군통신 체계를 위한 해밍 부호화된 미지 신호의 부호 탐지 기법

이인석, 오성준, 고영채
고려대학교

요 약

본고에서는 해밍(Hamming) 부호화된 미지의 신호를 바탕으로 전송된 채널 코딩의 종류에 대해서 알아낼 수 있는 알고리즘을 제안한다. 우선 해밍 코드의 부호화, 복호화 과정을 설명한다. (n,k) 해밍 코드의 부호화 과정을 위하여 사용되는 제너레이터 행렬 G는 (k * n)의 크기를 갖게 되며 복호화 과정을 위하여 사용되는 패리티 체크 행렬 H는 (n-k * n)의 크기를 갖게 된다. 그리고 설계한 알고리즘의 동작원리를 수신 신호의 상태에 따라 설명한다. 수신 신호의 앞에 프리앰블이 있는 경우나 수신 신호가 Inverse 또는 Reverse 되어 있는 등 여러 경우에 대비한 알고리즘 설계 방법을 알아본다. 이렇게 설계한 알고리즘은 복잡도가 낮고 확장이 용이하며 수신 신호의 코드워드의 시작점을 쉽게 알 수 있는 장점을 가지고 있어 일반 사용자의 통신 시스템뿐만 아니라 군사용 통신 체계에 적용하기에도 적합하다.

Ⅰ. 서 론

본 연구는 미지의 통신 신호를 수신하고 전송된 신호의 복조를 수행하여 비트로 구성된 데이터와 프레임 길이에 대한 정보만을 획득한 경우에 이를 바탕으로 전송된 채널 코딩의 종류에 대해서 알아 낼 수 있는 알고리즘을 개발하여 이를 최종적으로 해석 가능한 데이터로 이용할 수 있도록 하는 것에 목표가 있다. 현존하는 채널 코딩의 방법은 무한히 많다고 볼 수 있는데 왜냐하면 각 사용자가 임의로 정하여 개발할 수 있는 여지가 많기 때문이다. 그러한 이유로, 본 연구에서는 일반 사용자의 통신 서비스를 위한 셀룰러 시스템뿐 아니라 특히 군사용 통신 및 미국의 NASA 연구소에서 Deep Space Project를 위해 고안되어 알려진 채널 코딩 방법들 등에 대해서도 채널 코딩을 알아낼 수 있는 알고리즘을 목표로 하고 있다. 미지의 신호를 수신하였을 때, 그 신호로부터 코드의 정보를 알아내고 미지의 비트를 검출할 수 있는 알고리즘은 군사적인 목적으로 이용되기 적합하며 군사 통신 체계에 적용하였을 때 그 효용성이 높다고 할 수 있다. 해밍 블록부호를 코드워드 길이에 따라서 분류하고 판별하는 알고리즘 및 소프트웨어를 개발하기 위해서 해밍 블록부호 중에서 다양한 표준에서 널리 사용되는 코드워드를 파악하고 각각의 구조에 대한 특성을 파악한다. 그 후, 코드워드 길이가 다른 해밍 블록부호가 혼재되어 있는 입력 정보를 생성한 후 각각의 코드워드를 분류해 낼 수 있는지 파악하며 완벽한 판별을 위하여 추가로 요구되는 정보를 판별한다. 또한, 시스테메틱 부호와 비시스테메틱 부호의 경우에 각각 판별하는 기준을 연구한다.

본론에서는 현재까지 개발한 미지의 해밍 코드로 부호화 되어 있는 미지의 비트로 이루어진 정보를 받았을 때 어떤 길이의 어떤 제너레이터 행렬 G를 갖는 해밍 코드에 의해 부호화 되었는지를 판별하고, 더 나아가 코드워드의 시작 위치를 찾아내는 알고리즘에 대해 기술하도록 하겠다. 이와 더불어 알고리즘의 활용도를 높이기 위하여 수신 신호의 앞부분에 프리엠블(preamble)이 붙어있는 경우 프리엠블을 제외한 순수 메시지 신호가 시작되는 위치와 제너레이터 행렬을 찾을 수 있도록 하였고, 수신 신호가 'Inverse' 되어 있는 경우나 수신 신호가 코드워드 단위로 'Reverse' 되어 있는 경우 수신 신호를 원래의 형태로 되돌린 후 제너레이터 행렬과 코드워드의 시작 위치를 찾아내도록 하였다. 또한, 수신 신호가 두 개의 서로 다른 제너레이터 행렬로 부호화 되어있는 경우 서로 다른 두 제너레이터 행렬과 두 번째 제너레이터 행렬에 의하여 부호화가 시작되는 위치를 찾아내도록 하였다. 마지막으로 이렇게 찾아낸 제너레이터 행렬과 코드워드의 시작 위치 또는 프리엠블이 끝나는 위치, 서로 다른 두 개의 제너레이터 행렬이 사용되었을 때 경계점 등의 정보를 바탕으로 복호화 과정을 통해 원래의 메시지를 찾아내도록 하였다.

