

소스 코더(CS-ACELP) 패킷 데이터의 중요도에 따른 터보 코드의 평처리링 방법 연구

준회원 장만종*, 정회원 이인성*

Puncturing of Turbo Codes According to Importance of Packet Data of Source Coder(CS-ACELP)

Man-Jong Chang* , InSung Lee* *Regular Members*

요 약

본 논문에서는 IMT-2000 시스템에 음성 코더로 고려중인 CS-ACELP(ITU-T G.729)와 채널 코더로 고려중인 터보 코드의 결합 방법을 제안한다. 음성 코더의 전송률은 8Kbps이며 채널 코더에서는 부호화율이 1/2인 터보 코드를 사용하였다. 기존의 터보 코드는 부호화율을 1/3에서 1/2로 하기 위해서 패리티 비트를 고대로 평처리링하지만 본 논문에서는 소스 코더에서 출력되는 패킷 데이터의 중요도에 따라 패리티 비트의 평처리링을 달리하여 부호화율을 1/2로 하는 방법을 제안한다. 패킷 데이터의 중요도는 실제 음성을 이용하여 소스 코더의 출력 패킷 데이터의 전송 에러에 대한 감응도를 측정하여 분석하였다. 모의 실험은 페이딩 채널에서 이루어졌으며, 제안된 평처리링 방법의 성능이 기존의 평처리링 방법보다 SNR(Signal to Noise Ratio)과 SegSNR (Segmental SNR)면에서 개선됨을 확인하였다. 주관적인 음질 평가 시험인 MOS(Mean Opinion Score)시험에서도 음질 향상을 확인하였다.

ABSTRACT

In this paper, we propose the turbo codes combined with the CS-ACELP speech coder. The CS-ACELP and turbo codes are being considered as speech coder and channel coder of IMT-2000 system. The CS-ACELP is 8 kbps transmission rate. The code rate of channel coder is given by 1/2. Usually, 1/2 rate turbo codes is obtained by uniformly puncturing 1/3 rate turbo codes. However, In this paper, the puncturing methods according to importance of the output packet data of the source coder is proposed for the turbo codes. The importance of the output packet data is analyzed by checking the sensitivity of speech parameters to channel errors. The sensitivity of source coder output packet data is measured in noisy channel. The simulation is executed over Rayleigh fading channel. The performance improvement of proposed puncturing method was verified in the informal subjective listening test and MOS test.

I. 서 론

소스 코더와 채널 코더의 결합에 있어서 최적의 방법을 찾는 연구는 Shannon이 정보 이론을 정립할 당시부터 제기된 문제로서 이론적으로 많은 연구가 수행되어 왔으며 아직도 많은 연구 결과가 발표되

고 있다^{[1][2]}.

소스 코더의 출력 패킷 데이터의 비트 에러 감응도(Sensitivity)에 따라 차등적으로 채널 코딩하는 방법은 여러 통신 시스템에서 적용되었으며 DPCM, Subband coding 방식의 영상 코더와 PCM, Subband coding, CELP등 음성 코더에서 사용되었다^{[2][3][4]}. 또한 터보 코드를 사용한 차등적인 채널

* 충북대학교 전파공학과 통신신호처리 연구실 (chang93@hanmail.net)
논문번호 : 99183-0514, 접수일자 : 1999년 5월 14일

코딩 방법도 소개되었다^{[5][6]}.

