

論文2002-39TC-3-2

## 직렬연접 길쌈부호의 새로운 연접방법

## (A New Concatenation Scheme of Serial Concatenated Convolutional Codes)

裊商宰\*, 朱彦熒\*

(Sang Jae Bae and Eon Kyeong Joo)

## 요약

본 논문에서는 직렬연접 길쌈부호의 새로운 연접방법을 제시하고 그 성능을 분석하였다. 제시된 직렬연접 길쌈부호에서는 외부부호의 정보비트와 패리티비트가 각각 인터리버와 역인터리버를 거쳐서 내부부호로 입력된다. 따라서 인터리버와 역인터리버의 크기는 입력프레임의 크기와 동일하다. 제안된 직렬연접 길쌈부호에서는 인터리버의 크기가 Benedetto 구조에 비하여 반으로 줄어들기 때문에 반복복호시 매번 거쳐야 하는 인터리버의 지연시간을 줄일 수 있다. 또한 제시된 구조에서는 다중화기와 역다중화기가 사용되지 않기 때문에 복호기의 복잡도가 Benedetto 구조보다 상대적으로 더 간단하다. 모의실험 결과 인터리버의 크기가 동일한 경우 제시된 직렬연접 길쌈부호의 성능이 Benedetto 구조의 성능보다 더 우수하였다. 또한 Eb/No가 증가할수록 제시된 구조와 Benedetto 구조의 성능차이가 더욱 증가하였다. 입력프레임의 크기가 동일한 경우에는 제시된 직렬연접 길쌈부호의 성능이 인터리버의 크기가 반으로 줄어들어도 불구하고 Benedetto 구조와 거의 동일한 성능을 나타내었다.

## Abstract

In this paper, a new concatenation scheme of serial concatenated convolutional codes is proposed and the performance analyzed. In the proposed scheme, each of information and parity bits of outer code is entered into inner code through interleaver and deinterleaver. Therefore, the interleaver size is same as the length of input frame. Since the interleaver size of proposed type is reduced to half of the conventional Benedetto type, the interleaver delay time required for iterative decoding is reduced. In addition the multiplexer and demultiplexer are not used in the decoder of the proposed type, the complexity of decoder can be also reduced. As results of simulation, the performance of proposed type shows the better error performance as compared to that of the conventional Benedetto type in case of the same interleaver size. And it can be observed that the difference of BER performance is increased with the increase of Eb/No. In case of the same length of input frame, the proposed type shows almost same performance with Benedetto type despite that the interleaver size is reduced by half.

**Key Words** : serial concatenated convolutional code, iterative decoding, interleaver delay time, decoder complexity, error performance,

\* 正會員, 終身會員, 慶北大學校 電子電氣工學部

(School of Electronic and Electrical Engineering, Kyungpook National University)

接受日字:2001年11月30日, 수정완료일:2002年1月10日

## I. 서론

Berrou 등에 의해서 발표된 병렬연접 길쌈부호 (parallel concatenated convolutional codes; PCCC)는 비교적 간단한 구조를 가지면서도 반복복호에 의해서

매우 우수한 성능을 나타낸다<sup>[1]</sup>. 병렬연접 길쌈부호의 부호기는 상태수가 적은 2개 이상의 구성부호가 인터리버를 사이에 두고 병렬로 연결된 형태로 구성된다. 병렬연접 길쌈부호의 복호기는 각 구성부호에 대응되는 연관정 출력이 가능한 MAP(maximum a posteriori)<sup>[2-7]</sup> 복호기가 순환적으로 연결된 형태로 구성되므로 반복복호가 가능하다. 따라서 병렬연접 길쌈부호의 경우 수 만 비트 이상의 큰 크기를 가지는 인터리버를 사용하고 복호시 반복복호를 5회 이상 수행하면  $10^{-5}$ 의 BER에서 거의 Shannon limit에 근접하는 오류성능을 얻을 수 있다<sup>[1,8]</sup>. 그러나 병렬연접 길쌈부호에서는 반복복호 횟수와 SNR을 증가시키더라도 BER이  $10^{-5} \sim 10^{-7}$  이하로 낮아지지 않는 오류마루(error floor)가 발생하는 문제점이 있다<sup>[8-11]</sup>.