Ⅱ. 해밍코드

1. 부호화 과정

(n,k) 해밍 코드의 부호화 과정을 위하여 사용되는 제너레이

터 행렬 G는 (k * n)의 크기를 갖게 된다. (7,4) 해밍 코드를 예로 들어보겠다.

$$G = \begin{matrix} 1 & 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 1 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 & 0 & 0 & 1 \end{matrix}$$

제너레이터 행렬을 위와 같이 정의하면 4 비트 메시지는 다음과 같이 코드워드로 변환된다.

메시지 * G = 코드워드

메시지	코드워드
0000	0000000
1000	1101000
0100	0110100
1100	1011100
0010	1110010
1010	0011010
0110	1000110
1110	0101110
0001	1010001
1001	0111001
0101	1100101
1101	0001101
0011	0100011
1011	1001011
0111	0010111
1111	1111111

위에서 예를 든 제너레이터 행렬은 시스테메틱 형태로서 코드워드의 형태가 ((n-k) 페리티 비트 + (k) 코드 비트)의 형태를 가지게 된다. 만약 제너레이터 행렬이 비시스테메틱 형태라면 코드워드는 특정한 형태를 가지지 않게 된다. 수신단에서 코드워드로부터 메시지를 정확히 검출해 내기 위해서 메시지와 코드워드는 각각 일대일로 결정된다. n 비트 길이의 코드워드 당 한 비트 에러 정정이 가능한 코드워드를 생성하기 위해서는 각 코드워드 간의 해밍 거리(Hamming-distance)[1]는 최소한 '3'이 되어야 한다. 비시스테메틱 형태의 해밍 코드인 경우 모든 가능한 (k * n) 행렬 중 (모든 원소가 '0'인 행렬로부터 모든 원소가 '1'인 행렬) 코드워드 간의 최소 해밍 거리가 '3' 이상인 행렬만이 제너레이터 행렬이 될 수 있는 조건을 만족한다[1].

2. 복호화 과정

(n,k) 해밍 코드의 복호화 과정을 위하여 사용되는 패리티 체크 행렬 H는 (n-k * n)의 크기를 갖게 된다. 패리티 체크 행렬 H의 특징을 살펴보면 전체 n 차원의 공간에서 제너레이터 행렬 G의 행 공간(k차원)을 뺀 (n-k)차원의 공간을 생성(span)하는 (n-k)의 서로 독립적인 기저들을 H의 행 벡터로 정의한다. 그러면 $G * H^T = 0$ 이 성립한다.

$$G = [\, P,\ I_k \,] \ \rightarrow\ H = [\, I_{n-k}\ ,\ P^T] \ ^{2}$$

$$H = \begin{matrix} 1 & 0 & 0 & 1 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 & 1 & 1 & 0 \\ 0 & 0 & 1 & 0 & 1 & 1 & 1 \end{matrix}$$

송신단에서 보낸 메시지를 m이라 하면 코드워드 v는 (m * G)의 형태이다. $G * H^T = 0$이므로 다음 관계가 만족한다.

$$\begin{aligned} v * H^T &= (m * G) * H^T \\ &= m * (G * H^T) \\ &= m * 0 = 0 \end{aligned}$$

이를 이용하여 수신단에서 에러 검출을 할 수 있다. 송신단에서 코드워드 v를 전송하면 채널을 거쳐 수신단에서 v + 에러 = v'가 도착한다.

$$\begin{aligned} v' * H^T &= (v + \text{에러}) * H^T \\ &= (v * H^T) + (\text{에러} * H^T) \\ &= \text{에러} * H^T \end{aligned}$$

에러 * H^T를 신드롬(syndrome)이라 정의하고 신드롬에 따라 그에 대응되는 에러를 정의할 수 있다. 신드롬의 크기는 (n-k * 1)로 총 $2^{(n-k)}$의 경우의 수를 가진다. 위 예제에서 보면 신드롬은 총 2^{7-4} = 8개를 가지므로, (0,0, … ,0)부터 (1,1, … , 1)까지 총 2^7개의 모든 에러를 표현할 수는 없다. 하지만 BSC(Binary symmetric channel)을 가정하고 비트 에러 확률이 낮다고 한다면 확률 상 7개의 코드워드 비트에서 1비트 에러가 나올 가능성이 나올 가능성이 가장 높기 때문에 신드롬은 1비트 에러만을 대표하도록 설정한다.

