2000년도 한국음향학회 학술발표대회 논문집 제 19 권 제 2(s) 호

# G.729 음성 압축기의 계산량 감소

최윤창, 박호종
광운대학교 전자공학과

# Complexity Reduction of G.729 Vocoder

Younchang Choi, Hochong Park
Dept. of Electronics Engineering
Kwangwoon University
e-mail : ycchoi@explore.kwangwoon.ac.kr

## 요약

ACELP는 우수한 음질을 제공하지만 최적의 코드 벡터를 찾기 위한 계산량이 상당히 많은 단점이 있고, 이로 인하여 모든 시스템과 단말기에서는 고성능 DSP칩을 사용하여 동작시킨다.

본 논문에서는 고속 ACELP 코드북 검색 방법인 펄스 교환 검색 방법을 G.729 음성 압축기에 적용시켜 G.729 음성 압축기의 계산량을 감소시키는 방법을 연구하였다. 적용된 방법은 두 단계 과정을 가지며, 첫 단계에서는 완전 순차적 검색 방법을 통하여 매우 빠르게 대략적인 코드 벡터를 찾는다. 두 번째 단계에서는 앞에서 선택된 코드 벡터의 각 펄스의 중요도를 계산하여 역할이 적은 펄스를 제거하고 새로운 펄스로 교환하는 펄스 교환 과정을 통하여 코드 벡터의 성능을 향상시킨다. 적용된 방법은 표준에서 사용하는 코드북 검색 방법보다 적은 계산량을 가진다. 적용된 방법의 성능은 표준보다 0.3~0.5dB 정도의 SNRseg 감소를 보이지만 Fast Algorithm인 G.729A보다는 우수한 음질의 코드 벡터를 찾으며, 다양한 음성신호를 이용한 모의 실험을 통하여 이 결과를 확인하였다.

## 1. 서 론

디지털 음성통신은 아날로그 음성신호를 디지털 신호(Bit)로 변환하여 전송하는 방법을 이용하며, 디지털 통신 시스템의 용량은 단위 시간에 전달할 수 있는 Bit의 양으로 표시된다. 따라서 원하는 음질을 가지면서 주어진 음성 신호를 보다 적은 양의 Bit로 표현하기 위한 음성 압축기가 개발되었고, 이는 디지털 통신 시스템의 품질과 용량을 결정하는 매우 중요한 부분이다.

현재 IMT-2000과 인터넷망을 이용한 음성신호 전송 및 저장을 위한 표준 음성 압축기들의 코드북 구조는 ACELP(Algebraic Code Excited Linear Prediction) 구조를 가지는 음성 압축기를 채택하고 있다. 이러한 대표적인 음성 압축기들로는 AMR[1], G.729[2], G.723.1[3] 등이 있으며, 이들 모두 ACELP 구조를 가지는 코드북으로 구성되어 있다.

G.729 음성 압축기는 1996년 ITU-T에서 표준으로 채택하였으며, 8kbps의 CS-ACELP(Conjugate Structure-Algebraic Code Excited Linear Prediction) 구조를 가지며, 주로 무선 통신 시스템과 멀티미디어 시스템을 주 응용 분야로 하고 있다. 또한 거의 Toll-Quality 음질의 성능을 가지며, 특히 코드북의 구조는 고품질 음성 압축기에 가장 널리 사용되고 있는 ACELP 구조를 가진다. ACELP는 우수한 음질을 제공하지만 최적의 코드 벡터를 찾기 위한 계산량이 상당히 많은 단점이 있고, 이로 인하여 모든 시스템과 단말기에서 고성능 DSP Chip을 사용하여 동작시킨다.

본 논문에서는 G.729에서 코드북을 검색하는 과정에 성능의 저하 없이 계산량을 줄이기 위하여 펄스 교환 검색 방법[4]을 적용했다. 펄스 교환 검색 방법을 적용하여 계산량을 줄이기 위한 목적은 첫째 하나의 DSP Chip으로 여러 통화를 동시에 처리 할 수 있게 하여 시스템 용량을 증가시키고, 둘째 동작 할 때의 소비전력을 줄여 단말기의 사용 시간을 늘리고, 마지막으로 PC에서 별도의 하드웨어 없이 구현을 가능하게 하여 많은 PC응용 분야에 이용되도록 한다.

