

화상 전화용 음성 보코더의 실시간 구현

남 일룡, 서성대, 남현도
 단국대학교 전기공학과

Real-Time Implementation of Speech Vocoder For Video Telephony

Il-Ryong Nam, Sung-Dae Seo, Hyun-Do Nam
 Dept. of Electrical Engineering, DanKook University

Abstract - This paper presents real-time implementation of speech vocoder for PSTN video telephony using ITU G.723 16Kbps ADPCM algorithm.

The ADPCM encoder accepts 8-bit PCM compressed signals and expands it to a 14-bit-per-sample. The predicted values are subtracted from encoded signals to produce difference signals. Adaptive quantization is performed on the difference signal to produce a 2-bit output for transmission over the channel.

Computer simulations and experiments were performed to evaluate the performance of the speech vocoder.

1. 서 론

우리가 흔히 접할 수 있는 음성, 화상, 영상과 같은 정보는 아날로그 형태로 존재하고 있으며, 이를 아날로그 전송망을 통하여 전달하는 전송방식이 사용되어 왔다. 그러나 아날로그 전송방식에서 발생하는 문제점들을 해소하기 위해 디지털 음성을 이용한 전송방식이 계속 증가하고 있는 추세이다.

아날로그 신호의 디지털 신호화 과정을 음성 신호에 적용한 음성 부호화 기술은 [2] 적용분야에 따라 매우 다양한 방식이 개발되었는데, 전송속도가 16~64Kbps로 비교적 높지만 복잡도가 낮고 통화의 품질이 우수하여 일반 음성 통신에 많이 이용되는 파형 부호화 방식 중에서 Adaptive Differential Pulse Code Modulation(ADPCM) 알고리즘은 음성 신호를 직접 양자화하는 Pulse Code Modulation(PCM) 알고리즘과는 달리, 음성 신호가 인접한 표본값과의 상관성이 강하다는 점을 고려하여 상관관계를 이용해서 만들어낸 예측 신호와 실제 신호사이에서의 차이만을 양자화한다. [1,3] 그러므로 음성 신호를 정확히 예측할 수 있다면, 훨씬 더 적은 비트 수로 정확히 양자화하는 것이 가능하다. 부호기에서는 음성 신호를 재구성하기 위해서 양자화된 차분 신호를 예측 신호에 더하게 된다. 또한, ADPCM 알고리즘은 부호화 될 음성 신호의 시변 특성에 적응하기 위해 적응 예측기와 적응 양자화기를 이용하고 있다.

ITU(International Telecommunication Unit)는 다양한 압축률의 ADPCM 알고리즘을 표준으로 규정하고 있는데 권고안 G.723은 16Kbps ADPCM 알고리즘으로서, 64Kbps PCM 신호를 4분의 1로 압축시켜 16Kbps의 ADPCM 신호를 얻어낼 수 있다. [4]

본 연구에서는 권고안 G.723으로 알려진 16Kbps ADPCM 알고리즘을 구현하고 컴퓨터 시뮬레이션을 통하여 그 성능을 확인해 보며, 특별한 목적으로 만들어진 하드웨어나 DSP 보드가 아닌 PC상에서 화상 전화용 음성 vocoder를 이용할 수 있도록 실시간으로 구현하여 원음에 대한 충실도를 확인해 보고자 한다.

2. ADPCM 알고리즘

본질적으로 음성 신호는 quasi-stationary 특성을 지니고 있으므로 source 출력의 분산과 자기상관함수는 시간에 대해서 매우 느리게 변화하는 것을 알 수 있다. PCM 알고리즘이나 Differential PCM (DPCM) 알고리즘에서 이용하고 있는 부호기는 source 출력이 stationary하다는 가정에 기반을 두고 설계되어져 있어 이러한 부호기들의 성능과 효율성을 향상시키기 위해서는 부호기로 하여금 음성 신호의 매우 느린 시변 특성에 적응하도록 하는 것이다. [2]

PCM 알고리즘이나 DPCM 알고리즘에서, quasi-stationary 특성을 갖는 음성 신호를 이용해서 동작하는 균일 양자화기에 의해 만들어진 양자화 오차는 시변 분산(양자화 노이즈 파워)을 갖게 된다. 이러한 양자화 노이즈의 동적 범위를 줄일 수 있는 방법으로는 적응 예측기를 이용하는 것이다. [1,3] 그림 1은 적응 예측기를 적용한 ADPCM 알고리즘의 블록 다이어그램을 보여주고 있다.

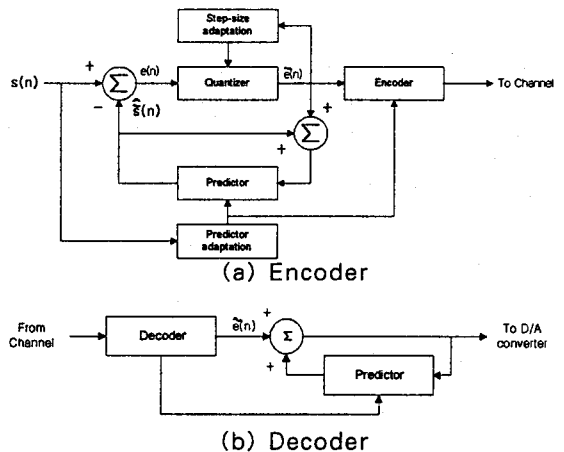


그림 1. ADPCM 알고리즘의 블록 다이어그램

2.1 적응 예측기

DPCM 알고리즘에서 이용하고 있는 예측기에 적응 알고리즘을 적용하면 적응 예측기를 만들 수 있다.

음성 신호는 시변 특성을 가지고 있기 때문에 예측기의 계수들도 계속적으로 변하는 신호의 통계적인 특성들을 예측하기 위해 반드시 주기적으로 변해야만 한다. 따라서 신호 전체의 자기상관함수를 통해 예측기 계수들을 구하지 않고 신호의 특성이 시간에 따라 변하는 것을 고려하여 전체 신호를 짧은 구간들로 나누어 각각에 대해 자기상관함수를 구하고 예측기의 계수들을 구한다. 이와 같은 방식을 이용하면 보다 정확한 적응 예측기를 구현할 수 있다.

본 연구에서는 적응 예측기의 계수들을 구하는 방법으로 Autocorrelation method를 소개한다. [1]

2.2.1 Autocorrelation Method

Autocorrelation method은 과거 P개의 표본값을 이용하여 현재의 표본값을 예측하는 것이다. 즉 $s(n)$ 을 현재의 표본값이라 하고 $\hat{s}(n)$ 을 예측된 값이라 하면, 아래와 같이 정의될 수 있다.

$$\hat{s}(n) = \sum_{i=1}^P a(i) s(n-i) \quad (1)$$

여기서 $\hat{s}(n)$ 은 과거 P개 표본값의 가중선형조합이고, $a(i)$ 는 예측기의 계수이다. $a(i)$ 는 $s(n)$ 과 $\hat{s}(n)$ 의 오차를 최소화 하는 값으로 선정되어야 한다.

수학적으로나 실질적으로 편리한 오차 함수로는 자승 오차의 합이 될 수 있다. 즉, 아래식의 값을 최소화 하는 $a(i)$ 의 값을 선택한다.

$$\begin{aligned} \epsilon_P &\equiv \sum_{n=1}^N e^2(n) = \sum_{n=1}^N \left[s(n) - \sum_{i=1}^P a(i) s(n-i) \right]^2 \quad (2) \\ &= r_{ss}(0) - 2 \sum_{i=1}^P a(i) r_{ss}(i) + \sum_{i=1}^P \sum_{j=1}^P a(i) a(j) r_{ss}(i-j) \end{aligned}$$

여기서 $r_{ss}(m)$ 은 $s(n)$ 의 자기상관함수로써 아래와 같이 정의된다.

$$r_{ss}(m) = \sum_{i=1}^N s(i) s(i+m) \quad (3)$$

$a(i)$ 에 대하여, ϵ_P 를 최소화 하는 것은 다음 식과 같이 선형방정식의 집합이 될 수 있다.

$$\sum_{i=1}^P a(i) r_{ss}(i-j) = r_{ss}(j), \quad j=1, 2, \dots, P \quad (4)$$

행렬의 형태로 나타내면 $Ra = r$ 이 되고, 여기서 R 은 자기상관행렬, a 는 예측기계수, r 은 자기상관벡터이다. 즉,

$$R = \begin{bmatrix} r_{ss}(0) & r_{ss}(1) & \dots & r_{ss}(P-1) \\ r_{ss}(-1) & r_{ss}(0) & \dots & r_{ss}(P-2) \\ \vdots & \vdots & \ddots & \vdots \\ r_{ss}(1-P) & r_{ss}(2-P) & \dots & r_{ss}(0) \end{bmatrix}, \quad (5)$$

$$r = \begin{bmatrix} r_{ss}(1) \\ r_{ss}(2) \\ \vdots \\ r_{ss}(P) \end{bmatrix}$$

으로 나타낼 수 있고, 다음 식으로부터 예측기 계수의 값을 얻을 수 있다.

$$a = R^{-1}r \quad (6)$$

실제적인 DPCM 알고리즘에서 예측기는 양자화기 주위로 폐루프를 형성하여 구현되고 있다.

$$\hat{s} = \sum_{i=1}^P a(i) \hat{s}(n-i) \quad (7)$$

여기서, $\hat{s}(n)$ 은 예측기의 입력으로서, 양자화 과정에 의해 수정된 신호 표본 $s(n)$ 을 나타내고 있다.

그리고 그 차이는 다음과 같이 표현한다.

$$e(n) = s(n) - \hat{s}(n) \quad (8)$$

$e(n)$ 은 양자화기의 입력을, $\hat{s}(n)$ 은 출력을 나타낸다. 양자화된 예측 오차 $\tilde{e}(n)$ 은 2진수로 부호화되고 채널을 통해 receiver로 전송된다. 양자화 오차 $\tilde{e}(n)$ 은 $\hat{s}(n)$ 을 만들어 내기 위해서 $\hat{s}(n)$ 에 더해진다.

receiver 측에서는 전송측에서 사용했던 동일한 예측기가 이용되고, 역시 $\hat{s}(n)$ 을 만들어 내기 위해서 $\hat{s}(n)$ 은 $\tilde{e}(n)$ 에 더해진다.

그림 2에서 이용하고 있는 양자화기는 $\hat{s}(n)$ 에서의 오차가 단순히 양자화 오차 $q(n)$ 이고 복호기에서는 이전의 양자화 오차가 이용되지 않는다는 것을 확실히 보여주고 있다. 즉,

$$\begin{aligned} q(n) &= \tilde{e}(n) - e(n) = \tilde{e}(n) - s(n) + \hat{s}(n) \quad (9) \\ &= \hat{s}(n) - s(n) \end{aligned}$$

그러므로,

$$\hat{s}(n) = s(n) + q(n) \quad (10)$$

추정값의 질을 향상시키기 위해서는 선형적으로 필터링한 과거의 양자화 오차를 이용한다. 따라서, $s(n)$ 의 추정치는 다음과 같이 표현할 수 있다.

$$\hat{s}(n) = \sum_{i=1}^L a(i) \hat{s}(n-i) + \sum_{i=1}^L b(i) \tilde{e}(n-i) \quad (11)$$

$b(i)$ 는 양자화 오차 $\tilde{e}(n)$ 에 대한 필터의 계수이다.

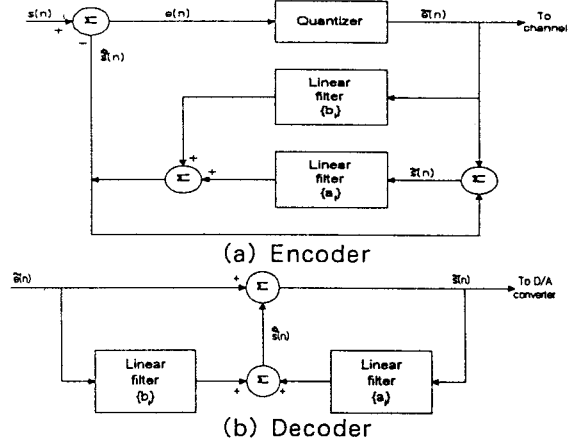


그림 2. 향상된 DPCM 블록 다이어그램

2.3 G.723(16Kbps ADPCM)

그림 3은 전화 채널로 음성을 전송하기 위해 ITU에서 권고한 G.723으로 규정한 16Kbps ADPCM의 부호기와 복호기를 블록 다이어그램으로 보여주고 있다.

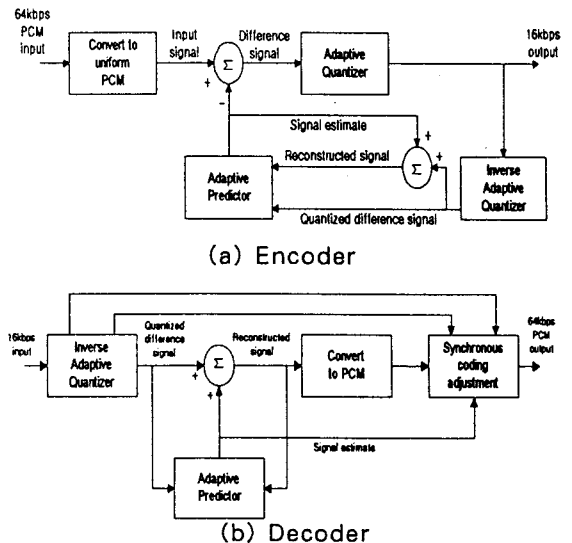


그림 3. PCM에 연결된 16kbps ADPCM 시스템

ADPCM 부호기와 복호기는 음성 신호에 직접 이용될 수 있지만, 현재 전화 네트워크에 널리 이용되고 있는 PCM 시스템과의 호환성을 유지하기 위해서 실제적으로는 PCM 시스템이 필요한 실정이다. 따라서, ADPCM 부호기는 64Kbps에서 신호 표본들을 압축한 8비트 PCM 신호를 받아들이도록 설계되어져 있고 16Kbps 채널에 대한 비트율을 감소시키기 위해서 적용 예측기와 2비트 양자화기를 이용하고 있다. 그리고 ADPCM 복호기는 부호기를 통해서 만들어진 16Kbps 데이터를 받아들여 다시 64Kbps 상에서 압축된 8비트

PCM 신호의 형태로 신호를 재구성하게 된다.

ADPCM 부호기는 8비트 PCM의 압축된 신호를 받아서 표본당 14비트 신호로 확장하고, 확장된 14비트 선형 신호에서 예측 신호를 빼서 양자화기로 들어가는 차분 신호 표본들을 만들어낸다. 이렇게 만들어진 차분 신호는 적응 양자화를 통해서 2비트 출력을 만들어내고 이것을 채널로 전송하게 된다.

부호기와 복호기는 모두 스스로 만들어낸 ADPCM 값에 근거를 두고 내부에서 이용되고 있는 변수들을 갱신시킨다. 결과적으로, 역 적응 양자화기를 포함하고 있는 ADPCM 복호기는 동일한 데이터에 근거를 두고, 내부에서 이용되고 있는 모든 변수들을 갱신하기 위해서 부호기 내에 속해 있다. 이것은 부호기와 복호기가 내부에서 사용하고 있는 변수들의 값에 대한 정보를 전송할 필요 없이 동시에 동작한다는 것을 말해주고 있다.

적응 예측기는 2개의 극점과 6개의 영점을 가진 필터로 구성되어 있다. 이 필터의 계수들은 각각의 새로운 입력 표본들에 대해서 적응성있게 갱신된다.

또한 부호기와 복호기 내부에 있는 2비트로 전송된 ADPCM 값은 역 적응 양자화기의 갱신에 이용되는데 역 적응 양자화기의 출력은 차분 신호의 dequantized 값이다. 이러한 dequantized 값은 음성 신호의 표본을 재생하기 위해 적응 예측기가 만들어 낸 신호에 더해지고 이 신호는 복호기의 출력이 된다.

3. 컴퓨터 시뮬레이션

G.723으로 알려진 16Kbps ADPCM 알고리즘의 실시간 구현을 하기 위해 필요한 시스템 구성도는 그림 4와 같다.

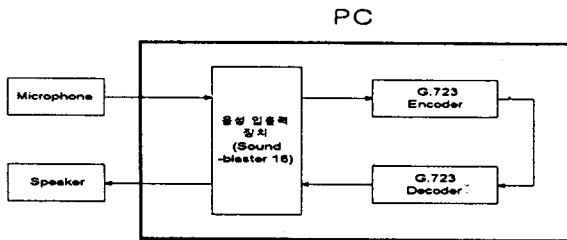
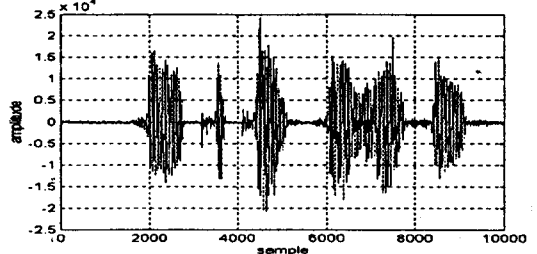


