

효율적인 블록 인터리버 파라미터 블라인드 추정 기법

정회원 정진우*, 최성환*, 종신회원 윤동원*, 정회원 박철순**, 윤상범***

Efficient Blind Estimation of Block Interleaver Parameters

Jinwoo Jeong*, Sunghwan Choi* *Regular Members*, DongWeon Yoon* *Lifelong Member*
 Cheolsun Park**, Sangbom Yoon*** *Regular Members*

요약

최근 블라인드 신호 복원에 대한 연구가 활발히 진행되고 있으며, 주로 블록 채널 부호화된 신호의 선형성에 대한 가우스-조던 소거(Gauss-Jordan elimination)를 적용하여 인터리버 파라미터를 추정한다. 그러나 가우스-조던 소거를 이용할 때 추정하고자 하는 인터리버의 주기가 커질수록 그 주기의 제곱배 이상의 연속적인 데이터가 필요하게 된다. 본 논문에서 제안하는 알고리즘은 기존의 인터리버 파라미터 추정 알고리즘에서 필요로 했던 입력 데이터 수의 15%만을 이용하며, 추정에 필요한 데이터를 충분히 확보하지 못했을 경우에도 적용이 가능하다. 또한 제안하는 알고리즘의 임의 신호 생성에 적용된 채널 부호화와 인터리버의 특징을 이용하면 기존의 알고리즘에서 분석해야 했던 인터리버 주기의 개수를 80% 가까이 줄일 수 있으며 인터리버의 종류와 행렬 크기뿐만 아니라 해당 채널부호화의 종류까지 추정 가능하다.

Key Words : blind estimation, Gauss elimination, block interleaver

ABSTRACT

Recently, much research on blind estimation of the interleaver parameters has been performed by using Gauss-Jordan elimination to find the linearity of the block channel code. When using Gauss-Jordan elimination, the input data to be calculated needs to run as long as the square multiple of the number of the interleaver period. Thus, it has a limit in estimating the interleaver parameters with insufficient input data. In this paper, we introduce and analyze an estimation algorithm which can estimate interleaver parameters by using only 15 percent of the input data length required in the above algorithm. The shorter length of input data to be calculated makes it possible to estimate the interleaver parameters even when limited data is received. In addition, a 80 percent reduction in the number of the interleaver period candidates increases the efficiency of analysis. It is also feasible to estimate both the type and size of the interleaver and the type of channel coding.

I. 서론

일반적으로 무선통신 시스템에서는 군집오류의 분산과 데이터의 암호화를 위해 인터리버가 사용된다. 무선채널 상에서 발생하는 군집오류는 채널부호

화 기법만으로는 보상하기 힘들며 인터리버를 추가함으로써 채널부호화의 성능을 개선시킨다^[1]. 인터리버는 채널 부호화된 심벌 단위의 부호어(code word) 내에서 발생하는 군집오류를 분산시키기 위해서 송신될 비트들의 순서를 특정 규칙에 따라 바

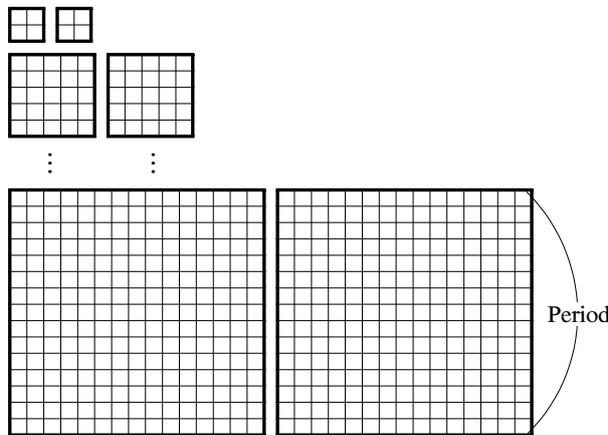
* 한양대학교 전자컴퓨터통신공학과 이동 및 우주통신 연구실(dwyoona@hanyang.ac.kr), (° : 교신저자)

** 국방과학연구소(csun@add.re.kr), *** LIG넥스원 주식회사(Sangbom.Yun@lignex1.com)

논문번호 : KICS2011-11-567, 접수일자 : 2011년 11월 30일, 최종논문접수일자 : 2012년 4월 19일

꾸어 놓는다. 따라서 수신 단에서 인터리버의 종류, 블록의 행, 열과 같은 인터리버의 파라미터 정보를 가지고 있지 않다면 순서가 뒤바뀐 비트들을 원래 상태로 디인터리빙 하기 어려우며 이러한 경우는 일종의 암호화 과정과 같다.

최근 미지의 신호에 대해서 인터리버의 파라미터 정보를 추출하고 이를 바탕으로 신호를 복원하는 연구가 활발히 진행되고 있다²⁻⁵⁾. 기존의 연구는 가우스-조던 소거를 이용하여 특정 블록 부호의 선형성을 찾는 방법으로써 채널 부호화된 데이터는 인터리빙 과정 후에도 선형성이 남아있다는 특성을 이용한다. 그러나 가우스-조던 소거를 이용할 때 인터리버의 주기가 커질수록 그 주기의 제곱배 이상의 연속적인 데이터가 필요하다²⁾.



