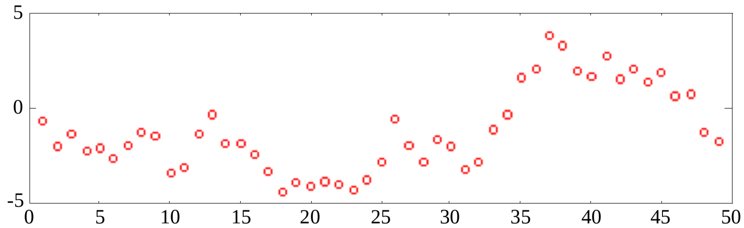
DPCM

(Differential Pulse Code Modulation)

Making scalar quantization work for a correlated source -- a sequential approach.

Consider quantizing a slowly varying source (AR, Gauss, $\rho = .95$, $\sigma^2 = 3.2$).

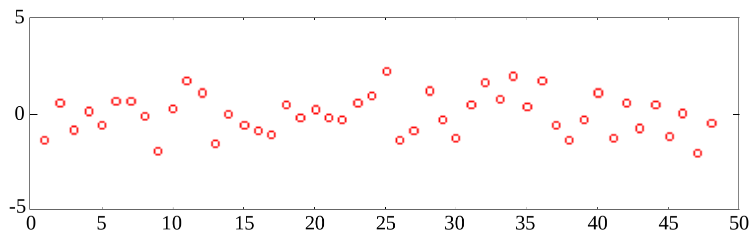


Direct scalar quant:

R=2 $\Rightarrow$ D = 1.2

However, a typical sample is quite similar to its predecessor.

Consider quantizing successive sample differences: $E(X_i - X_{i-1})^2 = \sigma^2/10 = 1.03 < 3.2$.

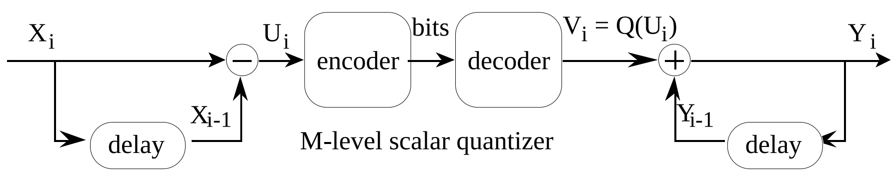


Scalar quant of $(X_i - X_{i-1})$:

$$R=2 \Rightarrow D = 1.2 \times \frac{1.03}{3.2} = .39$$

CODING SAMPLE DIFFERENCES

ATTEMPT 1: NAIVE DIFFERENTIAL CODING



Assume $X_0 = Y_0 = 0$

Key equation: $Y_i = Y_{i-1} + Q(X_i - X_{i-1})$

Consider the errors in the reproductions

$$X_1 - Y_1 = X_1 - Q(X_1)$$

$$X_2 - Y_2 = X_1 + U_2 - (Y_1 + V_2) = (X_1 - Y_1) + (U_2 - V_2)$$

$$X_3 - Y_3 = X_2 + U_3 - (Y_2 + V_3) = (X_2 - Y_2) + (U_3 - V_3)$$

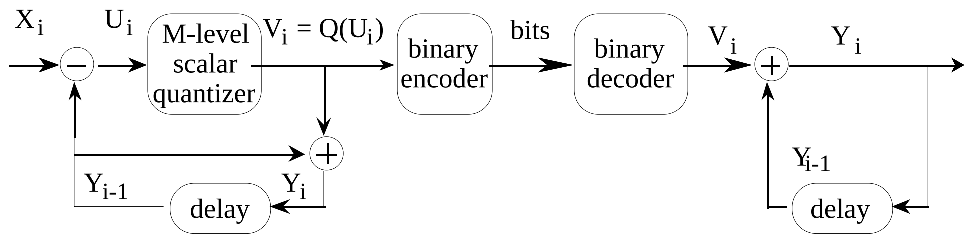
$$= (X_1 - Y_1) + (U_2 - V_2) + (U_3 - V_3)$$

$$X_i - Y_i = X_{i-1} + U_i - (Y_{i-1} + V_i) = (X_{i-1} - Y_{i-1}) + (U_i - V_i)$$

$$= (X_1 - Y_1) + (U_2 - V_2) + (U_3 - V_3) + \ldots + (U_i - V_i)$$

We see that the errors accumulate. This method doesn't work.