이러한 병렬연접 길쌈부호의 오류마루를 최소화하기 위하여 직렬연접 길쌈부호(serial concatenated convolutional codes; SCCC)가 Benedetto 등에 의해서 제안되었다<sup>[10]</sup>. 직렬연접 길쌈부호의 부호기는 병렬연접 길쌈부호와는 달리 외부구성부호(outer code)와 내부구성부호(inner code)가 인터리버를 사이에 두고 직렬로 연결된 형태로 구성된다. 그러므로 외부구성부호의 정보비트와 패리티비트 또는 패리티비트만 내부구성부호로 입력된다<sup>[10-12]</sup>. 따라서 직렬연접 길쌈부호의 MAP 복호기는 정보비트에 대한 LLR(log likelihood ratio)값 뿐만 아니라 패리티비트에 대한 LLR값도 계산하여야 한다<sup>[12]</sup>. Benedetto 등이 제안한 직렬연접 길쌈부호에서 인터리버의 크기는 입력프레임의 크기가 동일한 경우 병렬연접 길쌈부호에 비하여 두 배로 증가한다<sup>[10,11]</sup>. 또한 부호기와 복호기에서 다중화기(multiplexer; MUX)와 역다중화기(demultiplexer; DEMUX)가 추가적으로 사용된다<sup>[10,11]</sup>. 따라서 Benedetto 구조의 직렬연접 길쌈부호에서는 병렬연접 길쌈부호에 비하여 복호기의 복잡도와 인터리버의 지연시간 등이 증가하는 문제점이 발생한다.

그러므로 본 논문에서는 Benedetto 등이 제안한 직렬연접 길쌈부호의 문제점을 최소화하기 위하여 직렬연접 길쌈부호의 외부구성부호와 내부구성부호의 새로운 연결방법을 제시한다. 그리고 인터리버의 크기와 입력 프레임의 크기가 동일한 경우에 대하여 제안된 구조의 성능과 Benedetto 구조의 성능을 모의실험을 통하여 비교 및 분석한다.

II. 직렬연접 길쌈부호

1. 부·복호기 구조

직렬연접 길쌈부호에서는 연결방식에 따라 외부구성부호의 정보비트와 패리티비트 모두가 내부구성부호로 입력되거나 패리티비트만 내부구성부호로 입력된다<sup>[10-12]</sup>. Benedetto 등이 제안한 직렬연접 길쌈부호의 부호기와 복호기는 그림 1과 같다. 여기서  $x_k$ 는 외부구성부호의 정보비트를 나타내며, 역다중화기의 출력인  $x'_{1k}$ 와  $x'_{2k}$ 는 내부구성부호의 정보비트를 나타낸다. 그리고  $y'_k$ 는 내부구성부호의 정보비트  $x'_{1k}$ 와  $x'_{2k}$ 에 대한 패리티비트를 나타내며,  $d_k$ 는 복호기의 최종 출력비트를 나타낸다.

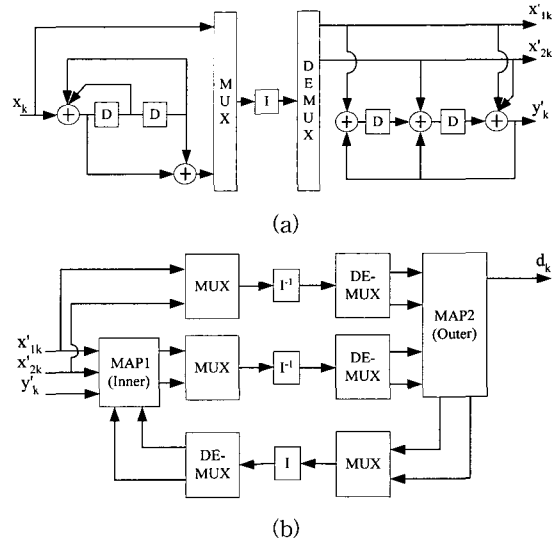


그림 1. Benedetto 형태의 직렬연접 길쌈부호

(a) 부호기

(b) 복호기

Fig. 1. Serial concatenated convolutional code of Benedetto type.

