

음성신호의 디지털화와 대역폭축소의 방법에 관하여 [I]—ADPCM 과 ADM

(On Speech Digitization and Bandwidth Compression Techniques [I]—ADPCM and ADM)

殷 鍾 官*
(Un, Chong Kwan)

要 約

本 論文에서는 음성신호의 디지털화와 대역폭축소의 한 방법으로 예측부호화 원리를 사용하는 adaptive differential pulse code modulation(ADPCM)과 adaptive delta modulation(ADM)에 관하여 고찰하였다. ADPCM에서 사용되는 대표적인 적응양자기의 원리를 설명하고 적응예측기의 계수를 얻는 두 방법, 즉 블록해석과 연차해석 방법을 검토하였다. 또한 ADM에서 사용되는 세가지 壓伸方法(instantaneous, syllabic, hybrid companding)을 구체적으로 설명하고 그의 성능을 비교하였다. 마지막으로 ADPCM 과 ADM 을 음성신호의 부호化器로 쓸 때의 성능과 장단점들을 비교 검토하였다.

Abstract

This paper deals with speech digitization and bandwidth compression techniques, particularly two predictive coding methods—namely, adaptive differential pulse code modulation(ADPCM) and adaptive delta modulation(ADM). The principle of a typical adaptive quantizer that is used in ADPCM is explained, and two analysis methods for the adaptive predictor coefficients, block and sequential analyses, are discussed. Also, three companding methods(instantaneous, syllabic, and hybrid companding) that are used in ADM are explained in detail, and their performances are compared. In addition, the performances of ADPCM and ADM as speech coders are compared, and the merits of each coder are discussed.

1. 序 論

지난 10년동안 디지털컴퓨터의 기술과 LSI(Large scale Integrated Circuits) 및 마이크로프로세서(micro-processor) 기술의 급격한 발전은 전자 계통의 여러 분야, 특히 통신분야에 많은 영향을 주고 있다. 그 결과 여러가지 통신 방식중 특히 음성통신은 종래는 어네로그(analog) 방식을 많이 써 왔지만 오늘 날은 디지털방식을 많이 사용하고 있다. 음성신호를 어네로그방법으로 송신하지 않고 디지털화해서 송신하면 여러 가지 장점들이 있다는 것은 잘 알려진 사실로써 이를 열거하면,

- 디지털신호는 어네로그 신호와는 달리 비교적 잡음이나 漏話(crosstalk), 歪曲(distortion)에 별로 영향을 받지 않는 점

- 장거리 통신에서 신호를 재생할 때 점차적으로 신호가 약화되지 않는 점
- 디지털화 된 신호는 암호화 하기가 쉬운 점
- 디지털통신 시스템은 LSI 회로와 microprocessor 기술에 의해서 소형으로 값싸게 만들 수 있는 점, 등이다.

이 論文에서는 음성신호를 디지털화해서 어떻게 좁은 대역폭으로 송수신 할 수 있나에 대해서 지난 70년대에 연구된 방법들을 비교 검토하고 최근 새로이 연구된 결과들을 소개 하고자 한다. 디지털화 된 음성신호는 컴퓨터 저장(computer storage)과 처리¹⁾, 자동음성회답(automatic voice response)²⁾, 話者認識(speaker recognition)³⁾, 등 여러 분야에 응용이 되지만 이 논문에서는 주로 음성통신에 관해서만 취급하겠다.

음성을 디지털화하는 데는 크게 분류해서 세 가지

*正會員, 한국과학원 전기및 전자공학과
(Dept. of Electrical Science KAIS)

接受日字: 1978年 4月 13日

방법이 있다. 첫째는 음성파형을 양자화하는 방법이고, 둘째는 vocoder 방법, 그리고 세째는 위의 두 가지 방법을 병행해서 쓰는 것이다. 첫째 방법은 직접적인 방법으로서 pulse code modulation(PCM)^{4,5,6)}, differential PCM(DPCM)⁶⁻¹²⁾과 delta modulation(DM)¹³⁻²⁰⁾을 예로 들 수 있고, 전송속도가 비교적 큰 반면에 시스템의 구성이 간단하다. 두번째 방법은 간접적인 것으로서 음성파형을 직접 양자화하는 것이 아니라 음성을 분석해서 그의 특성만을 송신하는 것이다. 예로서는 linear predictive coder(LPC)²¹⁾, channel vocoder²²⁾, formant vocoder^{23,23)}를 들 수 있다. 세번째 결합 방법은 앞서 기술한 두 방법들의 장점들을 택한 것으로서 residual excited linear prediction(RELP) vocoder²⁴⁾, adaptive predictive coder (APC)²⁵⁾ 등을 예로 들 수 있다.

