

# 효율적인 다채널 구현을 위한 ITU-T G.723.1 음성 부호화기 고속 알고리즘 제안

정성교\*, 박영철\*\*, 윤성완\*, 차일환\*, 윤대회\*

\* 연세대학교 전자·컴퓨터공학과 음향·음성 및 신호처리연구실

\*\* 연세대학교 신호처리연구센터

## A Proposal of Fast Algorithms of ITU-T G.723.1 for Efficient Multichannel Implementation

Sung-Kyo Jung\*, Young-Cheol Park\*\*, Sung-Wan Youn\*, Il-Chan Cha\*,  
and Dae-Hee Youn\*

\* ASSP Lab. Department of Electronic & Computer Engineering, Yonsei University

\*\* Center for Signal Processing Research, Yonsei University

e-mail : skjung@lethe.yonsei.ac.kr

### 요약문

최근 들어, 인터넷의 폭넓은 보급과 급속한 대중화에 따라 네트워크를 통하여 음성을 전송하거나 저장하려는 시도가 많이 이루어지고 있다.

본 논문에서는 네트워크를 통한 멀티미디어 전송에서 음성부호화 표준으로 널리 상용되는 ITU-T G.723.1 dual-rate speech coder의 효율적인 다채널 구현을 위한 고속 알고리즘을 제안한다. 고속 알고리즘은 부호화 과정에서 많은 계산량을 차지하는 적응 코드북 검색과 고정 코드북 검색 과정에 적용된다. 적응 코드북 검색 과정에서는 지연과 이득을 동시에 찾는 기존의 방법 대신, 지연과 이득을 순차적으로 검색함으로써 계산량을 개선하였다. 전송률에 따라 다른 알고리즘을 사용하는 고정 코드북 검색 과정에서는 다음과 같은 고속 알고리즘을 제안한다. MP-MLQ(Multi-Pulse Maximum Likely Quantization) 방법을 사용하는 높은 전송률(6.3 kbit/s)인 경우, 펄스를 등간격으로 검색함으로써 계산량을 줄였다. ACELP(Algebraic CELP) 방법을 사용하는 낮은 전송률(5.3 kbit/s)인 경우는 기존의 nested-loop 검색 방법 대신, 펄스를 쌍으로 나누어 순차적으로 찾는 depth-first tree 검색 방법을 적용하여 계산량을 감소시켰다.

제안된 고속 알고리즘에 대해 주관적 음질 평가 방법을 수행한 결과, 제안된 방법이 기존의 방법에 비해 음질의 저하가 없음을 확인하였다. 고정 소수점 DSP인 TMS320C6201을 사용하여 고속 알고리즘을 구현한 결과, 높은 전송률의 경우에는 10.29 MIPS, 낮은 전송률의 경우에는 8.70 MIPS의 연산량으로 구현 가능함을 확인하였다.

### 1. 서론

인터넷의 사용자가 급증하면서 데이터망을 통한 음성신호의 전송 및 저장의 필요성이 대두되고 있으며 그에 대한 여러 시도가 이루어지고 있다. 이를 위하여 다양한 음성 압축 알고리즘을 이용하여 데이터와 음성을 통합하고 있다. 대표적인 음성 부호화기로 널리 사용되는 있는 ITU-T G.723.1[1]과 G.729 CS-ACELP[2]를 그 예로 들 수 있다.

G.723.1은 분석에 의한 합성 방법을 사용하는 CELP(Code-Excited Linear Prediction)[3] 유형의 음성 부호화기이며 전체 부호화기의 복잡도 가운데서 대부분이 여기 신호를 양자화하기 위한 적응 코드북 검색 과정과 고정 코드북 검색 과정에서 요구되어진다. 그러므로 본 논문에서는 제안하는 G.723.1 음성 부호화기의 고속 알고리즘은 적응 및 고정 코드북 검색과정에 적용된다. 기존의 적응 코드북 검색 과정은 5차 피치추적기의 지연과 이득을 동시에 검색하므로 많은 계산량을 요구한다. 고정 코드북 검색 과정에서는 높은 전송률의 경우에는 다중 펄스 방법인 MP-MLQ 방법을 사용하고 낮은 전송률인 경우에는 대수코드북을 사용하는 ACELP 방법을 사용한다. 고속 적응 코드북 검색 알고리즘은 지연과 이득을 순차적으로 결정하므로써 요구되는 연산량을 개선하였다. 고정 코드북 검색과정에서 높은 전송률인 경우에는 펄스 위치 조합을 줄이기 위해 펄스 간격을 등간격으로 검색하였으며, 낮은 전송률인 경우에는 기존의 nested-loop search 대신에 depth-first tree search 방법을 사용하여 펄스를 검색하였다.

