디지탈통신 코딩의 경향

-어제, 오늘과 미래-

康 昌 彦
연세대학교 전자공학과 교수

1. 序 論

현대 산업 사회에서는 대규모의 고속 처리 기능의 발달과 더불어 효율적이고 신뢰성있는 데이터전송 및 저장(storage)시스템에 대한 필요성이 크게 대두되고 있다.

통신 시스템에서는 디지탈 데이터가 통신 채널을 통하여 전송될 때 수신 데이터에 에러가 발생하게 된다. 에러 정정코드는 에러에 아주 민감한 데이터에 대하여 에러제어(control) 코드로서 수신 데이터의 에러 허용치만큼 에러를 줄여서 디지탈 데이터를 보호하는 것을 목적으로 한다.

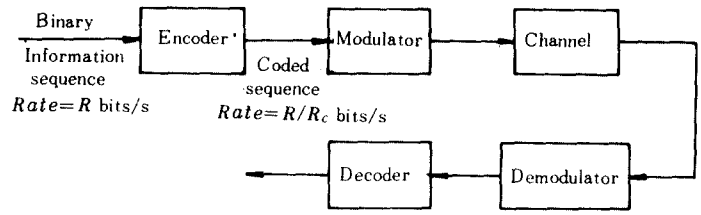
에러제어코드는 1948년 Claude Shannon에 의해 기본적인 이론이 제창되었는데, Shannon은 확률 개념을 도입하여 채널의 용량 C가 전송비율(transmission rate) R보다 크면 채널이나 저장 매체에 의한 출력 에러확률이 작아져서 에러제어 코드를 사용한 효율적인 통신 시스템을 구성할 수 있다는 것을 보여 주었다.

1950년대 이후, 코드는 크게 block코드와 convolutional코드로 나누어질 수 있게 되었는데, block코드는 1950년에 Hamming에 의해 최초로 소개되었다. 이 기간에는 한개의 에러정정코드가 개발되었고, 1959년과 1960년에는 Bose, Chaudhuri, Hocquenghem이 순환 코드(cyclic code)의 일종인 다수 에러정정코드(BCH 코드)를 개발하였다. 또한 1960년에는 Reed와 Solomon에 의하여 nonbinary코드인 Reed-Solomon 코드가 만들어졌다. 그 이후 encoder와decoder를 실현하기 위한 하드웨어와 소프트웨어의 실질적인 이론이 연구되어져 왔는데, 처음에는 Peterson에 의해 에러의 위치를 알아내는 효율적인 algorithm이 개발되어졌고 Berlekamp와 Massey 등에 의해 확장되어져서 새로운 디지탈 기술에 실제로 적용할 수 있게 되었다.

Block코드와 달리 확률적인 차원에서 encoding과 decoding방식을 연구하게 되었는데, 그 대표적인 것이 sequential decoding방식이다. Sequential decoding은 무한장(length)의 non-block 코드로서 tree로 표현될 수 있고 tree algorithm으로 decoding될 수 있으며, 가장 유용한 tree코드가 convolutional 코드이다. 이 convolutional코드는 1955년 Elias에 의해 소개되었고, 1950년 후반에 sequential decoding algorithm으로 성공적으로 decoding되었다. 1967년에는 더 간단한 algorithm인 Viterbi algorithm이 개발되어졌고, 오늘날 convolutional코드의 decoding방식으로 주로 이용되고 있다.

1970년대에는 block 코드와 convolutional코드를 조합하기 시작하였고, 1980년대에는 디지탈 통신시스템과 디지탈 저장 시스템에 encoder와 decoder가 이용되기 시작했다.

채널 encoding과 decoding을 행한 디지탈통신 시스템의 계통도는 다음 그림 1과 같다.



〈그림 1〉 채널 코딩을 행한 디지탈 통신시스템
Model of digital communication system with channel coding.

본 원고에서는 전송에러를 검출하고 정정하는 여러가지 에러제어 코드의 대략의 특성과 그 응용분야에 대하여 논하고자 한다.

2. 선형 block 코드

Block코드는 코드단어(code word)라 일컫는 고정된 길이 벡터의 집합으로 구성되고, 코드단어의 길이는 벡터내의 element의 갯수 n으로 표시된다. 그리고 element는 q개의 element로 표현되는데, 이때 q가 2이면 binary 코드, q > 2이면 nonbinary코드이다.

길이 n의 binary block 코드에서는 2^n개의 코드단어가 존재하게 되고, K개의 정보 비트는 길이 n의 코드단어로 mapping될 수 있으므로 보통 block 코드는 (n, k)코드로 표현되고, k/n는 코드율(code rate)로서 정의된다.

코드율외에 코드 단어에 대한 중요한 상수로서 코드단어가 포함하는 nonzero element의 수인 weight가 있다.

만약 Ci와 Cj를 (n, k)block코드의 코드단어라고 할 때 두 코드단어 사이의 차이를 Hamming거리(distance)라고 하고 dij로 표시하며 dij는 $0 < dij \leqq n$의 조건을 만족하게 된다. 코드단어에 대한 $|dij|$의 최소값을 코드의 최소거리 $d$min이라 한다.

선형코드는 zero코드단어를 가지고, α_1과 α_2라는 element에 대하여 $\alpha_1 Ci \oplus \alpha_2 Cj$ 또한 코드단어가 되는 코드이다. 이때 weight분포는 코드의 거리와 같은 특성을 가지므로 코드의 최소 거리 $d$min은 다음과 같다.

$$d_{\min} = \min_{r,\ r \neq 1} |w_r| \tag{1}$$

여기서 W는 weight를 나타내고, r는 정수이다.

