

디지털 통신 시스템에서의 터보부호의 성능분석

정회원 김 신 령*, 정 호 영**

The performance Analysis of Turbo Codes for Digital Communication System

Sin-Ryeong Kim *, Ho-Young Cheong** *Regular Members*

요 약

디지털 통신 시스템에서는 효율적인 통신을 수행하기 위하여 오류 정정 부호 방식을 사용하게 되는데 본 연구에서는 터보부호에 대하여 연구하고 그 성능을 분석해 보았다. 터보부호는 2개의 RSC부호를 병렬로 concatenation 한 부호로서 부가 백색 가우시안 채널에서 부호율이 1/2이고 구속장이 3인 경우에 대하여 시뮬레이션을 수행하였다.

시뮬레이션 결과 반복 횟수와 인터리버의 크기가 증가할수록 성능이 좋아지고, 랜덤 인터리버를 사용하였을 때 블록 인터리버를 사용한 경우보다 성능이 10 배 정도 우수함을 알 수 있었다.

ABSTRACT

In digital communication systems, error correcting codes have been used to control bit errors for effective communications.

In this paper, turbo code with rate 1/2 and constraint length 4 has been studied which has built using a parallel concatenation of two RSC(recursive systematic convolutional) codes.

The bit error rate performance has been simulated for BPSK on additive white Gaussian channel. As a result, it has been shown that turbo code had better performance as the number of iterations and interleaver size increased and random interleaving was better than uniform interleaving.

I. 서 론

오류 정정 부호(error correcting code) 기술이란 디지털 통신 시스템에서 채널상의 여러 요인에 의하여 발생하는 오류를 효율적으로 정정하기 위하여 사용되는 방식으로써^[1] 일반적으로 채널 부호(channel coding) 방식이라 한다. 오류 정정 부호 기술은 정보 비트에 임의의 잉여 비트(redundancy)를 덧붙여 부호어를 만들고, 수신단에서는 송신단에서 부가된 잉여 비트를 이용하여 수신된 부호어로 부터 오류가 없는 원래의 정보 비트를 복원하여 시스템의 성능을 높이는 방식이다.

오류 정정 부호 방식은 잉여 비트를 첨가하는 방식에 따라 블록 부호(block coding) 방식과 트리 부호(tree 또는 trellis coding) 방식으로 구별할 수 있다^[1].

블록 부호 방식은 정보 비트에 대수적인 연산을 적용하여 잉여 비트를 첨가하고 수신단에서는 역연산에 의하여 원래의 정보를 복원하는 방식으로서 1950년에 Hamming에 의하여 최초로 소개되었다.

트리 부호는 1955년에 Elias에 의하여 소개되었고 1960년대 말에 비터비(Viterbi) 알고리즘이 개발되어 오늘날까지 복호화 방식으로 주로 이용되고 있다.

트리 부호는 대수적인 연산이 부/복호 과정에 개

* 동서대학교 전자통신과(srkim@haksan.dsc.ac.kr)

** 남서울대학교 정보통신공학과(hycheong@www.nsu.ac.kr)

논문번호 : 99031-0922

접수일자 : 1999년 9월 22일

입되지 않기 때문에 복호 성능을 구하기가 어렵지만 연판정(soft decision)복호를 쉽게 수행할 수 있어 약 2dB 정도의 이득을 얻을 수 있다는 특징이 있다.

터보부호는 이러한 컨볼루션 부호를 병렬로 concatenation하여 만들어진 부호라고 할 수 있다. 터보부호는 1993년에 프랑스의 Berrou에 의하여 처음으로 소개되었고 Shannon의 극한 값에 도전하는 부호화 방식으로 집중적으로 연구되기 시작하였다^[2,3].

본 연구에서는 디지털 통신 및 위성 채널에 효율적인 오류 제어 방식인 터보부호에 대하여 연구하였고 그 성능을 분석하여 보았다.