위에 열거한 음성의 digitizer들○ 비교적 잘 알려진 PCM을 제외한 모든 방법들을 본 논문에서 취급할 것인 바, 지면 관계상 二部로 나누어 一部에서는 DM과 DPCM만 고찰하고 二部에서는 vocoding 방법을 토론하겠다.

2. Adaptive Differential Pulse Code Modulation(ADPCM)

ADPCM을 고찰하기 이전에 그림 1에 그려진 기본 DPCM을 간단히 설명하면 다음과 같다. DPCM에서는 帶域制限率 Nyquist rate로 sample이 된 입력의 음성신

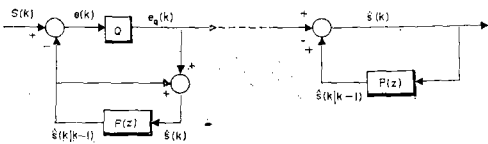


그림 1. 기본 DPCM 시스템
Fig. 1. Basic DPCM system.

호가 귀환회로(feedback path)의 예측기(predictor)에 의해서 예측이 되고 그 두 신호차 $e(k) = s(k) - \hat{s}(k/k-1)$ 를 L level의 양자기(quantizer)에 의해서 양자화하게 된다. DPCM의 양자화로 인한 잡음을 $n_q = e_q(k) - e(k)$ 라고 하자. 그러면,

$$\begin{aligned} s(k) &= \hat{s}(k/k-1) + e_q(k) \\ &= \hat{s}(k/k-1) + e(k) + n_q(k) \\ &= s(k) + n_q(k) \end{aligned} \quad (1)$$

를 얻는다. 위의 식 (1)은 양자기나 예측기가 適應(adaptive) 또는 非適應(nonadaptive)함에 관계없이 언제나 성립한다. 그러므로 decode된 신호 $s(k)$ 를 원래의 입력신호 $s(k)$ 와 가능한한 같게 하기 위해서는

양자화로 인한 잡음 $n_q(k)$ 를 어떻게 최소로 만드는가에 따르게 되는데 여기에서 예측기와 양자기를 adaptive로 만들 필요성이 생기게 된다.

이를 위하여 양자기의 step의 크기를 양자기의 입력신호의 分散(variance)와 비례하게 만들어 주면 양자화로 인한 잡음전력은 입력신호의 전력에 거의 비례하게 된다. 그러므로 예측기를 adaptive로 만들어 입력신호를 가능한한 정확하게 예측해서, 양자기의 입력신호 $s(k) - \hat{s}(k-1)$ 의 分散을, 줄이고, 더 나아가 양자기를 adaptive로 해서 이 分散과 match시키면 DPCM을 최적 시스템으로 만들 수 있다.

ADPCM에서는 양자기와 예측기를 반드시 둘다 adaptive로 만들 필요는 없고, 경우에 따라서는 둘중 하나만 adaptive로 만들고 다른 하나는 고정시킬 수 있다. 양자기를 adaptive로 만들 경우 예측기는 대개 tap이 하나인 累算器(accumulator)를 많이 쓴다.

예측기를 adaptive로 만들 경우에 어떻게 예측 오차(prediction error)를 최소화 하는가에 따라 tap의 계수들을 구하는 방법이 여러 가지 있는데, 크게 구분하면 tap계수의 수치를 정해진 시간마다 바꾸는 블록(block)해석 방법과, 매 sampling 시간마다 바꾸는 順次(sequential)해석의 방법이 있다.

A. 適應 量子器(adaptive quantizer)

양자기의 step의 크기를 입력신호와 같이 만들기 위해서는 양자기의 입력신호(즉 오차신호)를 추정해야 하는 데 그 방법에는 두 가지가 있다.

(1) 前方推定(feedforward estimation)

이 경우는 음성신호의 진폭을 비교기(comparator) 앞에서 계산하는 것으로 이를 위해서는 신호를 저장하기 위한 buffer가 필요할 뿐더러 또한 추측한 결과를 DPCM 출력신호(즉 양자화된 오차신호)와 다중화해서 송신해야만 한다.

(2) 歸還推定(feedback estimation)

양자기의 step의 크기를 양자화된 오차신호의 sample의 크기에 따라 귀환으로 결정하는 방법으로 물론 이 경우에는 추측된 결과를 송신할 필요가 없다.

위의 두 방법 중 전자를 쓰면 후자의 방법보다 정확하게 입력신호의 크기를 추정할 수 있으나 다중화 등 시스템이 복잡하게 되는 관계로 잘 안쓰고 후자의 방법을 많이 쓰고 있다.

양자기의 각 step level의 adaptation은 입력신호 진폭의 확률분포함수 [probability density function(pdf)]에 따라서 설계할 수 있다. 음성신호의 pdf는 실험으로 얻은 바에 의하면 Gamma pdf,

$$p(x) = \frac{\sqrt{k}}{2\sqrt{\pi}} \cdot \frac{e^{-k|x|}}{\sqrt{x}}$$

또는 Laplacian pdf,

$$p(x) = \frac{\alpha}{2} e^{-\alpha|x|}$$

에 근사하다.²⁶⁾ 여기서 k 와 α 는 입력신호의 rms 값 σ_x 와 반비례 되는데 Gamma pdf에 대해서는 $k = \sqrt{0.75}/\sigma_x$ 이고 Laplacian pdf는 $\alpha = \sqrt{2}/\sigma_x$ 이다. 음성신호의 실제 진폭의 분포를 Gamm pdf, Laplacian pdf와 비교한 것이 그림 2에 그려져 있다. PCM에서 μ_{th} law나 A_{th} law의 log 압축(companding)을 쓰는 이유도 위와 같은 pdf를 가졌기 때문이다.

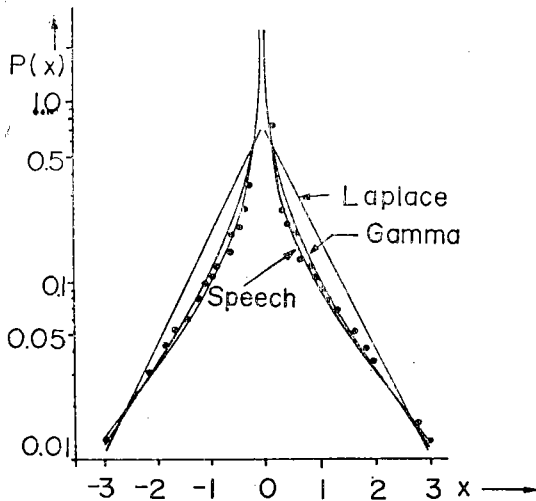


그림 2. 실제 음성의 진폭과 Gamma, Laplace 확률 분포의 비교²⁶⁾
Fig. 2. Comparison of real speech and theoretical Gamma and Laplace probability densities.²⁶⁾

ADPCM의 適應 量子器로 대표적인 예는 Bell 연구소에서 개발된 양자기를 들 수 있다.²⁷⁾ 이 양자기는 step의 크기를 입력신호의 分散에 맞추기 위해서 word의 memory를 쓴다. 예를 들어 B-bit의 均一(uniform) 양자기의 출력은,

$$Y_r = H_r \frac{\Delta_r}{2},$$

$$\pm H_r = 1, 3, 5, \dots, 2^B - 1 \quad (\Delta_r > 0, B \geq 2)$$

라 하자. 그러면 現 step의 크기 Δ_r 은 前 step의 크기 Δ_{r-1} 와 고정된 multiplier 함수 $M(H_{r-1})$ 를 곱해서 얻는다;

$$\Delta_r = \Delta_{r-1} M(|H_{r-1}|)$$

양자기의 level의 수가 4, 8, 16인 경우, 컴퓨터 simulation으로 얻은 multiplier 함수의 최적치[SQNR(signal-to-quantization noise ratio)을 最大化 하는 점에서]

들이 도표 1에 나타나 있다. 위의 한 예로서 그림 3에 3-bit(또는 8 levels)의 適應 量子器가 그려져 있다.

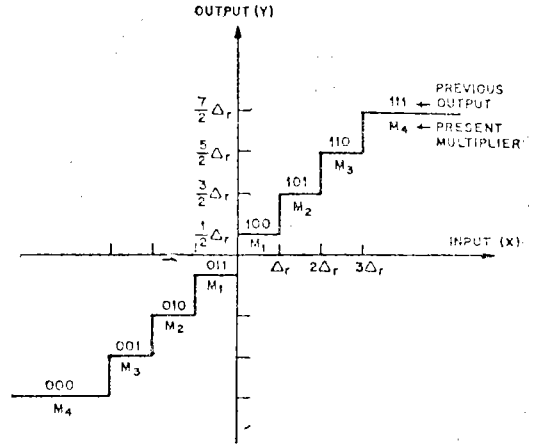


그림 3. ADPCM adaptive quantization²⁷⁾
Fig. 3. ADPCM adaptive quantization.²⁷⁾

표 1. ADPCM Quantizer Multiplication Factors²⁷⁾

	M ₁	M ₂	M ₃	M ₄	M ₅	M ₆	M ₇	M ₈
4 levels	0.8	1.6						
8 levels	0.9	0.9	1.25	1.75				
16 levels	0.9	0.9	0.9	0.9	1.2	1.6	2	2.4

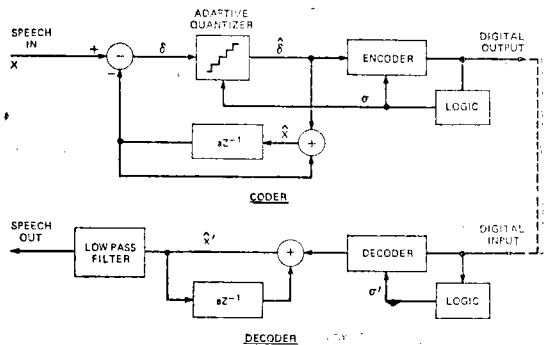


그림 4. ADPCM 블록도²⁷⁾
Fig. 4. ADPCM block diagram²⁷⁾

위에 기술한 量子器를 이용한 ADPCM의 블록(Block)도가 그림 4에 그려져 있다. 여기서 예측기는 first-order이며 또한 계수가 고정된 것으로 그의 최적치는 $a=0.85$ 이다. 그림 4에서 step의 크기를 조정하기 위한 σ 와 σ' , 그리고 신호 推定值 \hat{x} 와 \hat{x}' 은 channel의 오차가 없을 경우에는 동일한 것은 물론이다. 이

ADPCM은 Log PCM($\mu=100$)에 비해 같은 전송속도에서 SQNR이 약 8dB 가량 높다. 이 결과는 음성을 입력신호로 썼을 때 얻어진 것이지만 video 신호를 coding 하는 경우도 비슷한 결과를 얻을 수 있다.

B. 適應 예측기(adaptive predictor)

(1) 블록解釋(block analysis)

DPCM에서 오차신호의 分散을 최소화하기 위해서는 예측기를 입력신호에 적응하도록 만들 필요가 있다는 것을 앞서 이야기한 바 있다. 適應 예측기를 쓸 경우에는 tap을 약 4개 정도 쓰게 되며 tap의 계수는 블록 해석을 할 때 다음과 같이 구하게 된다. 그림 1에서 보는 바와 같이 DPCM의 예측 오차신호는,

$$e(k) = s(k) - \hat{s}(k) \quad (2)$$

이다. 예측기의 tap이 p 개인 경우에 예측된 신호 $\hat{s}(k)$ 는 전의 N 개의 음성 sample을 加重(weighting)해서 加함으로서 얻을 수 있다. 즉,

$$\hat{s}(k) = \sum_{i=1}^p a(i)s(k-i) \quad (3)$$

위 식에서 $a(i)$ 를 예측계수라 한다. 따라서 식(2) 다시 쓰면,

$$e(k) = s(k) - \sum_{i=1}^p a(i)s(k-i) \quad (4)$$

가 된다. 여기서 예측계수는 잘 알려진 minimum-mean-square-error 과정에 의해서 구해지는 데이터를 위해서 N 개의 음성 sample에 대한 오차 에너지를,

$$E = \sum_{k=0}^{N-1} e^2(k) = \sum_{k=0}^{N-1} [s(k) - \sum_{i=1}^p a(i)s(k-i)]^2 \quad (5)$$

가 된다. 위의 오차 에너지를 계수 $a(i)$ 에 관해서 최소화 하면 다음과 같은 식을 얻는다.

$$\phi \cdot \bar{a} = \Psi \quad \text{또는} \quad \bar{a} = \phi^{-1}\Psi \quad (6)$$

식 (6)에서의 vector 들은,

$$\phi \triangleq \begin{pmatrix} \phi_{11} & \phi_{12} & \dots & \phi_{1p} \\ \phi_{21} & \phi_{22} & \dots & \phi_{2p} \\ \vdots & \vdots & \ddots & \vdots \\ \phi_{p1} & \dots & \dots & \phi_{pp} \end{pmatrix}, \quad \bar{a} \triangleq \begin{pmatrix} a_1 \\ a_2 \\ \vdots \\ a_p \end{pmatrix}, \quad \Psi \triangleq \begin{pmatrix} \phi_{10} \\ \phi_{20} \\ \vdots \\ \phi_{p0} \end{pmatrix} \quad (7)$$

이다. 여기서,

$$\phi_{ij} = \sum_{k=0}^{N-1} s(k-i)s(k-j) \quad (8)$$

이다.

위의 블록해석의 경우에는 입력음성 sample을 약 20ms로 나누어 記憶機에 넣은 후 이 sample들을 分析해서 계수를 구하게 된다. 그렇게 함으로써 계수들의 수치를 매 20ms마다 다시 계산해서 송신하게 되는데, 물론 이 계수의 수치는 양자화된 오차신호와 함께 多重化된 후 송신 한다. 계수를 송신하기 위해서는 보통

線型으로 量子化한다. 이 경우에 coding의 정확도가 음질을 비교적 많이 좌우 하지 않기 때문에 계수를 coding하기 위한 bit수는 4개의 계수를 송신하는 경우 약 400bits/s에 불과하다.

DPCM에서 고정된(fixed) 양자기, 적응양자기, 또는 적응예측기를 사용했을 경우 PCM에서보다 얼마나 SQNR이 높아지는가를 그림 5에 나타냈다.¹²⁾ 여기에서 PCM은 μ_{16} law log PCM(4 σ loading)으로 SQNR 8.7dB를 기준으로 할 경우 양자기를 입력신호에 적응하도록 만들면 그로인한 利得(Gain)은 약 7dB(즉 SQNR이 15.7dB)가 된다. 그림에 나타난 entropy coding은 ADPCM 符號化 word의 길이를 일정하게 하지 않고, word의 확률 분포에 따라서 자주 나타나는 word는 짧게 만들고, 반대인 경우에는 길게 만들어 전체의 전송속도를 줄이거나 또는 SQNR을 높이는 방법이다. entropy codig을 쓰는 경우 SQNR을 3dB 정도 증가시킬 수 있다.

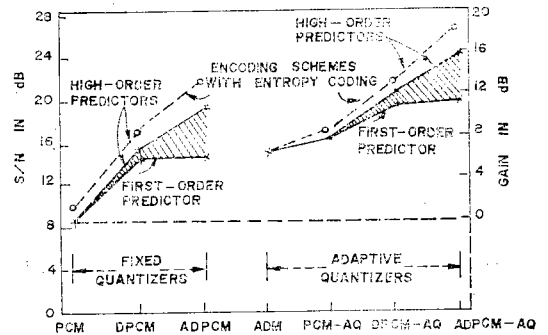


그림 5. 음성 coding 시스템들의 SQNR 値와 log PCM과 비교할 때의 利得¹²⁾

Fig. 5. SQNR values and gains over log PCM for different speech coding system. (AQ-Adaptive Quantization¹²⁾)

(2) 順次解釋(sequential analysis)

適應 예측기에서 계수를 계산하는 또 한 가지 방법으로는 順次解釋方法이 있는 데이터를 세분하면 adaptive gradient 방법과 Kalman 방법이 있다.^{8,10,11)} 이 順次解釋의 특징은 계수들의 수치가 일정한 시간마다 변경되는 것이 아니라 매 sampling 시간마다 계산되는 점이다. 또한 이 경우에는 예측계수를 직접 송신하지 않고, 수신기에서 decode된 음성 sample에 의해서 推定되는 것이 블록해석과 크게 다른 점이다.

順次해석에서 계수는 每 入力 sample마다 다음과 같이 推定하게 된다.

$$\hat{a}(k+1) = \hat{a}(k) + K(k+1)e_q(k+1) \quad (9)$$

여기에서

$$\hat{a}^T(k) = [a_1(k), \hat{a}_2(k), \dots, a_p(k)]$$

$$\hat{a}^T(o) = [o, o, \dots, o]$$

이고, $e_a(k+1)$ 은 量子化된 $k+1$ 째의 오차신호이다. 식 (9)에서 Gain $K(k+1)$ 은 順次 해석의 방법, 즉 adaptive gradient 나 Kalman 방법에 따라 달라지는 미전자의 방법을 쓸 경우에는,

$$K_{AG}(k+1) = \frac{0.08 \hat{S}_N(k)}{100 + S^T_N(k) S_N(k)} \quad (10)$$

이고, 후자의 방법을 쓸 경우에는,

$$K_K(k+1) = V \hat{a}(k+1/k) \hat{S}_N(k) [\hat{S}_N^T(k) V_a(k+1/k) \hat{S}_N(k) + 100]^{-1} \quad (11)$$

이다. 여기에서,

$$V \hat{a}(k+1/k) = [I - K_K(k) \hat{S}_N^T(k)] V \hat{a}(k/k-1) + V_w$$

$$V \hat{a}(o) = 0.01 \cdot I$$

$$V_w = 10^{-7} I$$

$$\hat{S}_N^T = [\hat{S}(k), \hat{S}(k-1), \dots, \hat{S}(k-N+1)]^T \quad (12)$$

이다.

앞서 기술한 適應量子器를 쓰고 예측기로 adaptive gradient 방법을 썼을 경우와 Kalman 방법을 썼을 경우에 ADPCM의 성능을 비교하면 그림 6에 나타난 바와 같다.¹¹⁾ 그림에서 보면 Kalman 예측기를 쓴 ADPCM이 adaptive gradient 예측기를 사용한 것보

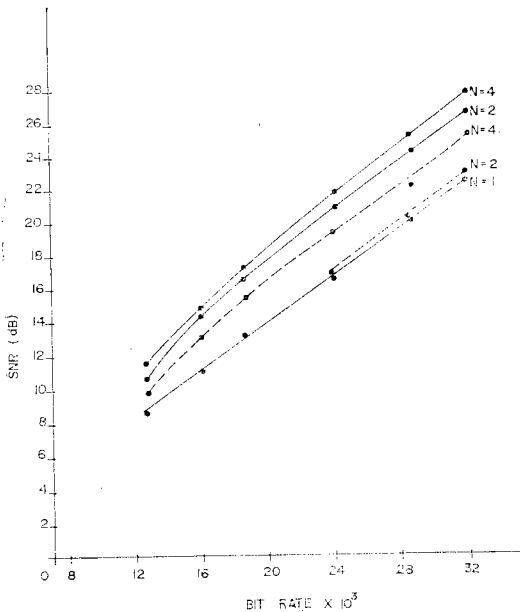


그림 6. 순차해석 ADPCM SNR 値 비교¹¹⁾
(--- Adaptive gradient predictor, — Kalman predictor, N은 tap의 수임)

Fig. 6. Comparison of SNR values of ADPCM's with sequential analysis.¹¹⁾

다 SQNR이 약간 좋지만 보다 많은 계산이 요구되는 단점이 있다.

이상 토의한 것들이 근래에 연구된 음성통신을 위한 ∇ DPCM의 주요 결과이다. 다음은 현재 음성신호의 coder로서 각광을 받고 있는 ADM을 고찰 하겠다.

3. Adaptive Delta Modulation(ADM)

DM은 DPCM의 특별한 경우로 생각할 수 있는데, 양자기의 level이 다만 돌인 반면 sampling rate를 Nyquist rate보다 몇배 늘려서 쓰는 것이 특징이다. ADM을 토의하기 전에 먼저 양자기의 step의 크기가 고정된 線型 DM(LDM)을 고찰하기로 하자.

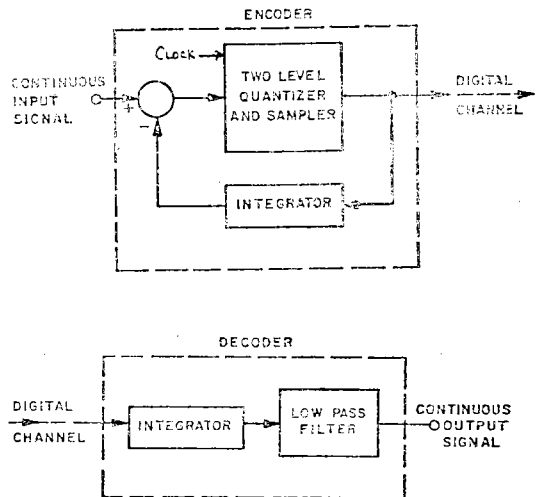


그림 7. 線型 delta modulation
Fig. 7. Linear delta modulation.

LDM이 그림 7에 그려져 있다. 먼저 입력신호가 들어오면 이는 예측기에 의해서 예측이 된 후, 이 차신호가 비교기에 의해서 비교된다. 입력신호가 예측된 신호보다 클 경우에는 비교기에서 binary 신호 "1"이 나오고 반대일 경우에는 "0"이 나온다. 물론 이 binary 신호의 signaling rate(전송속도)는 양자기의 clock rate와 同一하다. 예측기는 간단한 RC 적분기에 불과하며 binary 신호가 "1"인 경우에는 陽電流에 의해서 적분기가 충전되고, "0"일 경우에는 陰電流에 의해서 충전된다.

DM의 decoder는 encoder의 歸還回路(feedback path)와 同一하며 量子化로 인한 雜音을 줄이기 위해서 입력 filter와 같은 filter가 decode된 후에 쓰여진다. DM의 입력신호와 적분기의 出力, 양자기의 出力 binary 신호, 그리고 量子化로 인한 雜音이 그림 8에 그려져 있다.

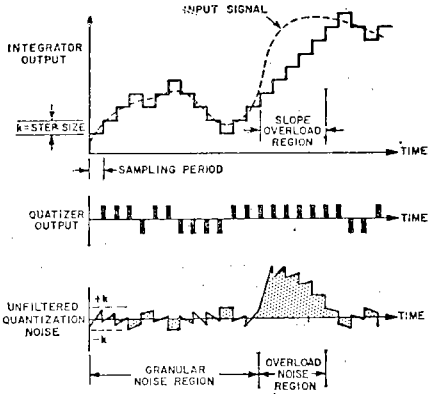


그림 8. DM 적분기 출력타형, 양자기출력 및 양자화잡음

Fig. 8. Waveforms of DM integrator output, Quantizer output and Quantization noise.

DM의 量子化로 인한 雜音은 두 가지로 나눌 수 있다. 하나는 binary 신호가 주로 010101...과 같이 계속적으로 바뀌게 될 때 생기는 粒狀雜音(Granular noise)인데 이것은 PCM이나 DPCM의 量子化 雜音과 성격이 같다.

또 하나는 입력신호가 갑자기 변할 때 DM의 적분기 출력신호의 기울기(slope)가 입력신호의 기울기를

따라가지 못할 경우에 생기는 slope overload 雜音이 있다. 가령 예를 들면 입력신호가,

$$v(t) = A \sin \omega t$$

라면, 그의 傾斜는,

$$v'(t) = A \omega \cos \omega t$$

이고 最大 傾斜는,

$$\max[v'(t)] = A\omega$$

이다. 한편 적분기 출력의 最大 傾斜는 量子器의 step의 크기 Q 와 sampling rate f_s 로 정해진다.

$$\max[v(t)] = Qf_s$$

그러므로 slope overload는 $A\omega > Qf_s$ 일 때에 생기고 그 잡음의 성격은 입력신호와 크게 상관되는(correlate) 특성이 있다.

위의 토의로써 DM의 두 가지 量子化 雜音을 작게 하기 위해서는 상반된 조건이 요하게 되는 것이 명백하다. 즉 주어진 전송속도에서 slope overload 잡음을 작게 하기 위해서는 量子器의 step의 크기 Q 를 늘 수 있으면 크게 하는 것이 좋지만 그렇게 되면 粒狀雜音이 많아지게 된다. 그러므로 DM system을 설계할 때는 이 상반된 조건을 절충해서, step의 크기의 최적치를 찾아 量子化로 인한 잡음을 최소화 시키는 것이 중요한 일이다.

(*다음 호에 계속.....)