Block코드는 위성통신 및 우주통신, 전자계산기의 기억장치에서의 에러검출 및 정정에 사용된다.

이제 이러한 선형 Block코드에 대한 특성들을 알아보기로 한다.

1. 생성행렬(generator matrix) 및 parity check행렬

$x_{m1}, x_{m2}, \cdots x_{mk}$를 정보비트라 할 때 encoder에 입력되는 정보 벡터는

$$\boldsymbol{X}_m = [x_{m1}, x_{m2}, \cdots x_{mk}] \tag{2}$$

이고, encoder의 출력 벡터는 다음의 식(3)과 같다.

$$\boldsymbol{C}_m = [c_{m1}, C_{m2}, \cdots c_{mn}] \tag{3}$$

encoding 과정은

$$\boldsymbol{C}_m = \boldsymbol{X}_m \boldsymbol{G} \tag{4}$$

으로 표현될 수 있는데, 이때 G를 코드의 생성행렬이라고 한다.

$$\boldsymbol{G} = \begin{bmatrix} g_1 \\ g_2 \\ \vdots \\ g_k \end{bmatrix} = \begin{bmatrix} g_{11}\ g_{12} \cdots g_{1n} \\ g_{21}\ g_{22} \cdots g_{2n} \\ \vdots\ \vdots \qquad \vdots \\ g_{k1}\ g_{k2} \cdots g_{kn} \end{bmatrix} \tag{5}$$

이 (n, k)코드의 생성행렬은 체계화된 형태(systematic form)로 다음과 같이 줄여질 수 있다.

$$G=[\mathrm{II}_k \vdots P]=\left[\begin{array}{ccc|c} 1 & 0\cdots & 0 & P_{11}P_{12}\cdots P_{1n-k} \\ 0 & 1\cdots & 0 & P_{21}\cdots\cdots P_{2n-k} \\ \vdots & & & \\ 0 & 0\cdots & 1 & P_{k1}\cdots\cdots P_{kn-k} \end{array}\right] \quad (6)$$

II_k는 $k\times k$ identity행렬이고, P는 $k\times(n-k)$ 행렬로서 parity check bit를 나타낸다. 체계화된 형태의 생성행렬에 의해 생성된 (n, k)코드를 체계화된 코드라고 한다.

이 두가지 형태의 생성행렬에 의한 코드는 동일하므로 모든 선형(n, k)코드는 (n, k)선형 체계화된 코드로 나타내어질 수 있다.

(n, k)코드의 dual코드는 $(n, n-k)$코드로서 생성 행렬을 H라고 할 때 다음과 같은 식이 성립된다.

$$\begin{aligned} &C_m H'=0 \\ &GH'=0 \end{aligned} \quad (7)$$

여기서 H를 parity check행렬이라고 한다. H는 $(n-k)$차이므로 다음과 같은 최소거리 dmin의 상한선을 가지게 된다.

$$d\min\leqq n-k-1 \quad (8)$$

2. 순환코드(cyclic code)

순환코드는 순환shift특성을 가지는 코드이다. 만약 $C=[C_{n-1}, C_{n-2}, \cdots, C_0]$가 순환코드의 코드단어라면 $[C_{n-2}, C_{n-3}, C_{n-4}, \cdots, C_0, C_{n-1}]$ 또한 코드단어인 특성을 가지므로 C의 순환 shift한 것들도 모두 코드단어가 된다.

순환코드에 대하여 코드단어 C는 $(n-1)$차 이하의 다항식 $C(p)$로 표현하는 것이 편리하다. 이때 $C(p)$는

$$C(p)=C_{n-1}p^{n-1}+C_{n-2}p^{n-2}+\cdots+C_1p+C_0 \quad (9)$$

이고, i번 shift한 코드단어 $C_i(p)$는

$$P^iC(p)=q(p)(p^n+1)+C_i(p) \quad (10)$$

에서 $P^iC(p)\bmod(p^n+1)$로 주어진다.

다항식 $g(p)$가 $(n-k)$차 다항식이고 (p^n+1)의 factor일 때, $(k-1)$차 이하의 $x(p)$는

$$x(p)=x_{k-1}p^{k-1}+x_{k-2}p^{k-2}+\cdots+x_1p+x_0 \quad (11)$$

로 나타나고 코드단어는 다음의 식(12)와 같이 쓸 수 있다.

$$C_m(p)=x_m(p)g(p) \quad (12)$$

(단, $m=1, 2, \cdots, 2^k$)

그러므로 다항식 $g(p)$를 코드의 생성 다항식이라 하고, 순환코드는 n차 벡터 space의 subspace가 된다.

이러한 순환생성 다항식을 체계적인 형태로 표현해 보면,

$$p^{n-l}=g_l(p)g(p)+R_l(p) \quad (13)$$

에서 $l=1, 2, \cdots, k$일 때 $p^{n-l}+R_l(p)$가 순환코드의 코드단어가 됨을 알 수 있다.

(1) 순환 Hamming코드

Binary순환 Hamming코드의 특성은 정수 m에 대하여 다음과 같다.

$$\begin{aligned} &n=2^m-1 \\ &K=2^m-1-m \end{aligned} \quad (14)$$

순환 Hamming코드의 parity check matrix H의 열들은 서로 선형 독립이고, (n, k)Hamming코드에 대하여 최소거리 dmin은 3이 된다.

(2) 순환 Golay코드

Golay코드는 다수 에러정정 binary코드로서, 실제로 사용되는 것으로는 dmin=7인 (23, 12)코드와 dmin=8인 (24, 12)코드가 있다.