고속 알고리즘에 대한 주관적인 성능 평가 결과, 기존의 방법에 비해 음질 저하가 크지 않음을 확인하였다. TI사의 고정 소수점 DSP인 TMS320C6201을 사용하여 구현한 결과, 전송률에 따라 부호화 과정에서 10.29 MIPS와 8.70 MIPS의 개선된 연산량으로 구현이 가능함을 확인하였다.

본 논문의 구성은 다음과 같다. 2장에서는 G.723.1 음성 부호화기의 알고리즘을 간단히 설명하고, 3장에서는 본 논문에서 제안된 고속 알고리즘을 적용 코드북 검색 과정과 고정 코드북 탐색 과정으로 나누어 살펴본다. 4장에서는 기존의 부호화기와 제안된 부호화기의 성능을 비교하고 제안된 부호화기의 실시간 구현 결과를 제시하고, 마지막으로 5장에서 결론을 맺는다.

## 2. ITU-T G.723.1 음성부호화기

멀티미디어 통신용 음성 부호화 표준안으로 제정된 G.723.1 음성 부호화기는 5.3 kbit/s와 6.3 kbit/s의 두 개의 전송률을 갖는 CELP 유형의 부호화기이다. 두 가지 전송률을 제공하므로 시스템 설계 과정에서 성능과 복잡도 면에서 유연성을 가질 수 있는 장점이 있다. 현재 G.723.1은 G.729와 함께 데이터망을 통하여 음성을 전송하고자 하는 VoN(Voice over Network) 응용 분야에 널리 사용되고 있다.

G.723.1 부호화기는 음성이나 오디오 신호를 30ms의 길이를 갖는 프레임단위로 부호화하며 7.5ms의 예견 구간이 있기 때문에 총 37.5ms의 알고리즘 지연이 생긴다. 또한 부호화기와 복호화기에서 데이터를 처리하는데 소요되는 실제 시간, 통신채널간의 전송시간, 멀티플렉싱을 위한 추가 버퍼링 등의 추가지연 등이 발생한다.

G.723.1 부호화기의 입력은 아날로그 신호를 일반전화기의 주파수대역으로 여과시키고 8000 Hz로 표본화시킨 다음, 16-bit 선형 PCM으로 양자화된 신호이다. 복호화기의 출력은 비슷한 방식으로 아날로그 신호로 전환된다. 이 부호화기는 합성에 의한 분석을 이용한 선형예측 기법을 사용하며 지각가중 오차신호를 최소화한다. 부호화기는 각각 240 개의 샘플(8kHz 표본화 주파수 시 30ms)마다 동작한다. 각각의 프레임은 처음에 끝대역 통과필터를 거쳐서 DC 성분을 제거한다. 그리고 4 개의 서브프레임으로 나뉘어 지는데 서브프레임의 길이는 60 샘플이다. 또한 서브프레임마다 10차의 선형 예측계수(LPC)를 계산하는데 마지막 서브프레임의 LPC는 예측분할 벡터 양자화를 이용하여 양자화하고 수신단에 전송한다. 양자화되지 않은 LPC 계수들은 단구간 지각가중필터를 만들 때 사용되며, 이러한 지각가중필터는 여기신호를 양자화하기 위해 필요한 목적신호를 만들 때 사용한다. 즉, 지각가중필터의 출력은 지각가중된 음성신호이다.

선형예측 분석이 끝나면, 두 개의 서브프레임(120 샘플)마다 개회로 피치 주기를 가중음성신호 도메인에서 구한다. 개회로 피치 주기의 검색 범위는 20에서 140 샘플이다. 이 후부터 음성신호는 서브프레임 단위로 처리된다. 추정된 피치주기를 이용하여 harmonic noise shaping filter를 구성한 후, LPC 합성필터, 포먼트 지각가중 필터와 조합으로 임펄스 응답을 계산하여 여기신호 양자화를 위한 목적신호 생성에 사용한다. 개회로 추정된 피치주기와 임펄스 응답을 이용해서 폐회로 피치 예측을 하며 5차의 피치예측기를 사용한다. 폐회로 피치 주기의 추정은 개회로 피치 주변에서 작은 편차를 갖는 범위 내에서 추정되며 피치예측기의 contribution은 초기의 타겟 벡터에서 제거되며 피치주기와 편차값은 복호화기로 전송된다. 적용 코드북 검색과정에서는