에러	신드롬
0000000	000
1000000	100
0100000	010
0010000	001
0001000	110
0000100	011
0000010	111
0000001	101

즉, 수신단에서 받은 v'에 H^T를 곱하여 신드롬을 확인하고 그에 해당하는 에러를 v'에 더하여 송신단에서 보낸 v로 복원하는

1 두 코드워드를 비교해 서로 다른 비트의 자릿수를 말함. 예를 들어 '110'과 '011'의 해밍 거리는 '2'가 된다.

2 I_k : 크기가 (k * k)인 단위행렬 / P^T : P 행렬의 행과 열이 바뀐 행렬

것이다.

제너레이터 행렬이 비시스테메틱 형태라면 위와 같은 신드롬에 기반한 디코딩 방법을 사용할 수 없다. 따라서 비시스테메틱 형태의 제너레이터 행렬에 의해 부호화 된 코드를 복호화 하기 위해서는 해밍 부호의 기본적인 성질을 이용하여 복호화 하여야 한다. (n,k) 해밍 코드가 에러를 수정해 원래 메시지로 복원하는 원리는 다음과 같다. 제너레이터 행렬 G가 만들어내는 코드워드 간의 해밍 거리는 (n-k)이다. 코드워드는 채널을 거치며 에러가 더해져 수신단에 도착하게 되고, 수신단에서는 에러가 더해진 코드워드를 가장 해밍 거리가 가까운 코드워드 중 하나도 바꾸어 준다. 이렇게 바꾸어 준 코드워드는 원래의 메시지로 바뀌어 지고 송신단에서 보낸 v로 복원되는 것이다. (7,4) 해밍 코드로 예를 들어보면,

송신 메시지 : 0 0 0 0
송신 코드워드 : 0 0 0 0 0 0 0
에러 : 0 0 0 0 0 0 1
수신 코드워드 : 0 0 0 0 0 0 1
↓
해밍 거리가 가장 가까운 코드워드 :
0 0 0 0 0 0 0
수신 메시지 : 0 0 0 0

하지만 만약 에러가 2 비트(0 0 0 0 0 0 1 1)가 생겨 수신 코드워드가 '0 0 0 0 0 1 1'이 되었다면 가장 해밍 거리가 가까운 코드워드는 '0 1 0 0 0 1 1'가 되어 수신 메시지를 '0 0 1 1' 로 잘못 복원하게 된다. 통상적으로 (n,k) 해밍 코드에서 에러의 개수가 $\left\lfloor \frac{n-k}{2} \right\rfloor$[3] 이하일 경우에만 에러 교정이 가능하다. $\left\lfloor \frac{n-k}{2} \right\rfloor$는 서로 인접한 두 코드워드 간 해밍 거리의 절반이 되는 값으로 에러가 더해진 신호와 원래의 코드워드와의 해밍 거리가 $\left\lfloor \frac{n-k}{2} \right\rfloor$보다 클 경우 복호화 과정에서 원래의 코드워드와의 해밍 거리보다 다른 코드워드와의 해밍 거리가 더 가깝기 때문에 수신 신호를 다른 코드워드로 판단하고 잘못된 추정 메시지를 반환하게 되어 에러가 발생한다[2][3].

Ⅲ. 알고리즘 동작원리

1. 수신 신호를 중간부터 받는 경우

미지의 해밍 코드의 크기를 (n,k)라고 하고, 제너레이터 행렬을 G라고 한다면 처음의 메시지와 코드워드는 일대일 대응이 되기 때문에 이 해밍 코드가 만들 수 있는 코드워드는 2^k개가 된다. 따라서 이 해밍 코드에 의해 부호화 된 미지의 비트로 이루어진 정보는 제너레이터 행렬 G가 생성할 수 있는 코드워드들의 연속이 된다. 즉,

