

길쌈 부호 복원 및 성능 분석

준회원 이재환*, 이현**, 강인식*, 정회원 윤상범**, 박철순***, 종신회원 송영준*

Recognition of Convolutional Code with Performance Analysis

Jae-hwan Lee*, Hyun Lee**, In-Sik Kang* *Associate Members*, Sang-bom Yun**,
Cheol-sun Park*** *Regular Members*, Young-joon Song* *Lifelong Member*

요약

통신, 컴퓨터를 포함한 많은 분야에서는 디지털 데이터를 신뢰성 있게 전달하기 위하여 채널 부호화를 사용하게 된다. 채널 부호화 된 신호가 전송 되었을 때 수신 측에서 채널 부호화 기법과 생성 파라미터를 알지 못한다면 복호는 어렵게 된다. 본 논문에서는 채널 부호화 기법 중 하나인 길쌈 부호(convolutional code)와 천공 패턴이 적용된 천공 길쌈 부호의 생성 파라미터 분석 방법을 제시한다. 또한 AWGN 채널 환경에서 시뮬레이션을 통하여 검증한다.

Key Words : 생성 파라미터, 길쌈 부호, 천공 패턴

ABSTRACT

The use of an error-correcting code is essential in communication systems where the channel is noisy. Unless a receiver has accurate channel coding parameters, it becomes difficult to decode the digitized encoding bits correctly. In this paper, we propose two algorithms for reconstructing convolutional codes: one for general convolutional codes and the other for punctured convolutional codes. And we also verify the algorithms by performing intensive computer simulation in additive white gaussian noise (AWGN) channel.

I. 서론

현대전에서는 전자전(EW: Electronic Warfare)이 전쟁의 승패를 좌우하는 매우 중요한 요인이 되고 있으며 특히 실시간 정보처리가 필수적인 요소로 부각되고 있다. 전자전이란 전자전 지원(ES: Electronic Warfare Support), 전자공격(EA: Electronic Attack), 그리고 전자 보호(EP: Electronic Protect)를 포함하는 전반적인 활동을 의미한다. 이 중에서 전자전 지원은 전자전의 가장 기본이 되는 부분으로 전장에

서 즉각적인 적 위협요소에 대한 탐지를 목적으로, 적에 의해 방사되는 전자파 에너지를 탐색→감청→방향 탐지→식별하여 필요한 정보를 생산하거나, 생산한 정보를 전자공격 활동에 제공하는 활동을 의미한다. 여기서 탐색은 감청 및 방향 탐지하고자 하는 적 전파의 주파수를 찾는 행위를 의미하는데 현대전에서는 광대역 주파수 범위의 위협신호를 탐지, 식별 및 전파 방해하는 기술이 매우 중요한 수단으로 사용되고 있다. 이러한 광대역 주파수 범위의 위협신호를 효과적으로 탐지, 식별하고 전파 방해를

* 금오공과대학교 전자공학부 모바일 통신 및 부호 연구실 (yjsong@kumoh.ac.kr) (° : 교신저자),

** LIG넥스원, *** 국방과학연구소

논문번호 : KICS2011-11-560, 접수일자 : 2011년 11월 30일, 최종논문접수일자 : 2012년 4월 16일

하기 위해서는 먼저 적의 전파 신호를 수신하고 수신된 신호를 복호하는 기술이 필요하다. 수신된 신호를 정확하게 복호하기 위해서는 수신 신호에 적용된 소스부호, 채널부호, 인터리빙 등 이러한 기법들에 대해 분석하고 분류할 수 있어야 한다. 여기서 채널 부호화 된 데이터에 대해 복호하기 위해서는 채널 부호의 생성 파라미터를 찾아내는 것이 중요하다 할 수 있다. 이처럼 수신 데이터로부터 사용된 채널부호의 파라미터를 추출하여 채널 부호를 복원하는 기법을 채널 부호의 복원 기법이라 한다. 본 논문에서는 채널부호의 한 종류인 길쌈 부호의 추정 기법에 대하여 알아본다.

길쌈 부호에서는 입력되는 정보 계열이 기억 소자를 통과하면서, 패리티 부분이 추가되어 부호어가 생성되는 구조를 가진다. 이때 한 번에 입력되는 정보 비트 수를 k , 출력 비트의 수를 n 으로 정의하고 부호화기에 사용되는 메모리의 개수를 m 이라고 하면 이러한 길쌈 부호를 (n, k, m) 부호라고 한다. (n, k, m) 부호는 $k \times n$ 개의 생성 수열로 정의되는데, 길쌈 부호의 경우 생성 수열에 모든 생성 파라미터가 포함되므로 생성 수열을 추정해 내는 것만으로도 복호에 필요한 정보를 대부분 알 수 있다 [1-3].

본 논문은 I장의 서론에 이어 II장에서는 길쌈 부호의 분석 방법에 대해 알아보고, III장에서는 천공 패턴이 추가된 천공 길쌈 부호의 분석 방법에 대해 알아본다. IV장에서는 시뮬레이션을 통해 제안된 방법의 실제 동작을 확인하고, 마지막으로 V장에서 결론을 맺는다.