선형 (23, 12)Golay코드는 다음 식(15)와 같은 생성 다항식 $g(p)$에 의하여 순환코드가 된다.

$$g(p)=p^{11}+p^9+p^7+p^6+p^5+p+1 \quad (15)$$

이 코드는 순환코드의 에러정정 능력을 t라고 할 때, t개를 정정할 수 있는 완전코드(perfect code)이다.

(3) BCH코드

BCH코드는 순환코드 중에서 에러정정 능력 t가 가장 큰 코드이다.

BCH 코드의 상수는

$$\begin{aligned} &n=2^m-1 \\ &n-k\leqq mt \\ &d\min=2t+1 \end{aligned} \quad (16)$$

이 되고, 생성 다항식은 p^{2m-1}의 factor로서 구할 수 있다.

이 BCH코드는 한개의 에러를 정정할 수 있는 Hamming코드를 다수 에러정정 코드로 확장시킨 것으로서, 비교적 간단하면서도 여러개의 에러를 정정할 수 있다는 장점이 있으므로 통신위성에 사용된다.

3. Nonbinary block코드와 concatenated block코드

Nonbinary block코드는 코드단어의 element가 $q(q=2^k)$개의 symbol로 이루어진 코드이다. Nonbinary코드단어의 길이를 N, 정보 symbol의 수를 K, 최소거리를 D_{min}으로 표기한다.

(1) Reed-Solomon코드

여러가지 nonbinary block코드 중에서 가장 실용적인 코드인 Reed-Solomon 코드에 대하여 알아보면, Reed-Solomon 코드는 BCH코드의 subset으로서 다음과 같은 상수를 가진다.

$$N=q-1=2^k-1$$
$$K=2^k-2t-1 \quad (K=1, 2, \cdots, N-1) \tag{17}$$
$$D\min=N-K+1$$

그러므로 에러정정 갯수 t는

$$t=\left\lfloor \frac{D_{min}-1}{2} \right\rfloor = \left\lfloor \frac{N-K}{2} \right\rfloor \tag{18}$$

이고, t개의 symbol에러를 정정할 수 있다.

Reed-Solomon코드를 실제로 이용하는 이유는 최소거리가 크다는 점과 코드 길이가 긴 코드를 효율적으로 실현하는 것이 가능하다는 점 때문이다. Reed-Solomon코드는 위성통신이나 컴퓨터 기억소자간의 통신에 이용된다.

(2) Concatenated block 코드

Concatenated코드는 코드길이가 길고 powerful한 코드를 얻기 위하여 두개의 코드를 조합한 코드이다.

보통 한 코드는 nonbinary 코드이고, 다른 코드는 binary코드를 사용하는 것으로서, 다음 그림과 같은 통신 시스템 계통도를 가진다.

외부 코드로는 (N, K)nonbinary코드인 Reed-Solomon코드를 사용하고, 내부코드로는 binary코드인 block코드 또는 convolutional코드를 사용한다.

Concatenated코드의 block길이는 N_n이 되고, 정보비트는 Kk가 되어 (nN, kK)의 긴 binary코드를 얻게 된다. 또한 최소거리는 외부코드의 최소거리가 $D\min$이고, 내부코드의 최소거리가 $d\min$일 때 $D\min\ d\min$으로 두 코드의 최소거리의 곱으로 나타난다.

Concatenated코드는 random에러와 burst에러가 동시에 발생하는 채널이나 에러가 많이 발생하는 채널에 효율적으로 사용될 수 있으며, 긴 코드를 구성하는 경우에 코딩 시스템의 복잡성을 줄일 수 있으므로 효율적이다.

4. Performance

시스템을 평가하는 기준인 performance는 soft decision decoding과 hard decision decoding할 때 에러 확률로서 구해질 수 있는데, hard decision decoding이 행해지는 것이고 soft decision decoding은 양자화되지 않은 신호를 decoding하는 것이다. 계산상의 복잡성을 줄이기 위하여 보통 Analog신호를 Digital화한 후 decoding을 수행하게 되는 hard decision decoding방식에 대하여 설명하다

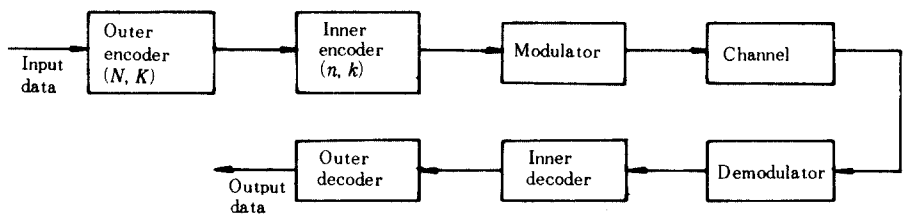
우리가 선택한 채널을 채널에러확률 p를 가진 BSC라고 할 때 다음과 같은 코드단어의 에러확률 P_M을 얻을 수 있다. BSC(Binary Symmetric Channel)은 memoryless채널이므로 비트에러는 독립적으로 나타나게 되고 n비트의 block에서 m개의 에러가 발생할 확률 $P(m, n)$은

$$P(m, n)=\binom{n}{m}P^m(1-P)^{n-m} \tag{19}$$

이 된다. 그리고 코드단어의 에러확률 P_M은

$$P_M \geqq \sum_{m=t+1}^{n} P(m, n) \tag{20}$$

이 되고, t개의 에러를 정정할 수 있는 코드는



〈그림 2〉 Concatenated 코드를 사용한 통신 시스템

Communication system employing a concatenated code.

$$2^k\sum_{i=0}^{t}\binom{n}{i}\leqq 2^n \qquad (21)$$

의 조건을 만족해야 한다.

Hard decision decoding방식보다 soft decision decoding방식이 performance는 좋지만, 계산이 복잡하므로 hard decision decoding방식을 이용한다.