II. 터보 부호

디지털 신호를 채널을 통하여 전송할 때 시스템의 성능을 높여서 효율적인 통신을 수행하기 위해서는 오류 제어 능력이 큰 부호를 필요로 하게 된다. 그러나 단일 부호로는 시스템의 구현이 어려워 지므로, 짧은 길이의 부호를 cascade하여 시스템의 복잡도를 줄이면서 성능을 개선시킬 수 있다^[4].

인터리버에 의하여 분리된 두 컨볼루션 부호의 병렬 concatenation을 터보부호라고 한다^[2].

터보부호는 한 번 복호를 수행한 후의 출력을 다시 복호기의 입력단으로 회귀시킨 후 이 회귀된 정보를 이용하여 이전보다는 좀 더 우수한 성능을 낼 수 있도록 하는 것이다.

터보부호에 대한 부호기의 구조가 그림 1에 나타나 있다. 부호기는 두 개의 RSC(recursive systematic convolutional) 부호와 이 두 RSC 부호를 연결시켜 주는 인터리버로 구성되어 있다. 여기서 RSC 부호란 컨볼루션 부호의 일종으로 순환성이 있고 정보가 부호어의 일부로 구성되어 있는 체계화된(systematic) 부호이다.

그림에는 부호화율이 1/2인 부호기가 나타나 있는데, 여기서 d_k 는 입력 데이터 비트이고 X_k 는 첫 번째 부호 비트, Y_k 는 두 번째 부호 비트를 각각 나타낸다. RSC 부호는 일반적으로 체계화되지 않은(nonsystematic) 부호보다 우수한 성능을 얻을 수 있다.

그림 1의 인터리버(interleaver)는 입력되어 들어오는 데이터를 분산시켜 두 번째 RSC 부호기로 보낼 수 있도록 해주는 기능을 수행한다.

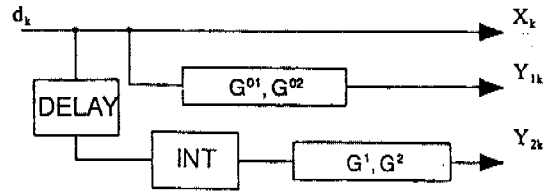


그림 1. 터보부호 부호기

터보부호의 부호화 과정을 살펴보면 다음과 같다. 입력 데이터 비트 d_k 가 들어오면 첫 번째 부호기의 출력 X_k 는 d_k 가 되고, 두 번째 출력 비트 Y_{1k} 는 첫 번째 RSC 부호에서의 부호 비트, 세 번째 출력 비트 Y_{2k} 는 입력 비트가 인터리버를 거친 후 두 번째 RSC 부호기로 입력되어 생성된 부호 비트가 된다. 따라서 그림 1에 나타난 터보부호에 대한 부호기로 부터 생성되는 부호어의 부호화율은 1/3이 된다.

이처럼 터보부호에서는 비교적 낮은 부호화율의 부호어가 생성되는데 Berrou의 터보부호에서 제안한 방식은 다음과 같이 부호화율 1/3의 터보부호로부터 부호화율 1/2의 터보부호를 생성하는 기법이였다. 그림 1에 나타난 부호기에서 생성되는 출력 ($X^1_k, Y^1_{1k}, Y^1_{2k}, X^2_k, Y^2_{1k}, Y^2_{2k}, \dots$)중 부호비트를 번갈아 삭제하면서 보내면 1/2의 부호어 ($X^1_k, Y^1_{1k}, X^2_k, Y^2_{2k}, \dots$)가 된다. 그리고 터보부호가 우수한 성능을 내기 위해서는 인터리버를 RSC 부호의 특성상 최대한 거리 특성이 우수한 부호어를 생성하기 위하여 비균일(non-uniform)인터리버로 하여야 한다.