송신 신호 = … , (2^k개의 코드워드 중 하나), (2^k개의 코드워드 중 하나), …

수신단에서는 송신 신호에 에러가 더해진 신호를 받게 되고, 어떤 미지의 해밍 코드로 부호화 되었는지를 판별해 내어야 한다. 우리가 개발한 알고리즘은 송신 신호가 2^k개의 코드워드들의 집합으로 되어있다는 사실에 기반을 두었다. 코드워드들은 n비트이기 때문에 코드워드가 될 수 있는 총 경우의 수는 2^n 개가 된다. 하지만 코드워드는 이 중 메시지 신호와 일대일 대응이 되는 2^k개만을 가지기 때문에 모든 n비트의 조합이 코드워드가 될 수 없다. 따라서 수신 신호를 n 비트씩 나누어 각각의 n 비트와 해밍 거리가 가장 짧은 코드워드를 2^k개의 코드워드들의 집합 중에서 선택한다. 그리고 이렇게 구한 최소의 해밍 거리들의 평균값을 구한다. 만약 수신 신호에 에러가 하나도 없고, 신호의 수신 위치가 정확히 코드워드의 시작점과 일치하며, 수신단에서 사용한 해밍 코드가 송신단에서 부호화하는데 사용한 해밍 코드와 일치한다면 해밍 거리의 평균값은 '0'이 될 것이다. 다른 조건은 동일한 채 수신 신호에 에러가 첨가되었다면 해밍 거리의 평균값은 BER[4] 값과 동일할 것이다. 이 BER 값이 어떠한 해밍 코드가 사용되었나를 판별하는 역치값(threshold)으로서 최적화 된 값을 사용해 알고리즘의 판별력을 높일 수 있다. 통상적인 통신시스템에서 목표 BER 값을 10^{-3} 정도로 설정해 시스템을 설계하기 때문에 신호의 수신 위치가 정확히 코드워드의 시작점과 일치하고, 수신단에서 사용한 해밍 코드가 송신단에서 부호화하는데 사용한 해밍 코드와 일치한다면 해밍 거리의 평균값은 최대 10^{-3}을 넘지 않는다.

수신단에서 사용한 해밍 코드가 송신단에서 부호화하는데 사용한 해밍 코드와 일치하지만 신호의 수신 위치가 정확히 코드워드의 시작점과 일치하지 않는다면 위의 알고리즘으로 구한 해밍 거리의 평균값은 BER의 값보다 커질 것이다. 반대로 신호의 수신 위치가 정확히 코드워드의 시작점과 일치하지만 수신단에서 사용한 해밍 코드가 송신단에서 부호화하는데 사용한 해밍 코드와 일치하지 않는다면 위의 알고리즘으로 구한 해밍 거리의 평균값도 BER의 값보다 커질 것이다. 무작위로 n 비트씩 자른 수신 신호들은 2^k개의 코드워드들 중 하나와도 정확히 일치(해밍 거리가 '0')하지 않을 확률이 크다. 오히려 해밍 거리

3 $\lfloor \ \rfloor$: 괄호 안의 값을 넘지 않는 최소의 정수를 반환한다.

4 bit error ratio의 약자로 에러 비트 수를 전체 비트 수로 나누어 준 값이다.

가 '0'이거나, '1'이거나, … , '$\left\lfloor \frac{n-k}{2} \right\rfloor$'중 하나의 값을 갖는다. 하지만 해밍 거리가 '0'일 확률은 낮고, 그 이외의 값을 가질 확률은 상대적으로 커 결과적으로 해밍 거리의 평균값은 대략 1/n 값을 갖는다. 따라서 정확한 해밍 코드와 코드워드 시작점으로 해밍 거리의 평균값을 계산하였을 경우에만 BER 값을 가지고, 그렇지 않은 경우에는 해밍 거리의 평균값은 BER 값보다 훨씬 큰 약 1/n 값을 갖는다. 역치값을 이 사이의 값 중 최적화 된 값으로 설정해 주어 해밍 거리의 평균값이 역치값보다 작다면 이때의 해밍 코드와 코드워드 시작점을 옳은 값으로 판단하고, 해밍 거리의 평균값이 이 역치값보다 크다면 이때의 해밍 코드와 코드워드 시작점은 틀린 값으로 판단한다.