5차의 피치예측기를 사용하며 첫 번째와 세 번째 서브프레임에서는 개회로 피치의  $\pm 1$  범위에서 폐회로 피치 래그를 찾아서 7비트를 사용하여 부호화한다. 두 번째와 네 번째 서브프레임에서는 이전 서브프레임의 양자화된 피치와 -1, 0, 1, 2 값 중의 하나를 2비트를 사용하여 편차부호화한다. 한편, 피치 예측기의 이득은 높은 비트율에선 85 또는 170개의 크기를 갖는 코드북으로 양자화하고, 낮은 비트율에선 170개의 크기를 갖는 코드북을 사용하여 양자화한다.

적용 코드북 검색 과정을 마치면 마지막으로 여기신호의 비주기적 성분을 모델링하는데 전송률에 따라 다른 모델을 사용한다. 높은 비트율에서는 MP-MLQ를 사용하며 낮은 전송률에선 ACELP 방법을 사용하여 여기신호를 양자화한다.

높은 전송률인 경우의 고정코드북 검색은 다중펄스 방법에 의해 이루어진다. 즉, 양자화 과정에서 발생하는 오차 신호는 다음과 같으며 이 값의 제곱값을 최소화하는 펄스 위치  $m_k$ , 부호  $a_k$  그리고 이득  $G$ 를 찾는다.

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

여기서  $r(n)$ 은 고정 코드북 검색을 위한 목적신호이며  $\hat{r}(n)$ 은 고정 코드북 검색을 통해 양자화된 신호이다. 최종적으로 오차신호  $err(n)$ 의 제곱의 평균값을 최소로 만드는 양자화된 변수들의 조합이 선택되고 전송된다. 펄스 위치  $m_k$ 는  $\binom{30}{M}$ 의 조합으로 부호화된다.

낮은 전송률의 고정 코드북 검색과정은 ACELP 방법에 기반하며 17비트의 대수코드북이 사용된다. 각 고정 코드북엔 4개의 펄스를 포함하고 있으며 펄스의 위치와 부호는 아래의 표와 같다. 이 경우도 위의 식(1)의 제곱값을 최소화하는 펄스 위치와 부호 그리고 이득을 찾는다. 가중 음성신호  $r(n)$ 과 가중 합성신호인  $\hat{r}(n)$  사이의 오차제곱평균이 최소화되도록 이루어진다. 탐색 과정은 4개의 펄스를 nested-loop 검색 방법으로 검색한다.

표 1. ACELP 여기 코드북 구조.

| Pulse | Sign    | Positions                       |
|-------|---------|---------------------------------|
| $i_0$ | $\pm 1$ | 0, 8, 16, 24, 32, 40, 48, 56    |
| $i_1$ | $\pm 1$ | 2, 10, 18, 26, 34, 42, 50, 58   |
| $i_2$ | $\pm 1$ | 4, 12, 20, 28, 36, 44, 52, (60) |
| $i_3$ | $\pm 1$ | 6, 14, 22, 30, 38, 46, 54, (62) |

## 3. 제안된 고속 알고리즘

5.3 kbps와 6.3 kbps G.723.1 음성 부호화기를 구현하기 위해서는 부호화기 전체 계산량은 약 16 MIPS 정도의 계산량을 요구되어진다[4]. G.723.1의 부호화 과정의 계산량을 비교하면 표 2와 같다. G.723.1 음성 부호화기의 계산량을 살펴보면 대부분의 계산이 적용 코드북과 고정 코드북 검색 과정에서 이루어짐을 알 수 있다. 그러므로 본 논문에서는 여기 신호를 양자화하는 적용 코드북 및 고정 코드북 검색을 위한 고속 알고리

들을 제안하고자 한다.