본 논문의 구성은 2장에서 G.729의 코드북 구조와 표준안에서 제안하는 코드북 검색 방법을 설명하고, 3장에서 계산량 감소를 위하여 적용된 방법을 설명하고, 4장에서 적용된 방법의 성능을 보여 준다. 그리고 5장에

서 결론을 맺었다.

## 2. G.729 코드북

### 2.1 코드북 구조

코드북은 복원기(Decoder)에서 음성신호를 복원할 때 필요한 최초의 여기 신호를 표현하는 코드 벡터들의 집합이며, 우수한 음질의 음성신호를 합성하기 위하여 다양한 신호 형태를 표현할 수 있어야 한다. 코드북의 크기와 구조는 최적의 코드 벡터를 찾기 위한 코드북 검색의 계산량 및 성능과 밀접한 관계가 있다.

ACELP는 임의의 위치에 펄스를 위치시켜 코드 벡터를 만드는 것으로서, Analysis-by-Synthesis 방법에 기초한 ACELP 코드북의 기본 검색 방법은 다음과 같다. 모든 가능한 코드 벡터들에 대하여 합성된 신호와 목표신호 사이의 오차신호의 에너지를 구하고, 이 값을 최소로 하는 코드 벡터를 최종 선택한다. 이 때 오차신호의 에너지를 최소로 하는 것은 식 (1)값을 최대로 하는 것과 동일하다.

$$Tk = \frac{C_k^2}{E_k} = \frac{d^t c^k}{c_k^t \Phi c_k} \qquad (1)$$

여기서 $C_k$ 는 첨자 $k$ 에 해당하는 코드 벡터를 가리키고, $d$ 는 목표신호와 충격응답과의 상관 벡터, $\Phi$ 는 충격응답의 상관행렬이다. 즉 $d$ 와 $\Phi$는 식 (2) 및 (3)와 같이 각각 나타내어 진다.

$$d(n) = \sum_{i=n}^{39} x'(i)h(i-n) \quad n=0,....,39 \quad (2)$$

$$\Phi(i,j) = \sum_{n=j}^{39} h(n-i)h(n-j)$$
$$i = 0,....,39 \quad j = i,....,39 \quad (3)$$

따라서 모든 $k$ 에 대하여 $T_k$ 값을 계산하여야 하며, 이 과정이 코드북 검색 과정 중에서 대부분의 계산량을 차지하는 부분이다.

ACELP에서는 가능한 코드 벡터의 수가 매우 많으며, 모든 가능한 경우에 대한 검색을 위하여 매우 많은 계산량이 필요하고 따라서 실제의 응용에서는 전체 검색을 통하여 최적의 코드 벡터를 찾을 수 없고 많은 제한을 두어 계산량을 줄이도록 한다.

G.729에서 사용하는 ACELP구조는 다음과 같다. 검색의 단위는 부프레임(Subframe)이고 각 부프레임의 길이는 40 샘플이다. 각 부프레임에 총 4개의 펄스를 할당하며, 따라서 각 부프레임별로 정하여진 코드벡터는 항상 4개의 펄스로 구성된 신호가 된다. 우선 각 부프레임에 대하여 $d(n)$은 절대값인 $d'(n) = |d(n)|$ 과 그 부호로 분리되며, 이때 부호값은 표 1에 보여진 40개의 가능한 펄스 위치에 대한 부호로써 미리 정해진다. 그리고 행렬 $\Phi$는 식 (4),(5) 와 같이 수정된다.

$$\Phi'(i,j) = sign[d(i)]sign[d(j)]\Phi(i,j)$$
$$i = 0,....,39 \quad j = i+1,....,39 \quad (4)$$

$$\Phi'(i,j) = 0.5\Phi'(i,j) \qquad i = 0,....,39 \quad (5)$$

이에 따라 식(1)의 분자인 $C$와 분모인 $E_k$는 식(6)과 식(7)으로 나타내어진다.