II. 길쌈 부호의 분석

2.1. 길쌈 부호의 생성

길쌈 부호는 블록 부호와는 다르게 부호기가 메모리를 가지고 있어 입력되는 정보 비트와 메모리에 저장된 이전 입력 비트와의 관계에 의해 부호어가 생성된다. 이때 한 번에 입력되는 정보 비트의 수를 k , 출력 비트의 수를 n 으로 정의한다. 하나의 심벌이 입력되었을 때 k 개의 입력 비트는 각각 다른 시프트 레지스터로 입력된다. 따라서 길쌈 부호화기는 k 개의 시프트 레지스터를 가지게 되는데 이때 가장 큰 메모리를 가지는 시프트 레지스터의 메모리 개수를 m 이라 한다. 또한 $k=1$ 일 때, 출력 심벌의 생성에 영향을 미치는 비트의 수, 즉 $m+1$

을 구속장(constraint length)이라 하며 K 로 정의한다. 길쌈 부호는 이러한 값들을 이용하여 (n, k, m) 부호로 표시되며 부호의 부호율은 k/n 이다[2]. (n, k, m) 부호를 발생하기 위하여 (1)과 같이 $k \times n$ 개의 생성 수열을 사용하며 생성수열은 생성다항식으로 표현 가능하다[10].

$$\begin{matrix} g_{1,1}, g_{1,2}, \dots, g_{1,n} \\ g_{2,1}, g_{2,2}, \dots, g_{2,n} \\ \vdots \\ g_{k,1}, g_{k,2}, \dots, g_{k,n} \end{matrix} \quad (1)$$

여기서 $g_{i,j}$ 는 i 번째 입력에 의한 j 번째 출력을 결정하는 생성 수열이며 각각의 생성 수열은 i 번째 시프트 레지스터와 같은 크기를 가진다. 길쌈 부호의 출력 수열은 입력 수열과 생성 수열 간의 길쌈 연산으로 계산되며 입력 수열을 i_1, \dots, i_k 라 하면 출력 수열은 (2)의 c_1, \dots, c_n 와 같이 계산된다[1][3].

$$\begin{matrix} c_1 = i_1^* g_{1,1} + i_2^* g_{2,1} + \dots + i_k^* g_{k,1} \\ \vdots \\ c_n = i_1^* g_{1,n} + i_2^* g_{2,n} + \dots + i_k^* g_{k,n} \end{matrix} \quad (2)$$

여기서 ‘+’ 연산은 modulo-2 덧셈을 의미하고, ‘*’ 연산은 길쌈 연산을 의미한다. 길쌈 연산은 선형 시스템에서 가능하므로 길쌈 부호는 선형 부호가 되며 길쌈 부호의 부호어는 (3)과 같이 각 출력 수열을 비트단위로 조합하여 생성된다.

$$C = (c_1^{(0)} \dots c_n^{(0)}, c_1^{(1)} \dots c_n^{(1)}, c_1^{(2)} \dots c_n^{(2)}, \dots) \quad (3)$$

2.2. 길쌈 부호의 생성 수열 추정

일반적으로 채널 부호화된 데이터를 수신하였을 때 그 부호의 종류와 생성 파라미터를 모르는 상태에서는 복호를 수행하는 것은 매우 어렵다. 따라서 채널 부호에 대한 정보 없이 생성 파라미터를 추정하는 연구는 중요한 의미가 있다. 길쌈 부호의 경우 생성 수열에 모든 생성 파라미터가 포함되므로 생성 수열을 추정해 내는 것만으로도 복호에 필요한 정보를 대부분 알 수 있다. 길쌈 부호의 생성 수열 복원에 대해서는 이미 많은 연구가 진행되어 왔다 [4-7]. 대표적으로 Filiol는 수신 부호와 생성 수열의 관계식에 의하여 길쌈 부호의 생성 수열을 복원하는 방법을 제안했다[4]. 본 논문에서는 다항식 기반

으로 연구된 Filiol의 알고리즘을 응용하여 길쌈 연산을 이용한 검증식을 유도한다. 간단한 예로 부호율 $R=1/2$ 인 길쌈 부호의 부호기를 살펴보면 그림 1과 같다.

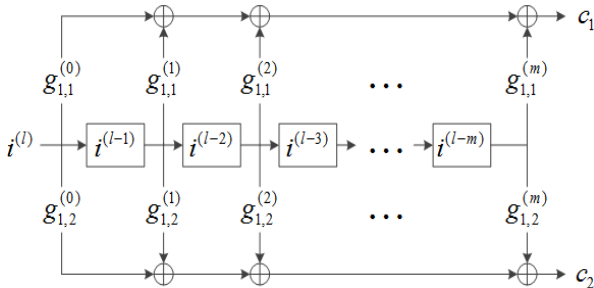


그림 1. 부호율 $R=1/2$ 인 경우의 길쌈 부호기.
Fig. 1. A rate $R=1/2$ convolutional encoder

그림 1에서 $i^{(l)}$ 은 현재 입력되는 정보 비트를 나타내며 $i^{(l-1)}, i^{(l-2)}$ 는 각각 현재 타이밍으로부터 첫 번째와 두 번째 이전에 입력된 정보 비트를 나타낸다. 만약 메모리가 0으로 초기화되어 있고 정보 비트가 모두 입력된 후 메모리가 0으로 채워질 경우 출력 수열 c_1 과 c_2 는 (4)와 같이 나타낼 수 있다.

$$\begin{aligned} c_1 &= i^* g_{1,1} = (c_1^{(0)} c_1^{(1)} c_1^{(2)} \dots) \\ c_2 &= i^* g_{1,2} = (c_2^{(0)} c_2^{(1)} c_2^{(2)} \dots) \end{aligned} \quad (4)$$