표 2. 부호화 과정의 계산량 비교.

| Encoder Module         | 5.3 kbps | 6.3 kbps |
|------------------------|----------|----------|
| LPC/LSP computation    | 7.1 %    | 6.9 %    |
| Open-loop pitch search | 11.0 %   | 10.6 %   |
| ACB search             | 31.9 %   | 28.3 %   |
| FCB search             | 45.4 %   | 51.9 %   |
| Others                 | 4.6 %    | 2.3 %    |
| Sum                    | 100 %    | 100 %    |

### 3.1. 고속 적응 코드북 검색

기존의 적응 코드북 검색 과정은 아래의 식과 같이 5차 피치예측기를 사용하여 지연과 이득을 동시에 검색한다.

$$\begin{aligned}
 MSE_{ACB} &= \sum_{n=0}^{N-1} \left( \hat{a}[n] - \sum_{i=-2}^2 G_i \hat{i}[n-L+i] \right)^2 \\
 &= \sum_{n=0}^{N-1} \hat{r}^2[n] - 2 \left( \sum_{i=-2}^2 G_i \left( \sum_{n=0}^{N-1} \hat{a}[n] \hat{i}[n-L+i] \right) \right. \\
 &\quad \left. - \frac{1}{2} \sum_{i=-2}^2 G_i^2 \left( \sum_{n=0}^{N-1} \hat{r}^2[n-L+i] \right) \right. \\
 &\quad \left. - \sum_{i=-2}^2 \sum_{j=-2}^2 G_i G_j \left( \sum_{n=0}^{N-1} \hat{i}[n-L+i] \hat{i}[n-L+j] \right) \right). \quad (2)
 \end{aligned}$$

$$L_{op} - 1 \leq L \leq L_{op} + 1 \text{ (or } L_{op} + 2).$$

여기서  $\hat{a}[n]$ 과  $\hat{i}[n]$ 는 각각 코드북 검색 목적신호와 가중된 합성신호이고  $L_{op}$ 는 개회로 피치이며  $G_i$ 는 5차 피치예측기의 이득이다. 그리고  $N$ 은 서브프레임 크기이다. 식 (2)을 최소화시키는 과정은 아래의 식 (3)를 최대화시키는 지연과 이득을 찾는 과정과 동일하다.

$$\begin{aligned}
 C_{ACB} &= \sum_{i=-2}^2 G_i \left( \sum_{n=0}^{N-1} \hat{a}[n] \hat{i}[n-L+i] \right) \\
 &\quad - \frac{1}{2} \sum_{i=-2}^2 G_i^2 \left( \sum_{n=0}^{N-1} \hat{r}^2[n-L+i] \right) \\
 &\quad - \sum_{i=-2}^2 \sum_{j=-2}^2 G_i G_j \left( \sum_{n=0}^{N-1} \hat{i}[n-L+i] \hat{i}[n-L+j] \right). \quad (3)
 \end{aligned}$$

제안된 적응 코드북 검색과정은 지연과 이득을 순차적으로 검색한다. 지연은 G.729처럼 1차 피치예측기와 같은 방법으로 찾고 구해진 지연에 대해 5차 피치예측기의 이득을 구한다. 즉, 지연 검색은 아래의 식과 같다.

$$\begin{aligned}
 MSE'_{ACB} &= \sum_{n=0}^{N-1} (\hat{a}[n] - g \hat{i}[n-L])^2 \\
 &= \sum_{n=0}^{N-1} \hat{r}^2[n] - 2g \sum_{n=0}^{N-1} \hat{a}[n] \hat{i}[n-L] \\
 &\quad + g^2 \sum_{n=0}^{N-1} \hat{r}^2[n-L] \quad (4)
 \end{aligned}$$

여기서  $g$ 는 1차 피치예측기의 이득이다. 폐회로 피치  $L_{cb}$ 는 아래의 식을 최대화시키는  $L$ 값이다.

$$C_{lag} = \frac{\left( \sum_{n=0}^{N-1} \hat{a}[n] \hat{i}[n-L] \right)^2}{\sum_{n=0}^{N-1} \hat{r}^2[n-L]} \quad (5)$$

지연  $L_{cb}$ 가 결정되면 5차 피치예측기의 이득 검색 과정은 다음을 최대화시키는  $G$ 를 찾는 과정이다.

$$\begin{aligned}
 C_{gain} &= \sum_{i=-2}^2 G_i \left( \sum_{n=0}^{N-1} \hat{a}[n] \hat{i}[n-L_{cb}+i] \right) \\
 &\quad - \frac{1}{2} \sum_{i=-2}^2 G_i^2 \left( \sum_{n=0}^{N-1} \hat{r}^2[n-L_{cb}+i] \right) \\
 &\quad + \sum_{i=-2}^2 \sum_{j=-2}^2 G_i G_j \left( \sum_{n=0}^{N-1} \hat{a}[n-L_{cb}+i] \hat{a}[n-L_{cb}+j] \right) \quad (6)
 \end{aligned}$$