(a) Encoder

(b) Decoder

그림에서 보는 바와 같이 Benedetto 구조의 부호기에서는 외부구성부호의 정보비트와 패리티비트가 다중화기와 인터리버를 거쳐서 내부구성부호로 입력된다. 따라서 Benedetto 구조에서 인터리버의 크기는 입력프레임 크기의 두 배가 된다. 즉, 입력프레임의 크기가 동일

한 경우 Benedetto 구조의 직렬연접 길쌈부호에서 인터리버의 크기는 병렬연접 길쌈부호에 비하여 두 배로 증가한다<sup>[10,11]</sup>. 또한 Benedetto 구조의 복호기에서는 부호기의 다중화기와 역다중화기로 인하여 그에 대응되는 역다중화기와 다중화기가 부가적으로 더 사용된다<sup>[10,11]</sup>. 그러므로 Benedetto 구조의 직렬연접 길쌈부호에서는 복호기의 복잡도와 인터리버의 지연시간 등이 병렬연접 길쌈부호에 비하여 증가하는 문제점이 발생한다.

Benedetto 구조의 직렬연접 길쌈부호에서 발생하는 문제점을 줄이기 위한 새로운 연결방법의 직렬연접 길쌈부호에 대한 부호기와 복호기는 그림 2와 같다. 여기서  $x_k$ 는 외부구성부호의 정보비트를 나타내며, 인터리버를 거쳐서 내부구성부호의 정보비트  $x'_{1k}$ 로 입력된다. 또한 외부구성부호의 패리티비트는 역인터리버를 거쳐서 내부구성부호의 정보비트  $x'_{2k}$ 로 입력된다. 그리고  $y'_k$ 는 내부구성부호의 정보비트  $x'_{1k}$ 와  $x'_{2k}$ 에 대한 패리티비트를 나타낸다.

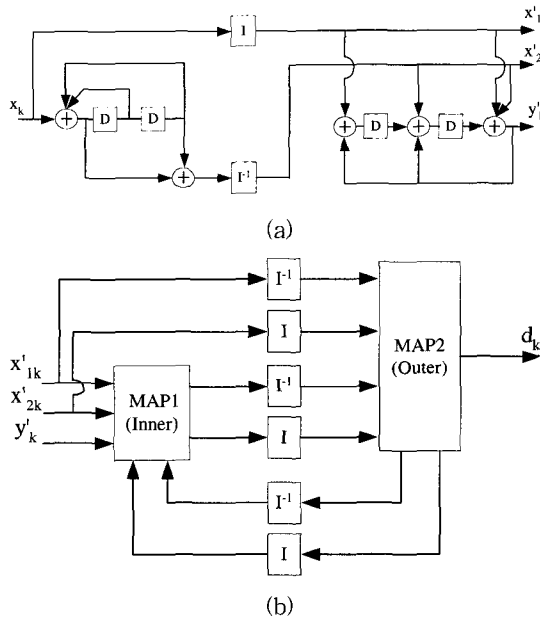


그림 2. 제안된 형태의 직렬연접 길쌈부호  
(a) 부호기  
(b) 복호기

Fig. 2. Serial concatenated convolutional code of proposed type.  
(a) Encoder  
(b) Decoder

그림에서 보는 바와 같이 외부구성부호의 정보비트

와 패리티비트가 각각 독립적으로 인터리버와 역인터리버를 거쳐서 내부구성부호로 입력된다. 따라서 제안된 구조에서 인터리버와 역인터리버의 크기는 입력프레임의 크기와 동일하다. 즉, 입력프레임의 크기가 동일한 경우 제시된 직렬연접 길쌈부호의 인터리버 크기는 Benedetto 구조의 인터리버 크기에 비하여 반으로 줄어든다. 그러므로 제안된 구조의 직렬연접 길쌈부호는 Benedetto 구조의 직렬연접 길쌈부호에서 발생하는 인터리버의 지연시간을 줄일 수 있다. 그리고 제안된 구조의 복호기에서는 Benedetto 구조의 복호기에서와는 달리 인터리버와 역인터리버만 사용된다. 즉, Benedetto 구조의 직렬연접 길쌈부호에서 사용되는 다중화기와 역다중화기가 제안된 구조에서는 사용되지 않는다. 따라서 제안된 직렬연접 길쌈부호의 복호기 복잡도는 Benedetto 구조에 비하여 상대적으로 더 간단하다.

