

# FS-CELP 음성 부호화기의 고정 소수점 성능 분석 및 구현

正會員 孫 鍾 瑞\*, 金 時 鉉\*, 姜 志 浪\*, 成 元 鎔\*

## Fixed-Point Performance Analysis and Implementation of the FS-CELP Vocoder

Jongseo Sohn\*, Seehyun Kim\*, Jiyang Kang\*, Wonyong Sung\* *Regular Members*

### 要 約

FS-1016 CELP(Code Excited Linear Prediction) 음성 부호화기의 유한 단어길이 효과(finite wordlength effect)를 분석하였고 구간 부동 소수점 방식(block floating-point method)을 적용하여 성능을 향상시켰다. 이때 스케일링에 필요한 정수 단어길이(integer wordlength)를 실행 시간(run-time)에 추정하는 알고리즘을 개발하였으며, TMS320C50 고정 소수점 디지털 신호처리기를 이용하여 전체 음성 부호화기의 성능 향상을 실시간으로 검증하였다. 고정 소수점 어셈블리 프로그램은 자동 스케일링을 지원하는 Autoscaler 개발환경을 이용하여 개발되었다.

### Abstract

Finite wordlength effects of the FS-1016 CELP(Code Excited Linear Prediction) vocoder algorithm is analyzed, and a block floating-point implementation method is employed to improve the fixed-point performance. An efficient run-time integer wordlength estimation algorithm is developed, and the overall system performance is verified in real-time using a TMS320C50 emulation board. Autoscaler software that conducts simulation-based automatic scaling to provide a floating-point like programming environment is used for this application development.

### I. 서론 및 개발 환경

고정 소수점 디지털 신호처리기는 전력 소모나 가

격 면에서 부동 소수점 디지털 신호처리에 비해 매우 유리한 반면 유한 단어길이 효과와 적절한 스케일링의 필요성 때문에 응용 프로그램의 개발이 더 어렵다. 또한 주어진 하드웨어의 모든 비트들을 최대한 활용하면서 오버플로우(overflow)를 방지하기 위해서는 스케일링뿐만 아니라 연산의 순서도 신중하게 고려되어야 한다. 본 연구에서 구현한 4.8Kbps의 낮은

\*서울대학교 공과대학 전기공학부 및 반도체 공동 연구소  
School of Electrical Engineering and Inter-University Semiconductor Research Center, Seoul National University  
論文番號: 95408-1125  
接受日字: 1995년 11월 25일

선충률을 갖는 FS-1016 CELP<sup>[11]</sup> 알고리즘은 코드북 검색과 이득 결정시 큰 다이내믹 레인지(dynamic range)를 갖는 변수들이 많기 때문에 고정 소수점 구현에 따른 유한 단어길이 효과에 의한 성능 저하가 매우 심각하다. 이러한 문제들을 해결하기 위하여 구간 부동 소수점 방식을 사용함으로써 계산량이 작으면서도 고정 소수점 효과가 작은 고정 소수점 FS-1016 CELP 알고리즘을 개발하였다.

본 개발에 사용된 고정 소수점 응용 프로그램 개발 환경이 그림 1에 나타나 있다.<sup>[12]</sup> 첫 단계로 알고리즘을 검증하기 위하여 부동 소수점 C 프로그램이 개발되고 모의 실험이 행해진다. 다음 단계에서는 부동 소수점 C 프로그램이 변수들의 형변환(type conversion)을 통하여 고정 소수점 C 프로그램으로 바뀌고 연산자 오버로딩(operator overloading)에 의하여 모든 연산들은 고정 소수점 클래스에 정의된 대로 고정 소수점으로 행해진다. 이 고정 소수점 클래스에서는 변수나 상수들에 전체 단어길이, 정수 단어길이(integer wordlength), 저장 형태, 반올림 여부와 오버플로우 처리 방법 등에 대한 정보들을 할당하여 고정 소수점 모의 실험을 행한다. 이러한 모의 실험 과정에서 고정 소수점 구현시 전체 시스템의 성능을 저하시키는 부분을 찾은 후, 알고리즘의 변환을 통하여 고정 소수점 구현에 따른 성능 저하를 보상한다. 이 과정에서 코드북 검색과 이득 결정의 구현을 위한 세가지 알고리즘-직접 나눗셈 방법, 대각 곱셈 방법, 그리고 재공간 방법을 비교하였다. 세번째 단계로, Autoscaler에서 제공하는 가상의 부동 소수점 명령어들을 이용한 어셈블리 프로그램이 개발되고 Autoscaler의 부동 소수점 모의 실험기를 통하여 검증된다. 가상의 부동 소수점 어셈블리 프로그램은 모의 실험중 부동 소수점 C 프로그램의 결과와 직접 비교함으로써 쉽게 검증될 수 있는데 이는 어셈블리 프로그램의 개발을 매우 편리하게 한다. 이러한 모의 실험 과정에서 모든 변수들의 범위가 측정된다. 네번째 단계에서는 부동 소수점 어셈블리 프로그램이 고정 소수점 어셈블리 프로그램으로 자동적으로 변환된다. 이때 스케일링을 위하여 필요한 쉬프트(shift) 횟수는 부동 소수점 모의 실험 동안에 측정된 범위 정보를 가지고 자동적으로 결정된다. 최종적으로 이 고정 소수점 어셈블리 프로그램은 에뮬레이션 보드(emulation board)에서

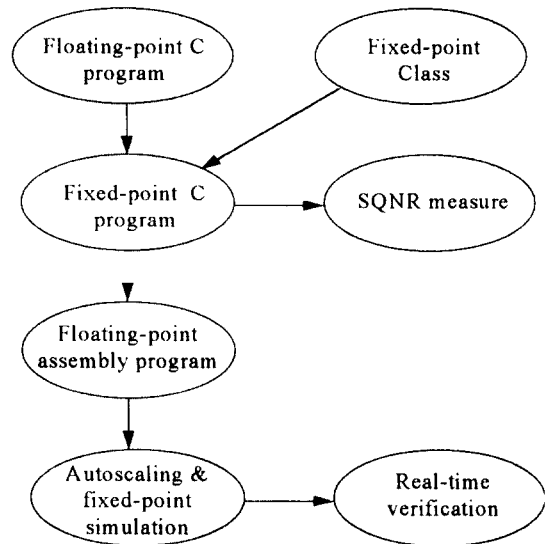


그림 1. Autoscaler를 이용한 개발 과정  
Fig 1. Development procedure with Autoscaler

실시간으로 검증된다.

이처럼 Autoscaler 개발환경에서는 프로그래머가 오직 부동 소수점 C 프로그램과 부동 소수점 어셈블리 프로그램만을 개발하게 되고, 고정 소수점 프로그램은 자동적으로 생성되므로 고정 소수점 프로그램 개발이 매우 편리해지며 개발 시간을 줄일 수 있게 된다.

본 논문의 2장에서는 FS-CELP 알고리즘이 간단히 설명이 된다. 3장에서는 각 구현 방법에 따른 고정 소수점 효과가 분석이 되며, 4장에서는 구간 부동 소수점 방식을 사용한 고정 소수점 알고리즘이 제시된다. 그리고 5장에서는 TMS320C50을 이용한 실시간 구현 결과를 보이며, 6장에서 결론을 맺는다.

## II. FS-CELP 알고리즘

FS-CELP 음성 부호화기는 LPC(Linear Prediction Coefficient) 분석, LPC와 LSP(Line Spectrum Pair) 계수간의 변환, LSP 계수들의 선형 보간(linear interpolation), 그리고 적응형 코드북(adaptive codebook)과 불규칙 코드북(stochastic codebook)의 검색 등으로 구성된다.<sup>[13]</sup> FS-CELP 알고리즘에서는 240개의 음성

샘플(sample)로 구성된 프레임 단위로 음성신호에 대한 처리가 이루어지는데, 부호화기에서는 각 프레임마다 음성 데이터의 LPC 분석을 통해 10차의 합성 필터(synthesis filter)의 계수들을 구하게 된다. 이 계수들은 예러에 강하고 선형 보간시 그 특성이 좋은 LSP 계수로 변형된 후 양자화되어 부호화기로 전송된다. 부호화기에서 양자화된 LSP 계수들은 60개의 음성 샘플로 구성된 4개의 각 부분 프레임(subframe)들에 대하여 전후 프레임의 LSP 계수들과 선형 보간된 후 다시 각 부분 프레임들마다 LPC 합성 필터의 계수들로 변환된다. 부호화기에서는 부호화기에서와 같은 방법으로 필터의 계수를 구하게 되므로 서로 같은 합성 필터를 사용하게 된다.

합성 필터의 입력으로 사용될 여기 신호(excitation signal)는 합성에 의한 분석(analysis by synthesis)과정을 통하여 결정된다. 즉, 미리 코드북에 저장되어 있는 일련의 여기 신호들을 각각 합성 필터를 통과시키면서 복원된 음성들을 생성해 내게 되는데, 그중 원래 음성 신호와 가장 가까운 복원 음성을 만들어내는 여기 신호의 변호(index)와 양자화된 이득값이 부호화기로 전송된다. 여기 신호들은 두개의 코드북에 저장되는데 하나는 장기 예측, 즉 피치 검색을 위한 적응형 코드북이고 다른 하나는 단기 예측을 위한 불규칙 코드북이다.<sup>[6]</sup> 장기 예측을 위한 코드북은 음성의 주기성(pitch period)을 모델링하기 위한 것으로서 그 원소 신호들이 매 부분 프레임마다 갱신되는 반면 불규칙 코드북은 그 원소들이 미리 고정되어 있다. 부호화기에서는 부호화기에서 최종적으로 선택된 두 여기 신호에 각각 이득값을 곱한 후 이를 더하여 만들어진 최종 여기 신호를 합성하여 출력하며, 이 최종 여기 신호는 부호화기에서 적응형 코드북의 갱신에도 이용된다. 검색을 위해서는 코드북에 저장된 모든 여기 신호들에 대하여 합성을 해야 하기 때문에 코드북 검색 과정은 CELP 알고리즘에서 가장 많은 계산량을 필요로 하게 된다. FS-CELP 알고리즘에서는 이러한 코드북 검색을 간단하게 하기 위하여 불규칙 코드북의 경우 각 여기 신호가 서로 두 샘플씩 쉬프트되어 중첩된 구조의 코드북을 사용한다.<sup>[11]</sup>

그림 2는 코드북 검색 과정의 블록 다이어그램을 나타낸다. 주관적으로 더 나은 음질을 얻기 위하여, 합성된 신호와 음성 신호간의 오차를 오차 가중화 필

터(error weighting filter)를 통과시켜 노이즈 웨이핑(noise shaping)을 한 후, 그 가중화된 오차의 에너지를 최소화하는 코드의 번호를 찾게 된다.<sup>[6]</sup> 계산량을 줄이기 위하여 오차 가중화 필터링은 그림 3과 같이 오차 가중화 합성 필터(weighted synthesis filter)를 사용하여 합성과 동시에 수행된다.<sup>[11]</sup> 이 오차 가중화 합성 필터는 극점만을 가진(all-pole) 무한 충격 응답(infinite impulse response) 필터이지만 검색중에는 적당한 길이( $R$ )의 충격 응답(impulse response)을 구하여 유한 충격 응답(finite impulse response) 필터( $h$ )로 근사화시켜 사용한다.<sup>[8]</sup> 그러나 부호화기에서는 부호화기에서와 마찬가지로 합성 필터의 계수들을 구한 후 direct form 무한 충격 응답 필터로 직접 음성을 합성해내게 된다.<sup>[9]</sup>

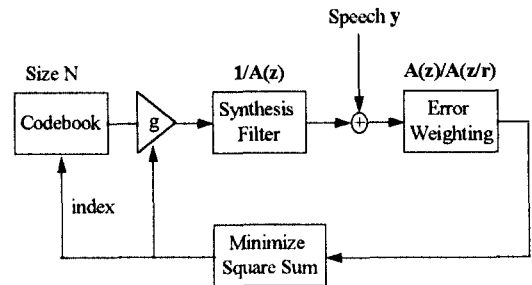


그림 2. 코드북 검색 과정  
Fig 2. Codebook search procedure

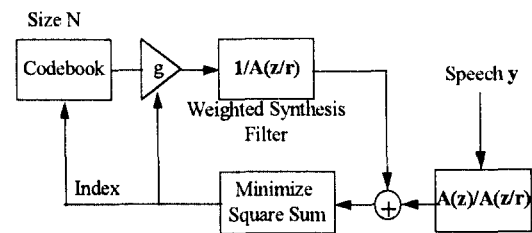


그림 3. 변형된 검색 과정  
Fig 3. Modified search procedure

각 신호들을 한 프레임의 표본 수와 같은 차원의 벡터들로 해석하면, 길이가  $N$ 인 한 프레임에 대하여 필터링을 위한 컨볼루션(convolution)은 다음과 같이 정의된  $N \times N$  행렬과의 곱으로 표현될 수 있다.

$$H = \begin{bmatrix} h_0 & 0 & 0 & \dots & \cdot & 0 \\ h_1 & h_0 & 0 & \dots & \cdot & \cdot \\ \cdot & \cdot & \cdot & \dots & \cdot & \cdot \\ h_{R-1} & h_{R-2} & h_{R-3} & \dots & \cdot & \cdot \\ 0 & h_{R-1} & h_{R-2} & \dots & \cdot & \cdot \\ \cdot & \cdot & \cdot & \dots & \cdot & \cdot \\ \cdot & \cdot & \cdot & \dots & \cdot & h_0 \\ \cdot & \cdot & \cdot & \dots & \cdot & h_1 \\ \cdot & \cdot & \cdot & \dots & \cdot & \cdot \\ \cdot & \cdot & \cdot & \dots & h_{R-1} & h_{R-2} \\ 0 & \cdot & \cdot & \dots & 0 & h_{R-1} \end{bmatrix} \quad (1)$$

위의 식에서  $h_i$ 는 오차 가중화 합성 필터의 중격응답 계수이다. 이때 음성 벡터  $s$ , 무입력 응답(zero-input response) 벡터  $z$ 와 양자화되지 않은 코드북의 원소 벡터  $x$ 에 대하여 다음 식이 성립한다.

$$s = Hx + z = t_0 + z \quad (2)$$

양자화는 코드북 원소 벡터들을 합성시킨 신호중 목표 신호  $t_0$ 에 가장 가까운 원소 벡터의 번호를 전송함으로써 행해진다. 검색은 적응형 코드북과 불규칙 코드북에 대하여 두번 행해지는데, 먼저 적응형 코드북에 대한 검색이 실시된 후 불규칙 코드북 검색이 행해진다. 불규칙 코드북 검색시에는 식 (2)의 목표 신호  $t_0$ 에서 적응형 코드북을 이용하여 합성한 신호를 뺀 값이 다시 목표 신호가 되어 검색을 반복하게 된다. 복호화기에서 최종적으로 음성을 합성해내기 위해서는 선택된 적응형과 불규칙 코드북의 여기 신호들을 더하여 합성 필터의 입력으로 사용한다. 목표 신호  $t_0$ 와 합성된 신호들과의 거리는 오차의 에너지로 표현되는데, 주어진 목표 벡터  $t_0$ 와 코드북 원소 벡터  $x_i$ 를 오차 가중화 합성 필터  $h$ 를 통과시켜 얻어지는 합성 신호  $y_i$ 에 대하여 오차 벡터는 다음과 같다.

$$e_i = t_0 - g_i y_i \quad (3)$$

여기서의  $g_i$ 는 이득값이다. 이때 오차의 에너지,  $E_i = e_i^T e_i$ 를 최소화하는 코드북 번호와 그때의 이득값을 결정해야 한다. 이득은 목표 신호와 합성된 신호사이의 상관값(correlation)과 합성된 신호의 에너지(energy)를 이용하여 구할 수 있다.<sup>[7]</sup>

$$g_i = \frac{t_0^T y_i}{y_i^T y_i} = \frac{correl_i}{eng_i} \quad (4)$$

이때 오차의 에너지는 아래와 같다.

$$E_i = t_0^T t_0 - \frac{correl_i^2}{eng_i} \quad (5)$$

식 (5)에서 첫번째 항은 코드북 원소 번호  $i$ 에 대하여 상수이므로 전체 오차의 에너지를 최소화하는 것은 적합지수(match score)<sup>[8]</sup>라고 하는 두번째 항( $\frac{correl_i^2}{eng_i}$ )을 최대화하는 것이다. 선택된 번호에 대한 이득값은 양자화된 후 번호와 함께 복호화기로 보내진다.

### III. 코드북 검색에서의 고정 소수점 구현 효과

FS-1016 CELP 음성 부호화기는 알고리즘이 고정 소수점 구현에 대한 고려없이 기술되어 있기 때문에 이를 고정 소수점으로 구현할 때 구현하는 알고리즘과 구조에 따라서 성능이나 특성이 많이 달라지게 된다. 본 논문에서는 전체 알고리즘에서 가장 많은 계산량을 필요로 하고 부호화기의 성능에 결정적 영향을 미치는 코드북 검색 과정을 새가지 고정 소수점 구현 방법에 대하여 분석하였다. 각 방법에 대한 성능 평가는 고정 소수점 C 프로그램을 이용한 모의 실험을 통하여 행해졌다. 모든 변수들에는 16비트를 할당하였고 누산기(accumulator)를 모델링하기 위하여 중간 결과들은 32비트의 단어길이를 갖는 임시 변수에 저장하였다. 이 코드북 검색에서는 위의 식 (5)에서 적합지수의 최대치를 찾는 것을 수행하는데, 이때 필요한 상관값,  $correl_i$ 와 에너지,  $eng_i$ 의 각 구간에 따른

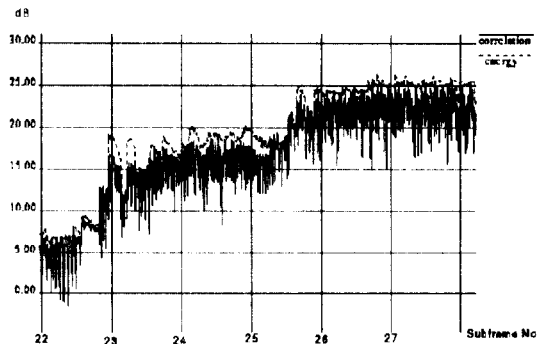


그림 4. 상관값과 에너지 다이내믹 레인지  
Fig 4. Dynamic range of correlation value and energy

다이내믹 레인지가 그림 4에 보인다. 이 그림에서 보는 바와 같이  $correl_i$ 와  $enrg_i$ 는 각각 상당히 큰 다이내믹 레인지를 가지나, 둘의 크기는 서로 비슷하게 움직인다.

1. 직접 나눗셈 방법(direct division method)

식 (5)에서의 적합지수를 다음의 식과 같이 직접 비교하는 방법이다.

$$\frac{correl_i^2}{enrg_i} < \frac{correl_j^2}{enrg_j} \quad (6)$$

이 적합지수의 다이내믹 레인지는 범위 추정기(range estimator)<sup>4)</sup>를 이용한 모의실험에 의하면 그림 5에서 보는 바와 같이 120dB 정도로 상당히 제한되어 있는데, 이는 그림 4에서 보는 바와 같이 각 프레임에서 상관도와 에너지가 매우 높은 상관관계를 갖기 때문이다. 이때, 상관도를 에너지로 나눈 후 다시 상관도를 곱하는 순서( $\frac{correl_i}{enrg_i} * correl_i$ )로 적합지수를 계산함으로써 더 높은 정밀도를 유지할 수 있다. 이처럼 제한된 다이내믹 레인지 때문에 적합지수를 통한 검색 방법은 표 1에서처럼 상대적으로 좋은 고정 소수점 성능을 나타낸다. 그러나, 이 방법은 코드북의 크기가  $M$ 인 경우 총  $M$ 번의 나눗셈과  $M$ 번의 곱셈 연산이 필요하다. 대부분의 고정 소수점 신호처리는 하드웨어 나눗셈기가 없으므로 이는 상당한 계산량의 증가를 가져오게 된다. TMS320C50에서 나눗셈은 조건부 뺄셈(conditional subtract)을 이용하여 구현되는데

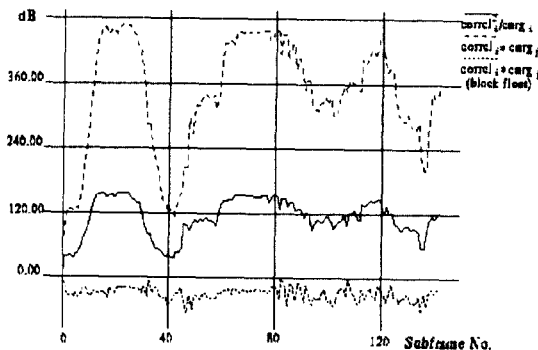


그림 5. 임시 변수들의 최대 값들

Fig 5. Maximum values of the temporary variables.

약 25 사이클(cycle)이 소요된다.<sup>1)</sup>

표 1. 세가지 방법의 성능 비교

Table 1. Performance comparison of the three methods

	나눗셈	대각 곱셈	제곱근
부동 소수점	11.17 dB	11.17 dB	11.71 dB
고정 소수점 (16bit)	9.62 dB	6.88 dB	8.73 dB
연산 수 (코드북 크기 N)	N 곱셈, N 나눗셈	3N-2 곱셈, 1 나눗셈	2N-2 곱셈, N 제곱근
계산량 (MIPS)	30.3	25.5	34.8

2. 대각 곱셈 방법(cross multiplication method)

다음과 같이 분모의 에너지를 대각으로 곱함으로써(cross multiplication) 연산 시간이 긴 나눗셈을 없애는 방법이다.<sup>1)</sup>

$$correl_i^2 enrg_j < correl_j^2 enrg_i \quad (7)$$

이 방법은 코드북의 크기가  $M$ 인 경우 오직 1번의 나눗셈과  $3M-2$ 번의 곱셈만을 필요로 하지만, 그림 5에서  $correl_i^2 enrg_j$ 를 저장하기 위한 임시 변수들의 다이내믹 레인지가 360dB 이상으로 커지는 것을 볼 수 있다. 이는 서로 상관관계가 큰  $correl_i$ 와  $enrg_i$ 를 서로 곱하였기 때문이다. 따라서, 이 임시 변수들을 제대로 표현하기 위해서는 내부 레지스터가 60비트 이상이 필요하게 되고, 16비트만 사용하였을 때 표 1에서와 같이 이 방법의 고정 소수점 구현시의 SNR은 상당히 낮은 것을 알 수 있다. 그러므로, 계산이 용이한 대각 곱셈방법을 사용하기 위해서는 부동 소수점 연산을 사용하거나 이 임시변수들의 다이내믹 레인지를 줄이도록 알고리즘을 바꿔야 한다.

3. 제곱근 방법(square root method)

대각 곱셈 방법에서 상관도의 절대값과 에너지의 제곱근을 곱하여 비교하는 방법이다.

$$|correl_i| \sqrt{enrg_j} < |correl_j| \sqrt{enrg_i} \quad (8)$$

이 방법에서 내부 변수들의 다이내믹 레인지는 대각 곱셈 방법에 비해 반으로 줄어뜨려,  $2M-2$ 번의 곱셈과  $M$ 번의 제곱근 연산을 필요로 한다. 제곱근 연산은 TMS320C50의 경우 약 160 싸이클이 소요된다. 따라서, 표 1에서 보는 바와 같이 성능은 대각 곱셈 방법에 비하여 향상되지만 제곱근 연산을 위한 계산량의 증가가 너무 크므로 실제 구현에 적합하지 않다.

#### IV. 구간 부동 소수점 방법을 이용한 코드북 검색 과정

CELP 알고리즘에서의 코드북 검색 과정은 다이내믹 레인지가 큰 변수들이 많기 때문에 유한 단어길이 효과에 의한 성능 저하가 매우 심각하다. 또한, 코드북 원소 벡터를 합성한 후 오차의 에너지를 구하는 과정은 매우 반복적으로 수행되기 때문에 이 부분의 수행 시간은 전체 부호화기의 수행 시간에 결정적인 영향을 미치게 된다. 그러므로, 고정 소수점 성능 향상을 위하여 배정도 연산(double precision arithmetic)이나 부동 소수점 연산등을 사용하는 것은 너무 큰 계산량의 증가를 가져오기 때문에 적합하지 않다.<sup>10)</sup> 본 연구에서는 검색 과정 전에 필요한 정수 단어길이를 추정하는 구간 부동 소수점 방법을 제안하였다. 반복되는 검색 과정의 바깥에서 정수 단어길이를 추정하므로 계산의 오버헤드가 매우 적다.

##### 1. 구간 부동 소수점 방법

유한한 크기의 메모리에 고정 소수점 형태로 변수들을 저장할 때 오버플로우를 방지하기 위해서는 그 변수의 범위(range)에 대한 정보가 필요하다. 얻어진 정보가 실제 범위에 가까울수록 주어질 크기의 메모리를 효과적으로 사용하게 되므로 더 나은 성능을 얻을 수 있다. 이러한 범위에 대한 정보를 이용하여 변수가 할당된 메모리상에 가상의 이진 소수점(binary point)의 위치를 결정하며, 이는 고정 소수점 프로그램에서 변수들 간의 연산시 스케일링, 즉 이진 소수점의 위치를 일치시키기 위한 쉬프트 횟수로서 내포된다. 고정 소수점 구현시 변수( $x$ )를 나타내는 고정 소수점 데이터의 이진 소수점의 위치, 즉 정수 단어길이는 그 변수의 범위 정보( $R(x)$ )를 이용하여 다음 식으로 구할 수 있다.<sup>11)</sup>

$$W_{int}(x) = \lceil \log_2 R(x) \rceil \quad (9)$$

이때  $\lceil x \rceil$ 는  $x$ 보다 작지 않은 최소 정수부를 의미한다.  $R(x)$ 의 값은 프로그램의 개발시에 미리 결정되는데 주로 모의 실험을 통하여 결정된다. 정수 단어길이는  $R(x)$ 의 값이 변수  $x$ 의 상계(upper bound)일때 오버플로우가 발생하지 않는 것을 보장할 수 있다.

신호 처리용 응용 프로그램은 일정한 시간 간격으로 들어오는 입력에 대하여 같은 과정의 처리를 반복하게 된다. 고정 소수점 형태의 변수들은 이진 소수점의 위치가 입력에 대하여 변하지 않는, 즉 시간상으로 고정된 정적인 스케일링(static scaling)에 의한 신호의 표현이라고 할 수 있다. 그러나, 이때 변수의 다이내믹 레인지가 클 경우 이진 소수점의 위치를 고정시키려면, 최악의 경우에 발생할 수 있는 오버플로우를 방지하기 위하여 변수의 정수 단어길이를 크게 할당하여야 하고, 따라서 대부분의 입력에 대하여 그 변수는 매우 작은 갯수의 유효 비트만을 갖게 되어 심각한 성능 저하를 일으키게 된다.

구간 부동 소수점 방식은 입력 신호를 적절한 구간으로 나누고, 각 구간의 신호의 크기에 따라 변수들의 이진 소수점의 위치를 적절하게 변화시키는, 즉 동적으로 스케일링(dynamic scaling)을 하는 방법이다. 따라서, 정수 단어길이가 정해진 고정 소수점 방식에 비해 양자화 오차를 크게 줄일 수 있고, 매 구간마다 한번만 스케일링을 하면 되므로 매 연산마다 스케일링을 해야 하는 부동 소수점과 비교하면 계산량이나 하드웨어 비용면에서 매우 유리하다. 그러므로, 구간 부동 소수점 방식은 고정 소수점과 부동 소수점 방식 양쪽의 장점을 모두 살린 절충된 형태의 신호의 표현이라고 할 수 있으나, 프로그램의 실행시(run-time)에 각 구간마다 내부 변수들의 범위를 알아야 하는 어려움이 있다. 즉, 구간 부동 소수점 방식에서의 범위 정보는 변수 이름  $x$ 와 입력 신호의 구간 번호  $n$ 에 대한 함수  $R(x, n)$ 으로 표현되므로, 각  $n$ 에 대한  $R(x, n)$ 의 값을 모두 구해야 한다.  $R(x, n)$ 의 값을 각  $n$ 에서 변수  $x$ 에 저장되는 값들 중 최대값을 찾아 구할 수도 있지만 그 변수값의 갱신이 많은 경우 최대값을 찾기 위한 추가 계산량이 너무 크므로  $R(x, n)$ 을 추정하여 사용하는 것이 더 유리하다. 추정은 각  $n$ 에 대하여 오직 한번 행해지고 얻어진 범위 정보는 매  $x$ 의 참조 때

마다 동일하게 적용된다. 주어진 구간에 대하여는 고정 소수점의 경우와 마찬가지로 스케일링되기 때문에 효율적인 추정 알고리즘을 적용할 수 있다면 구간 부동 소수점 방식의 적용에 따른 추가 계산량은 매우 작다.

본 연구에서는 코드북 검색에서 다이내믹 레인지가 매우 큰 변수인 식 (4)의 상관도,  $correl_i$ 와 에너지,  $enrg_i$ 에 대하여 구간 부동 소수점 방식을 적용하였으며, 이를 위하여 프로그램의 실행중에  $R(correl, n)$ 와  $R(enrg, n)$ 을 추정하기 위한 효율적인 알고리즘을 제시하였다. 추정은 부분 프레임에서 오직 한번 행해지므로 추가 계산량은 0.1 MIPS 정도로 매우 작다. 그림 6에서 보는 바와 같이 제한된 범위 추정 알고리즘을 사용하여 구간 부동 소수점 방식을 적용시켰을 때 식 (7)에서의 임시 변수들의 다이내믹 레인지가 60dB 정도로 매우 감소되었음을 알 수 있다. 계산이 용이한 대각 곱셈 방법을 사용하였는데도 다이내믹 레인지는 나눗셈 방법의 절반 정도로 줄어들었다.

2. 정수 단어길이 추정 알고리즘

각 부분 프레임에서 상관도와 에너지의 범위의 한계를 목표 신호  $t_0$ , 오차 가중화 합성 필터의 충격 응답 벡터  $h$ 와 코드북 원소 벡터  $x_i$ 들의  $L_2$  norm을 이용하여 유도하였다. 식 (4)에서 상관도는 앞서 정의된 두  $N$  차원 벡터  $y_i$ 와  $t_0$ 의 내적이기 때문에 다음 식을 만족한다.

$$|correl_i| \leq \|t_0\| \|y_i\| \tag{10}$$

$\|\cdot\|$ 는 벡터의  $L_2$  norm을 의미한다. 이제  $\hat{y}_i$ 를 앞서 정의된  $N$  차원 벡터  $x_i$ 와  $N$  차원 벡터  $h$ 의 컨볼루션이라고 하자. 그러면  $\hat{y}_i$ 는  $N + L - 1$ 의 차원을 갖게 되고 다음 식이 성립하게 된다.

$$y_{i,k} = \hat{y}_{i,k} \text{ for } k = 1, 2, \dots, N \tag{11}$$

마찬가지로  $\hat{h}$ 와  $\hat{x}_i$ 를 각각  $h$ 와  $x_i$ 에 0을 추가함으로써(zero padding)  $N + L - 1$  차원으로 확장된 벡터라고 하자. 그러면  $\hat{y}_i, \hat{h}, \hat{x}_i$ 의  $N + L - 1$  point DFT (Discrete Fourier Transform)인  $\hat{Y}_i, \hat{H}, \hat{X}_i$ 에 대하여 다음이 성립한다.

$$\hat{Y}_{i,k} = \hat{H}_i \hat{X}_{i,k} \text{ for } k = 1, 2, \dots, N + L - 1 \tag{12}$$

또한  $\|\hat{Y}_i\| \leq \|\hat{H}\| \|\hat{X}_i\|$ 이 성립하게 되며 Parseval의 정리<sup>[11]</sup>에 의하여

$$\|y_i\|^2 \leq \|\hat{y}_i\|^2 \leq N \|\hat{h}\|^2 \|\hat{x}_i\|^2 = N \|h\|^2 \|x_i\|^2 \tag{13}$$

이 성립한다. 식 (10)과 (13)을 결합하면

$$|correl_i| \leq \sqrt{N} \|t_0\| \|h\| \|x_i\| \tag{14}$$

이는 상관도의 절대값의 상계를 나타내는 것이며, 이를 바탕으로 상관도의 정수 단어길이를 구하면

$$W_{IWL}(correl, n) = \lceil \log_2(\sqrt{N} \|t_0(n)\| \|h(n)\| \max\{\|x_i\|\}) \rceil \tag{15}$$

이때,  $n$ 은 프로그램의 반복에 따른 번호(index)이고,  $\lceil x \rceil$ 는  $x$ 보다 작지 않은 최소 정수를 의미한다. 그러므로 상관도의 정수 단어길이를 구하기 위해서는 매 부분 프레임마다 목표 벡터와 오차 가중 합성 필터의  $L_2$  norm이 계산되어야 한다. 이는 MAC(Multiply and Accumulate) 연산을 이용하여 쉽게 계산될 수 있으며, 반복되는 코드북 검색 과정의 바깥에서 행해지는 것이므로 이에 따른 계산량의 증가는 매우 작다.

불규칙 코드북에서는 모든 여기 벡터(excitation vector)  $x_i$ 들이 고정되어 있으므로  $x_i$ 들의 최대  $L_2$  norm이 미

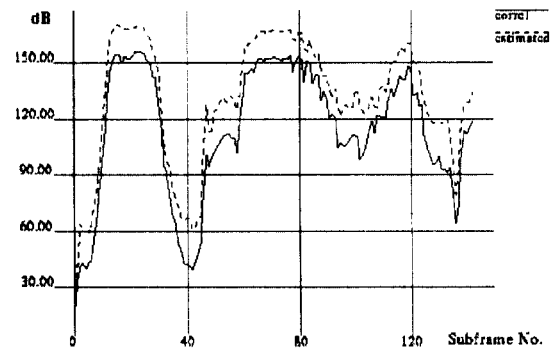


그림 6. 추정된 상관도의 상계  
Fig 6. Estimated upper bound of correlation

리 계산될 수 있다. 그러나 피치 검색에서는 매 부분 프레임마다 여기 벡터들이 변하기 때문에  $\max \|x_i\|$ 를 계산하는 것이 불가능하다. 그러므로 계산을 용이하게 하기 위하여 이전 부분 프레임의 목표 벡터의  $L_2$  norm이 식 (10)에서의  $\|y_i\|$  대신 사용된다. 그림 6은 추정된 정수 단어길이에 한 비트의 보호 비트(guard bit)를 추가하여 고정 소수점 모의 실험을 한 결과로서 피치 검색의 경우에도 제안된 방식에 의한 추정값이 항상 상관도의 상계임을 보여준다.