기존의 적응 코드북 검색 과정에서는 프레임마다  $13 \times L_{ACB}$  개의 코드북을 검색하며 제안된 검색 과정에서는 프레임마다  $4 \times L_{ACB}$  개의 코드북을 검색한다. 여기서  $L_{ACB}$ 는 적응 코드북의 이득 코드북을 나타내며 그 크기는 170이다.

### 3.2. 고속 고정 코드북 검색

높은 전송률의 고정 코드북 검색과정에서 다중 펄스 방법을 이용하여 여기 신호를 모델링한다. 고정 코드북 탐색 과정을 살펴보면, 다중 펄스를 찾기 위한 양자화 과정에서 개회로 검색 과정으로 찾은 코드북 이득  $G_{oi}$ 의 주위값인  $[G_{oi}-2, G_{oi}+1]$ 에 대해 4번의 폐회로 검색을 수행한다. 이러한 과정을 홀수번째 펄스 위치와 짝수 펄스위치에 대해 각각 반복한다. 고속 검색을 위해 ACELP와 같이 검색하는 펄스의 위치를 등간격을 유지하도록 검색을 수행하였다.

낮은 전송률에서는 대수코드북으로 이루어진 ACELP 방법에 기반하며 각 서브프레임을 4개의 track으로 나누고 각 track마다 1 개의 펄스를 검색한다. 각 track마다 펄스가 존재할 수 있는 가능한 위치의 경우의 수는 8개이다. 기존의 검색 과정에서는 nested-loop search를 사용하여 4 개의 펄스를 찾으므로  $8 \times 8 \times 8 \times 8 = 4096$  개의 펄스 위치 조합에 대해 분석-합성 방법으로 위치를 검색한다. 실제적인 구현에서는 마지막 loop에 대한 검색을 제한하여 계산량을 줄인다. G.729A[5]나 EVRC[6]에서처럼 펄스를 여러 쌍으로 나누어 순차적으로 검색하는 depth-first tree search를 사용하면 가능한 펄스 위치의 조합을 줄일 수 있다. 2개의 track씩 묶어서 검색하면 가능한 펄스의 위치의 조합은  $2 \times \{(8 \times 8) + (8 \times 8)\} = 256$ 가지로 줄어들게 된다. nested-loop search 대신 depth-first tree search를 사용하면 약 0.3dB 정도 SNR이 감소하는 것으로 알려져 있다.[5]

## 4. 실시간 구현 및 성능 평가

제안된 고속 알고리즘에 대해 주관적 음질평가를 통하여 성능 평가를 수행하였다. 평가 방법은 10명의 평가자에게 헤드셋을 통하여 남녀 각각 2명의 화자의 문장을 2번씩 들려주고 기존의 방법과 제안된 방법의 음

질의 우열을 가리는 선호도 평가를 수행하였다. 표 3은 기존의 부호화기와 제안된 부호화기의 선호도 평가 결과를 보여준다. 높은 전송인 경우, 30%가 기존의 부호화기의 음질을 선호하였고 나머지 70%는 음질의 우열을 가리지 못하거나 제안된 부호화기의 음질을 선호하였다. 낮은 전송률인 경우에도 이와 비슷한 결과를 보여준다. 45%가 음질의 우열을 가리지 못하였으며 35%와 20%는 각각 기존의 부호화기와 제안된 부호화기의 음질을 더 선호하였다. 주관적 음질 평가를 통하여 제안된 고속 알고리즘은 음질 저하가 거의 없음을 확인하였다.

표 3. 선호도 평가 결과.

| A-B Test      | Original (A) : Fast (B) |          |
|---------------|-------------------------|----------|
|               | 6.3 kbps                | 5.3 kbps |
| Preference A  | 30                      | 35       |
| Preference B  | 15                      | 20       |
| No preference | 55                      | 45       |

제안된 G.723.1 고속 알고리즘을 TI사의 범용 DSP인 TMS320C6201을 사용하여 구현하였다. TMS320C62x 프로세서 C compiler의 환경에 적합하도록 최적화 하였다. 최적화 방법은 최종적으로 얻어진 어셈블리 프로그램을 비교, 대비함으로써 가장 효율적인 어셈블리 코드를 생성할 수 있는 C 프로그램 구조를 만들어 나갔다. 이 과정에서 loop unrolling 과 double word (32비트) loading 등의 기본적인 테크닉과 함께 C 프로그램 각 루틴의 구조를 통합, 분리하는 방법이 사용되었다. 그 결과 C 프로그램의 상당 부분의 구조가 변경되었다.