(Period × Period × 2)

그림 1. 인터리버 주기에 따른 연산에 필요한 비트수 증가
Fig. 1. The increasing bits for operation according to the interleaver period

그림 1은 인터리버 주기에 따른 연산에 필요한 비트수 증가를 나타낸 것이다. 분석에 필요한 입력 데이터의 수는 추정하려는 블록 인터리버의 주기가 커질수록 주기의 제곱배 이상으로 늘어난다. 데이터를 연산하는데 걸리는 시간의 경우 하드웨어의 성능에 따라 단축될 수 있지만 입력 데이터의 경우 필요한 비트수를 충족시키지 못하게 되면 가우스-조던 소거 연산이 불가능하므로 기존의 인터리버 파라미터 추정 알고리즘을 사용할 수 없다.

본 논문에서는 인터리버 파라미터 분석에 필요한 미지의 데이터 수를 기존의 인터리버 파라미터 추정 알고리즘에서 필요로 하는 데이터 수의 15%만을 이용하여 분석에 필요한 입력 데이터 수가 부족한 상황에서도 인터리버 파라미터 추정이 가능하도록 한다. 또한 기존의 인터리버 파라미터 추정 알고리즘에서 분석해야 했던 인터리버 주기의 개수를 80% 가까이 줄여 분석의 효율성을 높인다. 마지막으로 기존의 인터리버 파라미터 추정 알고리즘에서는 인터리버의 파라미터 추정과 복원이 가능한 반면, 본 논문에서 제안하는 알고리즘은 인터리버의 파라미터 추정 및 복원은 물론 해당 채널부호화의 종류까지 추정이 가능하다.

본 논문의 구성은 다음과 같다. II장에서는 추정하고자 하는 블록 인터리버의 특징을 살펴보고 III장에서는 임의로 생성한 데이터를 이용하여 블록 인터리버의 파라미터와 채널 부호화를 추정하는 기법을 제안한다. IV장에서는 진행된 모의실험 결과를 분석하며 V장에서 결론을 맺는다.

II. 블록 인터리버의 특징

인터리버의 종류는 크게 블록 인터리버와 컨볼루션 인터리버로 나눌 수 있다. 컨볼루션 인터리버는 병렬로 구성된 시프트 레지스터(shift register)를 이용하여 데이터의 순서를 바꾼다. 이에 비해 블록 인터리버는 데이터의 순서를 바꾸기 위해 블록 형태의 저장 공간을 만들어 데이터를 저장하고 특정 규칙에 따라 순서를 바꾸어 출력한다. 따라서 블록 인터리버의 종류와 블록의 행과 열의 크기 따라 각각 다른 결과를 얻는다.

본 논문에서 다루고자 하는 인터리버는 매트릭스 인터리버와 헬리컬 스캔 인터리버로써 두 인터리버 모두 블록 인터리버에 속한다. 그림 2는 매트릭스 인터리버의 출력 방식을 나타낸 것이다. 매트릭스 인터리버는 특정 크기의 행과 열을 갖는 매트릭스에 데이터를 행 방향 순서로 저장하고 열 방향 순서로 출력한다. 즉, 출력은 [1, 5, 9, 13, 17, 2, 6, ... 12, 16, 20]으로 나타난다.

1	2	3	4
5	6	7	8
9	10	11	12
13	14	15	16
17	18	19	20

그림 2. 매트릭스 인터리버의 출력
Fig. 2. The output of matrix interleaver

헬리컬 스캔 인터리버의 경우 매트릭스 인터리버와 같이 특정 크기의 행과 열을 갖는 매트릭스를 사용하지만 데이터의 순서를 바꾸는 방법에서 행과 열의 방향이 아닌 대각선 방향을 이용한다. 그림 3은 헬리컬 스캔 인터리버의 출력 방식을 나타낸 것이다. 행 방향 순서로 저장된 데이터는 대각선 방향 순서로 출력이 된다. 즉, 출력은 [1, 6, 11, 16, 5, 10, 15, ... 17, 2, 7, 12]으로 나타난다.

1	2	3	4
5	6	7	8
9	10	11	12
13	14	15	16
17	18	19	20

그림 3. 헬리컬 스캔 인터리버의 출력
Fig. 3. The output of helical scan interleaver

일반적으로 채널 부호화 과정에서는 k 비트의 메시지어(message word)에 n-k 비트의 잉여 비트(redundant bit)를 추가시켜 n 비트의 부호어를 발생시킨다. 그림 4는 채널 부호화와 인터리버 과정을 나타낸 것이다. 채널 부호화를 거친 데이터는 인터리빙 과정을 거쳐 데이터의 순서가 뒤바뀌게 된다.



그림 4. 채널 부호화와 인터리빙 과정
Fig. 4. The process of channel coding and interleaving

그림 2~3에서 알 수 있듯이 블록 인터리빙 과정은 원래의 데이터를 변화하거나 특정 비트를 삽입하지 않고 데이터의 순서만 변화시킨다. 또한 일반적으로 블록 인터리버의 행렬 크기는 신호의 원활한 송수신을 위해 부호어의 배수로 정해진다^[2]. 즉, $S = N \times n$ 와 같이 나타낼 수 있으며 여기서 S는 인터리버 행렬의 크기, N은 부호어의 개수 그리고 n은 부호어를 나타낸다. 따라서 채널 부호화를 거친 데이터가 인터리빙 과정을 통해 그 순서가 뒤바뀌더라도 부호어의 특성은 모두 남아 있게 된다. 이와 같은 블록 인터리버의 특징을 이용하여 인터리버의 파라미터를 추정 할 수 있다^[3].