$g_{1,1}$ 과 $g_{1,2}$ 를 구하기 위하여 (4)의 c_1 에 생성 수열 $g_{1,2}$ 을 길쌈 연산 취하고 c_2 에 생성 수열 $g_{1,1}$ 을 길쌈 연산을 취하면 다음과 같다.

$$c_1^* g_{1,2} = i^* g_{1,1}^* g_{1,2} \quad (5)$$

$$c_2^* g_{1,1} = i^* g_{1,2}^* g_{1,1} \quad (6)$$

길쌈 연산은 교환법칙이 성립하기 때문에 (5)와 (6)은 동일한 값이 된다. 따라서 (7)과 같은 $R=1/2$ 부호의 검증식을 얻을 수 있으며 모든 $1/2$ 길쌈 부호는 항상 (7)을 만족하게 된다.

$$c_1^* g_{1,2} + c_2^* g_{1,1} = i^* g_{1,1}^* g_{1,2} + i^* g_{1,2}^* g_{1,1} = 0 \quad (7)$$

따라서 수신된 부호를 c_1 과 c_2 수열로 분리한 뒤 예상되는 생성 수열을 (7)에 대입함으로써 실제 부호에 사용된 생성 수열인지 확인할 수 있다. 만약 수신된 데이터의 부호율이 $1/2$ 이고 분석 측에서

$g_{1,1}$ 과 $g_{1,2}$ 를 정확하게 가정하였다면 (7)은 항상 성립할 것이다.

2.2.1. $k=1$ 인 경우 길쌈 부호의 검증식

$k=1$ 인 단일 입력 길쌈 부호 중 부호율이 $1/2$ 인 경우에는 앞서 설명한 바와 같이 (7)의 만족 여부로 부호 추정을 수행한다. 부호율이 $R=1/3, 1/4$ 인 경우도 위와 같은 특성을 이용하여 분석한다. 부호율이 $1/3$ 인 경우의 검증식은 다음과 같다.

$$c_1^* g_{1,2} + c_2^* g_{1,1} = 0 \quad (8)$$

$$c_1^* g_{1,3} + c_3^* g_{1,1} = 0 \quad (9)$$

$$c_2^* g_{1,3} + c_3^* g_{1,2} = 0 \quad (10)$$

수신된 모든 부호어에 대하여 검증식 (8)~(10)을 동시에 만족하는 생성 수열 $g_{1,1}, g_{1,2}, g_{1,3}$ 을 찾는다면 생성 수열을 추정하였다고 할 수 있다. 같은 방법으로 부호율 $1/4$ 인 경우 또한 다음과 같은 검증식을 이용하여 생성 수열을 추정할 수 있다.

$$c_1^* g_{1,2} + c_2^* g_{1,1} = 0 \quad (11)$$

$$c_1^* g_{1,3} + c_3^* g_{1,1} = 0 \quad (12)$$

$$c_1^* g_{1,4} + c_4^* g_{1,1} = 0 \quad (13)$$

$$c_2^* g_{1,3} + c_3^* g_{1,2} = 0 \quad (14)$$

$$c_2^* g_{1,4} + c_4^* g_{1,2} = 0 \quad (15)$$

$$c_3^* g_{1,4} + c_4^* g_{1,3} = 0 \quad (16)$$

2.2.1. $k \geq 2$ 인 경우 길쌈 부호의 검증식

k 가 2 이상인 경우에는 k 가 1일 때 보다 복잡도가 급격히 증가한다. 그러나 길쌈 부호 분석 모듈 내부의 연산이 단순한 연산이기 때문에 연산시간이 크게 차이 나지는 않는다. 먼저 k 가 2이상인 복수 입력인 경우 분석 알고리즘 소개에 앞서 생성 구조에 대해 간단히 설명한다.

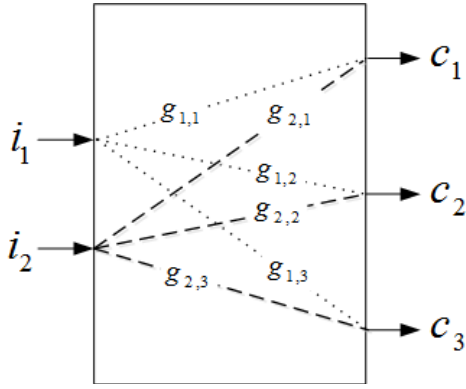


그림 2. 부호율 $R=2/3$ 일 때의 길쌈 부호기
Fig. 2. A rate $R=2/3$ convolutional encoder

그림 2는 $R=2/3$ 인 간략화 된 길쌈 부호기를 나타내고 있다. 입력이 2개, 출력이 3개인 구조이다. 즉 입력 2개가 출력 3개에 모두 영향을 주는 구조이다. 따라서 생성 수열도 총 6가지가 사용된다. 이러한 경우에도 k 가 1인 경우처럼 길쌈 특성을 이용하여 검증식이 만들어 지는데 그 과정이 매우 복잡하다. 그림 2에서는 입력을 i_1, i_2 라 정의하였고 출력을 c_1, c_2, c_3 생성 수열을 $g_{1,1} \sim g_{2,3}$ 로 정의 하였다.