III. 터보 부호의 복호 방식

터보부호의 복호기는 터보부호 부호기의 역기능을 수행하여야 할 것이다. 터보부호는 컨볼루션 부호를 병렬 concatenation하였으므로 복호 방식도 컨볼루션 부호에 대한 복호 방식을 사용하게 된다.

그림 2에는 그림 1에 나타나 있는 터보부호 부호기에 대응하는 복호기가 나타나 있다. 두 개의 RSC 부호기에 해당하는 복호기가 각각 나타나 있고 그 사이를 인터리버가 연결시켜주고 있다. 두 번째 복호기에서 나오는 출력은 전송된 채널에서 오류가 전혀 없었다 원래의 정보 비트와 동일한 값이 될

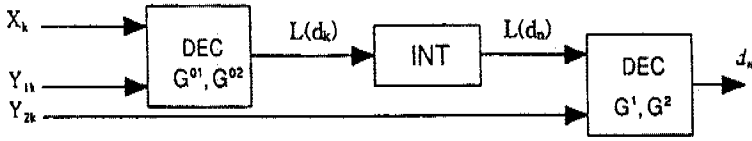


그림 2. 터보부호 복호기

것이다. 그러나 실제 전송 채널에서 오류가 많이 발생하였다면, 두 개의 복호기를 통과한 후의 출력 값이 원래의 전송 신호와는 다를 것이므로, 터보부호의 복호기에서는 반복적으로 복호를 수행함으로써 복호 후에 발생한 오류를 점점 개선하는 방식을 취했다.

터보부호에서의 인터리버는 시스템의 성능을 효율적으로 하기 위하여 사용하게 되는데 인터리빙을 하지 않았을 경우에는 비트 오류율이 10^{-1} 정도로 나타나게 되어 그 시스템을 이용할 수 없게 된다.

인터리버는 균일 인터리버(uniform interleaver)와 랜덤 인터리버(random interleaver)로 나눌 수 있다.

균일 인터리버는 블록 인터리버(block interleaver)라고도 하며 데이터를 인터리빙하는 방법중 가장 간단한 방법이다⁵⁾. 블록 인터리빙에서는 $M \times M$ 행렬로 비트열을 표현하게 되고 이 때 M 은 일반적으로 2^n 이 되며 행으로 읽어서 열로 출력하게 된다. 일반적으로 프레임 사이의 데이터를 흩어지게 할수록 즉, 확산시킬수록 성능은 좋아지나 블록 인터리빙은 간단하지만 상관 관계가 나타나게 되어 원하는 성능을 얻을 수 없다.

터보부호에서 랜덤 인터리버를 사용하면 연접 오류를 랜덤 오류로 확산시켜 좋은 성능을 가지게 된다. 랜덤 인터리버는 비균일(nonuniform) 인터리버라고 하며 데이터를 최대한 흩어지게 하는 조건과 인터리빙된 데이터의 순서가 최대한 바뀌어 지도록 하는 조건을 가져야 한다. 그렇게 함으로써 터보부호의 최소 거리가 균일 인터리빙의 경우보다 훨씬 더 증가될 수 있다. 일반적으로 인터리빙과 디인터리빙이 같은 장치로 수행될수 있도록 인터리버는 대칭이 되게 만든다.

터보부호의 복호 방식에는 MAP 알고리즘을 사용한다.

MAP 알고리즘은 Bahl 등에 의하여 처음으로 제안되었으나, 부호 단어가 많아지면 복호할 때 복잡도가 증가하고 비터비 알고리즘보다 시간 지연이 길어지게 되는 단점을 갖고 있다⁶⁾.

▶ 메모리를 가진 부호화기에서 시간 k 의 천이

레지스터 상태를 S_k 라고 정의한다면, 이 S_k 는 입력 d_k 를 포함하게 된다.

이 때 부호화되지 않은 비트는 $d_k(=I_k)$ 이며, 부호화된 비트는 C_k 이고 BPSK나 QPSK 변조 후 부가 가우시안 채널(AWGN)로 전송한다.

각각의 복호화된 비트 \hat{d}_k 의 대수화된 근사화 비(logarithm of likelihood ratio:LLR) 을 정리하면,

$$L(\hat{d}_k) = \log \frac{P(d_k=1 | R_o^{N+\nu-1})}{P(d_k=0 | R_o^{N+\nu-1})} \quad (1)$$

이다. 이를 다시 정리하면 다음과 같다⁷⁾.

$$L(\hat{d}_k) = \log \frac{\sum_{m=0}^{2^\nu} \alpha_k^1(m) \beta_k^1(m)}{\sum_{m=0}^{2^\nu} \alpha_k^0(m) \beta_k^0(m)} \quad (2)$$

식(2)에서 $\alpha_k^i(m)$ 와 $\beta_k^i(m)$ 를 반복적인 형태로 바꾸어 정리하면, 각각 다음의 식과 같이 표현이 가능하다.

$$\alpha_k^i(m) = \sum_m \sum_{j=0}^{2^\nu} \alpha_{k-1}^j(m) \gamma_{i,j}(R_k, m, m') \quad (3)$$

여기서,

$$\gamma_{i,j}(R_k, m, m') = P(d_k=i, S_k=m, R_k | d_{k-1}=j, S_{k-1}=m') \quad (4)$$

이다. 그러므로, $\alpha_k^i(m)$ 은 시간 k 에서 상태가 m 이고 입력이 i 일 때의 순방향 순환 상태 메트릭(forward recursion state metric)이 된다.

$$\beta_k^i(m) = \sum_m \sum_{j=0}^{2^\nu} \beta_{k+1}^j(m) \gamma_{j,i}(R_{k+1}, m', m) \quad (5)$$

여기서, $\beta_k^i(m)$ 은 시간 k 에서 상태가 m 이고 입력이 i 일 때의 역방향 순환 상태 메트릭(backward recursion branch metric)이라 한다.

확률변수 $\gamma_{i,j}(R_k, m, m')$ 과 $\gamma_{j,i}(R_{k+1}, m, m)$ 는 이산 가우시안 비기여 채널의 천이 확률과 부호화 기 트렐리스의 천이 확률에 의해서 결정된다.

대수적인 근사화 비를 정의하여, 반복적인 복호의 원리는 다음과 같이 전개될 수 있다.

$$L(I_k) = \log \frac{P(I_k=1)}{P(I_k=0)} \quad (6)$$

$L(I_k)$ 의 부호는 경관정 값이 되고, 크기 $|L(I_k)|$ 는 연관정 값으로 신뢰도(reliability)를 나타낸다. 이 랜덤 값을 수신단에서 보게 되면 조건부 랜덤 값이 된다. 정보 비트 I_k, C_k 가 전송된 후 채널을 통과한 값을 R_k 라고 한다면,

$$\begin{aligned} L(I_k | R_k) &= \log \frac{P(I_k=1 | R_k)}{P(I_k=0 | R_k)} \\ &= \log \left[\frac{\exp(-E_s(R_k - r)^2/N_0)}{\exp(-E_s(R_k + r)^2/N_0)} \right] + \log \left[\frac{P(I_k=1)}{P(I_k=0)} \right] \\ &= L_c \cdot R_k + L(I_k) \\ &\triangleq L(\hat{I}_k) \end{aligned} \quad (7)$$

이 된다. 여기서 $L_c = 4r \cdot E_s/N_0$ 이고, r 은 부가 가우시안 채널인 경우 I_k 가 0, 1에 상관없이 1이다.

이 복호기는 모든 정보 비트의 a priori 값 $L(I_k)$, 부호화 된 비트가 채널을 통과한 후의 값 $L_c \cdot R_k$ 와 이전의 반복에서 출력된 외부 정보 $L_e(I_k)$ 를 이용하여 복호를 하게 된다. 그러므로, $L(\hat{I}_k)$ 는 다음의 세 항으로 다시 표현이 가능하다.

$$L(\hat{I}_k) = L_c \cdot R_k + L(I_k) + L_e(I_k) \quad (8)$$

이것은 정보 비트를 복호하는 최대 근사화 비가 세 가지의 독립적인 항으로 표현이 가능하다는 것을 의미한다.

터보부호에서는 매 복호 비트마다 이처럼 전체 수신된 신호에 대한 값을 고려해 주어야 한다. 터보 부호에서 한 번의 복호가 끝난 후에는 각 지점에서 복호된 값이 어느 정도의 신뢰성을 갖는가를 나타 내주는 각 수신된 신호에 대한 $L(\hat{I}_k)$ 값을 복호기의 입력으로 순환시켜 준다.

IV. 성능 분석 및 결과 고찰

이 장에서는 부호화율이 1/2인 터보부호 시스템

에 대한 BER 성능을 컴퓨터 시뮬레이션을 통하여 분석해 보았다.

터보부호의 성능을 좌우하는 요소로는 RSC 부호의 생성 함수(generator)와 RSC 부호기 내의 메모리 개수가 있다. RSC 부호기의 메모리 개수가 증가할수록 성능은 향상되지만 부호기의 복잡성이 증가하게 된다. 터보부호의 성능을 좌우하는 또다른 요소는 두 RSC 부호를 연결하는 인터리버와 복호 횟수로서 복호 횟수가 증가할수록 성능은 향상되지만 그만큼 지연이 생기게 된다.

이 절에서는 구속장이 3이고 부호화율이 1/2인 반복 부호에 대한 성능을 분석하였다. 시뮬레이션 환경은 부가 가우시안 채널에서 블록 인터리버와 랜덤 인터리버를 각각 사용한 경우에 대하여 수행하고 비교하였다.

본 연구에서 사용한 변수는 다음의 표 1과 같다.

표 1. 사용한 실험 변수

구속장 길이	3
변조 방법	BPSK
부호화율	1/2
생성 다항식	$g_1 = 7(8), g_2 = 5(8)$
복호 알고리즘	MAP 알고리즘
인터리버 종류	블록, 랜덤
인터리버 크기	32 × 32 비트
채널	부가 백색 가우시안

시뮬레이션 결과로써 반복 횟수에 따른 BER 특성과 블록 인터리버와 랜덤 인터리버를 사용한 경우에 E_b/N_0 에 따른 BER 특성에 대하여 알아보았다.

반복 횟수에 따른 BER 특성은 다음 그림 3과 그림 4에 나타나 있다.