다음 장에서는 위와 같은 특징을 이용하여 블록 부호화와 블록 인터리버를 거친 미지의 신호를 분석하여 블록 부호와 블록 인터리버의 종류 그리고 행렬의 크기를 추정하는 알고리즘에 대해 설명한다.

III. 블록 인터리버 파라미터 및 채널 부호화 추정

3.1. 기존의 인터리버 파라미터 추정 기법

기존의 인터리버 파라미터 추정 기법은 인터리빙된 데이터 스트림에도 블록 부호에서 발생한 정보 비트와 잉여비트간의 선형성이 남아있는 특징을 이용한다. 가우스-조던 소거 연산 후 상삼각행렬(upper triangular matrix) 형태로 나타난 매트릭스에서 선형 조합인 행은 0으로 치환되고 선형 독립인 행은 1과 0이 랜덤하게 분포하게 된다. 이 특징을 이용하여 매트릭스 내의 1의 비율을 분석하면 인터리버 파라미터를 추정할 수 있다^[5]. 다음의 식(1)은 입력 데이터의 수와 추정하고자 하는 블록 인터리버의 행렬의 크기와의 관계식을 나타낸다^[6].

$$Bits = (Period^2) \times 2 \quad (1)$$

Bits는 가우스-조던 소거 연산을 위해 필요한 입력 데이터의 수를 나타낸다. Period는 추정하려는 블록 인터리버의 행렬의 크기 즉, 주기를 나타낸다. 인터리빙 주기는 인터리빙 과정이 반복되는 한 순환을 의미하며 블록 인터리버의 경우 행렬의 크기와 일치한다. 식(1)에서 보는바와 같이 인터리버 주기의 크기가 커질수록 필요한 입력 데이터의 수도 급격히 늘어남을 확인할 수 있다.

표 1. MIL-STD-188-110 A/B 인터리버 행렬 크기
Table 1. MIL-STD-188-110 A/B Interleaver matrix size

전송속도 (bps)	Long interleaver		Short interleaver	
	행의 수	열의 수	행의 수	열의 수
2400	40	576	40	72
1200	40	288	40	36
600	40	144	40	18
300	40	144	40	18
150	40	144	40	18
75	40	144	40	18

표 1은 군 통신 표준인 MIL-STD-188-110 A/B에서 적용하고 있는 인터리버의 행렬 크기를 전송 속도별로 나타낸 것이다. 이중 최대 주기는 행 40

비트와 열 576비트 곱인 23,040비트이므로 기존의 인터리버 파라미터 추정 기법으로 가우스-조던 소거 연산을 하면 $23,040 \times 2 = 1,061,683,200$ 비트가 된다^[7]. 실제 상황에서 연속으로 1,061,683,200비트를 수집하기는 어려움으로 본 논문에서는 분석에 필요한 미지의 데이터의 수 *Bits*를 아래와 같이 나타낼 수 있다.

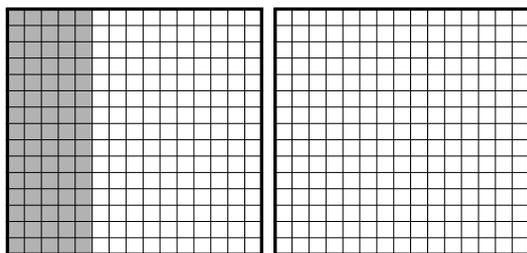
$$Bits = [0.3 \times Period] \times Period \quad (2)$$

식 (2)에서 입력 비트수는 식 (1)에서 나타난 입력 비트수의 15%로써 기존의 인터리버 파라미터 추정 알고리즘에서 필요로 했던 비트수를 85% 만큼 줄인 양이다.

다음 절에서는 입력 데이터 수를 줄이고도 신뢰성 있는 추정이 가능하도록 임의로 생성한 데이터를 적용하는 알고리즘에 대해 설명한다.

3.2. 임의로 생성한 데이터를 이용한 매트릭스 구성

기존의 인터리버 파라미터 추정 알고리즘은 인터리버의 파라미터를 추정하기 위해 가우스-조던 소거 연산을 이용한다. 이진 비트로 구성된 미지의 데이터는 가우스-조던 소거 연산을 위해 추정 주기와 같은 행 크기의 정사각행렬 매트릭스 두개로 구성되며 가우스-조던 소거 연산 후 블록 부호의 선형성을 찾는 방법을 이용한다. 그러나 인터리버의 주기가 커질수록 연산에 필요한 데이터 수가 급격히 늘어나게 되고 연산에 필요한 만큼의 입력 데이터의 수집이 되지 않은 경우에 가우스-조던 소거 연산의 적용이 불가능하게 된다. 따라서 분석에 필요한 데이터 수가 충족되지 못하였을 경우에 임의로 생성한 비트를 매트릭스 구성에 이용하는 방법을 고려한다. 이때 임의로 생성한 데이터는 미지의 데이터와 서로 상관성이 없어야 하며 미지의 데이터와 같이 채널 부호화 과정과 인터리빙 과정을 거친 데이터이어야 한다.