우선 c_1, c_2, c_3 는 아래 (17), (18), (19)와 같이 표현할 수 있다.

$$c_1 = i_1 * g_{1,1} + i_2 * g_{2,1} \quad (17)$$

$$c_2 = i_1 * g_{1,2} + i_2 * g_{2,2} \quad (18)$$

$$c_3 = i_1 * g_{1,3} + i_2 * g_{2,3} \quad (19)$$

단일 입력과는 다르게 부호 성분에 i_1, i_2 이렇게 2가지 성분이 존재하므로 두 성분 중 한 성분을 먼저 소거해 줘야 한다. 어느 성분을 소거하던 결과는 동일하다. (17)과 (18)을 조합하여 i_1 을 소거하기 위해 각각 $g_{1,2}$ 와 $g_{1,1}$ 을 길쌈 연산을 한다. 그 결과는 (20)이 된다.

$$c_1 * g_{1,2} + c_2 * g_{1,1} = (i_1 * g_{1,1}) * g_{1,2} + (i_2 * g_{2,1}) * g_{1,2} + (i_1 * g_{1,2}) * g_{1,1} + (i_2 * g_{2,2}) * g_{1,1} \quad (20)$$

그리고 (18)과 (19)을 조합하여 i_1 을 소거하기 위해 각각 $g_{1,3}$ 과 $g_{1,2}$ 를 길쌈 연산을 한다. 그 결과는 (21)이 된다.

$$c_2 * g_{1,3} + c_4 * g_{1,2} = (i_1 * g_{1,2}) * g_{1,3} + (i_2 * g_{2,2}) * g_{1,3} + (i_1 * g_{1,3}) * g_{1,2} + (i_2 * g_{2,3}) * g_{1,2} \quad (21)$$

여기에서 (20)과 (21)에 $g_{2,2} * g_{1,3} + g_{2,3} * g_{1,2}, g_{2,1} * g_{1,2} + g_{2,2} * g_{1,1}$ 을 각각 길쌈 연산하면 두식은 동일한 우변을 가지게 되며 (22)를 얻을 수 있다.

$$(c_1 * g_{1,2} + c_2 * g_{1,1}) * (g_{2,2} * g_{1,3} + g_{2,3} * g_{1,2}) + (c_2 * g_{1,3} + c_3 * g_{1,2}) * (g_{2,1} * g_{1,2} + g_{2,2} * g_{1,1}) = 0 \quad (22)$$

수식을 간소화하기 위해 G_1, G_2 를 (23)과 같이 정의하면 (24)와 같은 $R=2/3$ 길쌈 부호에 대한 검증식을 구할 수 있다.

$$G_1 = g_{2,2} * g_{1,3} + g_{2,3} * g_{1,2} \quad (23)$$

$$G_2 = g_{2,1} * g_{1,2} + g_{2,2} * g_{1,1}$$

$$(c_1 * g_{1,2} + c_2 * g_{1,1}) * G_1 + (c_2 * g_{1,3} + c_3 * g_{1,2}) * G_2 = 0 \quad (24)$$

확인하고자 하는 생성 수열의 부호율이 $2/3$ 인 경우 수신된 부호를 c_1, c_2, c_3 형태의 출력 수열로 재구성하고 출력 수열과 생성 수열에 대해 (24)을 만족하는지 확인한다면 해당 생성 수열이 사용된 부호인지 알 수 있다. 또한 동일한 방법으로 $2/3$ 부호 이외의 다른 k/n 부호에 대해서도 생성 수열 확인을 위한 검증식을 구할 수 있다.

2.3. 잡음이 존재하는 채널에서의 확인

일반적으로 길쌈 부호는 부호 길이가 매우 긴 경우가 많으며 이러한 경우 부호어 중 하나의 오류 비트만으로도 추정에 심각한 영향을 주게 된다. 따라서 잡음이 존재하는 채널에서는 부호어 전체가 아닌 부호어를 일정한 크기로 나누어 각 부분을 검증식을 통해 확인한 후 그 확률 값을 이용하여 생성 수열을 추정한다.

앞서 살펴본 검증식은 길쌈 부호어가 생성 수열과 입력 수열을 길쌈 연산으로 이루어지는 특성에 의해 성립된다. 만약 부호어 전체가 아닌 부호어의 일부분에 검증식을 적용하게 되면 그 결과 값은 완벽하게 일치하지 않으며 메모리 크기만큼의 오차가

발생하게 된다. 부호어 전체를 사용할 경우 메모리의 초기화와 테일 비트의 영향으로 일반적인 길쌈 연산과 동일한 형태가 되지만 일부분만을 보았을 경우에는 이전 입력 값이 메모리에 남아 영향을 주기 때문이다. 이러한 문제를 해결하기 위하여 길쌈 연산 후 수열의 앞부분과 뒷부분에서 메모리 크기만큼의 비트를 제거하게 되는데 이는 길쌈 연산의 시작 시점을 메모리 크기만큼 늦추고 종료시점을 같은 크기만큼 앞당김으로써 출력 수열이 메모리에 남아있는 값의 영향을 받지 않게 하기 위함이다. 이러한 방법으로 단축된 범위의 길쌈 연산을 이용하게 되면 부호어의 일부분만을 이용하였을 때에도 검증식을 만족하게 된다. 이러한 단축된 범위의 길쌈 연산을 사용할 경우 (25)와 같이 출력 수열을 나타낼 수 있다.

$$c^{(n)} = \sum_{k=n}^{n+m} i^{(n+m-k)} g^{(k-n)} \quad (25)$$

(25)에서 i 와 g 는 각각 입력 수열과 생성 수열을 나타낸다. 잡음이 존재하는 채널을 통과한 부호의 경우 이와 같이 작은 크기로 부호를 나누어 출력 수열을 생성한 후 검증식을 적용하게 되며 이러한 과정을 반복하여 검증식을 만족하는 확률을 구함으로써 생성 수열을 추정할 수 있다. 이 때 계산되는 출력 수열의 크기가 클수록 높은 정확도를 가지는 반면 오류의 영향을 많이 받게 된다.