본 논문에서는 최근에 ITU-T 표준안으로 채택된 CS-ACELP 소스 코더와 결합된 채널 코딩 방법을 제안한다. CS-ACELP(ITU-T G.729) 음성 코더는 8Kbps 전송률에서 명료한 음질(Toll quality)을 내며^{[7][8]}, IMT-2000 시스템의 음성 코더로 사용하기 위한 연구가 진행되고 있다. 이와 결합된 채널 코더로는 AWGN 채널에서 Shannon의 성능 극한에 근접하는 성능을 보이는 터보 코드를 사용하였다^[9]. 본 논문에서는 터보 코드의 부호화율을 1/3에서 1/2로 하기 위해서 패리티 비트를 평처링(Puncturing)할 때 CS-ACELP 음성 코더의 출력 패킷 데이터의 중요도를 검사한 후 각각의 중요도에 따라 중요한 비트에 대한 패리티 비트는 평처링을 하지 않고 대신 덜 중요한 비트에 대한 패리티 비트를 평처링하여 부호화율을 1/2로 하는 방법을 제안하였으며, 소스 코더의 출력 패킷 데이터의 특성과 관계없이 평처링하는 기존의 방법과 성능을 비교하여 본다.

본 논문은 다음과 같이 구성되어진다. 2장에서는 CS-ACELP(ITU-T G.729)에 대한 알고리즘을 설명하며, 3장에서는 채널 코딩 방법으로 사용한 터보 코드에 대해 설명한다. 4장에서는 소스 코더의 패킷 데이터의 중요도 분석과 중요도에 따른 평처링 방법을 설명하며, 5장에서는 모의 실험을 통해 기존의 방법과 제안된 방법의 결과를 비교하며, 6장에서는 결론을 맺는다.

II. CS-ACELP 음성 코더 알고리즘

8Kbps 전송률을 가지는 CS-ACELP (Conjugate Structure Algebraic Code Excited Linear Predicting)코더는 CELP 모델에 기본을 둔다. CS-ACELP 알고리즘의 부호화기는 10 ms 음성 프레임마다, 음성 신호를 분석하여 CELP 모델 파라미터를 추출한다. 추출된 음성 파라미터들은 프레임당 80 비트를 할당하여 부호화되어 전송되어진다. 부호화 과정에서 사용되는 모델 파라미터와 비트 할당은 표 1에 나타나 있다.

처음에 입력 음성 신호는 고역필터를 통과한 후 자기상관 방법을 사용하여 선형예측계수(LPC)를 추출한다. LPC 계수의 계산 및 전송은 10ms 프레임당 한번 수행되어진다. 음성 신호의 스펙트럴 포폭선 정보를 나타내는 LPC 계수의 전송을 위해 LPC 계수는 Line Spectrum Pairs (LSP) 파라미터로 변환된다. LSP 파라미터는 18비트를 사용하여 2

표 1. 8Kbps CS-ACELP 알고리즘의 비트할당

파라미터	코드워드	부프레임	부프레임	비트 수
		1	2	
LSP	L0,L1,L2,L3			18
적용코드복지연	P1,P2	8	5	13
피치 지연 parity	P0	1		1
고정코드복인덱스	C1,C2	13	13	26
고정코드복부호	S1,S2	4	4	8
코드복이득 1	GA1,GA2	3	3	6
코드복이득 2	GB1,GB2	4	4	8
Total				80

단 예측 벡터 양자화된다. MA 예측기 분리(Split) 벡터 양자화 방법이 사용된다. 적용 코드북과 고정 코드북 파라미터는 5 ms 부프레임 마다 검색되어 전송되어진다. 피치 지연의 검색시 계산량을 줄이기 위해 전 피치지연에 대해 폐구간 검색방법을 사용하지 않고 개구간 피치 검색 방법을 통해 대략의 피치 지연을 예측한 후 폐구간 검색을 수행한다. 폐구간 검색에서는 개구간 방법에 의한 예측된 피치 지연의 주변 값만을 검색하여 정확히 피치 지연을 예측한다. 검색되어지는 피치지연은 1/3 샘플 피치 지연 분해도를 갖는다. 피치지연 값은 첫 번째 부프레임에서 8 비트로 부호화되고, 두 번째 부프레임에서는 앞 부프레임 피치지연 값과의 차를 5 비트를 사용하여 부호화된다.