■ 분석에 필요한 데이터
□ 임의로 생성한 데이터

그림 5. 인터리버 파라미터 추정을 위한 매트릭스 구성
Fig. 5. Construction of matrix for Blind Estimation of interleaver parameters

그림 5는 인터리버 파라미터 추정을 위한 매트릭스 구성을 나타낸 것이다. 임의로 생성한 데이터는 분석에 필요한 입력 데이터 수가 부족할 경우 매트릭스의 빈 공간을 채우는 역할을 한다. 두 데이터의 매트릭스 내 위치는 결과에 영향을 주지 않으나 서로 다른 열에 구분되어 위치하여야 한다. 인터리버 주기가 15비트일 경우 음영으로 표시되어 있는 부분과 같이 $[0.3 \times Period] \times Period$ 즉, 75비트만을 이용하여 왼쪽부터 세로방향으로 위에서 아래로 채워지며 나머지 375비트를 임의로 생성한 비트로 왼쪽부터 세로방향으로 위에서 아래로 채우게 된다. 이러한 방식을 이용하면 분석에 필요한 데이터가 일정부분 부족하더라도 빈 공간을 임의로 생성한 데이터로 채움으로써 가우스-조던 소거 연산을 가능하게 한다.

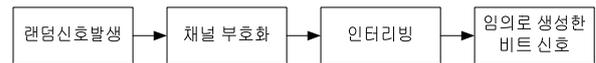


Fig. 6. 추정에 필요한 데이터 생성과정
그림 6. The process of creating data for estimation

그림 6은 추정에 필요한 데이터 생성과정을 나타낸 것이다. 임의의 데이터 생성 과정의 경우 먼저 랜덤 데이터가 생성되고 생성된 랜덤 데이터는 특정 채널 부호화와 인터리빙을 거친다. 랜덤 데이터를 이용하므로 미지의 데이터와 정보의 연관성은 없다. 그러나 생성한 데이터의 채널 부호화와 인터리버의 종류를 알고 있기 때문에 미지의 데이터와 결합된 매트릭스에서 가우스-조던 소거 연산을 통해 선형성 일치 여부를 비교 가능하다. 채널 부호화의 경우 블록 부호인 RM(Reed-Muller), Golay, BCH, Hamming 그리고 RS(Read-Solomon) 부호를 고려하였고 인터리버의 경우 블록 인터리버인 매트릭스 인터리버와 헬리컬 스캔 인터리버를 고려하였다. 채널 부호화의 종류, 인터리버의 종류 그리고 블록의 행렬 조합으로 나올 수 있는 모든 경우의 수를 고려하여 임의의 데이터를 생성한다. 이렇게 생성된 임의의 데이터들은 이후 미지의 데이터와 선형성 일치 여부를 판단하는데 사용된다.

효율적인 분석을 위해 제안하는 알고리즘의 임의 신호 생성에 적용된 채널 부호화와 인터리버간의 특성을 이용하여 기존의 인터리버 파라미터 추정 알고리즘에서 분석해야 했던 인터리버 주기의 개수

를 일정 부분 줄일 수 있다. 일반적으로 블록 인터리버의 주기는 부호어의 배수로 정해지기 때문에 블록 부호의 종류가 한정되어 있다면 인터리버의 주기 또한 한정된다. 표 2는 부호어 크기에 따른 인터리버 주기를 나타낸 것이다. 블록 부호의 경우 부호어의 크기는 정해져 있으며 부호어의 배수로 생성 가능한 인터리버의 주기는 1에서 1000비트까지 204개로 기존의 모든 주기를 고려했을 때 보다 그 대상을 약 80% 가까이 줄일 수 있다. 위와 같은 방법으로 주기를 한정시킨 상황에서는 204개의 주기만 분석하면 되므로 분석 처리 시간의 측면에서 큰 이득을 볼 수 있다.

표 2. 부호어 크기에 따른 인터리버 주기
Table 2. Interleaver period according to the code word

Code	종류	인터리버 주기(~1000bits)						
		7	98	207	320	506	713	
Hamming	(7,4)	8	104	208	322	510	720	
		14	105	210	330	512	736	
	(8,4)	15	112	216	336	525	744	
		16	115	217	345	528	756	
	(15,11)	21	119	224	352	529	759	
		23	120	225	360	540	765	
Reed Muller	(1,4)	24	126	230	368	544	768	
		28	128	231	375	552	782	
	(1,5)	30	133	232	378	555	792	
		32	135	238	384	560	800	
	(2,3)	35	136	240	390	567	805	
		40	138	245	391	570	810	
Golay	(2,4)	42	140	248	400	575	816	
		45	144	252	405	576	819	
	(23,12)	46	147	253	408	585	828	
		48	150	255	414	592	832	
	(24,12)	49	152	256	416	598	840	
		56	154	259	420	600	851	
BCH	(63,57)	60	160	264	432	608	864	
		63	161	266	435	621	874	
	(63,39)	64	165	270	437	624	882	
		69	168	272	441	630	888	
	(255,215)	70	175	273	448	640	889	
		72	176	276	450	644	896	
RS	(255,231)	75	180	280	456	648	897	
		77	182	285	460	667	912	
	(255,191)	80	184	288	464	672	920	
		84	189	296	465	675	928	
	(255,239)	88	192	299	480	690	936	
		90	195	300	483	693	945	
(255,247)	91	196	304	495	696	960		
	92	200	312	496	700	980		
		96	203	315	504	704	992	

3.3. 인터리버 파라미터 및 채널 부호화 추정

미지의 신호를 분석하여 블록 부호와 블록 인터리버의 종류 그리고 행렬의 크기를 추정하는 과정을 그림 7의 순서도에 나타냈다. 먼저, 미지의 데이터와 임의로 생성한 데이터는 가우스-조던 소거

연산을 위해 정사각행렬 매트릭스로 구성된다. 이때 매트릭스의 크기는 추정하는 인터리버 주기의 크기를 기준으로 구성된다. 미지의 데이터가 정사각행렬의 열 방향 순서대로 채워지며 빈 공간에는 임의로 생성한 데이터가 채워진다.

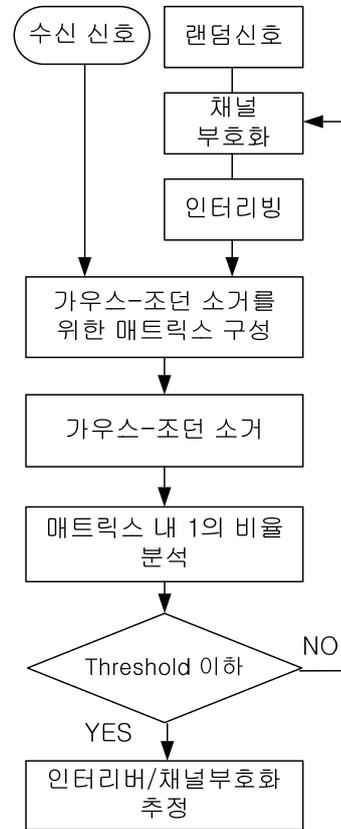


그림 7. 추정 순서도
Fig. 7. The estimating flow chart

그림 8은 추정이 성공 했을 때의 잉여 비트의 위치를 나타낸 것이며 그림 9는 추정이 실패 했을 때의 잉여 비트의 위치를 나타낸 것이다. 구성된 매트릭스는 가우스-조던 소거 연산을 통해 상삼각행렬 형태로 나타난다. 이때 만약 미지의 데이터와 임의로 생성한 데이터의 블록 부호와 블록 인터리버의 종류 그리고 행렬의 크기가 모두 일치하는 경우, 그림 8에서처럼 음영으로 표시된 잉여 비트가 하나의 행에 모여 분포하며 잉여 비트들은 모두 0으로 치환되고 정보 비트는 0과 1이 0.5의 확률로 나타나게 된다. 그러므로 매트릭스 내에 1의 비율은 잉여 비트가 정보 비트와 구별되지 않아 0으로 치환되지 않았을 때 보다 줄어들게 된다. 이와 반대로 미지의 데이터와 임의로 생성한 데이터의 블록 채널 부호, 블록 인터리버의 종류 그리고 행렬의 크기 중 하나

라도 일치하지 않았을 경우 그림 9와 같은 매트릭스가 구성되어 1의 비율이 0.5에 근접하게 된다. 이러한 경우 임의로 생성시킨 다른 조합의 데이터를 이용하여 1의 비율이 임계값 이하를 만족할 때까지 위와 같은 과정을 반복한다. 마지막으로 1의 비율이 임계값 이하인 후보 조합 중에서 1의 비율이 가장 낮은 조합을 최종 추정 조합으로 선택한다.

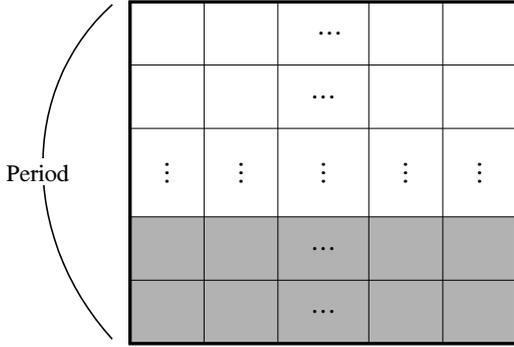


그림 8. 추정이 성공 했을 때 잉여비트의 위치
Fig. 8. The position of redundant bits when estimation succeeded

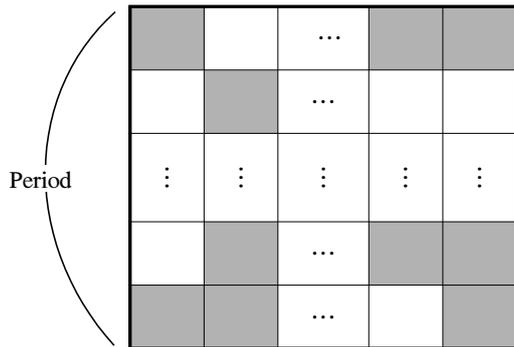


그림 9. 추정이 실패 했을 때 잉여비트의 위치
Fig. 9. The position of redundant bits when estimation failed

IV. 모의실험 및 성능분석

이 장에서는 블록 채널 부호와 블록 인터리버의 종류 그리고 행렬의 크기를 추정하는 모의실험 결과를 나타냈다. 그림 10은 (7, 4) 해밍 부호와 (7x2) 매트릭스 인터리버를 거친 데이터의 인터리버 파라미터 추정이 성공 했을 때의 매트릭스 결과이다. 분석에 필요한 데이터 수는 $[0.3 \times Period] \times Period$ 으로 총 56비트가 사용되었고 나머지는 임의로 생성한 데이터가 336비트 채워졌다. 두 데이터의 채널 부호화와 인터리버 조합이 일치하였을 경우에 그림 10의 음영으로 표

시한 부분과 같이 좌측은 상삼각행렬 형태로 나타나고 우측은 잉여 비트가 모두 0으로 치환된 형태

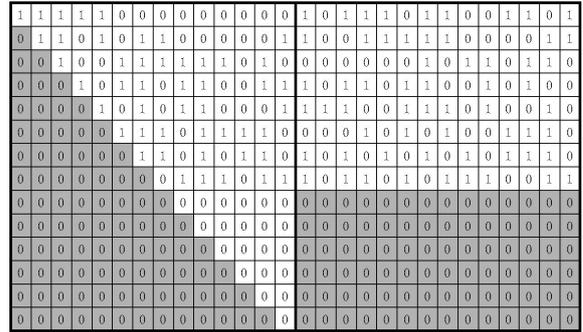


그림 10. 추정이 성공 했을 때의 매트릭스 형태
Fig. 10. The matrix form when estimation succeeded

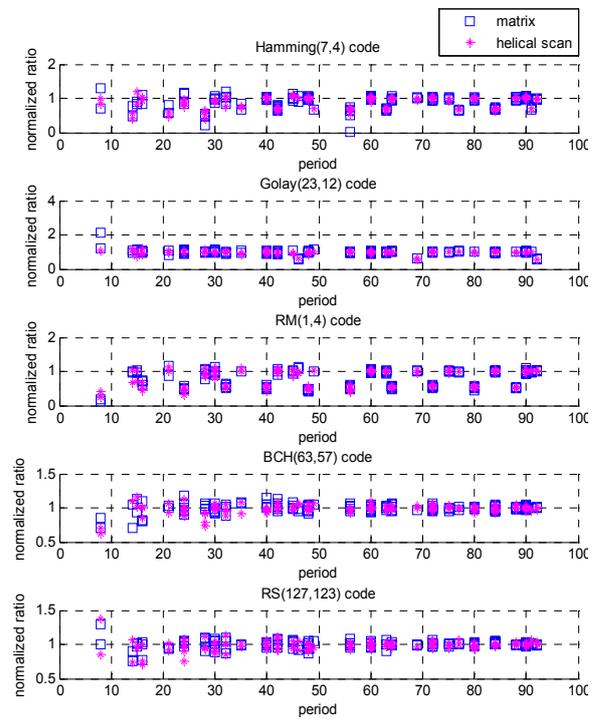


그림 11. 조합 추정치에 따른 정규화 된 1의 비율
Fig. 11. Normalized ratio of 1 according to the estimating combinations

그림 11은 (7, 4) 해밍 부호와 (8x7) 매트릭스 인터리버를 거친 데이터를 분석한 결과 그래프이다. 총 5가지 종류의 채널 부호화를 이용하였고 매트릭스 인터리버와 헬리컬 스캔 인터리버 기호를 다르게 하여 표시하였다. 매트릭스 내의 1의 비율은 그래프 상에서 비교하기 쉽도록 정규화 하여 가장 낮은 비율일 때를 0으로 0.5일 때를 1로 표기하였다.

각각의 주기마다 행렬 크기가 다른 여러 조합이 나타났으며 정규화 된 1의 비율은 대부분 1에 근접해 있음을 알 수 있었다. 그러나 추정 조합이 미지의 데이터의 블록 부호와 블록 인터리버의 종류

로 나타난다. 인터리버 주기가 14비트이고 부호어가 7비트이므로 한 열에는 0으로 치환된 잉여비트가 모두 6비트가 발생함을 확인하였다. 그리고 행렬의 크기와 모두 일치할 때는 1의 비율이 0에 근접한다. 그림 11의 첫 번째 그래프의 주기 56비트 지점의 1의 비율이 0에 가장 근접해 있음을 알 수 있고 이 지점의 조합이 미지의 데이터 조합과 일치함을 확인하였다.

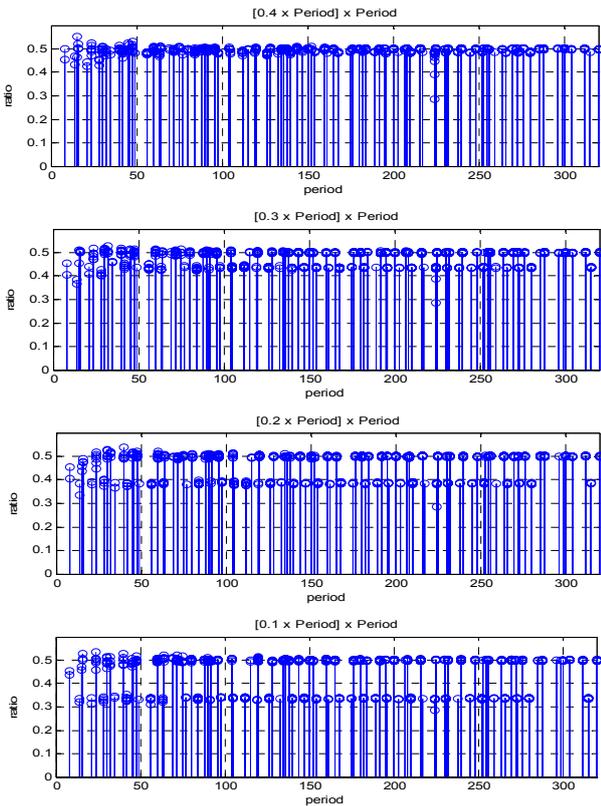


그림 12. 분석에 필요한 데이터 수에 따른 1의 비율
Fig. 12. Ratio of 1 according to the number of data for estimation

그림 12에서는 (7, 4) 해밍 부호와 (7x32) 매트릭스 인터리버를 거친 데이터를 분석하면서 그 수에 변화를 주어 1의 비율을 비교하였다. 첫 번째 그래프에서 미지의 데이터 수를 $[0.4 \times Period] \times Period$ 만큼 이용하였고 그림의 그래프마다 데이터 수를 차례로 데이터 수를 10%씩 줄여가며 실험하였다. 그래프에 나타난 것처럼 분석에 사용하는 데이터 수가 점차 줄어들수록 해당 주기 224비트의 추정 지점이 다른 주기 지점과 구분하기 힘들어진다. 그 이유는 미지의 데이터와 임의로 생성한 데이터를 조합하여 매트릭스를

만드는 과정에서 임의로 생성한 데이터의 비율이 커질수록 미지의 데이터의 채널 부호 특성이 임의로 생성한 데이터와 동일하게 상쇄되기 때문이다. 따라서 미지의 데이터의 수는 최소한 $[0.3 \times Period] \times Period$ 이상 되어야 블록 부호와 블록 인터리버의 종류 그리고 행렬의 크기 추정이 가능하다.

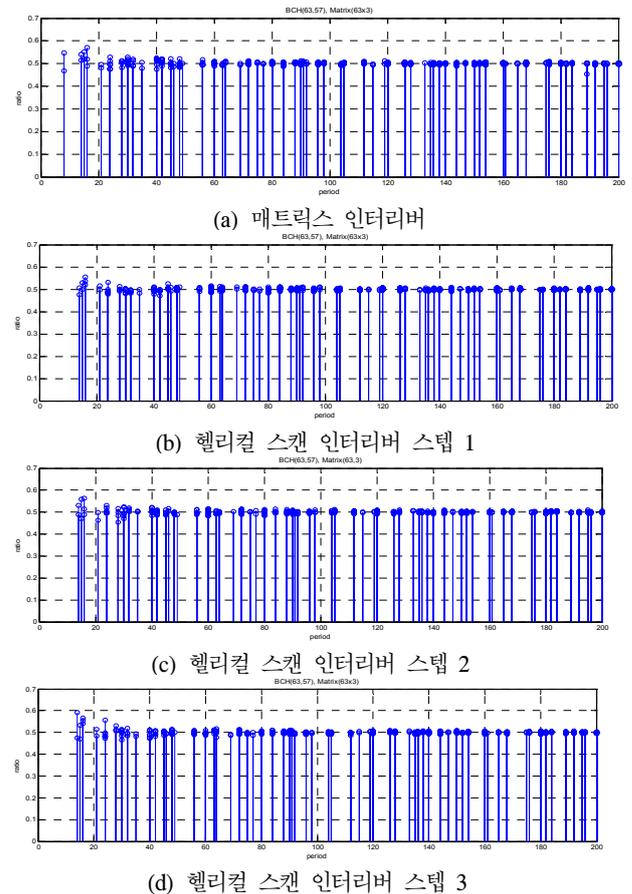


그림 13. BCH(63,57), Matrix(63x3) 추정
Fig. 13. BCH(63,57), Matrix(63x3) estimation

그림 13은 BCH(63,57)로 채널 부호화되고 매트릭스 인터리버(63x3)로 인터리빙된 데이터 스트림을 분석한 그래프이다. 헬리컬 스캔 인터리버의 총 3가지 스텝 크기와 매트릭스 인터리버로 인터리빙을 수행하여 최종 1의 비율차이를 살펴보았다. 그림 13 (b), (c), (d)에서는 모든 주기에서 1의 비율이 0.5에 수렴하여 잉여 비트가 한 행에 정렬되어 있지 않음을 알 수 있다. 그러나 그림 13 (a)에서는 추정하려는 인터리버의 주기 189에서 1의 비율이 가장 낮게 나타남으로써 해당 파라미터 추정이 가능하였다.

V. 결 론

일반적으로 무선통신 시스템에서는 인터리버가 사용되고 있으며 수신 단에서 인터리버의 파라미터 정보가 없다면 인터리빙 과정은 일종의 암호화 과정과 같다.

본 논문에서는 효율적인 블록 인터리버 파라미터 추정 알고리즘을 제시하였으며 블라인드 추정 성능을 분석하였다. 인터리버 파라미터 분석에 필요한 미지의 데이터 수를 기존의 인터리버 파라미터 추정 알고리즘에서 필요로 했던 데이터 수의 15%만을 이용하고 인터리버의 종류, 행렬 크기 그리고 해당 채널 부호의 종류를 추정하는 방법을 제안하였다. 분석에 필요한 미지의 데이터 수가 부족한 상황에서도 임의로 생성한 데이터를 이용하여 가우스-조던 소거 연산이 가능하도록 하였으며 제안한 알고리즘의 임의 신호 생성에 적용된 채널 부호화와 인터리버의 특징을 이용하여 기존의 인터리버 파라미터 추정 알고리즘에서 분석해야 했던 인터리버 주기의 개수를 80% 가까이 줄일 수 있었다. 모의실험을 통해 가우스-조던 소거 연산 후 매트릭스 내에서의 1의 비율을 비교하여 원하는 파라미터 추정이 가능함을 확인하였고 임의로 생성한 데이터와 분석에 필요한 데이터를 합한 매트릭스 구성에 있어서 필요한 미지의 데이터 수를 최소로 하여 분석이 가능한 비율을 제시하였다.

본 논문에서 제시한 기법은 미지의 신호에서 인터리버 파라미터 정보를 추정하고 분석하여 신호를 복원하는 블라인드 신호처리 기법 연구에 있어서 보다 효율적인 분석과 향상된 추정 성능에 기여할 것으로 기대한다. 또한 향후 실제 일반 통신환경에서 발생하는 오류를 고려하여 가우스-조던 소거 기법으로 매트릭스 내의 비트 간 선형성을 찾을 수 있는 최소한의 조건과 기존의 인터리버 파라미터 추정 알고리즘에서 필요로 했던 데이터 수를 얼마만큼 줄일 수 있는지에 대한 연구가 지속되어야 할 것이다.

참 고 문 헌

[1] B. Sklar, *Digital Communications: Fundamentals and Applications*, Prentice-Hall International Inc., 2001

[2] G. Burel and R. Gautier, "Blind estimation of

encoder and interleaver characteristics in a noncooperative context," *IASTED International Conference on Communications, Internet and Information Technology*, pp. 275-280, AZ, USA, Nov. 2003

[3] G. Sicot and S. Houcke, "Blind detection of interleaver parameters," *IEEE International Conference on Acoustics, Speech and Signal Processing*, Vol. 3, pp. 829-832, PA, USA, Mar. 2005

[4] Liru Lu, Kwok Hung Li and Yong Liang Guan, "Blind Detection of Interleaver Parameters for Non-Binary Coded Data Streams," *IEEE International Conference on Communications*, pp. 1-4, Dresden, Germany, Jun. 2009

[5] Jubyoung Lee, Jeonghoon Jeong, Dongweon Yoon, Sangbom Yun, Youngmi Park, Sunpil Nah, "Blind Reconstruction of Helical Scan Interleaver Parameters," *Agency for Defense Development General Conference*, pp.360-362, Agency for Defense Development, Aug. 2010

[6] G. Strang, *Linear Algebra and Its Applications*, Thomson Learning Inc., 1988.

[7] *MIL-STD-188-110 A/B*, IHS, pp. 29-46, April. 2000

정진우 (Jinwoo Jeong)

정회원



1999년 2월 한양대학교 전자통신 전파공학과 학사
 2001년 2월 한양대학교 전자통신 전파공학과 석사
 2001년~2007년 LG전자 디지털 미디어연구소 선임연구원
 2007년 9월~현재 한양대학교

전자통신공학과 박사과정

<관심분야> 이동통신, 위성통신, 전자전체계

최 성 환 (Sunghwan Choi)

정회원



2010년 2월 한양대학교 정보통신 공학과 학사
 2012년 2월 한양대학교 전자컴퓨터통신공학과 석사
 2012년~현재 현대자동차 정보통신 선형연구팀
 <관심분야> 차량IT, 전자전체계

박 철 순 (Cheolsun Park)

정회원



1989년 경기대학교 전자계산학과 학사
 1991년 인하대학교 전자계산공학과 석사
 1991년~현재 국방과학연구소 책임연구원
 1997년 전자계산 조직응용 기술사

2007년 충남대학교 정보통신공학과 박사
 <관심분야> 신호처리, 통신응용

윤 동 원 (Dongweon Yoon)

중신회원



1989년 2월 한양대학교 전자통신 공학과 학사
 1992년 2월 한양대학교 전자통신 공학과 석사
 1995년 2월 한양대학교 전자통신 공학과 박사

1995년 3월~1997년 8월 동서대학교 정보통신 공학과 전임강사
 1997년 2월~1997년 12월 한국전자통신연구소 초빙연구원
 1997년 9월~2004년 2월 대전대학교 정보통신 공학과 부교수
 2002년 11월~2005년 12월 한국전자통신연구원 초빙연구원
 2004년 3월~현재 한양대학교 융합전자공학부 교수
 <관심분야> 통신이론, 무선 및 이동통신, 위성 및 우주통신

윤 상 범 (Sangbom Yoon)

정회원



1999년 고려대학교 제어계측 공학과 학사
 2002년 고려대학교 전기공학과 석사
 2002년~현재 LIG넥스원 전자전 연구센터 수석연구원
 <관심분야> 통신, 신호처리