$$C = d'(m_0)+d'(m_1)+d'(m_2)+d'(m_3) \qquad (6)$$

$$\begin{aligned} E/2 = {} & \Phi'(m_0,m_0) \\ & + \Phi'(m_1,m_1)+\Phi'(m_0,m_1) \\ & + \Phi'(m_2,m_2)+\Phi'(m_0,m_2)+\Phi'(m_1,m_2) \\ & + \Phi'(m_3,m_3)+\Phi'(m_0,m_3)+\Phi'(m_1,m_3)+\Phi'(m_2,m_3) \end{aligned} \qquad (7)$$

| 트랙 | 펄스 | 부호 | 위치 |
|---|---|---|---|
| T0 | $i_0$ | $s_0$:±1 | $m_0$ : 0,5,10,15,20,25,30,35 |
| T1 | $i_1$ | $s_1$:±1 | $m_1$ : 1,6,11,16,21,26,31,36 |
| T2 | $i_2$ | $s_2$:±1 | $m_2$ : 2,7,12,17,22,27,32,37 |
| T3 | $i_3$ | $s_3$:±1 | $m_3$ : 3,8,13,18,23,28,33,38<br>4,9,14,19,24,29,34,39 |

표 1. G.729 코드북의 트랙 구조

### 2.2 코드북 검색 방법

2.1에서 설명한 G.729의 코드북 구조는 가능한 코드 벡터를 모두 검색함으로 많은 계산량을 필요로 한다. 즉, 한 트랙에서 한 개의 펄스를 선택하는 방법은 8 × 1 = 8 가지이다. 따라서 4개의 펄스를 선택하는 방법은 8 × 8 × 8 × 16 = 8192 가지가 된다.

너무 많은 계산량을 가지므로 G.729 표준안에서는 계산량을 줄이기 위하여 다음과 같은 방법을 이용한다. 마지막 트랙 즉, 표 1에서 T3 는 16개의 가능한 펄스 위치들로 구성되어 있기 때문에 앞의 세 트랙(T0, T1, T2)에 대하여 마지막 트랙(T3)에서 1개의 펄스를 선택하는 방법은 $N = 2^9 = 512$ 이다. 마지막 트랙에서의 펄스 검색을 위한 계산량을 줄이기 위하여 앞의 세 트랙의 최대 상관도 값( $C_{max}$ ) 과 평균 상관도 값( $C_{av}$ )

을 이용하여 이것에 대한 문턱 값( $C$th )을 미리 계산하여 앞의 세 트랙(T0, T1, T2)의 $C$값이 문턱 값을 넘을 경우에만 마지막 트랙을 검색하도록 한다. 앞의 세 트랙에 대한 최대 상관도 값은 식 (8)에 의하여 구하여진다. 식 (8)에서 $\max[d'(t_i)]$ 는 표1에서 앞의 세 트랙에 대한 $d'(n)$의 최대 값이다

$$C_{\max} = \max[d'(t_0)]+\max[d'(t_1)]+\max[d'(t_2)] \quad (8)$$

앞의 세 트랙에 대한 평균 상관도 값과 문턱 값은 식 (9)과 식(10)에 의하여 구하여진다.

$$C_{av} = \frac{1}{8}\left(\sum_{n=0}^{7} d'(5n) + \sum_{n=0}^{7} d'(5n+1) + \sum_{n=0}^{7} d'(5n+2)\right) \quad (9)$$

$$C_{th} = C_{av} + (C_{\max} - C_{av})\alpha_t \quad (10)$$

문턱 값은 코드북 검색 전에 미리 계산된다. (10)식에서 $\alpha_t$값은 마지막 트랙을 탐색할 수 있는 후보들의 수를 조절하는데, N=512개의 모든 후보가 T3 트랙을 탐색하는 경우의 성능과 같은 성능을 내기 위해서 $\alpha_t$=0.4를 쓰며, 이 값은 후보의 수를 평균 N=60으로 만들며, 단지 5%만이 N=90을 넘기게 한다. 또한 트랙 T3를 탐색하는 후보의 수를 $N_{max}$개로 제한한다. 첫 번째 부 프레임의 최대 후보 수를 $N_1$=105개로, 두 번째 부 프레임의 최대 후보 수는 180-$N_1$개로 제한한다. 이때 최악의 경우 부 프레임 당 평균 90개의 후보가 생기며, 이 경우 총 8192개의 조합 중 90×16=1440번의 탐색이 부 프레임 당 이루어진다.