3. Convolntional 코드

Convolutional코드는 encoding시 출력인 코드 sequence가 현재뿐만 아니라, 구속장(constraint length)에 제한을 받는 이전의 입력들과도 선형 결합된 형태로 나타난다는 점에서 block코드와 구별된다. 실용적인 면에서 볼 때 하드웨어의 복잡도가 같은 정도라면 convolutional코드가 block코드보다 에러정정 능력이 크고, 특히 위성 및 우주통신 분야의 응용에서 convolutional코드가 block코드보다 매우 우수한 것으로 평가되고 있다.

일반적으로 convolutional코드의 정보 비트 K와 코드 sequence n은 그 수가 작으며 코드율도 낮다.

이 코드의 표현 방법으로는 기억 소자에 의한 표현법, 행렬에 의한 표현법, trellis에 의한 표현법, tree에 의한 표현법 등이 있으며, decoding방법으로는 다수결논리decoding(majority logic decoding), sequential decoding, Viterbi decoding 등이 있다.

다수결논리decoding은 그 구조적 표현이 기억소자에 의한 표현과 행렬에 의한 표현을 사용하므로 대수적인 특성을 지닌다.

반면에 sequential decoding은 그 구조적 표현이 기억소자에 의한 표현을 사용하므로 위상학적이며, Viterbi decoding 또한 trellis에 의한 표현을 사용하므로 위상학적인 특성을 지닌다.

1. 생성 행렬을 이용한 convolutional 코드의 표현 방법

(n, k)convolutional코드의 코드sequence는 반무한대의 행렬인 생성행렬 $\boldsymbol{G}_\infty$에 의하여 다음과 같이 얻어질 수 있다.

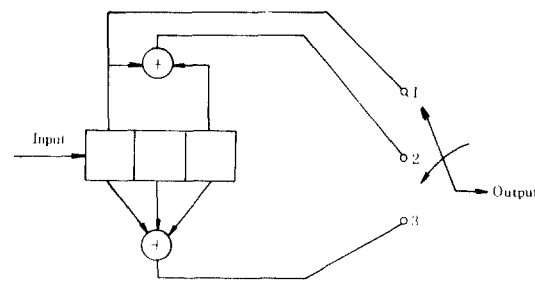
$$\boldsymbol{C}=\boldsymbol{M}\boldsymbol{G}_\infty \qquad (22)$$

이러한 표현 방법은 block코드와 같이 생성행렬에 의한 표현으로서, binary코드를 사용한다면 K정보 비트의 한 block이 입력될 때 구속장이 L인 encoder는 그때의 K정보 bit block뿐만 아니라 그 이전의 $(L-1)$정보 block들에 걸쳐서 선정된 각 기억소자들의 출력을 modulo-2가산기에 입력시켜 결합시킴으로써 출력이 k비트 보다 많은 n비트의 코드 sequence를 만든다.

생성행렬을 반무한대의 행렬로 표현하는 외의 다른 방법으로 n개의 modulo-2 가산기 각각을 vector로 하여 n개의 vector집합으로 나타내어 기능상에 있어서 등가인 표현을 사용한다. 각 vector는 Lk차원을 가지며, modulo-2 가산기와 encoder의 연결을 포함한다.

Vector에서 i번째 위치에 있는 1은 shift register의 해당 stage가 modulo-2가산기에 연결된 것을 가리키고, 0은 stage와 modulo-2 가산기 사이에 연결되지 않은 것을 가리킨다.

예를 들면, $L=3$, $k=1$, $n=3$인 convolutional코드의 encoder는 다음 그림 3과 같다.

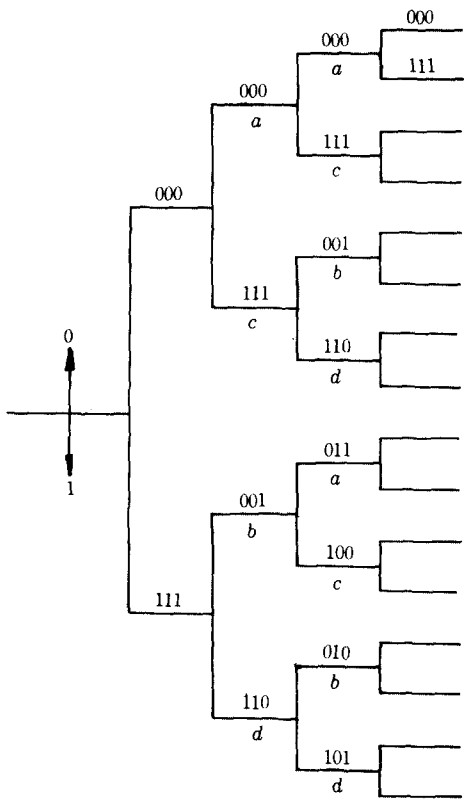


〈그림 3〉 $L=3$, $k=1$, $n=3$인 convolutional 코드의 encoder
$L=3$, $k=1$, $n=3$ convolutional encoder.

2. Tree diagram

초기에 encoder가, 모두 0인 상태라고 가정

했을 때, 어떤 특정한 입력 sequence에 따라 tree에서 특정 node로 가도록 주어진다고 가정하자. 이때 binary branching rule은 다음의 입력 비트가 1이라면 lower branch로 가고, 입력이 0이라면 node에서 upper branch로 가도록 하여 encoding을 수행하게 된다. 예를 들면, $L=3$, $k=1$, $n=3$인 convolutional코드의 tree diagram은 다음의 그림 4와 같다.

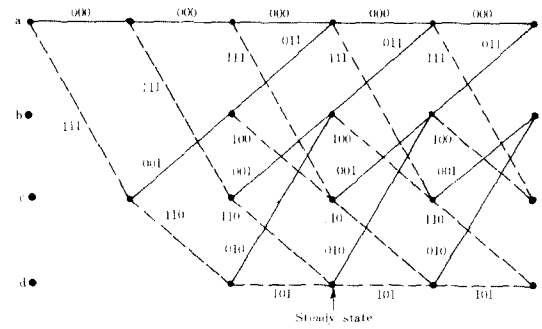


〈그림 4〉 $L=3$, $k=1$, $n=3$인 convolutional 코드의 tree diagram
Tree diagram for $L=3$, $k=1$, $n=3$ convolutional code.

각 stage에서의 출력 sequence는 입력 비트와 그 이전의 비트, 즉 shift register에서 처음 stage에 포함된 비트에 의해서 결정된다. Shift register의 마지막 stage에 있는 비트는 오른쪽에서 shift되어 나가서 출력에 영향을 미치지 않는다.

3. trellis diagram

$L=3$, $k=1$, $n=3$인 convolutional코드의 trellis diagram은 다음과 같다.



〈그림 5〉 $L=3$, $k=1$, $n=3$인 convolutional 코드의 trellis diagram code.
Trellis diagram for rate 1/3, $L=3$ convolutional code.

Tree diagram의 각 node를 shift register에서 가능한 4가지 상태에 상응하도록 명칭을 붙이면 tree의 세번째 단에서 a, b, c, d로 명칭이 된 node가 각각 2개씩 존재함을 알 수 있다.

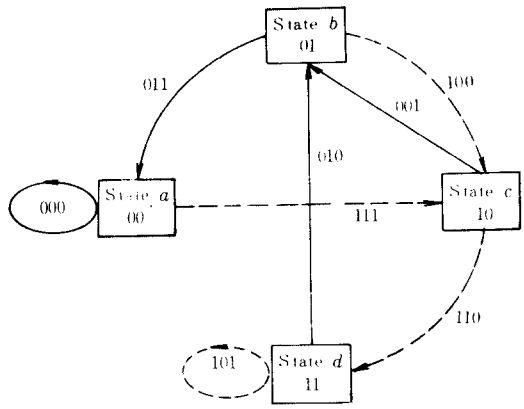
같은 명칭을 갖는 두 node에서 퍼져 나가는 모든 branch는 그것들이 같은 출력 sequence를 발생한다는 의미에서 서로 동일하다고 할 수 있다.

이것은 같은 명칭을 갖는 두 node는 하나로 합쳐질 수 있음을 의미하고, tree diagram에서 이와 같은 과정을 거치면 더욱 간결한 trellis diagram이 된다. Trell is diagram에서 굵은 실선은 입력 비트가 0일 때에 발생된 출력이고, 점선은 입력비트가 1일 때 발생된 출력 sequence를 나타낸 것이다.

4. State Diagram

Encoder의 출력은 입력과 encoder의 state에 의해 결정되므로 trellis보다 더욱 합축적인 state diagram이 요구된다. State diagram은 encoder의 가능한 state와 가능한 state전이(tra-

nsition)을 나타내는 그래프로서 다음 그림 6과 같다.



〈그림 6〉 $L=3$, $k=1$, $n=3$인 convolutional 코드의 state diagram

State diagram for rate 1/3, $L=3$ convolutional code.

그래프에서 점선은 입력비트가 1일 때이며, 굵은 실선은 입력비트가 0일 때를 나타낸다.

이 diagram에서 가능한 전이는 $a^0 \to a$, $a^1 \to c$, $b^0 \to a$, $b^1 \to c$, $c^0 \to b$, $c^1 \to d$, $d^0 \to b$, $d^1 \to d$ 이다. 여기서 $\alpha^1 \to \beta$는 입력비트가 1일 때 상태α에서 β로의 전이를 나타낸다.

일반적으로 코드율이 k/n, 구속장이 L인 convolutional코드는 tree diagram에서 각 node로부터 퍼져나가는 2^k개의 branch에 의해 특정지워 진다.

Trellis와 state diagram은 각각 $2^{k(L-1)}$개의 가능한 상태를 가지고, 각 상태로 입·출력되는 branch의 수는 2^k이다.

5. Decoding

Convolutional코드의 최적 decoding 방식으로는 viterbi algorithm이 있다. 이 viterbi algorithm은 간단한 decoding 기술로서, block코드의 decoding방식과 같이 수신된 코드 sequence와 2^k개의 모든 가능한 송신 sequence 사이의 거리, 즉 Hamming거리를 계산하여 수신된 코드 sequence에 가장 가까운 최소거리를 가지는 코드 sequence를 선택하는 방식이다. 이 방식은 또한 discrete memoryless채널에 대하여 최소 에러 확률을 가지게 되므로 적합하다.

이 viterbi algorithm은 maximum likelihood decoding 방식이지만, 긴 구속장일 때 계산상 복잡하고 많은 저장 매체를 요구하는 문제점이 있다.

Sequential decoding 방식은 확률적인 특성을 도입하여 tree 또는 trellis의 path 중에서 가장 확률이 큰 path를 선택하는 방식이다. 이 방식은 Fano가 제창한 것으로서 Fano sequential decoding algorithm은 여러 통신 시스템에 적용되고 있지만, viterbi algorithm과 에러 performance는 비슷하지만 적은 저장 매체를 필요로 하므로 큰 구속장을 가지는 convolutional코드에 적합하다. Sequential decoding방식은 viterbi algorithm에 비해 더 큰 decoding지연(delay)을 가지는 단점이 있다.