최종적으로 최적화된 어셈블리 코드의 복잡도를 클럭수로 측정하여 표 4에 정리하였다. 클럭수는 각 루틴 별로 측정하였으며, 전체 클럭수는 데이터 입출력을 제외한 부호화 과정 전체에 대한 것이다. 결과적인 DSP MIPS와 200MHz에서 동작하는 TMS320C62x로 수용 가능한 채널 숫자를 표 5에 정리하였다.

표 4와 표 5의 결과를 살펴보면 구성된 어셈블리 코드가 167MHz에서 동작하는 TMS320C6211에서 16채널 이상이 구현될 수 있음을 수치적으로 보여주고 있다.

표 4. 구현된 고속 부호화기의 연산량.

| Encoder Module         | Cycle    |          |
|------------------------|----------|----------|
|                        | 6.3 kbps | 5.3 kbps |
| LPC/LSP computation    | 30917    | 30917    |
| Open-loop pitch search | 40672    | 40672    |
| ACB search             | 128120   | 128120   |
| FCB search             | 103120   | 55408    |
| Others                 | 5989     | 5989     |
| Sum                    | 308818   | 261106   |

표 5. 표 4의 결과에 대한 MIPS 값.

| Encoder  | Simulator Clock Cycles per Frame | DSP MIPS | Maximum # of Channel with TMS320C6201 @200MHz |
|----------|----------------------------------|----------|---|
| 6.3 kbps | 261106                           | 8.70     | 20 (174MHz)                                   |
| 5.3 kbps | 308818                           | 10.29    | 17 (175MHz)                                   |

## 5. 결론

본 논문에서는 VoIP와 같은 멀티미디어 통신용 음성 부호화기로 널리 사용되는 G.723.1 음성부호화기의 효율적인 다채널 구현을 위한 고속 알고리즘을 제안하였다. 부호화 과정에서 대부분의 연산량을 차지하는 여기 신호 양자화 과정인 적응 코드북 검색과정과 고정 코드북 검색과정에서 음질의 저하없이 구현할 수 있는 고속 알고리즘을 제안하였다. 그리고 제안된 알고리즘에 대한 주관적 음질 평가를 통해 그 성능을 확인하였으며 TI사 TMS320C6201 DSP 프로세서를 사용하여 실시간으로 구현하였다.

제안된 G.723.1 고속 부호화기는 Gateway와 같은 패킷음성 전송을 시스템이나 다채널 음성을 저장하기 위한 음성 저장 시스템에 응용될 수 있다. 현재는 제안된 고속 알고리즘을 TMS320C6201을 이용하여 다채널 구현을 수행 중에 있다.

## 참고 문헌

- [1] ITU-T Rec. G.723.1, "Dual-rate Speech Coder For Multimedia Communications Transmitting at 5.3 and 6.3 kbit/s," 1996.
- [2] ITU-T Rec. G.729, "Coding of Speech at 8 kbit/s CS-ACELP Speech Coder," 1996.
- [3] B. Atal and M. R. Schroeder, "Stochastic coding of speech signal at very low bit rates," in *Proc. Int. Conf. on Communication*, pp. 1601-1613, May 1984.
- [4] Richard v. Cox, "Three New Speech Coders from the ITU Cover a Range of Applications," *IEEE Communication Magazine*, pp 40- 47, 1997.
- [5] Redwan Salami, Claude Laflamme, Bruno Bessette, and J. P. Adoul, "ITU-T G.729 Annex A: Reduced Complexity 8 kbit/s CS-ACELP Codec for Digital Simultaneous Voice and Data," *IEEE Communication Magazine*, pp 56-63, 1997.
- [6] QUALCOMM Inc., Proposed TIA/EIA/PN-3239 Standard, "Enhanced Variable Rate Codec, Speech Service Option 3 for Wideband Spread Spectrum Digital Systems," Official Ballot Version, April 19, 1996.