TMS320C50에서  $\log_2$ 의 가수(mantissa)와 지수(exponent)의 계산은 누산기의 정규화(normalization)를 행하는 명령어를 이용하여 쉽게 구현될 수 있다.<sup>12)</sup>

## V. TMS320C50을 이용한 구현

### 1. 구간 부동 소수점 방식을 이용한 LPC 분석 및 LSP 변환

부호화기에서의 코드북 검색이나 복호화기에서의 음성 합성 등 CELP 알고리즘에서 주된 부분들은 모두 LPC 분석 과정에서 얻어지는 합성 필터를 바탕으로 수행된다. 또한 필터링은 direct form 필터로 구현되기 때문에<sup>13)</sup> 필터링시의 안정성(stability)을 위해서도 필터의 계수는 매우 높은 정밀도(precision)를 필요로 한다. 그런데 LPC 분석 과정에서 Levinson-Durbin 알고리즘을 구현할 때 필요한 자기상관도(autocorrelation)는 다이내믹 레인지가 매우 크기 때문에 정적으로 스케일링할 경우 유한 단어길이 효과에 의한 성능 저하가 크다. 이를 극복하기 위하여 구간 부동 소수점 방식을 적용하였다. 그런데 자기상관도 함수(autocorrelation function)  $R(n)$ 은  $n=0$ 일때 최대값을 갖으므로 항상 최소 상계(least upper bound)를 알 수 있다. 그러므로 먼저  $R(0)$ 를 계산한 후 누산기를 정규화한 후 이를 바탕으로 다른 자기상관계수 함수값들을 스케일링하여 이용함으로써 매우 높은 정확도의 필터 계수를 얻을 수 있다.

LPC에서 LSP로의 변환은 LP 필터로부터 만들어지는 두 보조 다항식(auxiliary polynomial)의 근을 찾는 것인데, LSP 양자화기의 각 경계점(boundary point)에서의 보조 다항식의 값을 평가하여 부호의 변화를 찾음으로써 양자화와 동시에 근을 찾는다. 보조 다항식의 값은 Chebyshev 다항식을 이용하여 계산하였다.<sup>14)</sup>

이때 보조 다항식의 계수들은 배정도로 저장하였고, 다항식의 값을 평가하기 위한 중간 변수들은 덧셈 연산에 사용될 때는 배정도로 저장된 값을 사용하였으며 곱셈 연산을 위하여는 정규화된 값을 사용하였다.

### 2. Pre-shifting 방법

코드북 검색에서 여기 신호들을 합성하고 상관도와 에너지를 구해서 이들을 결정하는 과정은 TMS320C50으로 구현하였을때 전체 프로그램 크기의 약 20%인 반면 수행 시간은 90%를 차지한다. 그러므로, 전체 부호화기의 계산 요구량을 줄이기 위하여 이 과정에 해당하는 코드북을 최적화하는 것이 매우 중요하다. FS-CELP에서는 불규칙 코드북과 적응형 코드북 모두 중첩된 구조를 가지고 있기 때문에 여기 신호들과 오차 가중화 합성 필터와의 컨볼루션을 여기 신호들이 코드북에 저장된 순서대로 행함으로써 계산을 간단하게 할 수 있다. 즉,  $i+1$ 번째 여기 신호  $x_{i+1}$ 과 오차 가중화 합성 필터와의 컨볼루션의 결과  $y_{i+1}$ 은  $x_i$ 와의 컨볼루션의 결과  $y_i$ 를 한 샘플 쉬프트하고  $x_{i+1}$ 이  $y_i$ 에 대하여 달라진 샘플에 해당하는 부분만 교정해 주면 된다(end-point correction).<sup>15)</sup> TMS320C50에서는 쉬프트 연산을 위해서 1 사이클이 필요하지만 곱셈과 쉬프트 연산이 동시에 실행된다면 두 연산을 1 사이클만에 실행할 수 있다. 그러므로  $i$ 번째에서 상관도를 계산할때  $y_i$ 와  $t_0$ 를 곱하면서  $i+1$ 번째 여기 신호와의 컨볼루션을 위하여 미리  $y_i$ 를 쉬프트시킴으로써(pre-shifting) 전체 수행 시간을 줄일 수 있다. 이 과정을 매 부분 프레임마다 코드북의 크기만큼 반복되므로 pre-shifting 방법을 이용할 경우 계산량의 감소는 매우 크다. 여기 신호가 60 차원이고 불규칙 코드북의 크기가 256일때 약 3MIPS 정도의 계산량의 감소를 얻을 수 있다.

### 4. TMS320C50 구현 결과

구현된 음성 부호화기의 메모리 요구량과 계산량이 표 2에 나타나 있으며, 성능은 표 3에 나타나 있다. 표 3은 구간 부동 소수점 방법을 사용함으로써 고정 소수점 구현에 따른 성능 저하가 매우 줄어들었음을 나타낸다. 표 4에서 표 1의 대각 곱셈 방법과 구현된 음성 부호화기를 직접 비교하였다. 전체 계산량의 95% 정도인 25.6MIPS의 계산량을 필요로 하는 코드



복 검색에 구간 부동 소수점 방법을 적용하였는데도, 실행 시간 정수 단어길이 추정에 따른 계산량 증가는 0.1MIPS로 매우 작은 것을 알 수 있다.

표 2. 메모리와 계산 요구량

Table 2. Computation and memory requirement

	프로그램 (words)	데이터 (words)	계산량 (MIPS)
LPC & LSP	578	596	0.8
코드북 검색	1080	2130	25.6
복호기	155	611	0.8
기타	380	1076	0.6
합계	2193	4413	27.8
테이블(외부)	3149		

표 3. 구현된 부호화기의 성능

Table 3. Performance of the implemented coder

	구현된 부호화기	부동 소수점 모의 실험
남성1	10.97 dB	11.27 dB
남성2	11.16 dB	11.08 dB
여성1	10.88 dB	10.96 dB
여성2	11.14 dB	11.37 dB
평균	11.04 dB	11.17 dB

표 4. 구간 부동 소수점 방법의 적용 결과

Table 4. Result of the block floating-point method

	계산량 (MIPS)	성능 (SNR)
대각 곱셈 방법	27.7	6.88 dB
구간 부동 소수점을 이용한 대각 곱셈 방법	27.8	11.04 dB

구현된 음성 부호화기는 에뮬레이션 보드를 사용하여 실시간으로 검증되었는데, 전이중 방식(full-duplex)의 검증을 위해 부호화기의 출력을 그대로 복호화기의 입력으로 사용하였다. A/D 변환기를 통하여 양자화된 음성 샘플은 TMS320C50의 직렬 포트를 통하여 입력되고, 직렬 포트 인터럽트를 통하여 부호화기의 입력 버퍼에 저장된다. 복호화기의 출력도 직렬 포트

의 인터럽트를 통하여 D/A 변환기로 전달된다.