그림 4. 전체 구성도

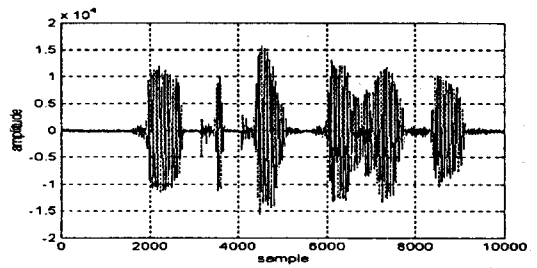
우선, 마이크로폰으로 입력받은 음성 신호를 사운드 카드에 내장되어 있는 ADC를 통하여 16bit 선형 PCM 데이터로 입력받고 이를 8bit Window PCM 포맷 형태의 파일로 저장한 후 16Kbps ADPCM 부호기를 거쳐 2bit 데이터로 압축한다. 그리고 이를 다시 16Kbps ADPCM 복호기로 복호화시켜 8bit Window PCM 데이터로 복원한 후 사운드 카드를 통하여 출력시킨다. 그림 5는 이렇게 복원된 신호를 보여주고 있다. 두 그림을 비교해 보면 잘 알 수 있듯이 원래 음성과 복원된 음성이 거의 일치하고 있음을 보여주고 있다.

그림 6은 16Kbps ADPCM 알고리즘을 PC 상에서 실시간으로 구현하는 과정을 블록 다이어그램으로 보여 주고 있다. 마이크로폰을 통해서 사운드 카드로 전달된 음성 신호는 매 0.6초 간격으로 샘플링 주파수가 8KHz인 8비트 Window PCM 파일로 저장되고 이렇게 저장된 파일은 1바이트씩 16Kbps ADPCM 부호기를 거쳐서 2비트의 ADPCM 데이터가 되며 다시 16Kbps ADPCM 복호기를 거쳐서 크기가 1바이트이고 동일한 샘플링 주파수의 8비트 Window PCM 파일을 생성하게 된다. 매 0.6초 간격으로 생성된 파일이 16Kbps ADPCM 부호기/복호기를 거쳐서 파일을 생성하는데 소요되는 시간은 대략 0.05초 정도이다. 따라서, 마이크로폰을 통해 흘러 들어간 음성 신호가 다시 스피커를 통해서 우리 귀에 들리기까지 소요되는 총 시

간은 모든 지연요소들을 감안하여 약 0.7초 정도이다.



(a) 원래 음성



(b) 복원된 음성

그림 5. 16Kbps ADPCM 부호기/복호기를 거친 음성

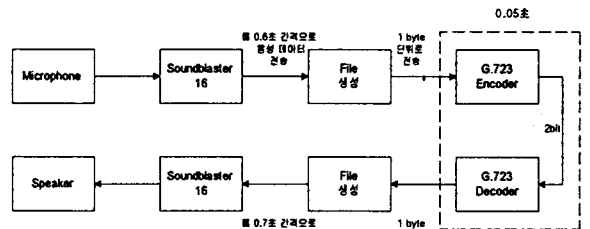


그림 6. 실시간 구현을 보여주는 블록 다이어그램

4. 결론

본 연구에서는 ITU 권고안 G.723으로 알려진 16Kbps ADPCM 알고리즘을 이용해서 화상 전화용 음성 vocoder를 PC상에서 실시간으로 구현하였다.

컴퓨터 시뮬레이션을 통하여 16Kbps ADPCM 알고리즘이 원래 음성과 거의 일치하는 음성 신호를 만들어 내어 원음에 대한 충실도를 확인하였고 이렇게 검증된 16Kbps ADPCM 알고리즘을 실시간으로 구현한 결과, 약간의 지연시간이 발생했지만 특별한 목적으로 만들어진 하드웨어나 DSP 보드가 아닌 PC상에서도 화상 전화용 음성 vocoder의 실시간 구현이 가능함을 입증하였다.

(참고 문헌)

- [1] Vinay K. Ingle and John G. Proakis, "Digital Signal Processing Using MATLAB V.4", PWS Publishing Company, 1997
- [2] John R. Deller Jr., John G. Proakis and John H. L. Hansen, "Discrete-Time Processing of Speech Signals", Prentice-Hall, 1987
- [3] L. R. Rabiner and R. W. Schafer "Digital Processing of Speech Signals", Prentice-Hall, 1978
- [4] R. V. Cox, B. G. Haskell, Y. LeCun, B. Shahraray, and L. Rabiner, "On the Applications of Multimedia Processing to Communications", Proc. IEEE, vol. 86, no 5, pp.768-772, May, 1998