2. 직렬연접 길쌈부호의 MAP 복호알고리즘

본 논문에 제시된 직렬연접 길쌈부호의 복호알고리즘은 병렬연접 길쌈부호의 MAP 알고리즘과 거의 동일하다. 그러나 외부 복호기 MAP2에서 정보비트에 대한 LLR값뿐만 아니라 패리티비트에 대한 LLR값도 계산하여야 한다.

외부 복호기 MAP2에서  $k$ 시간의 정보비트  $u_k$ 에 대한 LLR값은 다음 식과 같이 정의된다.

$$\Lambda(u_k) = \log \frac{P(u_k=1 | \mathbf{r}_1^N)}{P(u_k=0 | \mathbf{r}_1^N)}. \tag{1}$$

여기서  $\mathbf{r}_1^N$ 은  $r_1$ 부터  $r_N$ 까지의 수신된 비트를 나타낸다. 그리고 순방향 및 역방향 매개변수  $\alpha_k^i(m)$ 과  $\beta_k(m)$ 을 사용하면 식 (1)은 다음과 같이 정리된다.

$$\Lambda(u_k) = \log \frac{\sum_m \alpha_k^1(m) \cdot \beta_k(m)}{\sum_m \alpha_k^0(m) \cdot \beta_k(m)}. \tag{2}$$

$$\alpha_k^i(m) = P(u_k=i, S_k=m, \mathbf{r}_1^k) = \sum_m \sum_{j=0}^1 \gamma^i(\mathbf{r}_k, m', m) \cdot \alpha_{k-1}^j(m'), \tag{3}$$

$$\beta_k(m) = P(\mathbf{r}_{k+1}^N | S_k=m) = \sum_m \sum_{i=0}^1 \gamma^i(\mathbf{r}_{k+1}, m, m') \cdot \beta_{k+1}(m'). \tag{4}$$

여기서  $m$ 과  $m'$  및  $m''$ 은  $k$ 시간과  $k-1$ 시간 및  $k+1$ 시간에서의 각 상태를 나타내며,  $\gamma^i$ 는 정보비트가  $i$ 일 경우의 상태전이확률을 나타낸다.

한편  $k$ 시간에서 패리티비트  $p_k$ 에 대한 LLR값은 식 (1)과 (2)에서  $u_k$ 대신  $p_k$ 를 대입함으로써 구할 수 있다. 그리고  $p_k$ 에 대한 순방향과 역방향 매개변수는 식 (3), (4)와 동일하다. 그러나 실제 순방향 매개변수  $\alpha_k^i(m)$ 와 역방향 매개변수  $\beta_k(m)$ 의 계산과정에서는 이전 상태번호를  $p_k$ 에 대한 결로 바꾸어 주어야 한다. 따라서  $p_k$ 에 대한 상태전이확률은  $u_k$ 에 대한 상태전이확률과는 다르게 계산된다.

본 논문에서 구성부호기로 사용한 생성다항식이 [7, 5]이고 메모리가 두 개인 순환 체계적 길쌈부호(recursive systematic convolutional code)의 정보비트와 패리티비트에 대한 격자도 (trellis diagram)는 그림 3과 같다.