최적 피치지연 값을 찾아내면 음성의 여기신호 추출을 위한 고정 코드북 검색을 수행한다. 고정 코드북 파라미터는 5 ms 부프레임 마다 분석/합성 검색 방법을 사용하여 검색된다. 고정 코드북 검색을 위한 목적(Target)신호는 선형예측 잔여신호에서 적용 코드북(adaptive codebook)의 기여분을 제거함으로써 얻어진다. 고정 코드북 여기(fixed codebook excitation)신호를 발생하기 위해 대수적(algebraic) 코드북을 사용하며 펄스 위치에 14 비트 펄스 부호에 4비트 총 17비트를 할당한다. 각 부프레임당 4개의 펄스만이 지정된 위치 중에 +1, -1의 값을 갖게 되어 고정 코드북 검색시 계산량을 줄일 수 있다. 적용 코드북과 고정 코드북의 이득 양자화에는 Conjugate 구조를 갖는 2개의 코드북을 사용한다. 각각의 코드북은 8개와 16개의 요소 값을 가지며 구해진 이득값과 미리 선택된 4개와 8개의 요소 값들에 대해서만 검색이 이루어져 계산량을 줄일 수 있다. 또한 프레임간의 상관성을 이용한 4차 MA(Moving Average) 예측기를 사용한다. 부호화

의 최종적 단계에서는 다음 프레임의 목적신호를 구하기 위해 합성 및 가중필터의 메모리 갱신을 수행한다.

전체적 CS-ACELP 알고리즘의 부호화 과정은 그림 1에 나타나 있다.

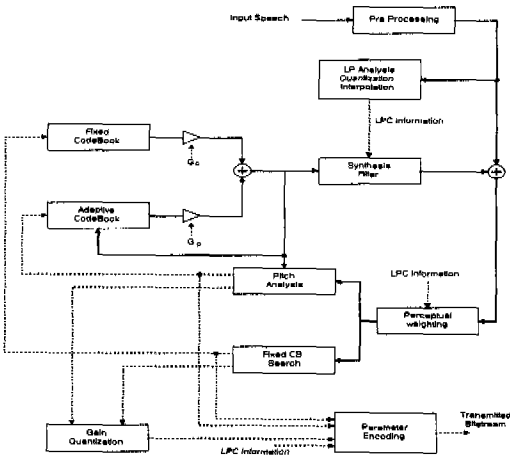


그림 1. CS-ACELP 부호화 과정

10ms 음성프레임에 해당하는 복호화기의 파라미터를 얻기 위해서, 이들 파라미터 색인(parameter index)들을 복호화한다. LSP계수들은 보간된 후 각각의 부프레임에 대한 선형 합성 필터 계수로 변환된다. 적응 코드북과 고정 코드북의 여기신호는 이득이 곱해진 후 단구간 합성필터 및 후단여파기를 통과하여 음성신호로 합성되어진다.

CS-ACELP 알고리즘의 복호화 과정은 그림 2에서 보여진다.

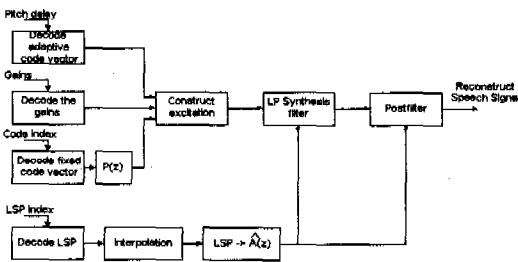


그림 2. CS-ACELP 복호화 과정

III. 터보 코드의 구조

터보 코드의 부호화기의 기본 구조는 두 개의

RSC(Recursive Systematic Convolutional) 부호화기를 병렬로 연결한 구조를 가진다. 그림 3은 터보 부호화기의 구조이다. 그림에서 부호화기1은 들어오는 입력 비트열 d_k 를 그대로 부호화하여 체계적인(systematic) 정보 X_k 와 부호화된 출력 Y_{1k} 를 발생시킨다. 그리고 같은 입력 비트열 d_k 는 인터리버를 통과한 후 부호화기2에서 부호화된 출력 Y_{2k} 를 발생시킨다. 부호화기1과 부호화기2에서 만들어 내는 패리티 비트 Y_{1k} 와 Y_{2k} 를 모두 전송하게 되면 부호화율이 1/3이 되고 그림에서처럼 교대로 패리티 비트를 평처리하여 전송하면 부호화율이 1/2가 된다.

