

실시간 음성 부호화기 구현을 위한 이득테이블 조정에 관한 연구

A Study on the Gain Table Optimazation for Real-Time Speech Codec

김 남 시*, 이 성 권**, 강 준 길***, 김 순 협**

(Nam Si Kim*, Seong Kwon Lee**, Jun Gil Kang***, Soon Hyob Kim**)

*본 논문은 1997년도 광운대학교내 신기술 연구소의 연구 지원에 의해 연구된 것입니다.

요 약

본 논문은 음성 부호화기인 MPMLQ(Multi Pulse Maximum Likelihood Quantization)를 고정 소숫점 범용 DSP에 실시간으로 구현할 때 발생하는 계산량을 줄이기 위한 변형된 형태의 MPMLQ에 관한 것이다. MPMLQ는 음성 신호에서 선형 예측 계수와 피치 정보를 추출하고 남은 잔여 신호와 가장 유사한 여기 신호를 표현할 때 상관법을 이용한다. 상관법은 DSP상에 구현할 때 계수 승산 오버플로우를 발생시킬 수 있으므로 연산후 항상 점검하여야 한다. 이것은 MPMLQ 구현시 전체 계산량의 많은 부분을 차지한다. 본 논문은 이러한 문제점에 착안하여 계수 승산 오버플로우가 발생하지 않도록 입력 음성신호의 크기를 2비트 반쯤줄이고, 이로 인하여 같은 크기로 줄어든 잔여 신호를 고려하여 MPMLQ에서 여기신호의 크기를 표현하는 고정 코드북 이득표를 적절히 조절하였다. 실험결과 변형된 MPMLQ의 SSNR은 0.040325dB(실험 data기준) 향상되었으며, 계산량에 있어서도 17.7%의 처리속도 향상이었다. 따라서 고정 소숫점 범용 DSP에 실시간 구현이 가능하였다.

ABSTRACT

The purpose of this thesis is to reduce the computational load on a general purpose DSP chip with the optimized MPMLQ(Multi Pulse Maximum Likelihood Quantization) method so that the MPMLQ speech codec can be implemented in a real time. In the process of MPMLQ, the residual signal extracted from the original speech signal can be represented with the most similar excitation signal using the correlation method. The correlation method is likely to make multiple overflows in DSP operations and they should be checked after every operation. Not to mention, the processing time dealing with overflows takes many portion of total MPMLQ calculation time. So in this paper, I reduced the magnitude of the input speech signal by 2 bits not to make multiple overflows which results in the reduced magnitude of a residual signal and properly modified the fixed codebook gain table which represents the magnitude of an excitation signal in MPMLQ. The result showed 0.040325 dB increment in SSNR, and 17.7% better performance in computations. It was possible to implement a real time operation on fixed point general purpose DSP.

I. 서 론

정보 및 통신문화가 급속히 발달함에 따라 의사전달의 중요한 수단인 음성처리에 관한 연구가 활발히 진행되고 있다. 음성처리에 관한 연구는 크게 음성부호화, 음성인식, 음성합성, 음성변환으로 나눌 수 있다. 이중 음성부호화는 최근의 멀티미디어 환경에서 크게 부각되는 분야 중 하나가 되고 있다.

음성부호화에서 가장 크게 고려하는 문제는 적은 수의 비트를 할당하여 원래 음성에 가까운 음질을 얻는 것이다. 대역폭이 좁은 통신 채널을 통한 장거리 음성통신은 이런 능력을 요구한다. 장거리 상에서 아날로그 음성의 전송은 일반적으로 리피터를 세 개 이상 통과하면 너무 많은 노이즈 때문에 급속한 음질의 저하가 나타난다. 비록 디지털 음성의 전송은 이러한 음질의 저하를 극복할 수는 있지만 아날로그를 디지털로 표현한 PCM(Pulse Code Modulation)형태에서 전화수준의 디지털 음성은 64 kbits/s가 요구되고 이것은 6KHz 채널인 HF와 30KHz 채널인 VHF밴드를 통해서만 실시간으로 전송할 수 없게된다[1]. 여기서 실시간 전송이란 세그먼트 단위의 디지털 데이터

* 삼성전자

** 광운대학교 컴퓨터공학과

*** 광운대학교 전자공학과

접수일자 : 1998년 1월 14일

를 세그먼트 단위 시간 이내에 처리하고, 전송지연을 포함한 총 전송시간이 50msec를 넘지 않아야 한다[2]. 이러한 필요성에 의해 나타난 선형 예측 부호화(Linear Predictive Coding)와 같은 기술은 음성을 2.4 kbit/s까지 줄일 수 있다. 그러나 심한 음질 저하로 인하여 일반적인 통신에서는 제외되고 있다. 더욱이 음성편지, 화상회의, 인터넷 폰, 멀티미디어와 같은 통합된 서비스의 특징은 자연스러운 음질을 유지하면서 음성표현의 비트 수를 줄일 수 있는 음성부호화기술의 중요성은 더욱 증가되고 있다. 음성편지와 멀티미디어는 음성을 저장하는 효과적인 방법이 요구되는 음성저장장치를 필요로 하며, 화상회의는 이미지 전송과 함께 채널에 음성을 효과적으로 표현하는 기술이 요구된다.

오늘날의 이러한 요구에 의해 8 kbit/s 이하의 음성부호화기에 관한 연구가 활발히 진행되고 있다. 8 kbit/s 이하의 음성부호화에서 최근연구의 대부분은 예측부호화(Predictive Coding)기술을 근간으로 하고 있다. 이 기술이 바탕이 된 가장 일반적인 기술은 1985년 M.R. Schroeder와 B.S. Atal에 의해 처음 제안된 Code Excited Linear Prediction(CELP)[3]과 1983년 T. Arazeki, K. Ozawa, S. Ono, K. Ochiai에 의해 제안된 Multi Pulse Excitation(MPE)이다. 이 두 방법은 전체 구조와 수행되어지는 방법에 있어 매우 유사하지만 여기신호를 표현하는 방법에 있어 예측필터에 다소 차이가 있다.

CELP가 코드북으로 부터 선택된 랜덤 노이즈형태의 여기신호를 사용하는 반면 MPE는 임펄스 여기신호를 사용한다. MPE와 CELP는 7.2 kbit/s이상에서는 뛰어난 성능을 발휘하지만 4.8 kbit/s이하에서는 상당한 음질의 손실을 가지고 있다. CELP시스템은 귀에 거슬리는 쉼 소리와 노이즈가 생기고 MPE 시스템은 거친 소리가 나며 무성음을 잘 처리하지 못한다[4].

CELP기술은 4.8 kbit/s에서 가장 뛰어난 음질을 얻을 수 있고, 압축률도 13:1 이상된다. 이러한 개념 정립에도 불구하고 CELP구현은 아직 복잡한 문제가 있다[5]. CELP 알고리즘은 코드북을 탐색하는데 있어 분석-생성 방법인 폐회로 프로시저를 채택하고 있다. 이것은 모든 후보의 여기신호들을 필터링한 후 생성필터의 출력과 원 입력신호를 비교하여 mean-square 에러를 최소화하는 후보의 열을 찾는 방법으로 탐색을 수행한다[6]. 이것은 매우 많은 곱셈과 덧셈 연산이 필요하게 된다. 따라서 이 프로시저를 직접 구현하는데는 실시간보다 너무 많은 계산시간을 요구한다. 실시간 프로시저는 음성부호화기가 처리하고자 하는 프레임단위 분량의 데이터를 프레임단위 이내에 처리해야만 한다. 따라서 실제적인 CELP시스템은 CELP 계산구조를 지닌 빠른 알고리즘을 써야만 한다. 그러나 빠른 알고리즘을 구현하는데 있어 계산의 복잡성이 크게 늘어나게 된다[7]. 반면 MPE는 분석-생성 모드 또는 상단 탐색 모드에서 수행되어 질 수 있다. 연산량은 CELP에 비해 약 10배 정도 적다. 그러나 여기신호의 표현에 있어 CELP에 비해 덜 정교하기 때문에 CELP에 비해 음질이 다소 떨어진다.

최근 수년간 이 두 가지 기술의 문제점을 극복하기 위한 연구가 활발히 진행되고있다. 또한 디지털 시대의 도래와 함께 디지털 통신을 위해 국제 통신 표준화 기구인 ITU(International Telecommunication Union)에 의해 표준화 작업 또한 활발히 진행되어 왔다. CELP를 축으로 하여 연구는 Adaptive CELP(7.2 kbit/s)[8][9], Low-Delay CELP(16 kbit/s)[10], Algebraic CELP(5.3 kbit/s)[11], Conjugate-Structure CELP(8 kbit/s)등이 있다. MPE를 축으로 한 연구는 Multipulse Linear Predictive Coding(4.8 kbit/s), Multipulse Maximum Likelihood Quantization(6.3 kbit/s)[11], Regular Pulse Excitation[12] 등이 있다.

최근 화상전화와 인터넷폰에서 사용되는 압축률이 6.3 kbit/s인 음성부호화기 MPMLQ (Multipulse Maximum Likelihood Quantization)가 ITU-T에 의해 권고되었다. 이것은 좋은 음질을 유지하면서도 높은 압축률을 지니고 있어 적은 양의 저장장치에 많은 양의 음성을 저장할 수 있는 음성부호화기이다. 본 연구에서는 이를 범용 DSP에 실시간으로 구현하여 각종 멀티미디어 장치에서 사용할 수 있도록 하기 위해 MPMLQ의 고정 코드북 이득 테이블을 조정하여 실험함으로써 최적의 테이블을 선택하였고 이를 이용 범용 DSP에 실시간 구현하였다. 원 MPMLQ와 조정된 MPMLQ에 대한 음질 평가는 음성부호화기가 세그먼트단위로 음성을 처리함에 근거하여 SSNR(Segment Signal to Noise Ratio)로 평가하였으며 평가를 위한 실험 음성 데이터는 남성화자 2명, 여성화자 2명으로 하였다.

본 논문은 5장으로 되어 있으며, 2장에서는 CELP와 MPE의 기본구조 및 IUT-T 표준인 G.723.1의 MPMLQ에 대해 알아보고, 3장에서는 본 논문에서 계산량을 줄이기 위해 변형시킨 MPMLQ에 대해 논한다. 4장에서는 구현 및 실험 결과를 기술하고, 5장에서 이에 대한 결론을 맺는다.

II. 음성부호화기

2.1 CELP(Code Excite Linear Prediction)의 개요

CELP알고리즘은 적은 비트수를 할당하여 좋은 음질의 음성부호화기를 구현하기 위해 벡터 양자화(Vector Quantization)를 이용한다[3]. 그림 2.1은 CELP의 기본적인 블록도이다. 그림 2.1에서 보는 바와 같이 음성 $S(n)$ 은 N개의 샘플단위로 블록화 되어 분석되어진다. 각 블록에 대한 분석은 코드북 CB에 저장되어 있는 K 코드북 열 $x_i(n)$, ($n=1, \dots, N$; $i=1, \dots, K$)로 부터 가장된 mean-squared-error(MSE)기준에 따라 최적의 여기신호를 생성하는 인덱스 k 와 이득 G_k 가 얻어진다.

CELP는 전체 코드북을 탐색하여 최소의 MSE를 출력하는 여기신호 $x_i(n)$ 을 선택한다. 선택된 여기신호에 해당하는 이득 G_k 와 인덱스 k 가 선형 예측필터 $1/A(z)$ 의 계수와 함께 수신단으로 전송되어진다. 이러한 프로시저는 코드북의 모든 열들을 선형 예측필터와 가중필터에 통과시켜야 하기 때문에 많은 계산상 부담이 요구된다[8].

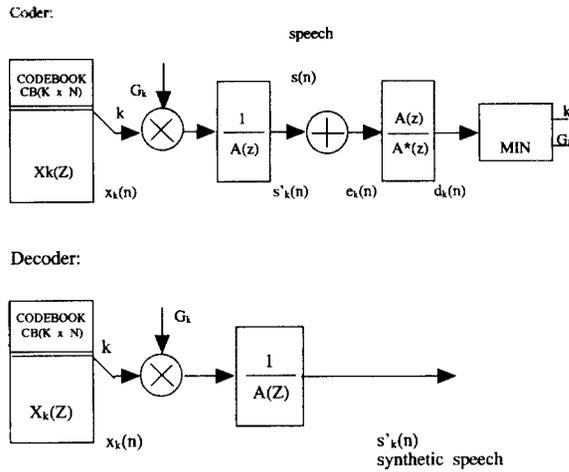


그림 2.1 기본적인 CELP부호화기의 블록도
Fig. 2.1 Basic block diagram of CELP.

2.2 MPE(Multi Pulse Excitation)의 개요

MPE의 기본적인 구조는 CELP와 같지만 잔여신호를 코드북에서 조사하여 최적의 여기신호를 찾아 그 인덱스를 보내는 것이 아니라 원 입력 음성신호에서의 선형 예측계수와 피치정보를 추출하고 남은 잔여신호를 이용하여 상관 탐색모드에서 수행하며 완벽하지는 않지만 대표 하는 큰 값들이 비슷하게 표현된 여기신호의 펄스들의 위치와 이득을 보내는 것이다[13].

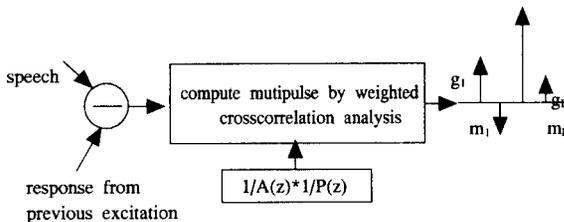


그림 2.2 Multipulse 벡터 생성기의 블록도
Fig. 2.2 Block diagram of Multipulse vector generator.

그림 2.2는 MPE가 CELP의 코드북 대신에 여기신호를 표현하는 Multipulse 벡터생성기의 기본적인 구조를 보여 주고 있다. CELP가 잔여신호를 최적으로 표현하는 여기신호의 코드북 인덱스를 수신단으로 전송하는 반면에 MPE는 multipulse 벡터생성기로부터 나온 여기신호에서의 펄스들의 위치와 이득을 수신단으로 전송한다. 이것은 CELP가 코드북 내의 모든 열을 조사하여 가지는 계산상 부담을 가지지 않으므로 계산상 많은 이득을 볼 수 있다[2].

2.3 ITU-T 표준안 G.723.1

ITU-T 권고안 G.723.1은 very low bitrate을 위한 멀티미디어 서비스(ITU-T 권고안 H.324)에서 음성이나 오디오를 압축하기 위한 음성부호화기이다. 본 부호화기는 5.3kbps와 6.3kbps 두가지를 지원하며 30ms 프레임 간격

으로 선택적으로 사용 가능하다. 또한 불연속 전송과 비음성구간동안 노이즈를 채워 전송하는 가변전송은 선택 사양으로 되어 있다. 본 코더는 제한된 양의 복잡성을 이용하는 위 두 rate들에서 높은 질의 음성을 표현하도록 최적화 되어있다. 음악이나 다른 오디오 신호들은 음성만큼 충실하게 표현하지는 못하지만 부호화와 복호화가 가능하다. 본 부호화기는 30msec 프레임의 음성이나 다른 오디오 신호들을 부호화 한다. 추가적으로 여기에는 7.5 msec의 분량의 선행 데이터가 있어 결과적으로 전체 알고리즘 지연은 37.5msec이다. 본 부호화기의 구현과 수행에 있어 모든 추가적인 지연은 부호화와 복호화에서 데이터를 처리하는데 걸린 실제시간, 커뮤니케이션 링크에서의 전송시간 그리고 멀티플렉싱 프로토콜을 위한 추가적인 버퍼링 지연에서 기인한다. 입력신호는 먼저 아날로그 입력을 전화대역으로 필터링(ITU-T 권고안 G.712)한 다음 8000Hz로 샘플링하고 16 비트 선형 PCM으로 변환한 디지털 신호이다. 이것은 인코딩 전에 16 비트 선형 PCM 데이터로 변환되어야한다. 출력 신호는 위와 비슷한 방법으로 반대로 변환하여 아날로그 신호를 만든다. 그림 2.3은 G.723.1의 전체 블록도이다.

프레임어(Framer)는 연속적인 음성신호 $y[n]$ 을 버퍼링하여 코더가 처리하는 240 샘플의 프레임 $s[n]$ 으로 만든다. 각 프레임은 피치 평가(Pitch Estimation)를 위해 120 샘플씩 두 부분으로 나누어진다. 각 부분은 다시 두개로 나누어 마침내 각 60 샘플씩의 4개의 서브프레임으로 나누어진다.

하이패스필터(High Pass Filter)는 입력 음성 $s[n]$ 으로부터 DC성분을 제거한다. 이 필터의 출력은 $x[n]_{n=0.239}$ 이다.

선형 예측부호화 분석(LPC analysis)은 음성 부호화기들에서 주로 사용하는 10차 선형분석을 이용한다. 이것은 우선 각 서브프레임에 대해 180샘플의 해밍(Hamming)윈도우를 중앙에 씌워 11개의 자기상관계수(autocorrelation coefficient)를 계산한다. 선형 예측계수들은 Levinson-Durbin recursion 방법에 의해 모든 입력 프레임에 대해 각 서브프레임마다 하나씩 모두 4개의 선형 예측계수세트가 계산된다. 이 선형 예측계수들은 short-term 인자 가중필터를 만드는 데 이용된다. 선형 예측 생성필터는 다음과 같다.

$$A_i(z) = \frac{1}{1 - \sum_{j=1}^i a_{ij} z^{-j}}, \quad 0 \leq i < 3 \quad (2.3.1)$$

여기서 i 는 서브 프레임 인덱스이다.

LSP 양자화기(Line Spectrum Pair Quantizer)는 먼저 LPC 계수들에 대해 7.5Hz만큼의 대역 확장을 수행한다. 이것은 계수 0.994로 LPC 생성필터의 폴(pole)을 스케일 한다. 그 다음 $A_3(z)$ 선형 예측필터의 결과는 세 개의 서브벡터로 구성된 8 비트 코드북 형태의 예측 분할 벡터 양자화기(predictive split vector quantizer)를 이 양자화 한다. 여기서 얻어진 인덱스들은 채널을 통해 전송된다.

LSP 디코더는 우선 세 개의 서브벡터를 10차 벡터형태로 디코딩 한다. 예측된 벡터에 디코딩된 벡터와 DC백

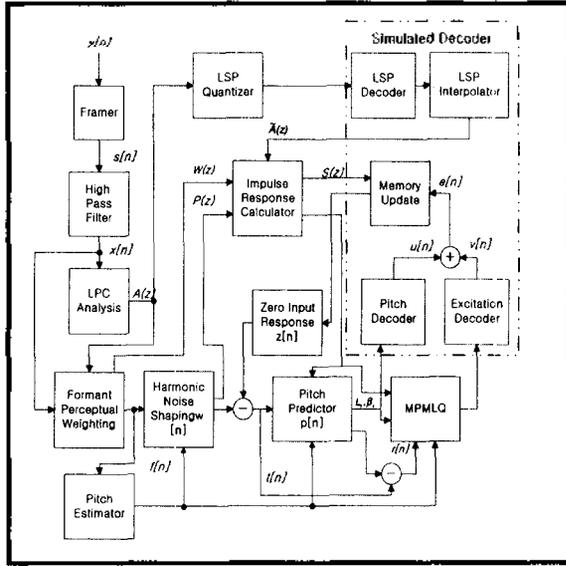


그림 2.3 G.723.1의 블록도
Fig. 2.3 Block diagram of the G.723.1.

터를 더하여 디코드된 LSP벡터를 구한다. 디코드된 LSP 벡터를 식 (2.3.2)의 조건에 따라 안정성(stability)을 점검 한다.

$$\hat{p}_{j+1,n} - \hat{p}_{j,n} \geq \Delta_{\min}, \quad 1 \leq j \leq 9 \quad (2.3.2)$$

여기서 Δ_{\min} 은 31.25 Hz이고, \hat{p}_n 은 n번째 디코드된 LSP벡터이다. 만약 위 식의 조건에 만족하지 않으면 \hat{p}_j 와 \hat{p}_{j+1} 을 아래 방법으로 수정한다.

$$\hat{p}_{avg} = (\hat{p}_j + \hat{p}_{j+1})/2 \quad (2.3.3-1)$$

$$\hat{p}_j = \hat{p}_{avg} - \Delta_{\min}/2 \quad (2.3.3-2)$$

$$\hat{p}_{j+1} = \hat{p}_{avg} + \Delta_{\min}/2 \quad (2.3.3-3)$$

조건을 만족할 때까지 수정하며 계속수행하고 10번 반복해서도 안정성 체크조건에 만족하지 못하면 이전의 LSP 벡터를 사용한다.

선형 보간법(Linear Interpolation)은 각 서브프레임에 대해 디코드된 LSP벡터와 이전의 LSP벡터 사이에서 수행된다. 보간된 4개의 LSP벡터들은 LPC벡터로 변환된다. 양자화된 LPC 생성필터는 디코드된 음성신호를 만드는데 이용된다.

포먼트 인지(Formant Perceptual)가중필터는 양자화되지 않은 LPC계수를 이용하여 각 서브프레임에 대하여 만들어진다. 입력 음성프레임 $\{x[n]\}_{n=0.239}$ 는 4개의 서브프레임으로 나누어진 다음 이 필터를 이용하여 필터링 되어져서 가중된 음성신호 $\{f[n]\}_{n=0.239}$ 가 얻어진다.

피치추정(Pitch estimation)은 매 프레임마다 앞의 두 서브프레임에서 하나, 뒤의 두 서브프레임에서 하나 총 2개

의 피치 추정 값이 계산된다. 이 추정 피치주기는 가중된 음성신호 $f[n]$ 을 이용 계산되어진다. 피치주기는 상호상관 기준 (Crosscorrelation criterion) $Col(j)$ 를 최대로 하는 인덱스 j 가 된다. 이식은 다음과 같다.

$$Col(j) = \frac{\left(\sum_{n=0}^{148} f[n] \cdot f[n-j] \right)^2}{\sum_{n=0}^{148} f[n-j] \cdot f[n-j]}, \quad 18 \leq j \leq 142 \quad (2.3.4)$$

피치주기는 일반적인 실험 결과에 의해 18에서 145범위에서 찾는다. 최상의 피치주기를 찾기 위하여 다음과 같은 조건에 의해 찾아진다. $Col(j)$ 의 최대 값 조사는 $j=18$ 에서부터 시작한다. 찾아진 모든 최대 값 $Col(j)$ 에 대해, 이 값은 이전에 찾아진 최대 값 $Col(j')$ 과 비교한다. 만약 인덱스 j 와 j' 의 차가 18보다 작고 $Col(j) > Col(j')$ 이면 새로운 최대 값이 선택되고 또 인덱스의 차가 18보다 크거나 같으면 $Col(j')$ 이 $Col(j)$ 보다 1.5dB 큰 경우에만 새로운 최대 값이 선택된다. 여기서부터의 모든 계산은 서브프레임을 기본 단위로 수행한다.

하모닉 노이즈표현 (Harmonic Noise Shaping)은 인코딩된 음성의 질을 개선하기 위하여 이용된다. 하모닉 노이즈 필터계수를 계산한 후 가중된 음성 $f[n]$ 은 통과시킴 결과벡터 $w[n]$ 을 얻는다.

임펄스 응답 계산기는 폐회로 분석에서 다음과 같은 결합필터 $S_i(z)$ 을 사용한다. 이 필터는 다음과 같다.

$$S_i(z) = \hat{A}_i(z) \cdot W_i(z) \cdot P_i(z), \quad 0 \leq i \leq 3 \quad (2.3.5)$$

임펄스 응답은 $\{h_i[n]\}_{n=0.59, i=0.3}$ 으로 된다.

결합필터 $S_i(z)$ 의 영 입력응답(Zero input response)은 입력신호가 모두 영인 샘플일 경우 그 필터의 출력 값을 계산함으로써 얻어진다. 공명 뱀셈(Ringing subtraction)은 하모닉 가중 음성벡터에서 영 입력응답을 뺀으로써 얻는다.

피치 예측기(Pitch predictor)는 5차 피치 예측기이며 앞의 피치 추정기에서 찾은 추정 피치 값을 기본으로 하여 서브프레임 각각에 대해 좀더 정확한 피치 값을 예측한다. 서브프레임 0과 2에 대하여서는 피치 추정 값 주위의 +1, -1범위에서 선택되고 7비트를 이용하여 코딩한다 (피치 추정 값은 전송되지 않음). 서브프레임 1과 3에 대해서는 이전 서브프레임에서 계산된 예측 값과 단지 -1, 0, +1, +2만큼만 다르기 때문에 별도로 2비트를 이용 코딩한다. 피치 예측기에서 찾은 피치의 이득은 85개로 구성된 코드북이나 170개로 구성된 코드북중 적절한 코드북을 선정 벡터양자화 한다. 코드북은 서브프레임 0과 1에 대한 피치 예측 값(양자화 되고 디코드된 pitch lag)이 만약 58보다 작거나, 서브프레임 2와 3에 대한 피치 예측 값이 58보다 작으면 피치이득 양자화에 85개로 구성된 코드북을 사용한다. 피치 예측기에서 만든 출력 값을 공명 뱀셈에서 나온 값에서 뺀으로써 잔여신호 $\{r[n]\}_{n=0.59}$ 을 얻는다.

2.3.1 MPMLQ(Multi Pulse Maximum Likelihood Quantization)

MPMLQ는 피치 예측기를 통과하여 나온 잔여신호 $\{r[n]\}_{n=0..59}$ 를 양자화 한다. 양자화 과정은 잔여신호 $r[n]$ 을 $r'[n]$ 으로 근사화 하는 것이다.

$$r'[n] = \sum_{j=0}^{M-1} h_j[n] \cdot v[n-j], \quad 0 \leq n \leq 59 \quad (2.3.1.1)$$

여기서 $v[n]$ 은 임펄스 응답 $h[n]$ 을 가진 결합필터 $S(z)$ 에 대한 여기이고 다음과 같이 정의한다.

$$v[n] = G \sum_{k=0}^{M-1} a_k \delta[n - m_k], \quad 0 \leq n \leq 59 \quad (2.3.1.2)$$

여기서 G 는 이득인수이고, $\delta[n]$ 은 디랙함수, $\{a_k\}_{k=0..M-1}$ 은 부호(\pm)이고, $\{m_k\}_{k=0..M-1}$ 은 디랙함수의 위치들이고 M 은 펄스의 수로 짝수 서브프레임은 6, 홀수 서브프레임은 5이다. 펄스의 위치는 모두 짝수이거나 모두 홀수라는 제한이 있고 이것을 grid bit으로 나타낸다. 모르는 파라메타인 G , $\{a_k\}_{k=0..M-1}$, $\{m_k\}_{k=0..M-1}$ 을 추정하는 것은 에러신호 $err[n]$ 의 mean square를 최소화하여 구할 수 있다.

$$err[n] = r[n] - r'[n] = r[n] - G \sum_{k=0}^{M-1} a_k h[n - m_k] \quad (2.3.1.3)$$

파라메타 추정과 양자화 방법은 분석 생성(analysis-by-synthesis) 방법을 기본으로 한다. G_{max} 파라메타는 다음과 같이 추정되고 양자화 된다. 먼저 상호 상관함수 $d[j]$ 는 임펄스응답 $h[n]$ 과 잔여신호 $r[n]$ 사이에서 계산되어진다.

$$d[j] = \sum_{n=j}^{59} r[n] \cdot h[n-j], \quad 0 \leq j \leq 59 \quad (2.3.1.4)$$

추정된 이득은 다음에 의해 주어진다.

$$G_{max} = \frac{\max\{|d[j]|\}_{j=0..59}}{\sum_{n=0}^{59} h[n] \cdot h[n]} \quad (2.3.1.5)$$

그 다음 추정된 이득 G_{max} 는 로그리즘 양자화기(logarithmic quantizer)에 의해 양자화 되어진다. 이 이득 양자화기는 3.2dB간격으로 24단계로 구성되어있다. 이렇게 양자화된 값 G_{max} 주위에서 추가적인 이득 값들이 범위 $[G_{max} - 3.2, G_{max} + 6.4]$ 내에서 선택되어진다. 이 이득 값들 각각에 대해 펄스의 부호와 위치가 순차적으로 최적화 된다. 이 프로시저는 음과 양 그리드 모두에 대해 반복된다. 그리고 최종적으로 $err[n]$ 의 최소의 mean square가 되는 양자화된 파라메타의 조합이 선택된다. 펄스의 위치와 이득이 최적인 조합이 전송되어진다. 짧은 피치 주기를 가진 음성의 질을 개선하기 위해 다음과 같은 추가적인 프로시저를 사용한다. 만약 서브프레임 0과 1에 대해 L_0

가 58보다 작거나 서브프레임 2와 3에 대해 L_2 가 58보다 작으면 각각의 위치 m_k 를 위해 위의 양자화 프로시저에서의 single 디랙함수 대신에 피치인덱스 L_0 나 L_2 의 주기인 디랙함수의 열이 사용된다. 잔여신호를 표현하는데 있어 single 디랙함수나 디랙함수들의 열이나의 선택은 mean square에러 계산을 근간으로 한다. 최소의 mean square에러가 되는 구성요소가 선택되면 이것의 파라메타 인덱스들이 전송된다.

III. 제안된 MPMLQ

MPMLQ 이전에 선형 예측계수나 피치 정보를 추출은 일반적으로 상관법을 이용한다. 뿐만 아니라 MPMLQ는 앞장에서 언급한 것과 같이 잔여신호를 이용 상호상관법(Crosscorrelation Method)에 의해 가장 큰 이득을 가지는 펄스의 위치부터 순차적으로 짝수프레임에서는 6개 홀수는 5개를 찾는다. 이렇게 찾아진 펄스의 조합이 여기신호가 된다. 여기서 상호 상관법은 계수승산연산이 루프안에서 계산되어 진다[8][12]. 이것은 저렴한 가격대의 16비트 고정 소숫점 범용 DSP에서 쉽게 구현할 수 있다. 그러나 이 범용 DSP는 가산기만을 추가적인 파워없이 오버프로우를 자동 점검한다. 32비트 결과 레지스터를 가진 계수승산의 경우 16비트×16비트×60샘플의 32비트가 가질 수 있는 값을 크게 넘어서기 때문에 항상 오버프로우를 점검하여야 계산 결과 값의 변형을 막을 수 있다. 즉 이것은 계수승산에 있어서는 프로그래머가 직접 오버프로우를 점검할 수 있도록 프로그램을 구현해야한다는 뜻이 된다. 대부분의 범용 DSP가 한 사이클로 계수승산을 수행하지만 오버프로우 점검을 위해 한 사이클 이상을 할당해야하고 이것이 실제구현에 있어 많은 계산량을 차지 실시간 구현에 장애요인이 될 수 있다. 본 논문에서는 상호 상관법 구현 시 오버프로우 점검을 하지 않으면서 권고된 MPMLQ와 같은 음질이 유지되는 변형된 MPMLQ를 연구하였다.

원 MPMLQ의 입력은 16비트 크기의 PCM 데이터이다. 이것이 표현할 수 있는 범위는 -32768~+32767이다. 본 논문에서 제안된 변형된 MPMLQ에서는 입력신호의 크기를 14비트를 할당 입력 값의 범위를 -8192~+8191로 한다. 이것은 14비트×14비트×60 샘플의 결과가 32비트가 표현할 수 있는 수의 범위를 넘지 않는다. 이 경우 원 MPMLQ에 입력된 음성과 변형된 MPMLQ에 입력된 음성은 값의 범위에서 차이가 있을 뿐이지 음성 파형 형태에서는 변화가 없다. 그림 3.1은 원 음성신호와 2비트 할당을 준인 음성신호의 파형을 보여주고 있다.

원 MPMLQ와 변형된 MPMLQ에서 음성신호가 지닌 특징적인 파라메타 즉 선형예측 계수와 피치정보를 추출하는 방법은 동일하고 추출된 파라메타의 값을 보면 거의 동일함을 알 수 있다. 결국 선형 예측계수로 만든 선형 예측필터와 피치정보로 만든 long-term 필터를 통과시켜 나온 잔여신호는 입력 음성 값의 LSB성분이 2비트 줄어든것기 때문에 다소의 손실이 예상되는 신호가 된다[17].

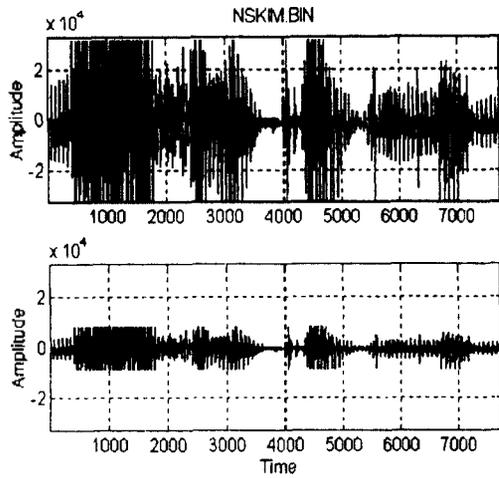


그림 3.1 원 음성신호와 2비트 축소된 음성신호
Fig. 3.1 Original speech signal and 2bit reduced speech signal.

결국 변형된 MPMLQ에 입력될 잔여신호는 입력음성과 마찬가지로 값의 범위가 2비트 줄어들음을 알 수 있다. 결국 잔여신호에 대부분 포함되어 있는 노이즈 성분과 무성음의 양적인 손실이 예상되어진다.

원 MPMLQ는 앞장에서 언급한 것과 같이 3.2dB 간격으로 24 개의 고정 코드북 이득 테이블을 가지고 있다. 이 테이블은 아래 수식과 같은 대수적 양자화기(logarithmic quantizer)의 값을 가지고 있다.

$$20 * \log_{10}(x) = 3.2 * i, \quad i = 1, \dots, 24 \quad (3.1)$$

여기서 x가 테이블을 구성하는 값이 된다. 이것은 잔여신호 중 가장 큰 값부터 찾아진 펄스(짝수 샘플은 6개, 홀수 샘플은 5개)의 대표 이득을 표현하는데 사용된다. 대표이득은 최초 가장 큰 펄스의 이득을 찾고 이득에서 -3.2dB에서 +6.4dB 구간의 이득 값을 펄스의 이득으로 하여 잔여신호와의 차가 최소인 이득을 최종 이득으로 하고 있다.

변형된 MPMLQ는 줄어든 펄스의 값을 표현하는데 사용되는 고정 코드북 이득 테이블을 변형시켜 보다 자세하게 펄스의 값을 표현하였다. 실험을 통하여 얻어진 2.7dB를 간격이 2비트가 줄어든 입력 음성신호의 잔여신호를 최적으로 함을 알 수 있었고 따라서 식 (3.1)은 아래와 같은 식으로 변형되어 24단계의 새로운 고정 코드북 이득 테이블을 만들어졌고 최종 이득을 찾는 범위도 실험을 통하여 -2.7dB에서 +5.4dB로 최적화 하였다.

$$20 * \log_{10}(x) = 2.7 * i, \quad i = 1, \dots, 24 \quad (3.2)$$

아래 그림 3.2는 원 MPMLQ와 변형된 MPMLQ에서 사용된 대수적 양자화기의 테이블 값을 그래프로 나타낸 것이다.

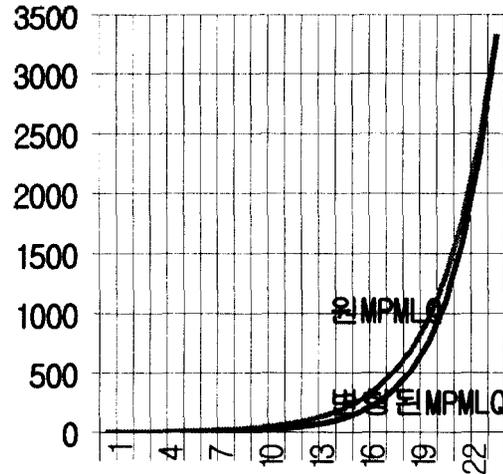


그림 3.2 원 MPMLQ와 변형된 MPMLQ의 대수적 양자화기 비교 그래프

Fig. 3.2 Graph about the comparison of original MPMLQ and modified MPMLQ.

IV. 실험 및 고찰

본 논문에서 변형한 MPMLQ가 원 MPMLQ와 계산량은 많이 줄이면서 음질의 저하는 최소가 되도록 하기 위해 변형시킨 고정 코드북 이득 테이블을 3.2dB에서 0.05dB씩 변화시켜 보면서 테이블을 만들고 C언어로 구현한 시뮬레이션 프로그램에 적용 각각의 SSSNR(Segment Signal to Noise Ratio)를 계산하여 원 MPMLQ와 최소의 SSSNR을 의 차를 보이는 고정 코드북 이득테이블을 선정하였다. 실험에 적용한 음성은 남자 2명, 여자 2명이다. 그리고 이것을 실제 16비트 고정 소숫점 범용 DSP인 ADSP2181에 실시간 구현하였다.

4.1 고정 코드북 이득 테이블 구성

고정 코드북 이득테이블은 대수적 양자화기로서 앞장의 식 (3.1)과 같다. 여기서 3.2는 간격 dB이고 i는 고정 코드북 이득 테이블 원소의 수이다. x가 테이블을 이루는 원소이다.

원 MPMLQ에서 이 테이블의 최대 값은 13246으로 이 값은 잔여신호가 가질 수 있는 최대 범위를 나타내므로 2비트 범위가 줄어든 변형된 MPMLQ에서는 이 값을 2비트만큼 줄인 3312를 최대로 하여 고정 코드북 이득테이블을 구성하였다. 표 4.1과 4.2는 3.2dB 간격을 0.05dB만큼 차를 두도록 하여 새로운 간격을 만들고 이 간격을 이용 실험을 위한 새롭게 구성된 고정 코드북 이득테이블이다.

4.2 고정 소숫점 범용 DSP를 이용한 실시간 구현

본 논문에서는 변형된 MPMLQ를 실시간에 구현하여 보고자 하였다. 고정 소숫점 범용 DSP는 부동 소숫점 범용 DSP에 비해 지가이므로 가능하다면 고정 소숫점 범용

표 4.1 실험을 위한 고정 코드북 이득 테이블 1
Table 4.1 Fixed codebook gain table 1 for experiment.

단위 : dB

간격	3.30	3.20	3.15	3.10	3.05	3.00	2.95
0	1	1	1	1	1	1	1
1	1	1	1	1	1	2	2
2	1	1	2	2	2	2	3
3	2	2	2	3	3	3	4
4	2	3	3	4	4	5	5
5	4	4	5	5	6	7	7
6	5	6	7	8	8	9	10
7	8	9	10	11	12	13	14
8	11	13	14	16	17	19	20
9	16	19	21	22	24	26	29
10	24	28	30	32	34	37	40
11	35	40	43	46	49	52	56
12	51	58	61	65	70	74	79
13	74	83	88	93	99	105	111
14	108	120	127	133	140	148	156
15	158	174	182	191	200	209	219
16	232	251	262	272	283	295	307
17	339	363	376	389	403	417	432
18	495	525	540	556	572	589	606
19	724	759	776	794	813	832	851
20	1059	1096	1116	1135	1155	1175	1195
21	1549	1585	1603	1622	1641	1660	1679
22	2265	2291	2304	2317	2331	2344	2358
23	3311	3311	3311	3311	3311	3311	3311

표 4.2 실험을 위한 고정 코드북 이득 테이블 2
Table 4.2 Fixed codebook gain table 2 for experiment.

단위 : dB

간격	2.90	2.85	2.80	2.75	2.70	2.65	2.60
0	2	2	2	2	3	3	3
1	2	2	3	3	4	4	5
2	3	3	4	4	5	5	6
3	4	5	5	6	7	7	8
4	6	6	7	8	9	10	11
5	8	9	10	11	12	14	15
6	11	13	14	15	17	19	20
7	16	17	19	21	23	25	28
8	22	24	26	29	31	34	37
9	31	33	36	39	43	46	50
10	43	47	50	54	58	63	68
11	60	65	69	74	79	85	91
12	84	90	95	102	108	115	123
13	117	124	132	140	148	157	166
14	164	173	182	192	202	213	224
15	229	240	251	263	275	288	302
16	320	333	347	361	376	391	407
17	447	462	479	495	513	531	550
18	624	642	661	680	700	720	741
19	871	891	912	933	955	977	1000
20	1216	1237	1259	1281	1303	1326	1349
21	1698	1718	1738	1758	1778	1799	1820
22	2371	2385	2399	2413	2427	2441	2455
23	3311	3311	3311	3311	3311	3311	3311

DSP에 구현하는 것이 상용화를 위한 시스템들에서 효율적이다. 따라서 본 논문에서는 고정 소숫점 범용 DSP를 이용 구현하고자 하였다. 현재 시중에는 다수의 고정 소숫

점 범용 DSP가 나와 있다. 본 실험에서 선택한 고정 소숫점 범용 DSP는 Analog Device사의 ADSP2181[19]이었다. 이 DSP는 33MIPS로 동작하며 32K 데이터 메모리와 32K 프로그램 메모리를 내부에 가진 범용 DSP이다. ADSP2181은 33MIPS의 처리속도를 가지므로 음성을 30msec 프레임 단위로 비퍼링하여 처리하는 MPMLQ는 99만 싸이클 이내에 연산을 마쳐야한다. 나눗셈을 제외한 모든 명령이 1 싸이클에 끝나는 ADSP2181은 최대 99만개의 명령을 30msec의 음성을 처리해야만 실시간 처리가 되는 것이다. 실시간 구현의 제약조건은 결국 DSP가 가지는 최대 처리속도 이내에 구현이 가능해야하고 이와 더불어 시스템의 가격측면을 고려하여 어셈블 소스의 코드 사이즈를 가능한 최소로 하여 구현하여야한다. 본 실시간 구현 실험에서는 계산량과 코드 사이즈를 줄이는 것을 동시에 고려하여 구현하였으며 구현된 코드는 범침에 침부하였다.

그림 4.1은 ADSP2181에 구현된 MPMLQ와 변형된 MPMLQ의 계수승산의 예를 보인 것이다. 예로 보인 수식은 아래와 같다.

$$\text{수식 예 : } x = \sum_{n=0}^{60} h[n] \cdot h[n] \quad (4.3.1)$$

그림 4.1에서 보는 것과 같이 서브프레임 하나에 대한 구현에서만 약 60개 명령이나 차이가 있음을 알 수 있다.

원 MPMLQ	변형된 MPMLQ
<pre> m1 = 1; i0 = *h; i0 = 0; i1 = *h; i1 = 0; mr = 0; cntr = 60; do until sum; mx0 = dm(i0,m1); my0 = dm(i1,m1); mr = mr + mx0 * my0(ss); sum = if mv sat mr; </pre>	<pre> m1 = 1; i0 = *h; i0 = 0; i1 = *h; i1 = 0; mr = 0; cntr = 60; do until sum; mx0 = dm(i0,m1); my0 = dm(i1,m1); sum = mr = mr + mx0 * my0(ss); </pre>

그림 4.1 고정 소숫점 DSP에서의 실시간 구현 예
Fig. 4.1 Example of real-time implementation on fixed point DSP.

4.3 실험 결과

4.3.1 고정 코드북 이득 테이블 조정실험

본 연구는 원 MPMLQ와 가장 작은 차이의 SSNR을 가지는 고정 코드북 이득 테이블을 선정하기 위한 실험으로 표 4.1, 4.2의 값을 각 간격마다 테이블로 하여 시뮬레이션 프로그램에 적용 SSNR을 구했다. 입력 음성으로 남성 2명과 여성 2명 총 4명의 테스트 음성을 입력하여 보았다. 표 4.3은 실험에 사용된 입력 음성 데이터의 특성 나타낸 표이다.

표 4.3 실험에 사용된 입력 음성신호의 특성
Table 4.3 The characteristic of input speech signal.

Input Speech Signal Type	Frame 수 (Frame: 30msec)
남 1	512
남 2	656
여 1	983
여 2	674

16비트 범위의 입력음성과 14비트 범위의 입력 음성으로만 하였을 경우에는 LPC계수와 Pitch등에 아주 적은 값의 변화를 가지고 있었다. 이것은 실제 입력 파형의 변화 즉 입력 신호의 크기 변화는 있지만 선형 예측의 특성은 거의 변화하지 않음을 알 수 있었다. 표 4.4는 입력 신호에 따른 LPC 계수의 변화를 보이고 있다.

음질의 측면에서 입력 신호의 다운 스케일링은 실험 데이터의 경우에 있어 원 MPMLQ에서 약 0.1dB의 음질 저하가 발생하였다. 이것은 음성을 평가하는 방법중 하나인 주관적 평가에서는 판단하기 힘든 정도의 음질의 저하였다. 표 4.6의 실험 결과표에서 보는바와 같이 변형된 MPMLQ에서 2.70dB의 간격을 지닌 고정 코드북 이득테이블을 적용한 경우가 원 MPMLQ와 가장 근사한 SSNR을 보였다. 따라서 본 논문에서는 변형된 MPMLQ의 고정 코드북 이득테이블로 2.7dB간격을 채택하였다. 표 4.5에 나타난 결과는 2.7dB간격을 채택한 변형된 MPMLQ와 원 MPMLQ의 SSNR을 비교한 것이다. 이 결과에 의하면 변형된 MPMLQ는 실험 데이터에 있어서 약 0.0403252dB(평균)의 음질향상을 보였다. 그림 4.2는 시뮬레이션을 통해 SSNR의 결과 값을 보인 것이다.

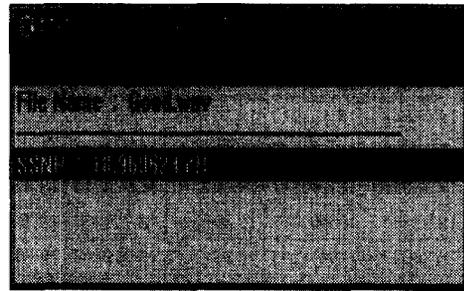


그림 4.2 시뮬레이션 프로그램
Fig. 4.2 Simulation Program.

4.3.2 실시간 구현 실험

MPMLQ를 그림 4.2의 시뮬레이션 프로그램으로 시뮬레이션 하여 각 모듈을 예측한 결과는 표 4.7에 보였다. 이 표는 그림 2.3의 블록도를 기본으로 G.723.1 전체에서 MPMLQ의 계산량을 백분율로 나타낸 것과 각 모듈을 실제 ADSP2181에 구현하였을 때의 계산량과 백분율을 나타내고 있다.

표 4.5 MPMLQ와 변형된 MPMLQ 음질비교
Table 4.5 Speech quality comparison between MPMLQ and the proposed MPMLQ.

단위 : dB

구분	남1	남2	여1	여2	평균
원 MPMLQ	18.960831	18.528168	19.805878	19.143625	19.109626
변형된 MPMLQ	18.960623	18.704473	19.792524	19.142183	19.149951
차이	-0.000208	0.176305	-0.013354	-0.001442	0.04032525

표 4.4 입력신호에 따른 LPC 계수
Table 4.4 LPC coefficient according to input signal.

Frame No.	Sub No.	Input Data Type	0'	1'	2'	3'	4'	5'	6'	7'	8'	9'
Frame1	Sub Frame1	16bit	1.16	-0.28	0.14	-0.07	0.05	-0.06	-0.01	0.02	0.01	-0.01
		14bit	1.16	-0.28	0.14	-0.07	0.05	-0.07	-0.00	0.02	0.01	-0.01
	Sub Frame2	16bit	1.19	-0.25	0.12	-0.07	0.04	-0.07	-0.01	0.04	0.00	-0.02
		14bit	1.19	-0.25	0.12	-0.06	0.04	-0.07	-0.01	0.03	0.01	-0.02
Frame2	Sub Frame1	16bit	1.04	0.01	0.04	-0.01	-0.05	0.00	-0.02	0.04	-0.02	-0.03
		14bit	1.03	0.02	0.04	-0.01	-0.05	-0.01	-0.02	0.04	-0.02	-0.03
	Sub Frame2	16bit	0.98	0.06	0.07	-0.25	-0.02	-0.02	0.07	0.17	-0.09	0.00
		14bit	0.98	0.06	0.08	-0.26	-0.03	-0.02	0.07	0.16	-0.08	0.00
Frame3	Sub Frame1	16bit	1.04	-0.07	0.07	-0.23	0.25	-0.17	-0.01	0.06	-0.11	0.14
		14bit	1.04	-0.07	0.07	-0.23	0.25	-0.16	-0.01	0.06	-0.10	0.14
	Sub Frame2	16bit	1.17	-0.34	0.03	0.02	0.15	-0.31	0.25	0.05	-0.17	0.11
		14bit	1.17	-0.34	0.04	0.01	0.15	-0.31	0.25	0.06	-0.16	0.10
Frame4	Sub Frame1	16bit	1.03	-0.41	0.47	-0.37	0.39	-0.36	0.15	-0.04	0.03	0.02
		14bit	1.02	-0.40	0.47	-0.38	0.39	-0.35	0.15	-0.04	0.03	0.02
	Sub Frame2	16bit	0.76	0.10	-0.09	-0.15	0.25	-0.18	0.01	0.05	0.18	-0.16
		14bit	0.75	0.11	-0.10	-0.15	0.25	-0.17	0.01	0.05	0.18	-0.16

표 4.6 고정 코드북 이득테이블 변경 실험 결과
Table 4.6 Result of the modified fixed-codebook gain table

간격	남1	남2	여1	여2	평균
3.30	18.930483	18.504933	19.015878	19.083725	18.883755
3.20	18.862725	18.431285	19.696878	19.060795	19.012921
3.15	18.885565	18.440141	19.001210	18.984715	18.827908
3.10	18.838737	18.360917	18.899321	18.975052	18.768507
3.05	18.880138	18.401235	18.881705	19.010492	18.793393
3.00	18.895964	18.400103	18.991735	18.945787	18.808397
2.95	18.800673	18.398784	19.015147	19.002738	18.804336
2.90	18.665758	18.331247	18.892154	19.022310	18.727867
2.85	18.768745	18.339046	19.412915	19.038763	18.889867
2.80	18.882442	18.411210	19.431759	18.996191	18.930151
2.75	18.834839	18.691732	19.518333	19.122577	19.041870
2.70	18.960623	18.704473	19.792524	19.142183	19.149951
2.65	18.780226	18.500174	19.703118	19.035414	19.004733
2.60	18.911410	18.314914	18.766943	18.775210	18.692119

표 4.7 G.723.1 각 모듈의 계산량
Table 4.7 Computation-time about each module of G.723.1.

Module	Profile(%)	Implement	
		%	MIPS
High Pass Filter	0.416	0.418	0.117
LPC Analysis	4.009	4.008	1.122
LSP Quantizer	3.495	3.493	0.978
LSP Decoder	0.022	0.020	0.006
LSP Interpolation	0.275	0.274	0.077
Formant Perceptual Weighting	1.705	1.708	0.478
Pitch Estimator	9.559	9.560	2.677
Harmonic Noise Shaping	1.345	1.348	0.387
Impulse Response Calculation	2.789	2.787	0.780
Zero Input Response	2.840	2.839	0.795
Pitch Predictor	33.661	32.640	9.139
MPMLQ	36.302	37.303	10.445
Excitation Decoder	0.523	0.533	0.149
Pitch Decoder	0.109	0.108	0.030
Memory Update	2.950	2.970	0.832

표 4.7은 결과는 실험에 사용된 입력 음성 데이터 모두를 실험하여 얻은 평균값이고 얻어진 계산량 즉 MIPS는 표 4.8의 구현 결과인 28MIPS를 기본으로 하고 있다. 표 4.8은 구현된 MPMLQ의 계산량과 사용된 메모리량을 보여주고 있다.

표 4.8 실시간 구현 결과
Table 4.8 Result of real-time implementation

*ADSP2181 Program Memory 1Word 24bit, Data Memory 1Word 16bit

구분	계산량	구현된 알고리즘의 실시간 처리 여부	프로그램 메모리 크기	데이터 메모리 크기
원 MPMLQ	34MIPS	불가능	8852Word*	13497Word
조정된 MPMLQ	28MIPS	가능	8814Word	13497Word

표 4.8에서 보는 것과 같이 원 MPMLQ에서 모든 오버프로우를 짐작하도록 하여 구현하였을 경우에는 34MIPS의 계산량을 가지므로 실험에서 구현코자한 ADSP2181에서 실시간 구현이 불가능하였다. 그러나 제안된 MPMLQ의 결과는 28MIPS로 실시간 구현이 가능함을 알 수 있다. 실제로 다른 최적화 알고리즘으로 구현한다면 실시간 구현이 가능하나 본 실험에서는 입력 데이터의 표현 범위를 2비트 줄이고 MPMLQ의 코드북 테이블을 입력 데이터의 범위에 맞도록 수정하여 오버프로우 연산을 줄임으로써 쉽게 17.7%의 계산량 성능향상이 가능하였다.

4.4 고찰

조정되어 최적의 고정 코드북 이득 테이블 선택된 2.7dB 간격의 테이블에서 시험 데이터의 평균 SSNR은 원 MPMLQ와 거의 같은 수준의 음질을 유지하였다. 남 1, 여 1 그리고 여 2 데이터는 원 MPMLQ의 SSNR보다 아주 작은 정도(0.005dB)의 음질 저하를 안고 있지만 거의 같은 수준으로 볼 수 있었다. 그러나 남 2에서는 조정된 MPMLQ에서의 SSNR이 특별히 좋아져 전체 실험 데이터의 평균은 좋아진 것으로 나타났다. 그 원인은 입력 데이터의 특징으로 살펴본 결과 남 2의 데이터는 음성 데이터의 크기 범위 즉 전체적인 에너지 레벨이 타 실험 데이터보다 상당히 높았고 이것은 고정 코드북 이득 테이블의 표현 범위가 줄어들게 됨으로써 원 MPMLQ보다 작은 음질저하를 나타낸 것으로 분석되었다.

본 실험에서 입력 음성데이터의 범위를 줄여 17.7%의 처리속도 성능향상을 얻어 실시간 구현을 쉽게 할 수 있도록 하고 이로 인한 음질의 손실을 최소화 하기 위해 조정된 고정 코드북 이득테이블을 가진 MPMLQ의 음질이 0.040325dB의 성능향상을 가져온 것은 다른 음성부호 화기의 설계 시에도 충분히 고려하여 설계할 수 있는 방법을 제시하였다.

V. 결론

본 연구에서 변형된 MPMLQ를 제안하고 고정 소숫점 범용 DSP에 실시간 구현하였다. 변형된 MPMLQ는 원 MPMLQ보다 계산량을 17.7% 줄이면서도 미미하지만 0.040325dB의 음질향상을 보였다. 본 논문에서는 음성의 특징 파라메타 즉 선형 예측계수와 피치정보의 추출은 ITU-T 권고안 G.723.1을 따라 구현하였다. 변형된 MPMLQ는 음성 값의 범위가 작은 화자에 대해서는 원 MPMLQ와 동등한 수준의 음질을 보였다. 본 논문은 여기신호의 이득만을 고려하여 반영하였으나 음성 값의 범위에 따라 같이 달라질 수 있는 피치이득을 표현하는 피치이득 테이블도 함께 조정하여 준다면 좀 더 좋은 성능의 향상을 보리라 본다. 음질 평가에 있어 본 논문에서는 SSNR만을 고려하였으나 음성이 단순히 정량적 압축방법만을 취하지 않고 인간의 감각특성을 고려한 것이므로 MOS(Mean Opinion Score), DAM(Diagnostic Acceptability Measure) 등과 같은 주관적인 평가가 함께 되었으면 좋으리라 본다.

참고 문헌

1. W. Grieder, A. Langi, and W. Kinsner, "Codebook searching for 4.8kbps CELP speech coder", Proc. IEEE conf. WESCANEX'93, pp.397-406, May. 1993.
2. ITU-T Recommendation G.114, "General Characteristics of International Telephone Connection and International Telephone Circuits", Feb. 1996.
3. M.R.Schroeder and B.S.Atal, "Code-excited linear prediction (CELP):high quality speech at low bit rates", Proc. of ICASSP, Mar. 1985.
4. R. L. Zinser and S. R. Koch, "4800 and 7200 bit/sec Hybrid Codebook Multipulse Coding", IEEE Trans. Acoust., Speech, Signal Processing, ICASSP-89 vol.2, pp.747-750, May. 1989.
5. P. Kroon and K. Swaminathan, "A High-quality multirate real-time CELP coder", IEEE J. Selected Areas in Comm. vol.10, pp.850-857, June 1992.
6. A. Langi and W. Kinsner, "Code-Excited Linear Predictive Speech Processing for Digital Transmission and Storage", IEEE Western Canada Conf. WESCANEX'91, pp.205-209, May. 1991.
7. F. F. Tzeng, "Multipulse Excitation Codebook Design and fast search methods for CELP speech coding", IEEE conf. GLOBECOM'88. vol.1, pp.590-594, Dec. 1988.
8. S. Yang and X. Chen, "Improved Adaptive CELP Coder at 5.2kbit/s", Proc. TENCON'93 IEEE Computer, Comm., Control, and Power Eng., vol.3, pp.299-302, Oct. 1993.
9. C. R. Galand, J. E. Menez, and M. M. Rosso, "Adaptive Code Excited Predictive Coding", IEEE Trans. Signal Processing, vol.40, pp.1317-1326, June 1992.
10. S. Taumi, K. Ozawa, and T. Nomura, "Low-delay CELP with Multi-pulse VQ and Fast search for GSM EFR", Proc. IEEE Int. Conf. Acoust., Speech, Signal Processing, vol.1, pp.562-565, May. 1996.
11. ITU-T Recommendation G.723.1, "Dual rate speech coder for multimedia communications transmitting at 5.3 & 6.3 Kbit/s", Study Group 15, Oct. 1995.
12. P.Kroon, E.F.Deprettere, and R.J.Sluyter, "Regular pulse excitation: a novel approach to effective and efficient multipulse coding of speech", IEEE Tran. ASSP, vol.34, Oct. 1986.
13. H. C. Woo and J. D. Gibson, "Multipulse-based Codebooks for CELP coding at 7kbps", IEEE Int. Conf. Acoust., Speech, Signal Processing, vol.1, pp.136-139, May. 1989.
14. M. J. Bae, H. Y. Whang, and H. Y. Yoo, "On a Fast Pitch Searching by using a Simple Correlation Technique in the CELP Vocoder", Proc. IEEE Circuits Systems, vol.2, pp.1256-1259, Aug. 1995.
15. M. Copperi, "Efficient excitation modeling in a low bit-rate CELP coder", IEEE Int. Conf. Acoust., Speech, Signal Processing, ICASSP-91 vol.1, pp.233-236, Apr. 1991.
16. M.R. Suddle, A. M. Kondos, and B. G. Evans, "DSP implementation of Low bit-rate CELP based speech coders", IEEE Int. Conf. Digital Processing of Signals in Comm., pp.309-314, 1991.
17. U. Kipper, H. Reininger, and D. Wolf, "Improved CELP

coding using Adaptive excitation Codebooks", IEEE Int. Conf. Acoust., Speech, Signal Processing, ICASSP-91 vol.1, pp.237-240, Apr. 1991.

18. L. Martins de Silva and A. Alcaim, "A Modified CELP Model with Computationally Efficient Adaptive Codebook Search", IEEE, Signal Processing Letters, vol.3, pp.44-45, Mar. 1995.

▲김 남 시(Nam Si Kim)

현재 : 삼성전자 멀티미디어 연구소, 기술총괄, 전임연구원
 *주관심분야: 멀티미디어, DSP, 음성 합성
 e-mail : nskim@mmmd.sec.samsung.co.kr

▲이 성 권(Seong Kwon Lee)

음향 학회지 15권 2E호 참조
 현재 : 광운대학교 컴퓨터 공학박사
 *주관심분야: 음성인식, 멀티미디어, 디지털 신호처리, 신경망
 e-mail : joyer@explore.kwangwoon.ac.kr

▲김 순 협(Soon Hyob Kim)

음향 학회지 15권 2E호 참조
 현재 : 광운대학교 컴퓨터 공학과 교수
 *주관심분야: 음성언어 및 합성, 신경회로망, 디지털 신호처리, 데이터 통신
 e-mail : kimsh@daisy.kwangwoon.ac.kr

▲강 준 길(Jun Gil Kang)



현재 : 광운대학교 전자공학과 교수
 *주관심분야: ATM교환망, 컴퓨터 네트워크, 신호처리