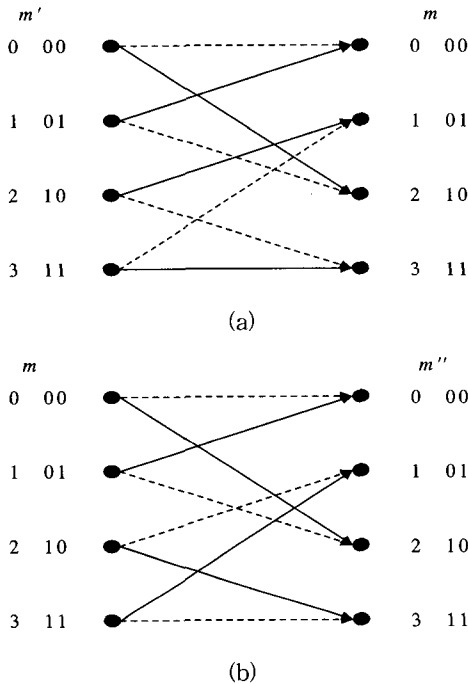


그림 3. 순환 체계적 길쌈부호의 격자도  
(a) 정보비트에 대한 격자도  
(b) 패리티비트에 대한 격자도

Fig. 3. Trellis diagram of recursive systematic convolutional code.  
(a) Trellis diagram of information bits  
(b) Trellis diagram of parity bits

여기서 점선과 실선은 입력이 각각 0과 1인 경우의 상태전이(state transition)를 나타낸다. 이 경우  $k$ 시간에서 정보비트 0과 패리티비트 0에 대한 LLR값을 구

하기 위하여 계산되는 순방향매개변수  $\alpha_k^0(m)$ 은 각각 다음과 같다.

$u_k = 0$  인 경우의  $\alpha_k^0(m)$

$$\begin{aligned}\alpha_k^0(0) &= \sum_{i=0}^1 \gamma_k^0(\mathbf{r}_k, \mathbf{0}, 0) \cdot \alpha_{k-1}^i(0) \\ \alpha_k^0(1) &= \sum_{i=0}^1 \gamma_k^0(\mathbf{r}_k, \mathbf{3}, 1) \cdot \alpha_{k-1}^i(3) \\ \alpha_k^0(2) &= \sum_{i=0}^1 \gamma_k^0(\mathbf{r}_k, \mathbf{1}, 2) \cdot \alpha_{k-1}^i(1) \\ \alpha_k^0(3) &= \sum_{i=0}^1 \gamma_k^0(\mathbf{r}_k, \mathbf{2}, 3) \cdot \alpha_{k-1}^i(2).\end{aligned}\quad (5)$$

$p_k = 0$  인 경우의  $\alpha_k^0(m)$

$$\begin{aligned}\alpha_k^0(0) &= \sum_{i=0}^1 \gamma_k^0(\mathbf{r}_k, \mathbf{0}, 0) \cdot \alpha_{k-1}^i(0) \\ \alpha_k^0(1) &= \sum_{i=0}^1 \gamma_k^0(\mathbf{r}_k, \mathbf{2}, 1) \cdot \alpha_{k-1}^i(2) \\ \alpha_k^0(2) &= \sum_{i=0}^1 \gamma_k^0(\mathbf{r}_k, \mathbf{1}, 2) \cdot \alpha_{k-1}^i(1) \\ \alpha_k^0(3) &= \sum_{i=0}^1 \gamma_k^0(\mathbf{r}_k, \mathbf{3}, 3) \cdot \alpha_{k-1}^i(3).\end{aligned}\quad (6)$$

위 식에서 보는 바와 같이 정보비트와 패리티비트에 대한 순방향 매개변수  $\alpha_k^i(m)$ 의 계산과정에서 이용되는 이전 상태번호가 서로 다름을 확인할 수 있다. 정보비트와 패리티비트에 대한 이전 상태번호가 서로 다른 것은 외부구성부호에서 입력되는 정보비트와 그에 대응되어 출력되는 패리티비트가 반드시 서로 일치하지 않기 때문에 발생한다.

### III. 모의실험 결과 및 분석

제시된 직렬연접 길쌈부호와 Benedetto 구조의 직렬연접 길쌈부호에 대한 성능을 비교하기 위하여 동일한 인터리버 크기와 입력프레임 크기에 대하여 AWGN 환경에서 모의실험을 하였다.

인터리버의 크기(N)가 동일한 경우, 즉 인터리버로 인한 지연시간이 동일한 경우에 대한 직렬연접 길쌈부호의 성능은 그림 4와 같다. 그림에서 점선은 제안된 직렬연접 길쌈부호의 성능을 나타내며, 실선은 Benedetto 구조의 직렬연접 길쌈부호에 대한 성능을 나타낸다. 그리고 제안된 구조와 Benedetto 구조에서 사용된 인터리버의 크기는 모두 1000이다.

그림에서 보는 바와 같이 인터리버의 크기가 동일한 경우 제시된 직렬연접 길쌈부호가 Benedetto 구조의 직렬연접 길쌈부호보다 더 우수한 성능을 나타냄을 알

수 있다. 예를 들면 BER이  $10^{-6}$ 인 경우 제시된 직렬연접 길쌈부호의 성능이 Benedetto 구조의 직렬연접 길쌈부호보다 약 0.5dB 정도 더 우수하다. 이러한 성능차이는 BER이 낮아질수록 더욱 증가함을 알 수 있다.

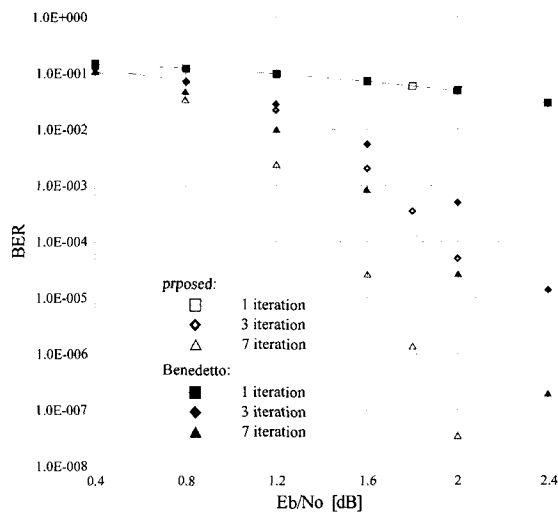


그림 4. 동일한 인터리버 크기에 대한 직렬연접 길쌈부호의 비트오율(N=1000)  
 Fig. 4. BER performance of SCCC with same interleaver size(N=1000).

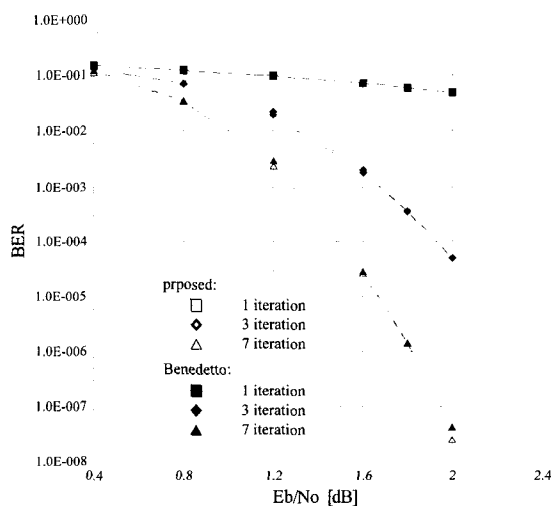


그림 5. 동일한 입력프레임 크기에 대한 직렬연접 길쌈부호의 비트오율(input frame=1000)  
 Fig. 5. BER of SCCC with same input frame length(input frame=1000).

입력프레임의 크기가 동일한 경우에 대한 직렬연접 길쌈부호의 성능은 그림 5와 같다. 그림에서 점선은 제

안된 구조의 직렬연접 길쌈부호를 나타내며, 실선은 Benedetto 구조의 직렬연접 길쌈부호를 나타낸다. 그리고 제안된 구조와 Benedetto 구조의 인터리버 크기는 각각 1000과 2000이다.

그림에서 보는 바와 같이 제시된 직렬연접 길쌈부호와 Benedetto 형태의 직렬연접 길쌈부호가 거의 동일한 성능을 나타냄을 알 수 있다. 그러나 II장에서 살펴본 바와 같이 제시된 직렬연접 길쌈부호의 인터리버 크기는 Benedetto 구조에서 사용되는 인터리버 크기의 반이다. 따라서 복호시 반복복호를 하기 위하여 매번 거쳐야 하는 인터리버의 지연시간을 줄일 수 있다. 또한 제시된 구조의 복호기에서는 다중화기와 역다중화기가 사용되지 않으므로 Benedetto 구조의 복호기보다 상대적으로 더 간단하다.

반복복호 횟수가 7회인 경우, 특정한 비트오율을 얻기 위해서 요구되는 직렬연접 길쌈부호의 Eb/No를 동일한 인터리버 크기와 입력프레임 크기에 대하여 각각 표 1과 2에 나타내었다.

표 1. 동일한 인터리버 크기에 대한 직렬연접 길쌈부호의 성능 비교(N=1000)

Table 1. Performance comparison of SCCC with same interleaver size(N=1000).

BER	Proposed type [dB]	Benedetto type [dB]
$10^{-3}$	1.28	1.58
$10^{-4}$	1.46	1.85
$10^{-5}$	1.65	2.09
$10^{-6}$	1.81	2.28

표 2. 동일한 입력프레임 크기에 대한 직렬연접 길쌈부호의 성능 비교(input frame=1000)

Table 2. Performance comparison of SCCC with same input frame length(input frame=1000).

BER	Proposed type [dB]	Benedetto type [dB]
$10^{-3}$	1.28	1.29
$10^{-4}$	1.46	1.47
$10^{-5}$	1.65	1.65
$10^{-6}$	1.81	1.81

표 1과 2에서 보는 바와 같이 인터리버의 크기가 동일한 경우에는 제시된 구조의 성능이 Benedetto 구조의 성능보다 더 우수함을 알 수 있다. 또한  $E_b/N_0$ 가 증가할수록 그 성능차이가 더욱 증가함을 알 수 있다. 그리고 입력프레임의 크기가 동일한 경우에는 제시된 구조의 인터리버 크기가 반으로 줄어들어도 불구하고 Benedetto 구조와 거의 같은 성능을 나타냄을 알 수 있다.

#### IV. 결 론

본 논문에서는 Benedetto 구조의 직렬연접 길쌈부호에서 발생하는 문제점을 해결하기 위하여 직렬연접 길쌈부호의 새로운 연결방법을 제시하였다. 그리고 인터리버의 크기와 입력프레임의 크기가 동일한 경우에 대하여 그 성능을 비교 및 분석하였다.

Benedetto 구조의 직렬연접 길쌈부호에서는 외부구성부호의 정보비트와 패리티비트가 다중화기와 인터리버를 거쳐서 내부구성부호로 입력된다. 따라서 Benedetto 구조에서 인터리버의 크기는 입력프레임 크기의 두 배가 된다. 그러나 제시된 직렬연접 길쌈부호에서는 외부구성부호의 정보비트와 패리티비트가 각각 독립적으로 인터리버와 역인터리버를 거쳐서 내부구성부호로 입력된다. 따라서 제시된 구조에서는 인터리버와 역인터리버의 크기가 입력프레임의 크기와 동일하게 된다. 즉, 제시된 직렬연접 길쌈부호에서 인터리버의 크기는 Benedetto 구조의 인터리버 크기에 비하여 반으로 줄어든다. 그러므로 복호시 반복복호를 하기 위하여 매번 거쳐야 하는 인터리버의 지연시간을 줄일 수 있다. 또한 제안된 직렬연접 길쌈부호에서는 다중화기와 역다중화기가 사용되지 않음으로 복호기의 복잡도가 Benedetto 구조에 비하여 상대적으로 더 간단해진다.

모의실험 결과 인터리버의 크기가 동일한 경우 제시된 직렬연접 길쌈부호의 성능이 Benedetto 구조의 성능보다 더 우수하였다. 또한  $E_b/N_0$ 가 증가할수록 제시된 구조와 Benedetto 구조의 성능차이가 더욱 증가하였다. 그리고 입력프레임의 크기가 동일한 경우에는 제시된 직렬연접 길쌈부호가 Benedetto 구조에 비하여 인터리버의 크기가 반으로 줄어들어도 불구하고 거의 동일한 성능을 나타내었다. 따라서 제시된 직렬연접 길쌈부호는 Benedetto 구조에서 발생하는 복호기의 복잡도와 인터리버의 지연시간 등을 줄일 수 있을 것으로 사료된다.

#### 참 고 문 헌

- [1] C. Berrou and A. Glavieux, "Near optimum error correcting coding and decoding: Turbo-codes," *IEEE Trans. Commun.*, Vol. 44, No. 10, pp. 1261~1271, Oct. 1996.
- [2] L. R. Bahl, J. Cocke, F. Jelinek, and J. Raviv, "Optimal decoding of linear code for minimizing symbol error rate," *IEEE Trans. Inform. Theory*, Vol. 20, No. 2, pp. 284~287, Mar. 1974.
- [3] S. S. Pietrobon and A. S. Barbulescu, "A simplification of the modified Bahl decoding algorithm for systematic convolutional codes," *Proc. IEEE ISITA'94*, Sydney, Australia, pp. 1073~1077, Nov. 1994.
- [4] P. Robertson, E. Villerun, and P. Hoeher, "A comparison of optimal and sub-optimal MAP decoding algorithms operating in log domain," *Proc. IEEE ICC'95*, Seattle, Washington, Vol. 2, pp. 1009~1013, June 1995.
- [5] S. Benedetto, D. Divsalar, G. Montorsi, and F. Pollara, "Soft-output decoding algorithms for continuous decoding of parallel concatenated convolutional codes," *Proc. IEEE ICC'96*, Dallas, Texas, Vol. 2, pp. 112~117, June 1996.
- [6] S. Benedetto, D. Divsalar, G. Montorsi, and F. Pollara, "Soft-output decoding algorithms in iterative decoding of turbo codes," *The TDA Progress report 42-124*, Jet Propulsion Laboratory, Pasadena, California, pp. 63~87, Feb. 15, 1996.
- [7] J. Hagenauer, "The turbo principle: tutorial introduction and state of the art," *Proc. Int. Symp. on Turbo Codes and Related Topics*, Brest, France, pp. 1~11, Sep. 1997.
- [8] S. Benedetto and G. Montorsi, "Unveiling turbo codes: Some results on parallel concatenated coding schemes," *IEEE Trans. Inform. Theory*, Vol. 42, No. 2, pp. 409~428, Mar. 1996.
- [9] D. Divsalar and F. Pollar, "On the design of

- Turbo codes," JPL TDA Progress Report, 42-123, pp. 99~121, Nov. 1995.
- [10] S. Benedetto, G. Montorsi, D. Divsalar, and F. Pollara, "Serial concatenation of interleaved codes: Performance analysis, design, and iterative decoding," JPL TDA Progress Report, Vol. 42, pp. 1~26, Aug. 1996.
- [11] S. Benedetto and G. Montorsi, "Serial concatenation of block and convolutional codes," Electron. Lett., Vol. 32, pp 887~888, May 1996.
- [12] A. Ambroze, G. Wade, and M. Tomlinson, "Iterative MAP decoding for serial concatenated convolutional codes," Proc. IEE Commun., Vol. 145, No. 2, pp. 53~59, Apr. 1998.

저 자 소 개



裴商宰(正會員)

1993년 2월 : 경북대학교 전자공학과 졸업(공학사). 1995년 2월 : 경북대학교 전자공학과 석사(M.S.). 1995년 3월~현재 : 경북대학교 전자공학과 박사과정. <주관심분야 : 터보부호, 다중 레벨 부호변조, 이동통신, WCDMA>



朱彦熒(終身會員)

1976년 : 서울대학교 공과대학 전자공학과 졸업(공학사). 1984년 : 미국 Ohio State University 전기공학과 졸업(M.S.). 1987년 : 미국 Ohio State University 전기공학과 졸업(Ph.D.). 1976년~1979년 : 해군통신전자 기술장교. 1979년~1982년 : 한국과학기술원 연구원. 1987년~현재 : 경북대학교 공과대학 전자전기공학부 재직(교수). <주관심분야 : Digital Communication Systems, Coding and Decoding, Modulation and Demodulation, Digital Signal Processing for Communications>