Attempt 2: Differential Pulse Code Modulation (DPCM)



Assume $Y_0 = 0$

Key Equations:

- $Y_i = Y_{i-1} + Q(X_i - Y_{i-1})$ i.e. new reproduction equals old reproduction plus quantized prediction error
- $X_i - Y_i = U_i - Q(U_i)$ i.e. overall error = error introduced by scalar quantizer

$$\left(X_i - Y_i = (Y_{i-1}+U_i) - (Y_{i-1}+Q(U_i)) = U_i - Q(U_i)\right)$$

Notes:

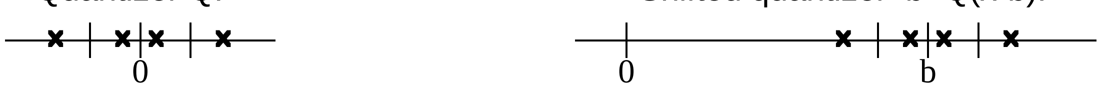
- This method workds well.
- $U_i \neq X_{i-1}$. However $U_i = X_i - Y_{i-1} \cong X_i - X_{i-1}$. So the U_i's usually have smaller variances than the X_i's.
- The predictive, differential style in DPCM is embedded in many other source coding techniques.

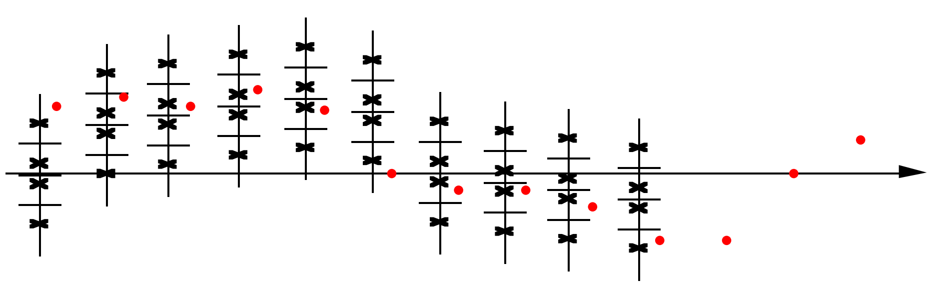
DPCM: Adaptive Quantization Viewpoint

Recall: $Y_i = Y_{i-1} + Q(X_i - Y_{i-1})$

$\Rightarrow$ X_i is quantized with shifted quantizer : $Y_i = b + Q(X_i - b)$, where $b = Y_{i-1}$

Quantizer Q: Shifted quantizer $b+Q(x-b)$:





DPCM is a kind of **backward adaptive** quantizer. The quantizer to use on X_i is determined by the reconstructions of past X's.

It is critical that the rule for determining how X_i is to be quantized depends only on what the decoder knows. Backward adaptation is a common theme in quantization.

Another example of backward adaption

Quantize X_i with quantizer Q scaled by Y_{i-1}, i.e. with $Y_i = Y_{i-1} Q(X_i / Y_{i-1})$.

Forward Adaptation

Example of Forward Adaptation:

Given $X_1,\dots,X_N$ (e.g. N = 100 or 1000) and a collection of scalar quantizers $\{Q_1, Q_2, ..., Q_S\}$ each with rate R_o, choose Q_k that gives least distortion on $X_1,\dots,X_N$, i.e. that minimizes $\sum_{i=1}^{N} (X_i-Q_k(X_i))^2$ (this is not avg. distortion!).