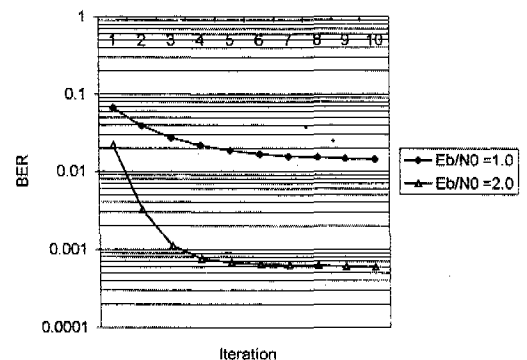


그림 3. 블록 인터리버를 사용한 경우의 반복 횟수에 대한 성능

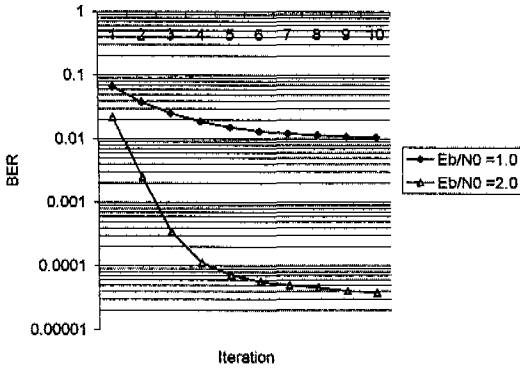


그림 4. 랜덤 인터리버를 사용한 경우의 반복 횟수에 대한 성능

결과 그림에서 반복 횟수가 많아지면 성능이 좋아짐을 알 수 있었지만 반복 횟수가 5이상에서는 성능 향상의 폭이 줄어들어 거의 비슷한 결과가 나옴을 볼 수 있다. 그러므로 다음 그림 5에서는 반복 횟수가 5인 경우에 대하여 블록 인터리버와 랜덤 인터리버를 비교해 보았다.

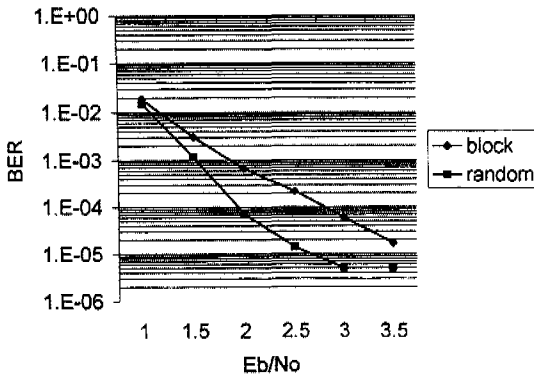


그림 5. AWGN 채널에서 R=1/2인 부호의 성능 비교

그림 5는 부호화율이 1/2인 터보부호를 부가 가우시안 채널에서 실험한 결과로서 랜덤 인터리버가 정보 비트간의 상관 관계가 적으므로 블록 인터리버에 비하여 성능이 10배정도 더 좋음을 알 수 있었다.

표2는 반복 횟수가 5이고 랜덤 인터리버를 사용한 경우의 1/2 터보부호의 성능과 두 부호를 cascade 하였을 때의 성능을 비교해 나타내었다. 이때 cascade 부호의 부호화율은 0.327로 하였다.

부호화율이 작아지면 성능이 좋아지지만 표2에서 볼 수 있듯이 터보부호의 부호화율이 더 클지라도 기존 부호에 비하여 성능이 훨씬 좋음을 알 수

있다.

표 2. 터보부호와 cascade 부호와의 성능 비교

Eb/No	BER	
	터보부호	cascade부호
0.5	0.095657	0.930116
1.0	0.016906	0.521703
1.5	0.003213	0.201431
2.0	0.000520	0.067934
2.5	0.000113	0.021927
3.0	0.000033	0.006997

그림 6은 인터리버 크기를 변화시켰을 때의 BER 특성을 나타낸다.

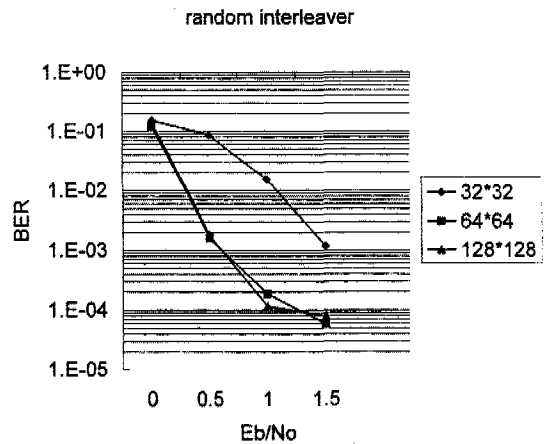


그림 6. BER 특성

그림 6에서 인터리버의 크기가 커지면 시스템의 성능은 좋아지지만 지연 등의 문제로 인하여 64 x 64 이상의 크기인 경우에는 거의 비슷한 성능을 가짐을 알 수 있었다.

V. 결론

본 연구에서는 위성 채널에서 효율적으로 오류를 제어하기 위한 터보부호에 대하여 연구하고 그 성능을 분석하였다.