2. 수신 신호 앞에 프리엠블이 있는 경우

프리엠블은 수신단에서 채널 추정이나 오프셋 등을 보정하기 위하여 보내는 송신단과 수신단 사이에 약속된 일정한 패턴의 신호이다. 수신단에서는 수신된 프리엠블과 이미 알고 있는 원래의 프리엠블 신호를 비교해 채널이나 오프셋에 관한 정보를 얻게 된다. 하지만 프리엠블 신호는 부호화 과정을 거치지 않고 약속된 일정한 패턴 그대로 송신되기 때문에 제안된 알고리즘에서는 필요하지 않고 버려져야 하는 부분이다. 따라서 제안된 알고리즘에서 수신 신호 앞에 프리엠블이 있는 경우 프리엠블을 제외한 순수 메시지 부분의 시작 위치와 제너레이터 행렬을 찾아내도록 설계되었다. 수신 신호를 n 비트씩 나누어 각각의 n 비트와 해밍 거리가 가장 짧은 코드워드를 2^k개의 코드워드들의 집합 중에서 선택한다. 그리고 이렇게 구한 최소의 해밍 거리들의 평균값을 구한다. 만약 수신 신호에 에러가 하나도 없고, 신호의 수신 위치가 프리엠블을 제외한 순수 메시지 부분의 시작 위치와 일치하며, 수신단에서 사용한 해밍 코드가 송신단에서 부호화하는데 사용한 해밍 코드와 일치한다면 해밍 거리의 평균값은 '0'이 될 것이다. 다른 조건은 동일한 채 수신 신호에 에러가 첨가되었다면 해밍 거리의 평균값은 BER 값과 동일할 것이다. 반대로 신호의 수신 위치가 정확히 프리엠블을 제외한 순수 메시지 부분의 시작 위치와 일치하지만 수신단에서 사용한 해밍 코드가 송신단에서 부호화하는데 사용한 해밍 코드와 일치하지 않는다면 위의 알고리즘으로 구한 해밍 거리의 평균값도 BER의 값보다 커질 것이다. 이 후의 알고리즘은 수신 신호를 중간부터 받는 경우와 동일한 과정을 거친다.

3. 수신 신호가 'Inverse' 되어 있는 경우

송신단에서 신호가 inverse 연산을 거친 경우 신호는 1의 보수 형태로 변하게 된다. 예를 들어 해밍 부호화를 거친 신호가 (1 0 0 1 0 1 …) 이라면 inverse 연산을 거친 신호는 (0 1 1 0 1 0 …) 이 된다. 수신단에서는 수신 신호를 inverse 연산을 거친 후 n 비트씩 나누어 각각의 n 비트와 해밍 거리가 가장 짧은 코드워드를 2^k개의 코드워드들의 집합 중에서 선택한다. 그리고 이렇게 구한 최소의 해밍 거리들의 평균값을 구한다. 만약 수신 신호가 inverse 연산을 거쳤고 에러가 하나도 없으며, 신호의 수신 위치가 코드워드의 시작 위치와 일치하고 수신단에서 사용한 해밍 코드가 송신단에서 부호화하는데 사용한 해밍 코드와 일치한다면 해밍 거리의 평균값은 '0'이 될 것이다. 다른 조건은 동일한 채 수신 신호에 에러가 첨가되었다면 해밍 거리의 평균값은 BER 값과 동일할 것이다. 신호의 수신 위치가 정확히 코드워드의 시작 위치와 일치하고 수신단에서 사용한 해밍 코드가 송신단에서 부호화하는데 사용한 해밍 코드와 일치하지만 inverse 연산을 거치지 않았다면 위의 알고리즘으로 구한 해밍 거리의 평균값은 BER의 값보다 커질 것이다. 따라서 inverse 연산을 거친 신호를 정확한 해밍 코드와 코드워드의 시작점으로 해밍 거리의 평균값을 계산하였을 경우에만 BER 값을 가지고, 그렇지 않은 경우에는 해밍 거리의 평균값은 BER 값보다 훨씬 큰 약 1/n 값을 갖는다. 역치값을 이 사이의 값 중 최적화 된 값으로 설정해 주어 해밍 거리의 평균값이 역치값보다 작다면 이때의 해밍 코드와 순수 메시지 부분의 시작점을 옳은 값으로 판단하고, 해밍 거리의 평균값이 이 역치값보다 크다면 이때의 해밍 코드와 순수 메시지 부분의 시작점은 틀린 값으로 판단한다.

4. 수신 신호가 'Reverse' 되어 있는 경우

송신단에서 신호가 reverse 연산을 거친 경우 신호는 코드워드 길이 단위로 신호가 뒤집히게 된다. 예를 들어 (7,4) 해밍 코드로 부호화 되었다면 reverse 연산을 거친 신호는

(1001010 1101000 1101011…)
↓ reverse 연산
(0101001 0001011 1101011…)

이 된다. 수신단에서는 수신 신호를 reverse 연산을 거친 후 n 비트씩 나누어 각각의 n 비트와 해밍 거리가 가장 짧은 코드워드를 2^k개의 코드워드들의 집합 중에서 선택한다. 이후의 알고리즘은 수신 신호가 'Inverse' 되어 있는 경우와 동일한 과정을 거친다.