## 3. 적용된 코드북 검색 방법

적용된 펄스 교환 검색 방법의 기본 개념은 두 단계 펄스 검색에 기초한다. 첫 단계에서는 매우 빠른 검색 방법을 통하여 대략적인 코드 벡터를 찾고, 두 번째 단계에서는 "펄스 교환 과정"을 통하여 펄스들을 재조정하여 최종적으로 최적에 가까운 코드 벡터를 찾도록 한다.

첫 단계에서는 각 트랙에서 한 개의 펄스를 독립적으로 찾는 "완전 순차적" 검색 방법을 이용하여 4개의 펄스를 찾는다. 즉 T0, T1, T2, T3에서 트랙 당 하나의 펄스를 선택하면 4개의 펄스를 찾게 된다. 완전 순차적 검색 방법의 계산량은 8 + 8 + 8 + 16 = 40 가 되어 매우 적지만 성능이 매우 저하되므로 이 결과를 그대로 최종코드 벡터로 이용 할 수 없고 두 번째 단계의 추가 처리가 필요하다.

두 번째 단계에서는 완전 순차적 방법으로 선택한 4개의 펄스에 대하여 펄스 교환 과정을 수행한다. 이 과정은 이미 순차적으로 선택된 4개의 펄스 중에서 한 개의 펄스를 버리고 새로운 펄스를 찾는 과정이다. 먼저 4개의 펄스에 대하여 각 펄스의 중요도를 계산한다. 즉 각 펄스를 제외하고 3개의 펄스만을 가질 때의 성능을 구하고, 이중에서 가장 성능이 우수한 경우에 해당하는 3개의 펄스로 구성된 코드 벡터를 만든 후, 한 개의 펄스를 새로 찾아 다시 4개의 펄스로 구성되는 코드 벡터를 최종적으로 구한다. 이 과정은 중요도가 적은 펄스를 제거하고 새로운 펄스로 교환하는 의미를 가지며, 새롭게 정의된 코드벡터는 교환전의 코드 벡터보다 우수하거나 같은 성능을 가진다. 이와 같은 펄스 교환 과정이 반복되면 계속적으로 코드 벡터의 성능이 향상된다.

| 펄스 조합 | 트랙의 조합 |
|---|---|
| 1 | T0-T1-T2 |
| 2 | T0-T1-T3 |
| 3 | T0-T2-T3 |
| 4 | T1-T2-T3 |

표 2. 중요도 계산을 위한 펄스 조합

적용된 코드북 검색 방법의 계산량은 다음과 같다. 완전 순차적 검색에 필요한 검색 수는 40이고, 한번의 펄스 교환에 대하여 펄스의 중요도를 계산하는 과정이 4번 필요하고 새로운 펄스를 하나 찾기 위하여 한 트랙을 다시 검색하므로 T0, T1, T2 트랙에 대하여는 8번, 그리고 T3트랙에 대하여 16번의 검색이 필요하다. 여기서 검색을 위한 계산은 $T_k$ 값을 계산하는 것이고, 각 펄스의 중요도를 계산하는 것도 역시 하나의 펄스를 제외한 후 3개의 펄스에 대한 $T_k$ 값을 계산하는 것이다. 따라서 중요도 계산을 위한 계산량은 한번의 펄스 검색을 위한 계산량과 거의 동일하다. 물론, 펄스 수에 따라 $T_k$를 계산하기 위한 계산량에는 약간의 차이가 있으나 이에 따른 차이는 무시한다. 따라서 제안된 검색을 위한 계산량은 최대 40 + (4 + 16) × (펄스 교환 횟수)이며 한번의 펄스 교환을 할 경우 60이 된다. 이는 표준이 제안하는 방식의 1440보다 매우 적은 계산량을 가진다.

## 4. 적용된 펄스 교환 검색 방법의 성능

적용된 검색 방법의 성능을 측정하기 위하여 여러 음성 데이터를 이용하여 여러 검색 방법에 의한 성능과 비교하였다. 한글과 영어 음성을 모두 사용하였고 여성

과 남성을 분리하여 성능을 측정하였으며, Fast Algorithm인 G.729A[5]와 성능을 비교하였다.