## VI. 결 론

CELP 알고리즘의 고정 소수점 구현시 발생하는 성능 저하를 막기 위하여 코드북 검색을 위한 세가지 구현 방법의 고정 소수점 성능을 분석하였다. 그리고, 효과적인 실행 시간 정수 단어길이 추정 알고리즘을 제안하여 구간 부동 소수점 구현 방법을 유도하였다. 이를 바탕으로 4.8Kbps의 낮은 전송률에서도 우수한 음질을 갖는 FS-1016 CELP 음성 부호화기를 Autoscaler를 이용하여 구현한 후 실시간으로 검증하였다. 구현된 부호화기는 256개의 여기 신호를 갖는 불규칙 코드북을 사용하였을 때 28MIPS 정도의 계산 속도를 필요로 한다. 개발된 유성 부호화기는 이동통신이나 자동 응답 전화기 등 여러 용도로의 사용이 기대된다.

## 參 考 文 獻

1. J. P. Campbell, Jr., V. C. Welch and T. E. Tremain, "An expandable error-protected 4800 bps CELP coder," in *Proc. Int. Conf. Acoustics, Speech, Signal Processing*, pp. 735-737, 1989.
2. U. S. Federal Standard 1016, "Analog to Digital Conversion of Radio Voice by 4,800 Bit/Second Code Excited Linear Prediction(CELP)," Feb. 1991.
3. Seehyun Kim and Wonyong Sung, "A floating-point to fixed-point assembly program translator for the TMS 320C25," *IEEE Trans. Circuits and Systems*, vol. 41, No. 11, pp. 730-739, 1994.
4. Seehyun Kim, Ki-Il Kum, and Wonyong Sung, "Fixed-point optimization utility for C and C++ based digital signal processing programs," in *Proc. IEEE Workshop on VLSI Signal Processing*, pp. 197-206, 1995.
5. Seehyun Kim and Wonyong Sung, "Fixed-point simulation utility for C and C++ based digital signal processing programs," in *Proc. Twenty-eighth Annual Asilomar Conference on Signals, Systems, and Computer*, pp. 162-166, 1995.

6. B. S. Atal, "Predictive coding of speech at low bit rates," *IEEE Trans. Communications*, vol. 30, No. 4, pp. 600-614, 1982.
7. P. Kroon and K. Swaminathan, "A high-quality multirate real-time CELP coder," *IEEE J. Select. Areas Commun.*, vol. 10, No. 5, pp. 850-857, 1992.
8. W. B. Kleijn, D. J. Krasinski, and R. H. Ketchum, "Fast methods for the CELP speech coding algorithm," *IEEE Trans. Acoustics, Speech, Signal Processing*, vol. 38, No. 8, pp. 1330-1342, 1990.
9. J. H. Chen, Y. C. Lin, and R. V. Cox, "A Fixed-point 16 KB/S LD-CELP Algorithm," in *Proc. Int. Conf. Acoustics, Speech, Signal Processing*, pp. 21-24, 1991.
10. W. Sung, J. Sohn, J. Y. Kang, and S. Kim, "Fixed-point Implementation of the FS-CELP Vocoder Using the Autoscaler for the TMS320C50," in *Proc. Int. Conf. Signal Processing Applications and Technology*, pp. 1883-1887, 1995.
11. A. V. Oppenheim and R. W. Schaffer, *Discrete-Time Signal Processing*, Prentice-Hall, Englewood Cliffs, NJ, 1989.
12. *TMS320C50 User's Guide*, Texas Instruments Inc., 1993.
13. P. Kabal and R. P. Ramachandran, "The Computation of Line Spectral Frequencies Using Chebyshev Polynomials," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. 34, No. 6, pp. 1419-1425, 1986.



孫 鍾 瑞(Jongseo Sohn) 정회원  
 1970年 12月 24日生.  
 1994年 2月: 서울대 제어계측공학과 졸업(공학사).  
 1996年 2月: 서울대 대학원 제어계측공학과 졸업(공학석사).  
 ※주관심 분야: 음성 신호처리 및 신호처리를 위한 CAD 등임.



姜 志 浪(Jiyang Kang) 정회원  
 1971年 12月 30日生.  
 1994年 2月: 서울대 제어계측공학과 졸업(공학사).  
 1996年 2月: 서울대 대학원 제어계측공학과 졸업(공학석사).  
 ※주관심 분야: 신호처리를 위한 CAD 및 컴퓨터 그래픽 등임.



金 時 鉉(Seehyun Kim) 정회원  
 1968年 2月 27日生.  
 1990年 2月: 서울대 제어계측공학과 졸업(공학사).  
 1992年 2月: 서울대 대학원 제어계측공학과 졸업(공학석사).  
 1996年 2月: 서울대 대학원 제어계측공학과 졸업(공학박사).  
 ※주관심 분야: VLSI 신호처리와 신호처리를 위한 CAD 등임.



成 元 鎔(Wonyong Sung) 정회원  
 1995年 4月 14日生.  
 1978年 2月: 서울대 전자공학과 졸업(공학사).  
 1980年 2月: 한국과학기술원 전기 및 전자공학과 졸업(공학석사).  
 1987年 7월: 미국 University of California, Santa Barbara 전기 및 컴퓨터공학과 졸업(공학박사).  
 1980年 2월~1983年 7월: (주) 금성사 중앙연구소 연구원.  
 1989年 2월~현재: 서울대 전기공학부 부교수.  
 ※주관심 분야: DSP용 CAD 소프트웨어 개발, VLSI를 이용한 고속 신호처리, 프로그래밍 및 DSP의 응용 등임.