수신된 부호에 대하여 검증식을 만족하는 확률을 구한 경우 일정한 임계값 이상이 되면 해당 생성 수열을 사용하였다고 추정하게 된다. 이 때 적절한 임계값을 결정하기 위해서는 실제 잡음이 존재하는 채널에서의 검증식 만족 확률을 알고 있어야 한다. 따라서 부호율과 SNR (Signal-to-Noise Ratio)에 따른 검증식 만족 확률을 모의실험을 통하여 살펴본다. 각 모의실험은 무작위의 정보 비트로 생성된 길쌈 부호어를 사용하였으며 0~10dB의 AWGN 채널에서 수행되었다. 또한 각 dB에서 1000개의 출력 수열에 대하여 해당 부호율에 대한 검증식을 적용하였을 때 검증식을 만족하는 확률을 그래프로 나타내었다.

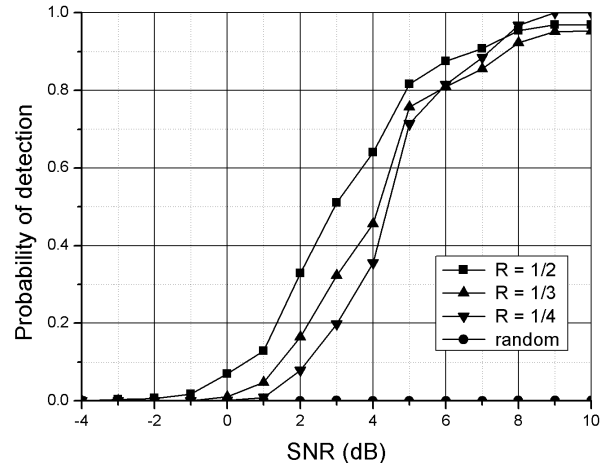


그림 3. $R=1/n$ 길쌈 부호의 검출 확률
Fig. 3. Probability of detection for a convolutional code with $R=1/n$

그림 3은 부호율이 $1/n$ 인 길쌈 부호에 대하여 생성에 사용된 생성 수열을 이용하여 검증식을 적용하였을 때 만족하는 확률을 나타낸 그래프이다. 또한 길쌈 부호가 아닌 무작위 비트에 대하여 검증식을 적용하였을 때의 만족 확률도 표시하였다. 각 부호율에서는 모두 메모리 개수가 8개인 생성 수열을 사용하였다.

그림 3의 그래프에서 올바른 생성 수열을 이용한 검증식을 적용하더라도 SNR에 따라서 검증식을 만족하는 확률이 달라지는 것을 볼 수 있다. 이는 앞서 알아본 바와 같이 확인하고자 하는 출력 수열에 오류가 포함된 경우 검증식을 만족하지 않기 때문이다. 하지만 SNR이 낮아지더라도 올바른 생성 수열을 사용할 경우에는 어느 정도의 확률 값을 가지는 반면 무작위로 생성된 비트는 SNR에 상관없이 거의 0에 가까운 확률 값을 나타내는 것을 볼 수 있다. 만약 사용된 생성 수열이 아닌 다른 생성 수열을 이용하여 검증식을 적용한다면 검증식과 상관 없는 무작위 값으로 볼 수 있으며 마찬가지로 0에 가까운 결과 값을 나타내게 된다.

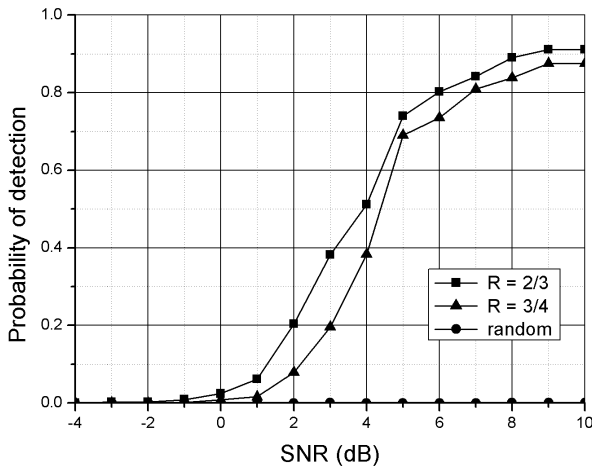


그림 4. $R=k/n$ 길쌈 부호의 검출 확률
 Fig. 4. Probability of detection for a convolutional code with $R=k/n$

그림 4는 부호율이 k/n 인 길쌈 부호와 무작위 비트에 대한 검증식 만족 확률을 나타낸 그래프이다. 그림 4의 모의실험에서는 모두 총 메모리 개수가 5개인 생성 수열을 사용하였다. $1/n$ 부호와 마찬가지로 SNR에 따라서 검증식을 만족하는 확률이 달라지는 것을 볼 수 있으며 무작위로 생성된 비트는 SNR에 상관없이 0에 가까운 확률 값을 나타내는 것을 볼 수 있다. 따라서 일정한 임계값을 설정한다면 잡음이 존재하는 채널에도 실제로 사용된 생성 수열을 구해 낼 수 있을 것이다. 또한 동일한 SNR에서 두 부호율의 확률 값이 다르게 나타나는 것을 볼 수 있으며 이는 $1/n$ 부호와 마찬가지로 출력 수열의 개수에 따른 것이다. 따라서 출력 수열의 개수가 같은 $1/3$ 과 $2/3$ 부호의 경우 동일한 SNR에서 비슷한 확률 값을 가지는 것을 볼 수 있다. 또한 k/n 부호에서도 0.2의 임계값을 적용할 경우 약 4dB이상의 채널 환경에서 모의실험에 사용된 모든 생성 수열을 추정해 낼 수 있을 것으로 보인다.

III. 천공 길쌈 부호의 분석

3.1. 천공 길쌈 부호의 구조

천공 길쌈 부호는 통신 시스템에서 부호율을 조정할 때 흔히 사용되는 기법이다. 예를 들어 부호율이 $R=1/2$ 인 부호를 (26)이라 하고 천공 패턴을 $[1\ 1\ 1\ 0]$ 이라 하면 천공된 부호는 (27)과 같다.

$$c = (c_1^{(1)}, c_2^{(1)}, c_1^{(2)}, c_2^{(2)}, c_1^{(3)}, c_2^{(3)}, c_1^{(4)}, c_2^{(4)}, c_1^{(5)}, c_2^{(5)} \dots) \quad (26)$$

$$c = (c_1^{(1)}, c_2^{(1)}, c_1^{(2)}, c_1^{(3)}, c_2^{(3)}, c_1^{(4)}, c_1^{(5)}, c_2^{(5)} \dots) \quad (27)$$

즉, 천공 패턴의 0이 되는 위치의 부호어는 전송 시 보내지 않게 된다. 수신 측에서는 이렇게 천공된 자리에 복호에 영향을 주지 않는 값으로 채워 넣어서 복호를 수행하게 된다. 천공 패턴이 $[1\ 1\ 1\ 0]$ 인 경우에는 부호어 4비트 중에서 3비트만 보내게 된다. 따라서 원래 부호율이 $1/2$ 인 부호어 4비트 중에서 3비트만 전송이 되므로 결과적으로 2비트가 입력되어 3비트가 출력되는 부호기를 사용한 결과가 되므로 부호율은 $2/3$ 이 되는 부호를 사용한 것이 된다.

3.2. 천공 길쌈 부호의 분석

천공 길쌈 부호의 분석은 앞선 길쌈 부호의 분석 알고리즘과 다르다. 천공된 자리가 존재하기 때문에 임의의 생성 수열과 직교를 수행함에 있어 어려움이 있다. 천공 길쌈 부호의 분석에 관해서도 많은 연구가 진행되어 왔는데 대표적으로 E.Filiol과 M.Cluzeau가 제안한 방법을 들 수 있다^[8,9]. 본 논문에서는 비터비 복호기와 길쌈 부호 부호기를 이용하여 천공 길쌈 부호를 분석하는 방법을 제시한다. 일반적으로 통신 시스템에 사용되는 천공 패턴의 종류는 거의 정해져 있기 때문에 길쌈 부호의 종류에 따라 몇 가지 천공 패턴을 고정하여 분석을 수행한다. 부호율 $R=1/2$, 구속장 7인 길쌈 부호를 예로 들면 먼저 수신 데이터에 대해서 임의의 천공 패턴을 가정하고, 비터비 복호기를 이용하여 복호를 수행한다. 이렇게 복호된 데이터를 다시 가정한 천공 패턴이 적용된 부호기에 입력으로 사용하여 부호화를 수행하는데 만약 파라미터 추정이 올바르게 되었다면 수신 데이터와 다시 부호화된 데이터가 일치할 것이다. 따라서 수신 데이터와 다시 부호화된 데이터를 비트 단위로 비교하여 일치한다면 수신 데이터를 천공 길쌈 부호로 판정한다.

그림 5는 $[1\ 1\ 1\ 0]$, $[1\ 1\ 1\ 0\ 0\ 1]$ 2가지의 천공 패턴에 대한 천공 길쌈 부호의 분석 모듈을 simulink 블록도로 나타낸 것이다. 만약 수신데이터에 오류가 포함되어 있는 경우에는 앞서 길쌈 부호의 분석과 마찬가지로 비트 단위로 비교할 때 일치하는 비율에 임계치를 설정하여 오류가 존재하는

경우에도 분석이 가능하도록 한다.

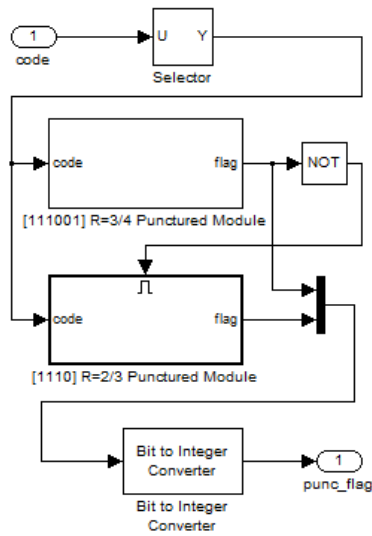


그림 5. 천공 길쌈 부호의 분석 모듈
Fig. 5. Analysis module for punctured convolutional codes

IV. 시뮬레이션

지금까지 수신된 부호의 정보를 알 수 없을 때 이 부호가 길쌈 부호임을 판별하고 생성 수열의 복원을 통한 파라미터를 추정하는 알고리즘을 살펴보았다. 이 장에서는 지금까지 살펴본 알고리즘을 여러 종류의 길쌈 부호 및 천공 길쌈 부호에 적용하였을 때 실제 잡음이 포함된 채널 환경에서 어떠한 결과를 나타내는지 알아본다. 시뮬레이션은 앞서 검증식을 유도해냈던 부호율 1/2, 1/3, 1/4, 2/3, 3/4 인 길쌈 부호에 대해 수행하였다. 또한 모든 부호율에 대하여 0.2의 동일한 입계값을 적용하였으며, 각각의 부호율에서 동일한 메모리 개수를 갖는 생성 수열을 사용하였다. 채널 환경은 AWGN 채널을 사용하였으며 -1dB에서 5dB까지 0.1dB 단위로 시뮬레이션을 수행하였다. 각각의 dB에서 1000번의 시뮬레이션을 통해 길쌈 부호의 부호 인식 확률을 그래프에 나타내었다.

그림 6은 수신 데이터가 1/n 길쌈 부호일 때 부호인식확률을 나타낸 그래프이다. 여기서 각 부호별로 사용된 생성 수열은 모두 메모리 개수가 8개인 생성 수열을 사용하였다. 동일한 SNR에서 R=1/2 일 때 가장 높은 성능을 보이며, 4dB이상의 채널 환경에서는 시뮬레이션에 사용된 모든 부호를 인식함을 볼 수 있다.

그림 7은 수신데이터가 k/n 길쌈 부호일 때 부호인식확률을 나타낸 그래프이다. 시뮬레이션에서 각 부호율은 모두 총 메모리 개수가 5인 생성 수열을 사용하였다. 그래프를 살펴보면 동일한 SNR에서도 부호율에 따라 추정 성능에 차이가 있는 것을 볼 수 있다. 부호율이 R=2/3인 경우와 R=3/4인 경우 모두 4dB 이상에서 모두 부호를 인식해 낼 수 있다.

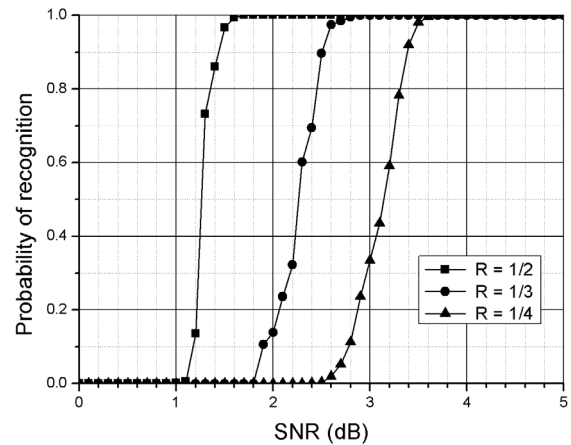


그림 6. R=1/2, 1/3, 1/4 길쌈 부호 인식 확률
Fig. 6. Probability of recognition for convolutional codes with R=1/2, 1/3, 1/4

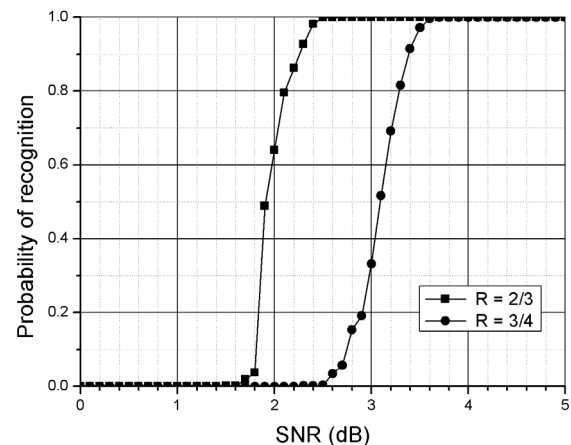


그림 7. R=2/3, 3/4 길쌈 부호 인식 확률
Fig. 7. Probability of recognition for convolutional codes with R=2/3, 3/4

그림 8은 부호율이 R=1/2이고, 천공 패턴 [1 1 1 0]과 [1 1 1 0 0 1]이 적용된 길쌈 부호에 대한 부호 인식 확률을 나타내고 있다. 마찬가지로 채널 환경은 AWGN 채널을 사용하였으며 -3dB에서 9dB까지 1dB 단위로 시뮬레이션을 수행하였다. 시뮬레이션 결과 적용한 천공 패턴에 따라 모두 5dB 이상에서 부호를 인식해 낼 수 있다.