본 논문에서 적용된 펄스 교환 검색 방법의 계산량과 검색 성능은 표준안이 제안하는 검색 방법, 즉 앞의 세 개의 트랙 T0, T1, T2 의 $C$ 값이 문턱 값을 넘을 경우 마지막 트랙을 검색하는 방법을 기준으로 하여 비교한다. 성능은 객관적 성능 비교에 많이 사용되는 Segmental SNR(SNRseg)를 이용하며 단위는 dB이다. SNRseg는 부프레임 단위로 SNR을 계산하고 이것의 전체 평균을 dB 단위로 계산하여 얻는다. SNRseg계산은 압축기의 입력신호의 DC 성분을 제거한 신호를 기준으로 합성된 신호와 비교하여 계산한다.

성능을 비교하기 위하여 총 5가지의 방법을 실시하였으며, 표 3에 검색 방법과 계산량이 정리되어 있으며, 각 검색 방법의 설명은 다음과 같다.

- G.729표준 : 표준안에 명시된 방법으로 4개의 트랙에서 펄스의 위치를 모두 검색
- G.729A표준 : 표준안에 명시된 방법으로 2개의 트랙씩 묶어서 펄스의 위치를 검색
- 적용0 : 각 트랙별로 완전 순차적 검색을 하며, 펄스 교환 없이 펄스의 위치를 검색
- 적용1 : 각 트랙별 완전 순차적 검색을 하며, 펄스 교환을 한번 실행하여 펄스의 위치를 검색
- 적용2 : 각 트랙별 완전 순차적 검색을 하며, 펄스 교환을 두번 실행하여 펄스의 위치를 검색

| 검색 방법 | 계산량(비율) |
|---|---|
| G.729표준 | 1140(100%) |
| G.729A표준 | 320(28.1%) |
| 적용0 | 40(3.50%) |
| 적용1 | 60(5.26%) |
| 적용2 | 80(7.0%) |

표 3. 검색 방법별 계산량

SNRseg(dB)로 측정한 각 검색 방법별 성능은 표 4에 정리되어 있다.

| 언어 | | 영어 | | 한글 | |
|---|---|---|---|---|---|
| 성별 | | 여성 | 남성 | 여성 | 남성 |
| 검색 방법 | G.729표준 | 12.74 | 11.65 | 11.57 | 12.73 |
| | G.729A표준 | 12.4 | 11.57 | 11.1 | 12.11 |
| | 적용0 | 12.28 | 11.24 | 11.08 | 12.18 |
| | 적용1 | 12.33 | 11.35 | 11.24 | 12.23 |
| | 적용2 | 12.39 | 11.37 | 11.26 | 12.42 |

표 4. 검색 방법별 코드북 검색 성능

## 5. 결론

본 논문에서는 G.729의 코드북 검색 과정의 계산량을 감소시키기 위하여 펄스 교환 검색 방법을 적용하였다. 두 단계 검색 방법을 사용하여 첫 단계에서는 각 트랙별로 하나의 펄스를 완전 순차적 방법으로 검색하여 4개의 펄스를 대략 적으로 찾고, 두 번째 단계에서는 펄스 교환 과정을 통하여 보다 최적에 가까운 코드 벡터를 찾는다. 적용된 검색 방법의 계산량은 표준에서 제안하고 있는 방법에 비하여 매우 적지만 표준 방법보다는 SNRseg 약간 떨어지는 것을 볼 수 있다. 그러나 Fast Algorithm인 G.729A 보다는 우수한 것을 볼 수 있었고 주관적 청취 음질에 있어서는 거의 차이가 없음을 확인하였다.

## 참 고 문 헌


[1] ETSI EN 301 704, "European digital cellular telecommunication system(Phase 2+); Adaptive Multi-Rate(AMR) speech transcoding", Version 7.2.1 Release 1998

[2] ITU-T Rec. G.729, "Coding of speech at 8kbit/s CS-ACELP Speech Coder", 1996

[3] ITU-T Rec. G.723.1, "Dual Rate Speech Coder for Multimedia Communication Transmitting at 5.3 and 6.3kbit/s ", 1996

[4] Hochong Park, "Efficient codebook search method for EVRC speech codec", IEEE Signal Processing Letters, vol.1, no1, pp.2-3, January, 2000.

[5] R. Salami et al.,"ITU-T G.729 Annex A: Reduced Complexity 8kb/s CS-ACELP Codec for Digital Simultaneous Voice and Data", IEEE Commun. Mag. pp.56-63, Sept. 1997