이러한 거리를 계산하는 방식 이외에 수신된 sequence의 syndrome을 계산하여 에러를 정정하는 방식이 있는데, 이것은 feedback decoder로서 다수결 논리 decoder 또는 threshold decoder가 있고 block코드의 decoding방식과 유사하다.

6. Performance

시스템을 평가하는 기준인 performance는 block코드의 경우에서 처럼 송신된 정보에 대하여 수신측에서 에러를 정정하지 못할 비트 에러확률 P_E로서 구할 수 있다.

여기서 채널을 BSC라고 하고 비트 에러가 서로 독립적으로 발생하며 채널의 에러확률을 p라고 할 때, 거리 d가 기수(odd number)라면 잘못된 path를 택할 확률 $P_2(d)$는 다음과 같다.

$$P_2(d) = \sum_{k=(d+1)/2}^{d} \binom{d}{k} P^k (1-P)^{d-k} \qquad (23)$$

그리고 거리 d가 우수(even number)라면 $P_2(d)$는

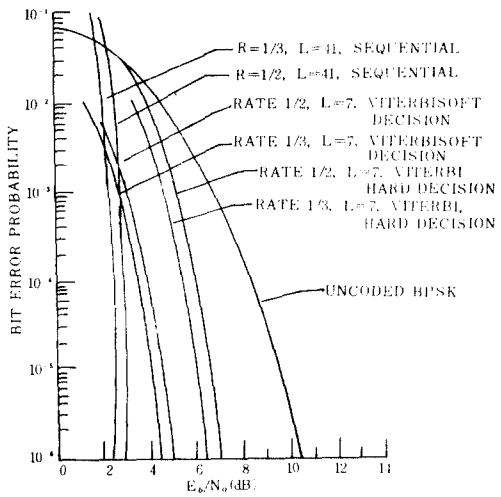
$$P_2(d) = \sum_{k=\frac{d}{2}+1}^{d} \binom{d}{k} P^k (1-P)^{d-k}$$

$$+\frac{1}{2}\binom{d}{d/2}P^{d/2}(1-P)^{d/2} \qquad (24)$$

이다. 그러므로 에러를 정정하지 못할 확률 P_E는 모든 가능한 path에서의 에러확률 $P_2(d)$의 합으로써 나타낼 수 있다.

$$P_E < \sum_{d=d_{free}}^{\infty} a_d P_2(d) \qquad (25)$$

여기서 $\{a_d\}$는 거리 $\{d\}$에 해당하는 path의 수를 나타내는 상수이다.



〈그림 7〉 Viterbi decoding과 sequential decoding의 performance

Performance of viterbi decoding and sequential decoding.

Viterbi decoding은 저장 매체가 구속장에 대하여 지수적으로 증가하므로 보통 구속장 $L=10$정도일 때 사용되고, sequential decoding에서는 구속장이 $L=40$ 정도로 크게 된다.

다음의 그림 7은 이러한 특성을 가지는 rate가 1/2과 1/3인 viterbi decoding과 sequential decoding의 performance이다. 여기에서 viterbi decoding의 구속장은 $L=7$이고, sequential decoding의 구속장은 $L=41$이다.

그림에서 알 수 있듯이 viterbi decoding은 속도는 빠르지만 sequential decoding보다 performance는 나쁘다.

그리고 sequential decoding에서 $P_B=10^{-5}$에 대하여 코딩 이득은 약 7[dB]가 된다.

4. Block 코드와 Convolutional 코드의 비교

Block코드에 대한 이론과 설계는 convolutional코드에 비하여 더 오래되었고 풍부하다.

코딩을 하지않은 BPSK시스템과 코딩을 행한 시스템의 에러확률을 10^{-5}~10^{-8}으로 했을 때, 차이점을 비교해 보면 다음의 표 1과 같다.

Data rate capability가 low라면 10[Kbps]이하를 말하고, moderate는 10[Kbps]에서 1[Mb

〈표 1〉 BPSK와 코딩기술의 비교

Comparison of coding techniques with BPSK

Coding Technique	Coding gain (dB) at 10^{-5}	Coding gain (dB) at 10^{-8}	Data rate capability
Concatenated (RS and Viterbi)	6.5~7.5	8.5~9.5	Moderate
Sequential decoding (soft decisions)	6.0~7.0	8.0~9.0	Moderate
Block codes (soft decisions)	5.0~6.0	6.5~7.5	Moderate
Concatenated (RS and short block)	4.5~5.5	6.5~7.5	Very high
Viterbi decoding	4.0~5.5	5.0~6.5	High
Sequential decoding (hard decisions)	4.0~5.0	6.0~7.0	High
Block codes (hard decisions)	3.0~4.0	4.5~5.5	High
Block codes-threshold decoding	2.0~4.0	3.5~5.5	High
Convolutional codes-threshold decoding	1.5~3.0	2.5~4.0	Very high

ps]를, high는 1[Mbps]에서 20[Mbps], very high는 20[Mbps]이상을 나타낸다.

다소 복잡할지라도 moderate와 high data rate에서는 viterbi decoding방식을 이용한 convolutional코드가 가장 유력하다.

Very high data rate에서는 Reed-solomon코드와 짧은 block코드를 concatenation한 코드가 viterbi decoding방식을 사용했을 때보다 덜 복잡하면서도 같은 코딩 이득을 가진다.

High speed에서 더 큰 코딩 이득을 가진 시스템에서는 sequential decoding이 유력하고, moderate data rate에서는 sequential decoding중 soft decision decoding을 행하는 것이 유력하다.

TDMA방식을 요구하는 시스템에서는 block코드가 더 유력하고 threshold decoding은 복잡하지 않고 very high speed에서 동작하는 시스템에 적합하다.