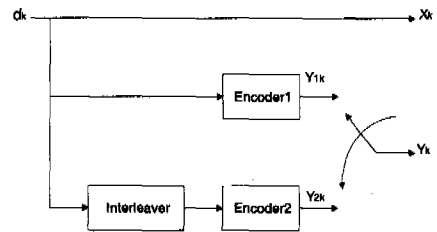


그림 3. 터보 코드의 부호화기

그림 4는 터보 부호화기에 대해서 부가 정보(extrinsic information)를 이용하여 반복 복호를 수행하는 터보 복호화기의 구조를 보여준다. 복호 알고리즘으로는 MAP(Maximum a Posteriori)과 SOVA(Soft Output Viterbi Algorithm) 복호 알고리즘이 사용되는데 MAP 복호 알고리즘이 성능에서 우수한 장점이 있으나 복잡도가 SOVA 복호 알고리즘 보다 높은 단점이 있다^[10]. 식 (1)은 복호화기가 출력하는 입력 비트 d_k 에 대한 로그 근사비(log likelihood ratio)이다.

$$L(d_k) = L_c x_k + L_a(d_k) + L_e(d_k) \quad (1)$$

첫 번째 반복 복호에서는 부호화기1에 사전 정보가 없으므로 $L_a(d_k) = 0$ 이 되며 부호화기1의 출력은 식 (2)와 같다.

$$L_1(d_k) = L_c x_k + L_e^{D_1}(d_k) \quad (2)$$

부호화기2에서는 부호화기1로부터 사전 정보를 가지므로 식 (3)과 같은 출력을 가진다.

$$L_2(d_k) = L_c x_k + L_e^{D_1}(d_k) + L_e^{D_2}(d_k) \quad (3)$$

여기서, $L_e^{D_2}(d_k)$ 는 다음 단계의 반복 복호에서 사전 정보 $L_e(d_k)$ 가 된다. 각각의 복호화기에서 발생된 부가 정보를 다음 복호에서 이용함으로써 터보 코드의 복호 성능을 향상시킬 수 있다.

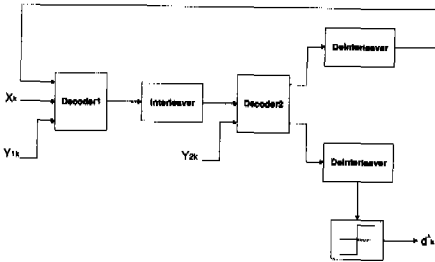


그림 4. 반복 복호 알고리즘을 적용한 터보 코드의 복호화기

IV. 소스 코더 패킷 데이터의 중요도 분석과 중요도에 따른 평처링 방법

소스 코더의 출력 패킷 데이터의 특성에 따라 터보 코드의 평처링을 달리하기 위해 소스 코더 패킷 데이터의 전송에러에 대한 특성을 분석하였다. 소스 코더의 성능을 평가하기 위해 객관적 성능 평가 방법인 SNR(Signal to Noise Ratio)와 SegSNR(Segmental SNR)을 사용하였다.

SNR 과 SegSNR은 다음과 같이 계산되어진다.

$$SNR(dB) = 10 \log \left[\frac{\sum_{n=1}^{N_s} s^2(n)}{\sum_{n=1}^{N_s} (s(n) - \hat{s}(n))^2} \right] \quad (4)$$

$$SegSNR(dB) = \frac{1}{N_b} \sum_{j=0}^{N_b-1} SNR(j) \quad (5)$$

여기서, N_b 는 20ms 크기를 갖는 블록의 수를 나타내며 SNR(j)는 j번째 블록의 SNR 값을 나타낸다.

실험에 사용한 음성 데이터는 남자 6명, 여자 6명의 한국어 음성과 영어 음성으로 구성된 약 153초 가량의 음성데이터를 사용하였다. 수신된 패킷 데이터의 각 파라미터 비트에 에러를 발생시킨 후 CS-ACELP로 복호화한 음성 출력과 원래 신호와의 SNR와 SegSNR을 계산하여 패킷 비트의 에러 감응도를 측정하였다. 실험에서 전송 에러에 민감한

파라미터 비트와 전송에러에 강인한 파라미터 비트 사이의 SNR와 SegSNR 는 각각 14.83dB, 14.57dB 정도로 심한 성능 차이를 보였다.

실험 결과는 그림 5에 나타나 있다. 일반적으로 코드북 이득, 피치 주기는 전송에러에 민감하였고, 코드북 인덱스는 전송에러에 강인함을 알 수 있다. 이 결과에 의해 중요도가 결정된다.

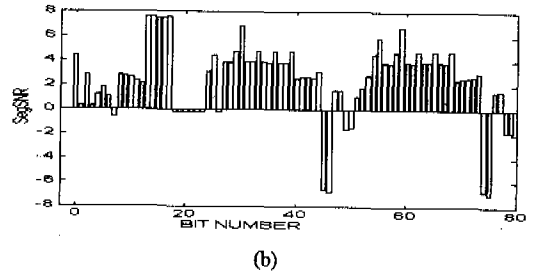
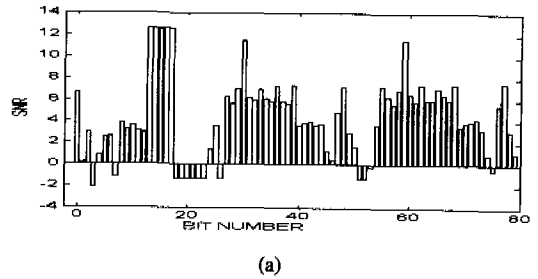


그림 5. CS-ACELP 패킷 데이터의 에러 감응도
(a)SNR (b)SegSNR

터보 코드의 평처링 방법은 분석된 비트들의 중요도에 의해 결정된다. 터보 코드의 경우 입력 비트열이 인터리버를 통과한 후 부호화기2에서 패리티 비트를 출력하므로 인터리버의 구조에 따른 평처링 방법을 적용해야 한다. 터보 코드의 입력 비트열은 꼬리(tail) 비트가 더해지므로 구속장 K=3인 경우, 음성 코더의 패킷 데이터 80비트와 2개의 꼬리 비트로 총 82개의 비트가 된다. 이것을 중요도에 따라 3개의 그룹으로 분류하였다. 본 논문에서는 프레임 크기가 작은 경우 우수한 성능을 보이는 순환 시프트(circular shift) 인터리버를 사용하였다^[11].

그림 6(a)의 경우 중요한 비트에 대한 부호화기1의 패리티 비트는 적게 평처링하고 덜 중요한 비트에 대한 패리티 비트는 많이 평처링하는 방법을 적용하였다. 또한 그림 6(b)의 경우 인터리빙 후의 입력 비트열에 대한 부호화기2의 경우에는 중요한 비트와 덜 중요한 비트가 인접한 경우 짧은 선으로 표시한 부분과 같이 중요한 비트에 대한 패리티 비

트는 평처리하지 않고 대신 덜 중요한 비트에 대한 패리티 비트를 평처리함으로써 전체 부호화율이 1/3에서 1/2가 되게 하였다. 이것을 그림 6에 0과 1로 나타내었다. 0인 경우는 평처리 되어지고 1인 경우는 평처리하지 않는다.

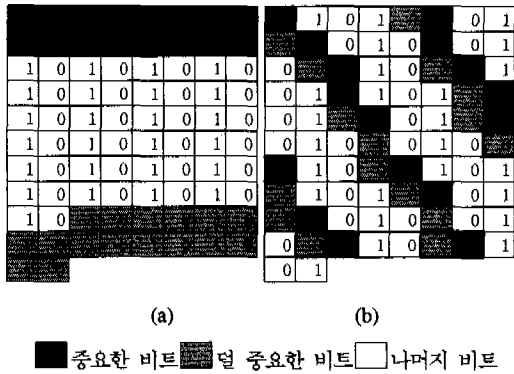


그림 6. 제안된 평처리 방법 (중요한 비트 16개, 덜 중요한 비트 16개의 경우)

V. 모의 실험 결과

소스 코더의 패킷 데이터의 중요도에 따른 터보 코드의 평처리 방법의 성능 평가를 하기 위한 실험 환경을 구성하였다. 전체적인 시스템은 그림 7과 같이 구성되어진다. 모의 실험에서 소스 코더 패킷 데이터는 터보 코드의 부호화기로 채널 코딩된 후 BPSK로 변조하여 디지털 채널로 전송된다. 채널 환경은 Rayleigh 분포를 갖는 페이딩 채널($f_d T = 0.01$)에서 실험하였다. f_d 는 도플러(Doppler) 주파수이고 T 는 부호화된 비트의 구간이다. 터보 코드는 구속장 $K=3$, 생성 다항식 (7, 5)인 부호화기와 SOVA 복호 알고리즘을 부호화기의 복호 알고리즘으로 사용하였으며 반복 횟수는 10이다. 또한 복호 알고리즘에 채널 부가 정보(channel side information)를 사용하지 않았다^[12].

터보 코드의 입력으로는 총 82비트(음성 코더 패킷 데이터 80비트+포리 비트)이며 측정된 패킷 데이터의 SNR과 SegSNR에 의한 중요도에 따라 3개의 그룹(포리 비트는 덜 중요한 비트에 포함)으로 나누었고 각 그룹은 앞장에서 제안된 방법으로 달리 평처리시켜 전체 부호화율을 1/2로 하였다. 이와

같은 방법으로 4가지의 경우로 분류하여 실험하였다. 그룹 당 중요한 비트와 덜 중요한 비트의 비트 수는 표 2에 나타나 있다.

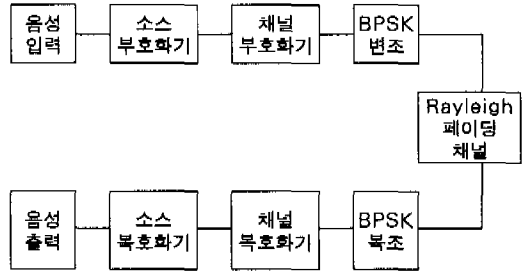


그림 7. 모의 실험된 통신 시스템

표 2. 중요한 비트와 덜 중요한 비트의 비트 수

	비트 수
Case 1	기존의 평처리 방법 사용
Case 2	중요한 비트 16, 덜 중요한 비트 16
Case 3	중요한 비트 24, 덜 중요한 비트 24
Case 4	중요한 비트 12, 덜 중요한 비트 24

터보 코드의 시뮬레이션에서 Rayleigh 분포를 갖는 페이딩 채널환경의 E_b/N_0 에 따른 BER은 표 3에서와 같이 측정되었다.

표 3. 페이딩채널 환경에서 터보 코드의 BER

E_b/N_0	Case 1	Case 2	Case 3	Case 4
6dB	0.02160	0.02078	0.02066	0.02159
7dB	0.01373	0.01337	0.01329	0.01369
8dB	0.00870	0.00813	0.00823	0.00834
9dB	0.00523	0.00492	0.00490	0.00509
10dB	0.00283	0.00283	0.00289	0.00299

각각의 터보 코드의 BER은 중요도에 따른 평처리를 하지 않은 부호화율 1/2(Case 1)경우와 비교하여 볼 때 E_b/N_0 에 따라 거의 비슷하거나 더 나아짐을 알 수 있다.

전체적인 소스 코더와 채널 코더의 성능평가는 페이딩 채널을 통과한 후 복호화된 음성 데이터와 원래 음성 데이터와의 SNR 및 SegSNR을 계산하여 이루어진다. 각 E_b/N_0 에 따른 SNR과 SegSNR의 값은 표 4에 나타나 있다. 기존의 중요도에 따른 평처리를 하지 않은 부호화율 1/2인 경우가 중요도에 따른 평처리를 한 부호화율 1/2에 비해 SNR과 SegSNR면에서 나쁘게 평가됨을 확인할 수 있다. Case 4인 경우는 각각의 E_b/N_0 에 대하여 전체적으

로 좋은 성능을 갖고 있으며 Case 2와 Case3도 Case1에 비해 좋은 성능을 갖고 있음을 알 수 있다. E_b/N_0 이 9dB인 경우를 보면 Case4가 Case1보다 SNR은 1.60dB이상, SegSNR은 0.26dB이상의 좋은 성능을 나타냄을 알 수 있다.

표 4. E_b/N_0 에 따른 SNR 및 SegSNR

		Case1	Case2	Case3	Case4
6dB	SNR	2.38	2.73	2.81	3.26
	SegSNR	1.83	2.11	2.34	2.21
7dB	SNR	3.82	4.12	4.37	4.80
	SegSNR	3.36	3.55	3.68	3.74
8dB	SNR	5.33	6.12	6.58	6.81
	SegSNR	4.60	4.93	5.01	5.06
9dB	SNR	6.56	8.06	7.88	8.16
	SegSNR	5.75	6.01	6.05	6.01
10dB	SNR	8.93	9.80	10.34	9.23
	SegSNR	6.65	6.71	6.77	6.80

객관적인 음질 평가 이외에 주관적인 음질 평가인 MOS 시험을 수행하였으며, 남자 음성 6개, 여자 음성 8개의 음성 데이터를 사용하여 10명에게 평가하였다. MOS 시험 결과는 그림 7에 나타나 있다. 그림 7은 거의 비슷한 BER을 갖는 Case 1과 Case 4인 경우에서 MOS시험을 한 결과이다. 그림 7에서 나타나듯이 모든 경우에서 Case 4가 Case 1보다 음질이 좋게 평가됨을 알 수 있다. 페이딩 채널에서의 E_b/N_0 값이 9dB일 때에는 Case 1인 경우보다 Case 4인 경우가 MOS 0.25정도로 음질이 향상됨을 확인하였다.

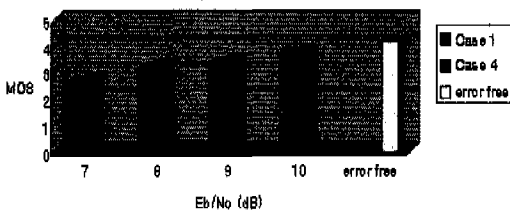


그림 7. MOS시험 측정값

VI. 결론

본 논문에서는 소스 코더의 패킷 데이터의 중요도에 따른 효과적인 터보 코드의 펄칭 방법을 제안하였다. 소스 코더는 CS-ACELP(ITU-T G.729)를

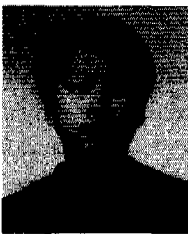
사용하였으며 터보 코드는 부호화율 1/2을 사용하였다. 소스 코더의 패킷 데이터의 에러 감응도를 측정하여 중요도에 따라 터보 코드의 펄칭을 달리하여 채널 코딩을 수행하는 방법을 제안하였다. 페이딩 채널 환경에서 터보 코드의 부호화율을 1/3에서 1/2로 하기 위해 패리티 비트를 펄칭 할 때 기존의 방법보다 본 논문에서 제안한 소스 코더의 패킷 데이터의 중요도에 따른 펄칭 방법이 E_b/N_0 가 9dB일 때 SNR과 SegSNR에서 1.60dB, 0.26 dB의 성능 향상을 얻을 수 있다. 실제 음성 청취에서도 성능 향상을 발견할 수 있다. 이와 같이 소스 코더의 패킷 데이터의 중요도에 따른 터보 코드의 펄칭은 기존의 방법보다 효과적인 채널 코딩 방법이 된다.

참고 문헌

- [1] J. Hagenauer, "Source-Controlled channel decoding", *IEEE Tran. on Commun.*, Vol. 43, No. 9, pp. 2449- 2456, Sept. 1995.
- [2] W.C.Wong, R.steele, W-E.W.sundberg, "Source-Matched Mobile Communications", *IEEE Press.*, 1995.
- [3] D.J Goodman, "Embedded DPCM for Variable Bit Rate Transmission", *IEEE Trans. Commun.*, COM-28, Nol.7, pp. 1040-1046, July 1980.
- [4] R.A. Salami, L. Hanzo and D. Appleby, "A computationally efficient CELP codec with stochastic vector quantization of LPC parameters", *Proc. URSI-ISSE '89.*, Erlangen, W-Germany, pp. 140-143, Sept 1989.
- [5] G. Caire and G. Lechner. "Turbo codes with unequal error protection", *Electronics letters.*, Vol. 32, No.7, pp. 629-631
- [6] Frank Burkert, Giuseppe Caire, Joachim Hagenauer, Thomas Hindelang and Guenther Lechner. "'Turbo' Decoding with Unequal Error Protection applied to GSM speech coding", *Proc. IEEE Globalcom '96.*, Vol. 3, 1996.
- [7] Salami R, Laflamme C, Adoul JP, Kataoka A, Hayashi S, Moriya T, Lamblin C, Massaloux D, Proust S, Kroon P, Shoham Y Design and Description of CS-ACELP - A Toll Quality 8

- kb/s Speech Coder”, *IEEE Trans. on Speech and Audio Processing*, Vol.6 No.2 , pp. 116-130 , Mar. 1998.
- [8] Redwan Salami, Claude Laflamme, Bruno Bessette, Jean-Pierre Adoul, “ITU-T G.729 Annex A: Reduced Complexity 8kb/s CS-ACELP Codec for Digital Simultaneous Voice and Data”, *IEEE Communications Magazine.*, pp.56-63, September 1997.
- [9] C. Berrou and A. Glavieux, “Near optimum error correcting coding and decoding: turbo-codes”, *IEEE Trans. Commun.*, Vol. 44, No. 2, pp. 1261-1271, Oct 1996.
- [10] J. Hagenauer and P. Robertson, “Iterative (“Turbo”) Decoding of Systematic Convolutional Codes with the MAP and SOVA Algorithms”, *ITG- Fachberichte.*, Vol. 130, pp. 21-29, 1995.
- [11] S. Dolinar and D. Divsalar, “Weight distributions for turbo codes using random and nonrandom permutation”, *TDA Progress Report* 42-122, pp. 56-65, August 15, 1995.
- [12] E.K. Hall and S.G. Wilson, “Design and analysis of turbo codes on Rayleigh fading channels”, *IEEE J. Select. Areas Commun.*, Vol. 16, 1998.
- 1986년 5월~1987년 7월 : 한국통신 연구개발단 전임연구원
 1989년 9월~1992년 8월 : Texas A&M University 연구조교
 1993년 2월~1995년 9월 : 한국전자통신연구원 이동통신기술연구단 선임연구원
 1995년 10월~현재 : 충북대학교 전기전자공학부 부교수
 <주관심 분야> 소스 코딩, 채널 코딩, 적응 필터, 이동 통신

장 만 증(Man-jong Chang) 준회원



1998년 : 충북대학교 전자공학과 (학사)
 1998년 2월~현재 : 충북대학교 대학원 전자공학과 석사과정
 <주관심 분야> 채널 코딩, 소스 코딩, 이동 통신

이 인 성(InSung Lee) 정회원



1983년 : 연세대학교 전자공학과 (학사)
 1985년 : 연세대학교 대학원 전자공학과 (석사)
 1992년 : Texas A&M University 전자공학과 (Ph.D)