5. 수신 신호가 두 개의 서로 다른 해밍 코드로 부호화 된 경우

수신 신호가 두 개의 서로 다른 해밍 코드로 부호화 되었다면 지금까지 제시한 임의의 제너레이터 행렬이 만들어낸 코드워드와의 최소 해밍 거리를 구하는 알고리즘으로는 부호화 과정에 사용된 제너레이터 행렬을 찾을 수 없게 된다. 부호화 과정에

사용된 두 개의 서로 다른 해밍 코드를 위한 제너레이터 행렬을 각각 A, B라 한다면 수신 신호는 다음의 세 가지 경우로 나누어 생각해 볼 수 있다.

Case 1) 제너레이터 행렬 A와 B에 의해 부호화 된 부분의 길이가 비슷한 경우

Case 2) 제너레이터 행렬 A에 의해 부호화 된 부분의 길이가 훨씬 더 길 경우

Case 3) 제너레이터 행렬 B에 의해 부호화 된 부분의 길이가 훨씬 더 길 경우

본 논문에서는 위의 세 가지 경우와 같이 수신 신호가 서로 다른 두 개의 제너레이터 행렬에 의해 부호화 되었다고 판단 될 경우, 새로운 알고리즘을 적용하여 두 개의 제너레이터 행렬을 모두 찾아내고 또한 두 번째 제너레이터 행렬에 의해 부호화가 시작 된 위치를 'location'이라고 할 경우 location 위치도 정확히 찾아낼 수 있도록 하였다.

새로운 알고리즘은 크게 4단계로 나눌 수 있다.

1) 10개의 코드워드를 기준으로 location 위치를 대략적으로 찾아낸다.

2) 각각의 코드워드를 기준으로 location 위치를 정확히 찾아낸다.

3) Location 위치보다 앞 부분에서 사용된 제너레이터 행렬 A를 찾아낸다.

4) Location 위치보다 뒷 부분에서 사용된 제너레이터 행렬 B를 찾아낸다.

새로운 알고리즘을 위와 같이 4단계로 나눈 이유는 알고리즘의 복잡도를 낮추고 알고리즘 수행 속도를 빠르게 하며 기존에 제작하였던 알고리즘을 이용할 수 있게 해주기 위함이다.

각 단계에 대하여 자세히 알아보면

1) 10개의 코드워드를 기준으로 location 위치를 대략적으로 찾아낸다.

첫 단계는 대략적인 location 위치를 찾아내는 과정이다. 여기서 location 위치란 제너레이터 행렬 A에 의해 부호화 된 부분과 제너레이터 행렬 B에 의해 부호화 된 부분의 경계 지점을 말한다. 우선 수신 신호의 처음 10개의 코드워드를 임의의 제너레이터 행렬이 만들어낸 코드워드와 비교해서 서로 다른 코드워드의 개수를 구한다. 기존의 알고리즘과 다른 점은 최소 해밍 거리를 구하는 것이 아니라 최소의 서로 다른 코드워드 개수를 구하는 것이다. 만약 부호화 과정에서 사용된 제너레이터 행렬을 이용하여 코드워드를 만들었지만 코드워드의 시작점이 정확히 일치하지 않는다면 10개의 코드워드 중 서로 다른 코드워드의 개수는 약 10개에 조금 못 미치지만 거의 10개에 근접한 개수일 것이다. 만약 코드워드의 시작점이 정확히 일치하지만 부호화 과정에서 사용된 제너레이터 행렬을 이용하여 코드워드를 만들지 않았다면 10개의 코드워드 중 서로 다른 코드워드의 개수는 약 10개에는 조금 못 미치지만 거의 10개에 근접한 개수일 것이다. 만약 부호화 과정에서 사용된 제너레이터 행렬을 이용하여 코드워드를 만들었고 코드워드의 시작점이 정확히 일치하며 채널을 거치는 동안 에러가 하나도 발생하지 않았다면 10개의 코드워드 중 서로 다른 코드워드의 개수는 0개일 것이고, 부호화 과정에서 사용된 제너레이터 행렬을 이용하여 코드워드를 만들었고 코드워드의 시작점이 정확히 일치하지만 채널을 거치는 동안 에러가 발생하였다면 BER이 낮은 상황에서 서로 다른 코드워드의 개수는 0개보다는 크지만 거의 0개에 가까운 값을 가질 것이다. 따라서 코드워드의 정확한 시작 위치와 부호화 과정에서 이용된 제너레이터 행렬을 가지고 위의 알고리즘을 수행할 경우 수신 신호 10개의 코드워드 중 비교가 되는 제너레이터 행렬에 의해 만들어진 코드워드와 서로 다른 코드워드의 개수는 매우 작을 것이다. 만약 서로 다른 코드워드의 개수가 일정한 역치값 이하라면 그때의 코드워드의 시작 위치와 제너레이터 행렬을 옳다고 판단하고 수신 신호의 다음 10개의 코드워드에 대해 같은 과정을 반복한다. 수신 신호의 다음 10개의 코드워드에 대해 같은 과정을 반복하였을 때 제너레이터 행렬에 의해 만들어진 코드워드와 서로 다른 코드워드의 개수가 일정한 역치값 이하라면 채널에 의해 발생한 에러를 고려하더라도 동일한 제너레이터 행렬에 의해 부호화 된 부분이라도 판단할 수 있다.

위의 과정을 반복하였을 때 처음으로 수신 신호의 10개의 코드워드 중 제너레이터 행렬에 의해 만들어진 코드워드와 서로 다른 코드워드의 개수가 일정한 역치값 이상이 나오게 되면, 그 순간 알고리즘은 수신 신호의 10개의 코드워드 혹은 그 전의 10개의 코드워드 중 한 부분에서 두 번째 제너레이터 행렬에 의해 부호화 된 부분이 시작된다고 판단을 내리고 다음 과정으로 넘어가게 된다.

2) 각각의 코드워드를 기준으로 location 위치를 정확히 찾아낸다.

두 번째 단계에서는 첫 번째 단계에서 구한 대략적인 location 위치를 기준으로 정확한 location의 위치를 찾아낸다. 첫 번째 단계에서 대략적인 location의 위치를 현재 10개의 코드워드 혹은 그 전 10개의 코드워드라고 알려주었기 때문에 두 번째 단계에서는 그 전 10개의 코드워드부터 다음의 과정을 수행한다. 그 전의 10개의 코드워드 중 마지막 역치값 이하의 코

드워드부터 두 번째 제너레이터 행렬에 의해 부호화 되었기 때문에 그 전 10개의 코드워드 중 마지막 역치값 이하의 코드워드부터 처음으로 첫 번째 제너레이터 행렬에 의해 만들어진 코드워드와 서로 다른 코드워드를 찾아낸다. 예를 들어 역치값이 '2'라면 그 전 10개의 코드워드 중 마지막 1개 혹은 2개의 코드워드가 두 번째 제너레이터 행렬에 의해 부호화 되었더라도 첫 번째 제너레이터 행렬과 서로 다른 그 전 10개의 코드워드와의 개수는 역치값 이하일 것이다. 따라서 두 번째 알고리즘에서는 그 전 10개의 코드워드 중 마지막 '역치값' 이하의 코드워드부터 처음으로 첫 번째 제너레이터 행렬에 의해 만들어진 코드워드와 달라지는 지점을 찾아낸다.

위와 같이 location 위치 추정 방식을 두 단계로 나누어 줌으로서 알고리즘의 수행 시간을 낮추어 줄 수 있고 (10개씩 처리하므로) 만약 채널에 의해 에러가 생겨 location의 정확한 위치를 찾지 못하게 되더라도 추정된 location의 위치는 정확한 location의 위치와 비교해 코드워드 10개의 거리 이하에 있게 되므로 원래 메시지와의 차이를 최소화 해 줄 수 있다.

3) Location 위치보다 앞 부분에서 사용된 제너레이터 행렬 A를 찾아낸다.

첫 번째 단계에서 제너레이터 행렬 A를 찾아내었지만 동일한 코드워드를 만들어 낼 수 있는 제너레이터 행렬의 후보군을 모두 찾아내기 위하여 세 번째 단계를 수행한다. 첫 번째 단계에서 코드워드의 정확한 시작 위치를 찾아내었기 때문에 기존의 알고리즘을 적용해 수신단에서 사용한 제너레이터 행렬이 송신단에서 부호화하는데 사용한 제너레이터 행렬 A와 일치하지 않는다면 기존의 알고리즘으로 구한 해밍 거리의 평균값은 BER의 값보다 커질 것이다. 수신단에서 사용한 제너레이터 행렬이 송신단에서 부호화 하는데 사용한 제너레이터 행렬 A와 일치한다면 기존의 알고리즘으로 구한 해밍 거리의 평균값은 BER 값과 동일할 것이다. BER 값은 기존 알고리즘의 역치값보다 낮으므로 이때의 제너레이터 행렬을 옳다고 판단한다. 이렇게 찾아낸 제너레이터 행렬 A를 이용하여 복호화 과정을 통해 원래 메시지의 앞 부분을 복원해 낼 수 있다.

4) Location 위치보다 뒷 부분에서 사용된 제너레이터 행렬 B를 찾아낸다.