이러한 비교는 오늘날 디지탈 IC기술에 의하여 영향을 받아왔고, 이 IC기술의 진전은 시스템의 복잡성과 data rate의 개선을 하였다. 그리고 여러가지 상황에서 에러 performance를 개선할 수 있는 코드의 선택을 가능하게 해 줄 것이다

5. 응용분야 및 연구 진행 방향

Block코드는 주로 디지탈 통신계와 전자계산기 계통에 이용되어져 왔는데, 디지탈 통신계에 이용된 몇가지 예를 들어 보면 다음과 같다. 미연합전략 정보망(JTIDS)에는 (31, 15) Reed-Solomon코드를 이용하였고, INTELSAT V통신위성에는 (127, 112)BCH코드, 미공군 위성통신에는 (7, 2)Reed-Solomon코드를 이용하였으며, 1972년과 1975년에 Mariner 우주선에서 화성으로부터의 영상을 전송하는데 (32, 6)Reed-Muller코드를 이용하였다.

전자계산기 계통에서는 기억장치에서 발생하는 에러의 검출 및 정정을 위하여 block코드를 사용하였는데, magnetic core 주기억장치에서 에러정정을 위하여 1961년 IBM7030에서는 두개의 에러를 검출하고 하나를 정정할 수 있는 Hamming코드를 사용하였다. 1970년대에는 신속하고 정확한 정보처리를 위하여 IBM370에서는 (72, 64) Hamming코드를 이용하였다.

그리고 burst에러를 정정하기 위하여 fire코드를 사용하였고, IBM3370 시스템에서는 GF(2^8)의 원소를 사용한 Reed-Solomon코드를 이용하였다.

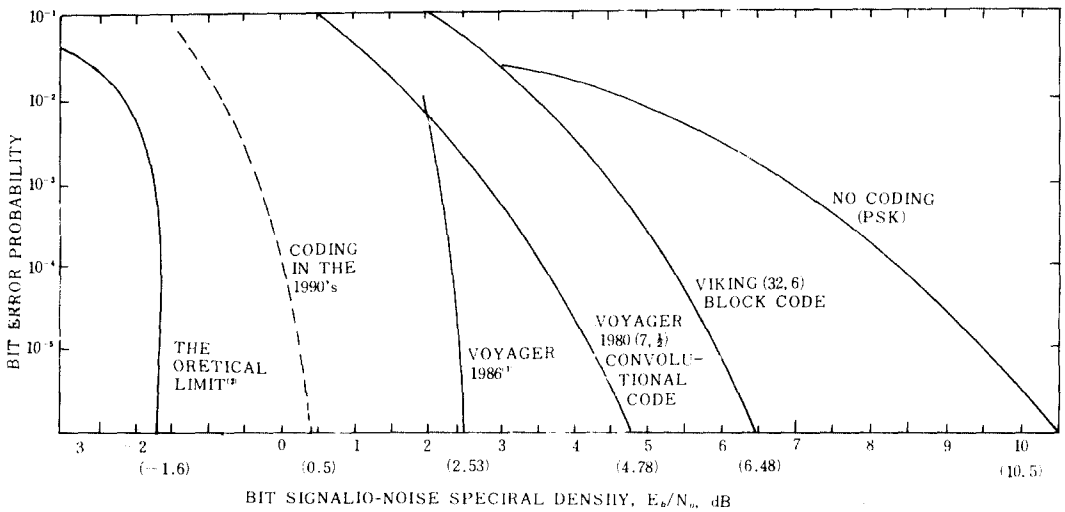
Block코드는 1980년대 이후로는 우주 통신에서 코딩 시스템의 일부분으로 concatenational되어 사용되고 있다.

Convolutional코드는 최근에 우주 및 위성통신에 널리 응용되고 있다. 1969년 INTELSAT Ⅳ 통신위성에는 $L=146$인 (8, 7)self orthogonal convolutional코드에 다수결 논리 decoding 방식을 적용하여 사용하였다.

미항공우주국(NASA)의 pioneer 9호에는 $L=25$의 (2, 1)체계화된 convolutional코드를 사용하였고, pioneer 10, 11, 12호는 $L=32$의 (2, 1)convolutional코드를 사용하였으며 sequential decoding방식을 채택하였다. 그리고 1977년 Voyager 우주탐사계획에는 $L=7$인 (2, 1), (3, 1) convolutional 코드를 maximum likelihood decoding, 즉 viterbi decoding방식을 채택하여 사용하였다.

현재 우주통신에서 사용되는 코드의 대부분은 viterbi decoding 방식을 적용한 구속장 $L=7$이고 rate 1/2인 convolutional코드이다. 미국방위위성 통신망(DSCS)에서도 (2, 1)convolutional코드를 사용중이고 1980년대 이후의 Voyager 우주계획에는 viterbi decoding 방식을 채택할 예정이다. 이때의 코딩 이득은 5×10^{-3}의 에러 확률에 대하여 3.5[dB]이다.

앞으로는 영상전송과 telemetry에는 약 10^{-6}의 에러 확률이 요구되므로, 1986년 천왕성과 1989년 해왕성으로부터의 Voyager우주탐사 계획에는 16문자의 에러를 정정할 수 있는 (255, 223) Reed-Solomon코드와 Viterbi decoding방식을 이용한 convolutional코드를 concatenation하여 적용할 예정이다. 이 코드는 10^{-6}에러 확률에 대하여 8[dB]의 코딩 이득을 가지고, 다



(1) $\left(7, \frac{1}{2}\right)$ CONVOLUTIONAL CODE (VITERBI DECODING) CONCOTENATED WITH A REED-SOLOMON (255, 223) OUTER CODE.
(2) INFINITE BANDWIDTH EXPANSION

〈그림 8〉 코딩을 행한 telemetry의 발달과정
Progress of coded telemetry.

음 그림 8에서는 여러 가지 계획들에 대한 코딩 이득을 나타내었다.

우리나라에서도 방송위성을 띄울 계획이 있으므로 coding의 필요성이 대두되었고, 다수 에러 정정을 위한 BCH 코드와 Reed-Solomon코드에 대한 연구가 진행되고 있다.

우주 및 위성통신에 널리 사용되는 convolutional코드와 더 나아가서 concatenated코드에 대한 연구가 계속되어져야 하고, 코드율을 높이면서 다수 에러를 정정할 수 있는 코드에 대한 연구가 진행되어야 하겠다. 그리고 신속하게 정보를 처리할 수 있는 방법에 대한 연구가 진행되어 우주 및 위성통신에 기여할 수 있도록 하여야 겠다.

참 고 문 헌

(1) S. Lin, Error Control Coding, Prentice-Hall, N. J., 1983.
(2) W. W. Peterson, E. J. Weldon, Error Correcting Codes, M. I. T. Press, Massachusetts.
(3) R. E. Blahut, Theory and Practice of Error Control Codes, Addison-Wesley, N. Y. 1983.
(4) G. D. Forney, Concatenated Codes, M. I. T. Press, Massachusetts, 1966.
(5) A. J. Viterbi, J. M. Omura, Principles of Digital Communication and Coding, McGraw-Hill, Inc, 1979.
(6) P. Elias, "Error Free Coding," IRE Trans. on Information Theory, PGIT-4, pp. 29-37, 1954.
(7) D. D. Falconer, "A hybrid sequential and algebraic decoding scheme", Ph. D. dissertation, Dep. Elec. Eng., M. I. T., Cambridge, 1966.
(8) J. P. Odenwalder, "Optimal Decoding of Convolutional Codes," Ph. D. dissertation, Dep. Elec. Eng., Univ. California, Los Angels, 1970.
(9) H. O. Burton, E. J. Weldon, Jr., "Cyclic Product Codes", IEEE Trans. Information Theory, IT-11, pp. 433-440, 1965.
(10) N. M. Abramson, "Cascade Decoding of cyclic Product Codes", IEEE Tran. on Communication Technology, COM-16, pp. 398-402, 1968.

(11) E. R. Berlekamp, Algebraic Coding Theory, McGraw-Hill, 1968
(12) I. S. Reed, G Solomon, "Polynomial Codes over certain Finite Fields", J. Soc. Indust. Appl. Math., vol. 8, No. 2, June, pp. 300-304, 1960.
(13) T. Kasami, S. Lin, W. W. Peterson, "Polynomial Coder", IEEE Trans. on Information Thery, vol. 14, No. 6, November, pp. 804-814, 1968.
(14) R.C. Singleton, "Maximum distance Q-Nary codes", IEEE Trans. on Information Theory, April, pp. 116-118, 1964.
(15) G. D. Forney, "Generalized Minimum Distance Decoding", IEEE Trans. on Information Theory, vol. IT-12, No. 2, April, pp. 125-131, 1966.
(16) F. Jelinek, "Fast Sequential Decoding Algorithm Vsing a Stack," IBMJ. RES. DEVELOP, November, 1969.
(17) K. Zigangirov, "Some Sequential Decoding Procedures", Probl. Peredachi Information, 2, pp. 13-25, 1966.
(18) J. L. Massey, Threshold Decoding, M. I. T. Press., Mass, 1963.
(19) P. Elias, "Coding for Noisy Channels", IRE Conv. Rec., Pt. 4, pp. 37-46, 1955.
(20) J. P. Robinson, "Error Propagation and Definite Decoding of Convolutional Codes", IEEE Trans. on Information Theory, Vol. 14, pp. 121-128, Jan., 1968.

〈용 어 해 설〉

Real Time Network (실시간 네트워크) : 실시간 네트워크는 단말부속 시스템, 전송 네트워크, 그리고 어떤 수의 이용자들이 주어진 시간과 주어진 간격 내에 그들의 요구를 동시에 충족시켜 줄 수 있는 부속 시스템의 처리과정으로 구성된다.
예 : 항공 또는 철도예매 시스템 : 수 초의 지연이 있다.
주문수신 시스템 : 수 초의 지연이 있다.
급여전달 시스템 : 주 또는 월 간격으로 수신간의 지연이 있을 수 있다. ⇨네트워크

Remote Data Processing (원격 데이터처리) : 모든 데이터 처리와 전기통신 기술이 전산화된 장치가 정보교환을 가능하게 한다. ⇨네트워크

Switch (교환) : 교환기는 회로에서 특히 전송 시작 전의 데이터 전송 네트워크에서 하나 또는 그 이상의 접속을 이루거나 끊어 주는 기기이다. 교환기는 여러 방법으로 조작된다. 수동식, 기계식, 전자식 또는 네트워크의 연결 부분에 위치한 마이크로컴퓨터에 의한 방법들이 있다. ⇨회선교환, 메시지 교환, 패킷 교환

Telecopier (전송 복사기) : 영상(글로 쓰인 내용, 도표, 사진)의 도식적 원격재생을 가능하게 해 주는 기계. 전송복사 장치는 전송자 속에서 영상의 전송은 광학적 기기에 의해 주사된다. 수신자 속에서 원 영상과 비슷한 영상은 또 다른 주사기기에 의해 사진감응이나 전자감응이 종이로 재현된다. 전송은 애널로그 신호나 디지탈 신호를 이용한다. 전화복사는 표준전화 네트워크를 이용한다. ⇨사무절차