논문 2011-48TC-9-2

길쌈 부호 복원 기법을 이용한 블라인드 블록 디인터리빙

(Blind Block Deinterleaving using Convolutional Code Reconstruction Method)

정 진 우*, 윤 동 원**, 박 철 순***, 윤 상 범****, 이 상 현*****

(Jinwoo Jeong, Dongweon Yoon, CheolSun Park, Sangbom Yun, and Sanghyeon Lee)

요 약

채널 부호화된 심볼은 연집 오류(Burst error)에 취약한 특성으로 인해 송신단에서 인터리빙 과정을 거쳐 송신된다. 수신단에서는 채널을 통하여 발생된 연집 오류를 디인터리빙 과정을 통해 랜덤 오류(Random error)로 변화시키고 채널 복호화를 통하여 오류 정정 효과를 높인다. 그러나 수신단에서 송신단의 인터리버 파라미터를 알지 못하는 경우에는 인터리빙은 특정 패턴에 의한 일종의 암호화로 볼 수 있으며 디인터리빙이 어렵게 된다. 최근 선형 블록 부호가 블록 인터리빙 되었을 때 선형 블록 부호의 선형성을 이용하여 인터리버 파라미터를 추정하는 기법들의 연구가 진행되었다. 그러나 길쌈 부호가 인터리빙 되었을 때, 선형 블록 부호와 달리 부호어 길이에 따라 구분되는 선형성을 이용할 수 없기 때문에 기존 선형 블록 부호의 선형성을 이용한 방법으로는 인터리버의 파라미터를 추정할 수 없다. 본 논문에서는 길쌈부호가 블록 인터리빙이 되었을 때 길쌈 부호 복원(Convolutional code reconstruction) 기법을 이용하여 블록 인터리버의 행과 열을 추정하는 블라인드 블록 디인터리빙 기법을 제안한다.

Abstract

Interleaving is applied to prevent from exceeding the error-correction capability of channel code. At the receiver, burst errors are converted into random errors after deinterleaving, so the error-correction capability of channel code is not exceeded. However, when a receiver does not have any information on parameters used at an interleaver, interleaving can be seen as an encryption with some pattern. In this case, deinterleaving becomes complicated. In the field of blind deinterleaving, there have recently been a number of researches using linearity of linear block code. In spite of those researches, since the linearity is not applicable to a convolutional code, it is difficult to estimate parameters as in a linear block code. In this paper, we propose a method of blind block deinterleaving using convolutional code reconstruction method

Keywords : Convolutional code, Blind block deinterleaving, Convolutioinal code reconstruction method

Ⅰ. 서 론

디지털 통신에서는 다양한 왜곡 요소를 갖는 채널로 인해 발생하는 오류를 정정하기 위하여 채널 부호화 기법이 요구된다. 송신단에서 정보비트에 중복비트를 추가해서 전송함으로써 잡음이나 다른 요소들로 발생된 오류들을 수신단에서 정정하게 된다. 그러나 채널의 특성에 따라 연집 오류(Burst error)가 발생하게되면 채널 부호의 오류 정정 효율이 떨어지게 된다. 이러한 현상

* 정회원, 한양대학교
(Hanyang University)
** 평생회원-교신저자, 한양대학교
(Hanyang University)
*** 정회원, 국방과학연구소
(Agency for Defense Development)
**** 정회원, LIG 넥스원
(LIG Nex1 Co., Ltd.)
***** 정회원, 대한항공
(Korean Airline Co., Ltd.)
접수일자: 2011년5월17일, 수정완료일: 2011년9월16일

을 막기 위해 대부분의 통신시스템에서 인터리버를 사용한다. 송신단에서 인터리빙을 수행하고 수신단에서는 디인터리빙을 수행함으로써 연집 오류를 랜덤 오류로 변화시켜 채널 부호의 오류 정정 효율이 떨어지는 것을 막을 수 있다. 그러나 수신자가 송신단의 인터리버에 대한 정보가 없는 경우, 인터리빙은 특정 패턴에 의한 일종의 암호화가 될 수 있다. 이러한 경우, 해당 수신자는 미지의 신호에 대한 정보를 알아내기 위해 송신단의 인터리버 파라미터에 대한 정보 없이 디인터리빙을 수행하여야 한다. 이와 관련하여, 최근 블라인드 디인터리빙 기법에 대한 다양한 연구가 진행되어 왔으며, 특히 블록 인터리빙된 선형 블록 부호에 대하여 선형 블록 부호가 가지는 선형성을 이용하여 인터리버의 파라미터를 추정하는 연구들이 주로 이루어졌다[1]-[3]. 그러나 블록 인터리빙된 길쌈 부호의 경우에는 부호어 길이에 따라 구분되는 선형성을 이용할 수 없기 때문에 기존의 선형 블록 부호에 대한 블록 블라인드 디인터리빙 기법으로는 인터리버 파라미터의 추정이 제한적이다.

이에 본 논문에서는 길쌈 부호를 복원하는 기법을 이용하여 블록 인터리빙된 길쌈 부호의 인터리버의 파라미터를 추정하는 기법을 제안한다. Ⅱ장에서는 기존 연구와 길쌈부호 및 복원 기법에 대해 설명하고 Ⅲ장에서 제안하는 기법과 모의실험을 통하여 도출된 결과를 분석하고 Ⅳ장에서 결론을 맺는다.

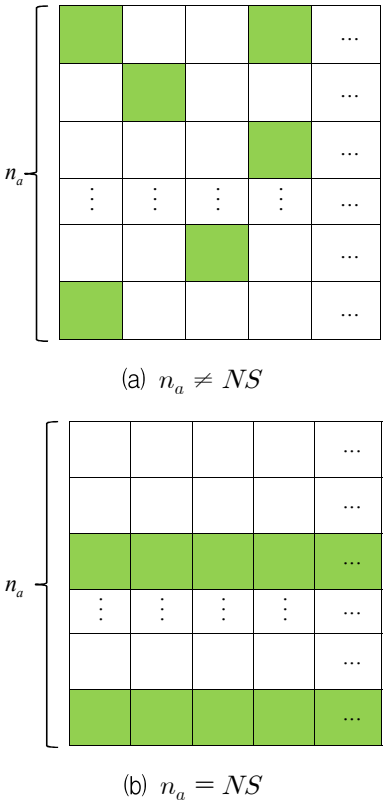
Ⅱ. 블라인드 블록 디인터리빙과 길쌈 부호 복원 기법

1. 블라인드 블록 디인터리빙

기존의 블라인드 블록 디인터리빙 기법은 선형 블록 부호가 블록 인터리빙 되었을 때, 인터리버 주기(n_a)를 추정하였다[1~3]. [1]에서는 Rank를 이용하여 선형 블록 부호의 선형성을 찾아내고, [2], [3]에서는 채널에 의한 잡음 및 오류를 고려하여 Gauss 소거법을 이용하였다. 이때, 인터리버 주기는 인터리빙 과정이 반복되는 하나의 블록 크기를 의미하고 복수개의 부호어로 구성된다.

그림 1은 (n,k) 선형 블록 부호가 기존 [1]-[3]에서의 인터리버 주기 추정 방식을 이용하여 인터리버 주기를 추정할 때, 추정값 n_a에 따른 행렬의 중복비트의 분포를 나타내었다. N은 인터리버 주기, S는 자연수를 나타내고 그림의 음영 부분은 중복비트를 나타낸다.

그림 1 (a)와 같이 $n_a \neq NS$ 인 경우, 매트릭스 내에서 중복비트가 정렬되지 않으나, 그림 1 (b)와 같이 $n_a = NS$인 경우에서는 중복비트가 동일한 행으로 정렬되어 행렬 내에 행간 선형성이 나타나며, 이로 인하여 Rank가 감소한다. 이때 추정된 주기가 송신단의 인터리버 주기가 되며, 이것을 이용하여 행과 열을 추정할 수 있게 된다.



(a) $n_a \neq NS$

(b) $n_a = NS$

그림 1. 인터리버 주기 추정에 따른 중복비트의 분포
Fig. 1. Distribution of redundant bit according to interleaver period estimation.

2. 길쌈 부호 복원 기법

길쌈 부호의 복원 기법은 부호율이 1/2이고 채널오류가 없는 제한적인 경우에 대해서 Rice에 의해 처음 제안되었고[4]. 채널 오류가 있는 경우의 모든 부호율에 대해서 Filiol에 의해 일반화되었다[5]. 본 절에서는 부호율에 따라 부호기의 출력과 생성 수열의 관계식을 이용한 부호율 판별식 유도과정을 살펴본다.

부호율이 1/2인 길쌈 부호기를 가정하면, 출력 C_1과 C_2는 다음과 같이 나타낼 수 있다.

$$C_1 = m * f_1 \tag{1}$$

$$C_2 = m * f_2 \tag{2}$$

여기서 f_1과 f_2는 부호기의 생성 수열을 나타낸다. 이 때 위의 두 식에서 메시지 m을 제거하기 위해서 다음 식과 같이 전개된다.

$$C_1 * f_2 = m * f_1 * f_2 \tag{3}$$

$$C_2 * f_1 = m * f_2 * f_1 \tag{4}$$

식 (3)과 식 (4)에서 보는 바와 같이 두 식의 결과는 같으며, 따라서 위의 두 식은 다음과 같이 m을 제거한 형태로 나타낼 수 있다.

$$C_1 * f_2 \oplus C_2 * f_1 = 0 \tag{5}$$

따라서 위의 식은 식 (6)과 같이 $Cf^T = 0$의 형태로 나타낼 수 있다.

$$\begin{pmatrix} c_{K-1}^{(2)} & c_{K-2}^{(2)} & \cdots & c_0^{(2)} & c_{K-1}^{(1)} & c_{K-2}^{(1)} & \cdots & c_0^{(1)} \\ c_K^{(2)} & c_{K-1}^{(2)} & \cdots & c_1^{(2)} & c_K^{(1)} & c_{K-1}^{(1)} & \cdots & c_1^{(1)} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ c_{t-1}^{(2)} & c_{t-2}^{(2)} & \cdots & c_{t-K-2}^{(2)} & c_{t-1}^{(1)} & c_{t-2}^{(1)} & \cdots & c_{t-K-2}^{(1)} \end{pmatrix} \begin{pmatrix} f_{1,0} \\ f_{1,1} \\ \vdots \\ f_{1,K-1} \\ f_{2,0} \\ f_{2,1} \\ \vdots \\ f_{2,K-1} \end{pmatrix} = 0 \tag{6}$$

부호율이 1/2인 길쌈 부호의 경우 식 (6)을 항상 만족시키며, 오류가 포함되어도 식 (6)을 정해진 일정 구간을 이동하면서 반복 수행하여 만족하는 빈도를 확인함으로써 길쌈 부호 여부를 판별할 수 있다. 이 때, 식 (6)이 해를 갖기 위한 조건은 행렬 C의 rank가 $2K-1$ 이하의 값 이어야 하며, 따라서 t는 $t \geq 3K-2$가 되도록 설정하여야 한다. 여기서 K는 구속장의 길이를 나타낸다. 따라서 복원을 위해서 필요한 수신 비트의 최소길이는 다음과 같다.

$$L_{\min} = 2(3K-2) \tag{7}$$

단, 식 (7)은 채널오류가 없는 경우로 가정한다.

부호율이 1/3, 1/4 인 경우에도 위의 부호율 1/2에서와 같은 특성을 이용하여 다음의 식 (8)과 식 (9)를 유도할 수 있다[5].

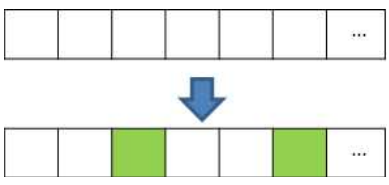
$$\begin{cases} C_1 * f_2 + C_2 * f_1 = 0 \\ C_1 * f_3 + C_3 * f_1 = 0 \\ C_2 * f_3 + C_3 * f_2 = 0 \end{cases} \tag{8}$$

$$\begin{cases} C_1 * f_2 + C_2 * f_1 = 0 \\ C_1 * f_3 + C_3 * f_1 = 0 \\ C_1 * f_4 + C_4 * f_1 = 0 \\ C_2 * f_3 + C_3 * f_2 = 0 \\ C_2 * f_4 + C_4 * f_2 = 0 \\ C_3 * f_4 + C_4 * f_3 = 0 \end{cases} \tag{9}$$

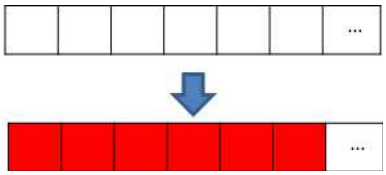
Ⅲ. 인터리빙된 길쌈 부호의 블라인드 블록 디인터리빙

길쌈 부호의 경우, 블록 또는 부호어의 길이가 선형 블록 부호와는 달리 부호율에 따라 정의되지 않는다. 길쌈 부호화 과정에서 부호기의 출력은 단지 현재 입력뿐만 아니라, 그 이전의 $K-1$개 입력에 의해서도 영향을 받는다. 즉, 선형 블록 부호는 그림 2 (a)와 같이 정보비트와 중복비트가 합해져서 특정 위치에 분포하게 되지만 길쌈 부호는 그림 2 (b)와 같이 정보비트가 특정한 패턴에 따라 완전히 다른 형태로 변하게 된다. 따라서 기존의 블록 블라인드 디인터리빙 기법에서와 같이 특정 위치에 있는 중복비트와 정보비트간의 선형성을 이용하여 인터리버 주기를 추정하는 것은 불가능하다.

따라서 본 논문에서는 인터리빙된 길쌈 부호의 인터리버 파라미터를 추정하기 위하여 앞서 언급한 길쌈 부호 복원 기법을 이용하는 새로운 기법을 제안한다. 즉, 수신된 스트림을 디인터리빙하여 식 (5), (8), (9)의 만족 여부를 확인한 후, 그 결과에 따라 송신단의 인터리버 행과 열을 추정하고 인터리빙 되기 전의 스트림으로 복원하는 기법이다.



(a) 선형 블록 부호화



(b) 길쌈 부호화

그림 2. 선형 블록 부호화와 길쌈 부호화
Fig. 2. Linear block coding and convolutional coding.

송신단에서 길쌈 부호 스트림이 부호율 $1/2$로 인터리빙 된 경우, 수신단의 디인터리버 파라미터가 송신단과 일치하는 경우에는 식 (5)를 만족하지만, 일치하지 않을 경우에는 판별식의 결과가 식 (10)과 같이 0이 아닌 값을 가지게 된다.

$$C_1 * f_2 \oplus C_2 * f_1 \neq 0 \tag{10}$$

또한 송신단에서 길쌈 부호 스트림이 부호율 1/3, 1/4로 인터리빙된 경우에도 수신단의 디인터리버 파라미터가 송신단과 일치하지 않으면 다음의 식 (11), (12)와 같이 하나 이상의 식이 0이 아닌 값을 가지게 된다.

$$\begin{cases} C_1 * f_2 + C_2 * f_1 \neq 0 \ \text{ or} \\ C_1 * f_3 + C_3 * f_1 \neq 0 \ \text{ or} \\ C_2 * f_3 + C_3 * f_2 \neq 0 \end{cases} \tag{11}$$

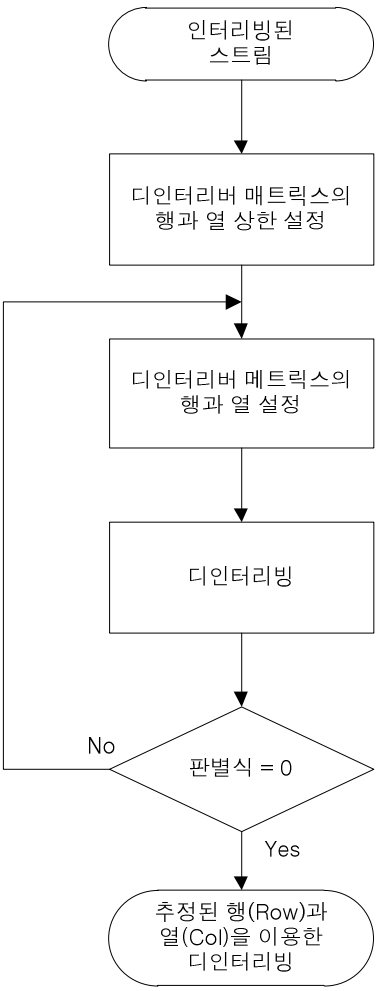


그림 3. 인터리빙된 길쌈 부호의 인터리버 파라미터 추출 순서도

Fig. 3. Flow chart for interleaver parameter extraction.

$$\begin{cases} C_1 * f_2 + C_2 * f_1 \neq 0 \ \text{ or} \\ C_1 * f_3 + C_3 * f_1 \neq 0 \ \text{ or} \\ C_1 * f_4 + C_4 * f_1 \neq 0 \ \text{ or} \\ C_2 * f_3 + C_3 * f_2 \neq 0 \ \text{ or} \\ C_2 * f_4 + C_4 * f_2 \neq 0 \ \text{ or} \\ C_3 * f_4 + C_4 * f_3 \neq 0 \end{cases} \tag{12}$$

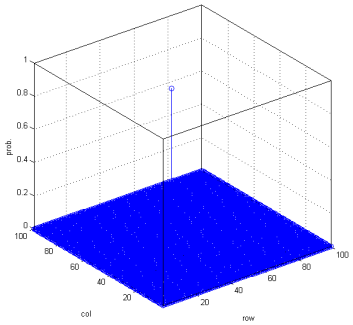
인터리빙된 길쌈 부호의 이러한 특성을 이용한 인터리버 파라미터 추출 순서는 그림 3과 같이 나타낼 수 있다.

그림에서와 같이 수신단에 인터리빙된 길쌈 부호 스트림이 수신되면, 먼저 수신된 스트림에 적용할 디인터리버의 행과 열의 상한 값을 설정한다. 그리고 수신된 스트림에 적용할 디인터리버의 행과 열을 앞서 설정한 상한값 내에서 변화시키면서 디인터리빙을 수행하고, 디인터리빙된 스트림에 대해서 판별식(식 (5), (8), (9))의 만족여부를 확인한다. 여기서 판별식 (5), (8), (9)를 만족하게 되면 수신단에서는 추정된 행과 열의 크기를 이용하여 디인터리빙을 수행하여 인터리빙되기 전의 스트림을 복원하고, 만족하지 않으면 디인터리버의 행과 열을 계속 변화시키면서 판별식을 만족할 때 까지 위의 과정들을 반복 수행한다.

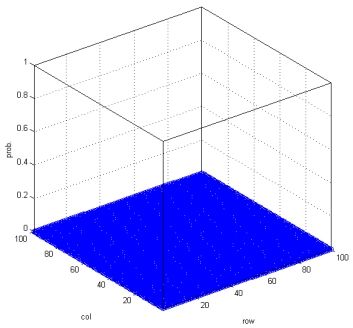
그림 4와 그림 5에 블록 인터리빙된 길쌈 부호의 인터리버 행과 열을 추정하는 모의실험 결과를 나타내었다. 두 실험 모두 신뢰성 확보와 결과 값의 검증을 위하여 실제 통신 시스템에서 사용 중이며 다양한 인터리버 크기를 권고하는 MIL-STD 110B 표준을 기준으로 진행되었다. MIL-STD 110B에서 권고하는 인터리버의 행과 열의 크기는 다음의 표 1과 같다.

실험 방법은, 먼저 부호율을 $1/2$, $1/3$, $1/4$ 중 하나를 랜덤 선택 후 표 1에 표시된 긴 인터리버와 짧은 인터리버 각각의 크기를 랜덤 선택하여 스트림을 생성한다. 여기에서 구속장의 길이 $K=7$로 설정한다. 생성된 스트림에 대하여 긴 인터리버의 경우에는 600×600을 인터리버 크기의 상한으로, 짧은 인터리버의 경우에는 100×100을 인터리버 크기의 상한으로 설정하여 판별식 (5), (8), (9)에 적용하여 인터리버 크기를 찾는다.

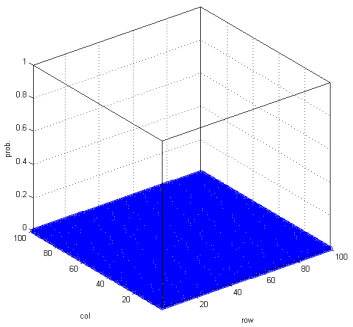
그림 4에는 표 1의 짧은 인터리버가 적용된 길쌈부호에 대해서 디인터리버의 행과 열의 크기를 100×100까지 증가시키면서 디인터리빙된 스트림을 식 (5), (8), (9)에 적용한 결과를 (a), (b), (c)에 각각 나타내었다. 위의 그림 (a)에서 보듯이 모의실험에 사용된 스트림이 판별식 (5)를 만족함으로써 부호율이 $r=1/2$이며 행과



(a) $r=1/2$



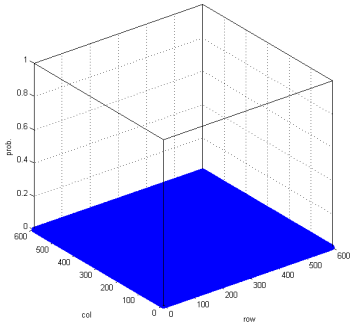
(b) $r=1/3$



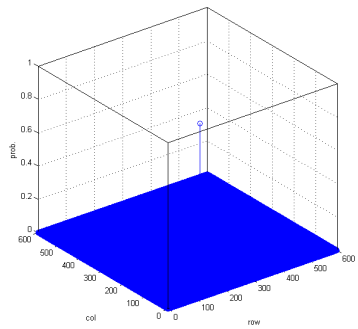
(c) $r=1/4$

그림 4. 40×36으로 인터리빙된 길쌈부호($r=1/2$, $K=7$)의 디인터리버 행과 열의 추정

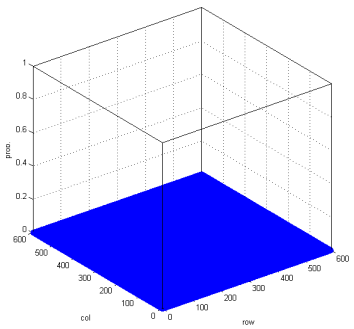
Fig. 4. Estimating deinterleaver row and column of convolutional code($r=1/2$, $K=7$) interleaved with 40×36.



(a) $r=1/2$



(b) $r=1/3$



(c) $r=1/4$

그림 5. 40×144로 인터리빙된 길쌈부호($r=1/3$, $K=7$)의 디인터리버 행과 열의 추정

Fig. 5. Estimating deinterleaver row and column of convolutional code($r=1/3$, $K=7$) interleaved with $40\times144\,40\times36$.

열이의 크기가 40×36으로 인터리빙이 되었음을 확인할 수 있다. 그러나 판별식 (8), (9)에 적용할 경우에는 (b), (c)와 같이 만족하지 않는 것을 확인할 수 있다. 그림 5에는 표 1의 긴 인터리버가 적용된 길쌈부호에 대해서 디인터리버의 행과 열의 크기를 600×600 까지 증가시키면서 디인터리빙된 스트림을 식 (5), (8), (9)에 적용한 결과를 (a), (b), (c)에 각각 나타냈었다.

위의 그림 (b)에서 보듯이 모의실험에 사용된 스트림이 판별식 (8)를 만족함으로써 부호율이 $r=1/3$이며 행과 열이의 크기가 40×144으로 인터리빙이 되었음을 확인할 수 있다. 그러나 판별식 (5), (9)에 적용할 경우에는 (a), (c)와 같이 만족하지 않는 것을 확인할 수 있다. 이와 같이 여러 가지 길쌈 부호의 부호율과 인터리버의 크게 대해서도 본 논문에서 제안하는 디인터리

표 1. MIL-STD 110B의 인터리버 매트릭스 크기
Table 1. Interleaver matrix size of MIL-STD 110B.

긴 인터리버		짧은 인터리버	
행의 수	열의 수	행의 수	열의 수
40	576	40	72
40	288	40	36
40	144	40	18
20	36	10	9

버 파라미터 추정 기법을 이용하면 블록 인터리버의 행과 열이 추정 가능하고, 추정된 인터리버 파라미터를 바탕으로 인터리빙되기 전의 스트림으로 정확한 복원이 가능함을 모의실험을 통하여 확인할 수 있다.

Ⅳ. 결 론

본 논문에서는 길쌈 부호 복원 기법을 이용하여, 인터리빙된 길쌈 부호의 인터리버 파라미터를 추정하는 기법을 제안하였다. 기존의 블록 블라인드 디인터리빙 연구에서는 부호어의 정보비트와 중복비트의 선형성을 이용하기 때문에 길쌈 부호와 같이 정보비트와 중복비트사이의 선형성이 보장 되지 않는 경우에는 인터리버 파라미터 추정이 제한적이었다. 그러나 본 논문에서 제안한 길쌈 부호 복원 기법을 이용하면 길쌈 부호가 인터리빙된 경우에 대해서도 수신된 스트림에 대해서 디인터리빙과 그에 따른 판별식의 만족 여부 확인을 거쳐서 인터리버 파라미터의 추정이 가능함을 모의실험을 통하여 확인하였다. 이와 더불어 향후, 인터리버의 메모리 크기가 데이터 프레임의 약수배로 구성되는 특징을 이용한다면 인터리버의 경우의 수와 그에 따른 인터리버 파라미터 추정 속도를 크게 단축할 수 있을 것으로 기대된다.

참 고 문 헌

[1] G. Burel and R. Gautier, “Blind estimation of encoder and interleaver characteristics in a non cooperative context,” *Proceedings of the Second IASTED International Conference on Communications*, Internet and Information Technology, AZ, USA, pp. 275-280, Nov. 2003.

[2] G. Sicot and S. Houcke, “Blind detection of interleaver parameters,” *Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing*, PA, USA, pp. iii/829-iii/832, Mar. 2005.

[3] L. Liru, et al., “Blind Detection of Interleaver Parameters for Non-Binary Coded Data Streams,” *Proceedings of IEEE International Conference on Communications*, Dresden, GER, pp. 1-4, Jun. 2009.

[4] B. Rice, “Determining the parameters of a rate convolutional code over GF(q),” *Proceedings of Third International Conference on Finite Fields and Applications*, Glasgow, UK, 1995.

[5] E. Filiol, “Reconstructions of convolutional encoders over GF(q),” *LNCS 1355*, Springer Verlag, pp. 100-110, 1997.

저 자 소 개

정 진 우(정회원)
1999년 한양대학교 전자통신 전파공학과 학사
2001년 한양대학교 전자통신 전파공학과 석사
2001년～2007년 LG전자 디지털 미디어연구소 선임연구원
2007년～현재 한양대학교 전자통신공학과 박사과정
<주관심분야 : 이동통신, 위성 및 우주통신>

윤 동 원(평생회원)-교신저자
1989년 한양대학교 전자통신 공학과 학사
1992년 한양대학교 전자통신 공학과 석사
1995년 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
<주관심분야 : 통신이론, 무선 및 이동통신, 위성 및 우주통신>

박 철 순(정회원)
1989년 경기대학교 전자계산학과 학사
1991년 인하대학교 전자계산 공학과 석사
1991년～현재 국방과학연구소 선임연구원
1997년 전자계산 조직응용 기술사
2007년 충남대학교 정보통신공학과 박사
<주관심분야 : 신호처리, 통신응용>

윤 상 범(정회원)
1999년 고려대학교 제어계측 공학과 학사
2002년 고려대학교 전기공학과 석사
2002년～현재 LIG넥스원 전자전 연구센터 수석연구원
<주관심분야 : 통신, 컴퓨터, 신호처리, 반도체>

이 상 현(정회원)
2007년 한양대학교 전자전기 컴퓨터공학부 학사 졸업
2009년 한양대학교 전자컴퓨터 통신공학과 석사 졸업
2009년～현재 대한항공연구소
<주관심분야 : 이동 통신, 무선 통신>