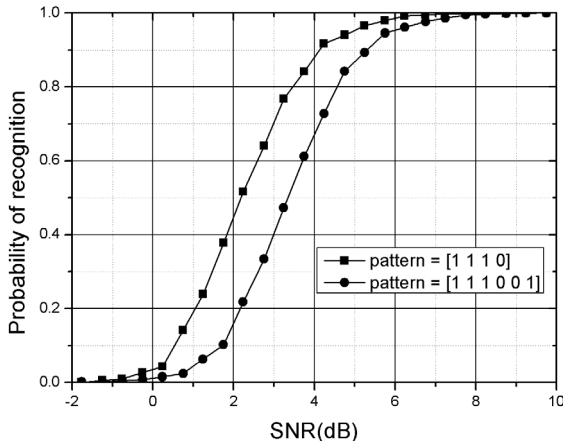


그림 8. 천공패턴 [1 1 1 0], [1 1 1 0 0 1]인 천공 길쌈 부호 인식 확률
 Fig. 8. Probability of recognition for punctured convolutional codes with puncturing pattern [1 1 1 0], [1 1 1 0 0 1]

V. 결 론

본 논문에서는 길쌈부호의 특성을 살펴보고 그 특성을 이용하여 생성 정보를 알 수 없는 미지의 부호가 수신되었을 때, 수신된 부호가 길쌈 부호 및 천공 길쌈 부호인지를 판별하는 방법을 알아보았다. 선행되어 연구된 Filiol의 길쌈 부호 생성 수열 추정 기법은 다항식을 이용하여 증명되었으나 본 논문에서는 길쌈 연산을 이용하여 생성 수열을 추정하기 위한 검증식을 유도하였다. 또한 채널 잡음으로 인하여 부호에 오류가 포함될 가능성이 있는 경우 생성 수열을 확인하는 방법을 알아보았다. 본 논문에서 제시한 방법은 길쌈 부호의 출력 수열이 생성 수열과 입력 수열의 길쌈 연산으로 이루어진다는 점을 기반으로 검증식을 유도하였으며 각 부호를 별로 SNR에 따른 검증식 만족 확률을 구하여 생성 수열 추정이 용이하게 하였다. 천공 길쌈 부호의 경우에는 비터비 알고리즘을 이용한 복호기와 길쌈 부호의 부호기를 이용한 분석 방법을 제시하였다.

또한, 시뮬레이션을 통해 실제 채널 환경에서 어떠한 결과를 나타내는지 검증하였다. 시뮬레이션을 통해 임계값을 0.2로 설정하였을 때 약 4dB이상에서 시뮬레이션에서 사용된 모든 생성 수열을 추정해 낼 수 있었다.

참 고 문 헌

[1] S. Lin and D. J. Costello, *Error Control*

Coding: Fundamentals and Applications, Prentice Hall, Inc., 1983.

[2] Robert H. Morelos-Zaragoza, *The Art of Error Correcting Coding*, John Wiley & Sons, Ltd, 2007.

[3] Y.J. Song, *Coding Theory for Communication Engineering*, Infinity Books, 2008.

[4] E.Filiol. "Reconstruction of convolutional encoder over GF(q)," *Lecture Note in Computer Science*, pp.101-109, 1997.

[5] M. Cote and N. Sendrier. "Reconstruction of convolutional codes from noisy observation," *ISIT2009*, pp. 546-550, 2009.

[6] J. Dingel and J. Hagenauer, "Parameter estimation of a convolutional encoder from noisy observations," *ISIT2007*, pp. 1775-1780, 2007.

[7] I. S. Kang, H. Lee, J. H. Lee, Y. J. Song, "Estimation of Generator Sequences for Convolutional Codes," *Proceeding of KICS Autumn Conference*, Vol. 46, pp. 163, 2011.

[8] E. Filiol. "Reconstruction of punctured convolutional encoders," *International Symposium on Information Theory and Applications*, pp. 4-7, 2000.

[9] M. Cluzeau and M. Finiasz, "Reconstruction of punctured convolutional codes," *Information Theory Workshop*, pp. 75-79, 2009.

[10] M.Y. Lee, *Error-Correction Coding Theory*, Hee-Jung Dang, 1985.

이 재 환 (Jae-Hwan Lee)

준회원



2011년 2월 금오공과대학교 전자공학과 학사

2011년 2월~현재 금오공과대학교 전자공학과 석사과정
 <관심분야> 이동통신 시스템, 부호 이론

이 현 (Hyun Lee)

준회원



2010년 2월 금오공과대학교
전자공학과 학사
2012년 2월 금오공과대학교
전자공학과 석사
2011년~현재 LIG 넥스원
구미공장 기술 6팀 연구원
<관심분야> 이동통신 시스템,
부호이론

박 철 순 (Cheol-sun Park)

정회원



1989년 2월 경기대학교 전자
계산학과 학사
1991년 2월 인하대학교 전자
계산공학과 석사
1991년~현재 국방과학연구소
선임연구원
1997년 전자계산 조직응용 기

술사

2007년 충남대학교 정보통신공학과 박사
<관심분야> 신호처리, 통신응용

강 인 식 (In-Sik Kang)

준회원



2010년 2월 금오공과대학교 전
자공학과 학사
2012년 2월 금오공과대학교 전
자공학과 석사
<관심분야> 이동통신 시스템,
부호이론

송 영 준 (Young-joon Song)

중신회원



1987년 2월 한양대학교 전자
통신공학과 공학사
1994년 2월 한양대학교 전자
통신공학과 공학석사
1999년 2월 한양대학교 전자
통신공학과 공학박사

윤 상 범 (Sang-bom Yun)

정회원



1999년 2월 고려대학교 제어
계측공학과 학사
2002년 2월 고려대학교 전기
공학과 석사
2002년~현재 LIG 넥스원 전
자전 연구센터 수석연구원
<관심분야> 통신, 컴퓨터, 신
호처리, 반도체

2006년 1월~2007년 1월 미국 하와이 주립대학교
방문학자

2002년 3월~현재 금오공과대학교 전자공학부 부교
수

<관심분야> 이동통신 시스템, 부호 이론