Binary output:

- $\lceil \log_2 S \rceil$ bits describing index s of the chosen quantizer.
- $e_s(X_1),\dots,e_s(X_N)$

$\Rightarrow$ rate $R = R_o + \dfrac{\lceil \log_2 K \rceil}{N}$ bits/samples

Implementation: (a) Try each quantizer to see which is best, or
(b) Use a selection rule, e.g. if quantizers are shifted versions of some basic quantizer, choose the quantizer whose "middle" is closest to $\frac{1}{K}\sum_{i=1}^{K} X_i$.

There are many possibilities for the Q_k's. Different shifts, different scales, different point densities. One might even redesign the quantizer based on $X_1,\dots,X_N$, and then lossy encode the quantizers.

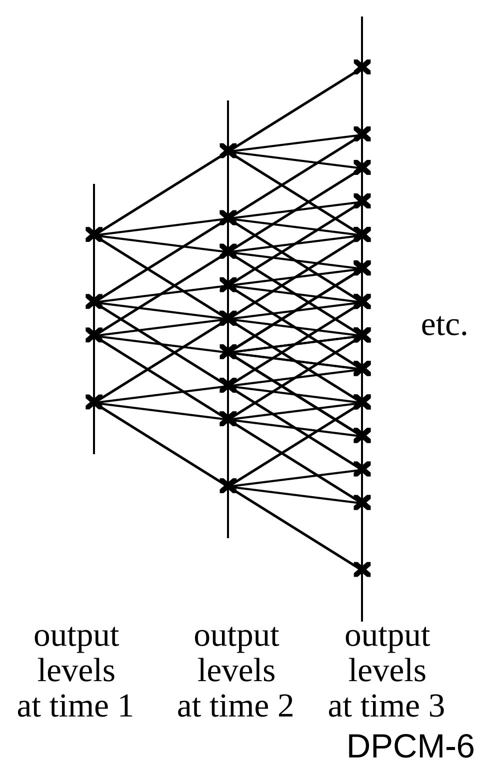
Choosing N large reduces $\lceil \log_2 K \rceil/N$, but decreases the benefits of forward adaptation.

Which is better, backward or forward adaption? No conclusive answer.

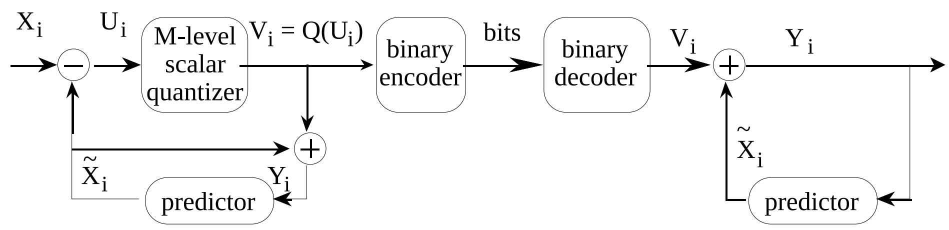
DPCM: Third Viewpoint -- Encoder Controls Decoder

- Recall: $Y_i = Y_{i-1} + V_j$ where $V_j \in C = \{w_1,\dots,w_M\}$ are scalar levels. $Y_0 = 0$.
- Knowing X_i and Y_{i-1}, encoder chooses V_j to be the level w_j that makes $Y_{i-1}+w_j$ closest to X_i; i.e encoder causes (controls) decoder to make the best possible output.
- This is a greedy encoding algorithm.
- Looking ahead might do better.
- Consider the **tree structure** of DPCM decoder outputs, i.e. of reconstruction sequences shown to the right.
- Tree-encoding: Given N, search the tree for the sequence of N quantization levels in the tree that is closest to the source sequence. Send the indices of the levels in the chosen sequence. Use the usual DPCM decoder.

  This method has never worked its way into practical use. Instead people use the usual DPCM greedy encoder.



DPCM WITH IMPROVED PREDICTION



$\tilde{X}_i = g(Y_{i-1}, Y_{i-2}, \ldots)$

- Example: $\tilde{X}_i = a\, