/4만을 선택하여 복호 한다. 여기서 콘볼루션코드와의 차이점을 찾을 수 있다. 정상상태 이후 선택 경로의 수는 기존의 콘볼루션코드의 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 복호하게 된다. 이러한 과정을 프레임의 길이 만큼 하게 되면 복호화 과정은 종료한다

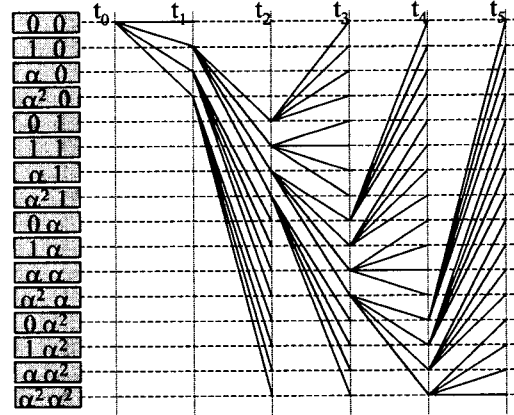


그림 5. s=2, K=2인 MLC의 격자도  
Fig. 5. Trellis diagram for (s=2,K=2)MLC

### 4. MLC Encoding Time 비교

결론적으로 MLC 방법은 코딩 처리 시간을 감소시킬 수 있다. MLC는 위의 그림 3에서 보는 바와 같이 부호어는 콘볼루션코드의 병렬 연결에 의해 출력 값이 결정되어지는 개념을 가지고 있다. 즉, MLC 기법은 코딩 시간에 있어서 연산이 병렬로 처리되는 만큼 그 코딩 출력 값을 빠르게 얻을 수 있다. MLC는 Tuple 이 가지고 있는 register의 수가 증가할수록 동일 조건의 콘볼루션 코드보다 코딩 시간을 줄일 수 있다. 콘볼루션코드와 MLC의 정상상태에 이르는 시간을 비교하면 다음과 같다. 여기서 정상상태라는 것은 모든 register(MLC의 경우 Tuple)의 상태까지 입력 값이 도달한 상태를 말한다. 콘볼루션코드의 경우에  $t = k$  (Constraint length : register의 수) 시간 이후에 정상상태에서 동작을 시작한다. (예 : k=4 인 경우 : 4), 그리고 MLC의 경우에  $(2^s)(2^K) = (2^s)^k$ 로 정의되어지며, 즉,  $t = K(\text{Tuple의 수})$  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 콘볼루션코드와 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 코딩 속도는 더욱(s배) 빠르게 진행됨을 알 수 있다.

### III. 시뮬레이션 결과

그림.6은 시뮬레이션에 사용된 encoder system들을 간략하게 도식화한 것이다. 그림.6 (a)는 기존의 콘볼루션 코드의 경우이며, 그림.6 (b)는 Modulo-operation의 MLC의 경우를 나타낸 것이고, 그림.6 (c)는 Galois Field를 이용한 MLC의 경우를 나타낸 것이다. 컴퓨터 시뮬레이션을 위한 채널환경으로 가우시안(Gaussian)채널로 설정하였다. 그림.7는 Modulo-operation을 이용하여 MLC를 구성했을 때 기존의 콘볼루션코드와의 성능을 비교한 것이다.

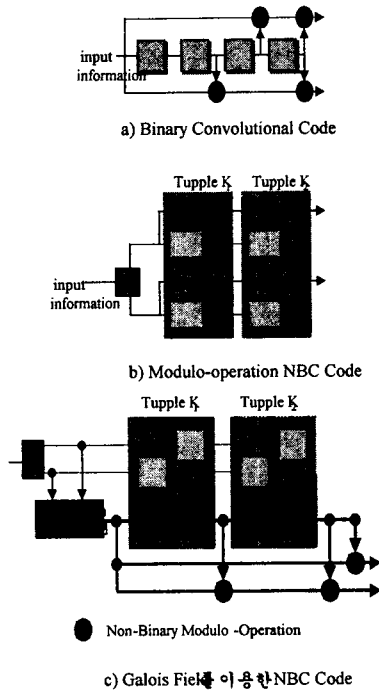


그림 6. MLC의 시뮬레이션 모델  
Fig. 6. Simulation model for MLC

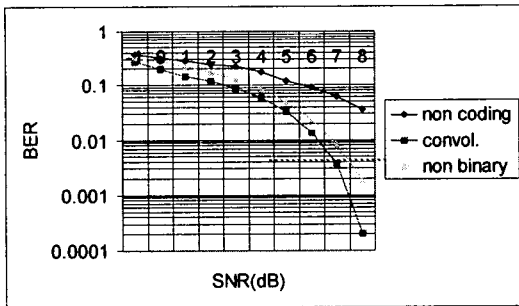


그림 7. 가우시안 환경에서의 MLC의 성능분석  
Fig. 7. Performance of MLC in Gaussian channel

그림.7은 가우시안 채널환경에서의 시뮬레이션 결과로 기존의 콘볼루션코드와 비교하여 약 0.1 ~0.2 dB의 코딩 이득(Coding Gain)의 차이를 보이고 있으며, 이러한 이유는 다음과 같이 정의되어 질 수 있다. 격자도에서 콘볼루션코드방법은 일정 시간 t에서의 register의 상태 값은 이전의 상태에서 2 가지 경로만을 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<sup>s</sup>가지가 되고, 2<sup>s</sup>가지에 대해서 해밍거리를 계산하여 최소의 값을 갖는 한 개의 경로만을 선택하여 생존경로로 선택하게 된다. 즉, 본 논문에서 s=2에 해당하는 MLC기법을 사용하였을 경우에는 모두 4( 2<sup>2</sup> )가지의 경로 중에서 최소의 거리를 갖는 경로를 검색하여 복호하

게 된다. 콘볼루션코드의 경우에 정확하게 복호 할 확률이 1/2이지만 NBC의 경우에는 1/4에 해당하게 된다. 이러한 이유로 코딩 이득의 차이를 가지게 된다식(1)은 콘볼루션코드와 MLC의 경우, 올바른 값일 확률을 나타내고 있다.

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

여기서, P<sub>T</sub>(c)는 각각의 코드에 대한 올바른 값일 확률을 나타내고 있다.

#### IV. 결론

MLC는 multi-level의 데이터가 처리되는 특성을 가지고 있으며 이러한 이유로 코딩 처리 시간의 단축에 있어서 기존의 콘볼루션코드방식에 비해 우수하다 할 수 있다. 결론적으로 코딩 처리 속도를 향상시키면서, 기존의 콘볼루션코드와 비교하여 크지 않은 성능차이를 보임을 시뮬레이션 결과로 알 수 있었다. 또한 ACA 블록은 Tuple 단위로 multi-level의 값을 이용하여 코딩을 처리하게 되며, 이러한 코딩 기법은 multi-level 값을 이용하여 코딩하기 때문에 전체적인 입력 데이터들에 대한 코딩 시간을 절약할 수 있다는 장점을 가지고 있다. MLC 방식의 특성은 다음과 같이 요약 할 수 있다.

- 코딩 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 코딩 처리를 할뿐만 아니라 소프트웨어상으로 구현이 간단하다.

본 연구는 비교적 단순한 ACA block를 이용하여 기존의 콘볼루션코드와 성능을 비교 분석하였다. 차후 연구 과제로 ACA block에 기존의 RS 개념을 도입하여 iteration 복호가 가능한 구조를 설계한다면, 속도 적인 측면뿐만 아니라 성능에서도 우수한 부호기를 설계하게 될 것이다. 그리고, 복호과정에서 SOVA(Soft-Output Viterbi Algorithm)와 같은 soft 복호개념의 적용가능성에 대하여 연구할 것이다.

#### V. 참고문헌

- [1] Stephen B. Wicker, "Error Control Systems for Digital Communication and Storage", School of Electrical and Computer Engineering Georgia Institute of Technology, pp. 264-355, 1995
- [2] C. E. Shannon, " A Mathematical Theory of Communication", Bell System Technical Journal, VOL. 27, pp. 379-428, July. 1948
- [3] Floyd M. Fardner, John D. Baker, " Models of Communication Signals and Process", JOHN WILEY & SONS, Inc, pp. 261-273
- [4] Rodney F. W. Coates, Gareth J. Janacek, " Monte Carlo Simulation and Random Number Generation", IEEE Journal on Selected Areas in Communications, VOL. 6, NO.1, pp 58-66, January, 1988