네 번째 단계는 location 위치보다 뒷 부분에서 사용된 제너레이터 행렬 B를 찾아내는 단계이다. 두 번째 단계에서 location의 정확한 시작 위치를 찾아내었기 때문에 기존의 알고리즘을 적용해 수신단에서 사용한 제너레이터 행렬이 송신단에서 부호화하는데 사용한 제너레이터 행렬 B와 일치하지 않는다면 기존의 알고리즘으로 구한 해밍 거리의 평균값은 BER의 값보다 커질 것이다. 수신단에서 사용한 제너레이터 행렬이 송신단에서 부호화 하는데 사용한 제너레이터 행렬 B와 일치한다면 기존의 알고리즘으로 구한 해밍 거리의 평균값은 BER 값과 동일할 것이다. BER 값은 기존 알고리즘의 역치값보다 낮으므로 이때의 제너레이터 행렬을 옳다고 판단한다. 이렇게 찾아낸 제너레이터 행렬 B를 이용하여 복호화 과정을 통해 원래 메시지의 뒷 부분을 복원해 낼 수 있다. 새로운 알고리즘의 세 번째와 네 번째 단계에서는 기존의 알고리즘을 그대로 이용해 주기 때문에 알고리즘의 활용도를 높여주었고, 이미 검증된 기존 알고리즘의 성능을 새로운 알고리즘에서도 똑같이 기대해 줄 수 있다. 또한 새로운 알고리즘 개발을 위한 복잡성도 낮추어 주었다.

지금까지 수신 신호가 두 개의 서로 다른 제너레이터 행렬에 의해 부호화 되었을 경우 적용하는 새로운 알고리즘에 대하여 알아보았다. 새로운 알고리즘을 이용하면 수신 신호가 어떠한 형태를 가지더라도 두 개의 제너레이터 행렬을 모두 알아낼 수 있고 처음 코드워드가 시작하는 위치, location의 위치를 모두 정확히 알아낼 수 있다. 따라서 이를 바탕으로 복호화 과정을 통해 원래 메시지를 추정해 낼 수 있다.

Ⅳ. 결 론

본고에서는 미지의 해밍 코드로 부호화되어 있는 미지의 비트로 이루어진 정보에서 코드의 정보를 알아내고 미지의 비트를 검출할 수 있는 정확도가 높은 알고리즘을 연구하였다. 이 알고리즘은 수신 신호 앞에 프리앰블이 있는 경우나 수신 신호가 Inverse 또는 Reverse 되어있는 등의 다양한 수신 신호의 상태에도 모두 대처할 수 있게 설계되었다. 복잡도가 낮고 확장이 용이하며 수신 신호의 코드워드 시작점을 쉽게 알 수 있는 장점을 가지고 있으며 두 개의 제너레이터 행렬에 의해 부호화된 신호도 판별할 수 있게 하여 활용도의 측면에서도 큰 강점을 가지는 알고리즘이라고 할 수 있다. 따라서 미지의 신호로부터 코드 정보를 알아낼 수 있는 본 알고리즘은 일반 사용자의 셀룰러 시스템 뿐 아니라 군사용 통신 체계에 이용되기 적합하며, 국내외 학계에서 활발히 연구되지 않은 분야인 만큼 관련 기술 수준의 시작점과 더불어 국방 기술 수준 향상에 상당히 기여할 것으로 본다.

참 고 문 헌

[1] Davies R. W.” The Data Encryption standard in

perspective,"Computer Security and the Data Encryption Standard, pp. 129-132.(http://www.nist.gov/aes).

[2] Shu Lin and Daniel J. Costello, Jr., Error Control Coding. Upper Saddle River, New Jersey: PEARSON Prentice Hall, 2004

[3] Jorge Castineira Moreira and Patrick Guy Farrell, Essentials of Error-Control Coding. New York: Wiley, 2006

약 력

이 인 석

2014년 고려대학교 정보통신대학 컴퓨터 · 통신공학부 학사
2014년~현재 고려대학교 일반대학원 전파통신공학전공 석 · 박사통합과정
관심분야: 무선통신, 네트워크

오 성 준

1991년 KAIST 전기 및 전자공학 학사
1995년 KAIST 전기 및 전자공학 석사
2000년 University of Michigan EECS 박사
2000년~2003년 Senior Engineer, Ericsson CDMA Systems, San Diego, CA, USA.
2003년~2007년 Staff Engineer, Qualcomm CDMA Technologies, San Diego, CA, USA.
관심분야: 무선통신, 네트워크, 군통신

고 영 채

1997년 2월: 한양대학교 공학사
1999년 5월: 미네소타대학교 공학석사
2001년 10월: 미네소타대학교 공학박사
2000년 12월 ~ 2004년 2월: Texas Instruments Inc. 책임연구원
관심분야: 무선 통신 시스템 설계