설계한 터보부호는 RSC 부호를 병렬로 concatenation 하여 만든 부호로서 부가 백색 가우시안 채널에서 구속장이 3 이고 부호화율이 1/2 인 경우에

대하여 시뮬레이션을 수행하였다. 변조 방식으로는 BPSK를 이용하였고 생성 다항식은 7/5 를 사용하였다.

터보부호의 성능은 인터리버 크기가 32×32 일 때 반복 횟수에 따른 비트 오류율 블록 및 랜덤 인터리버를 사용하였을 때 E_b/N_0 에 따른 비트 오류율에 대하여 분석하였다. 그리고 인터리버의 사이즈를 증가시켰을 때의 성능을 비교 분석하여 보았다.

시뮬레이션 결과 복호시 반복 횟수가 많아지면 성능이 좋아짐을 알 수 있었지만 반복 횟수가 5 이상일 경우에는 성능 향상의 폭이 줄어들어 포화 상태에 이르게 됨으로써 결과가 거의 비슷하게 나타남을 알 수 있었다. 그리고 랜덤 인터리버는 정보 비트간의 상관 관계가 적으므로 블록 인터리버에 비하여 성능이 더 좋을 수 있었고 인터리버의 크기가 커질수록 성능은 우수하지만 지연 등의 문제로 인하여 64×64 이상에서는 결과가 거의 같은 값을 가짐을 알 수 있었다.

앞으로는 성능 향상을 위한 최적의 인터리버 설계 등에 대하여 연구가 계속되어야 할 것이다.

참 고 문 헌

- [1] S.Lin, Error Control Coding, Prentice-Hall, pp.3-9, 1983.
- [2] Claude Berrou, Alain Glavieux and Punya Thitimajshima, "Near Shannon Limit Error Correcting Coding and Decoding:Turbo Codes," Proc. IEEE ICC, pp.1064-1070, May 1993.
- [3] D.Divsalar and F.Pollara, "Turbo Codes for PCS Applications," Proceedings of IEEE ICC '95, Seattle, Washington, pp.54-59, June 1995.
- [4] J.Proakis, Digital Communications, McGraw-Hill, pp.240-281, 1993.
- [5] Claude Berrou and Alain Glavieux, "Near Optimum Error Correcting Coding And Decoding : Turbo-Codes," IEEE Trans. on Commun., vol.44, No.10, pp.1261-1271, Oct. 1996.
- [6] L.Bahl, J.Cocke, F.Jelinek, and J.Raviv, "Optimal decoding of linear codes for minimizing symbol error rate," IEEE Trans. on Inform. Theory, vol. IT-20, pp.284-287, March 1974.
- [7] F. Malardel, "Simulation and optimisation of

the turbo decoding algorithm," EFREI Final Year Report, University of South Australia, Nov. 1996.

김 신 령(Sin-Ryeong Kim)

정회원

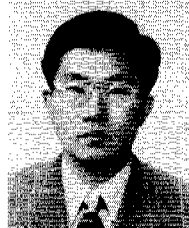


1983년 2월:경북대학교 공과대학
전자공학과(학사)
1985년 2월:연세대학교 본대학원
전자공학과(공학석사)
1990년 2월:연세대학교 본대학원
전자공학과(공학박사)

1992년 9월~현재: 동서울대학 전자통신과 조교수
<주관심 분야> 부호이론, 위성통신 등.

정 호 영(Ho-Young Cheong)

정회원



1986년 2월: 충남대학교
전자공학교육학과(학사)
1989년 8월: 연세대학교
본 대학원 전자공학과
(공학석사)
1994년 2월: 연세대학교
본 대학원 전자공학과
(공학박사)

1997년 4월~현재: 남서울대학교 전자정보통신공학
부 조교수
<주관심 분야> 부호이론, 변복조이론, 위성통신,
CDMA 등.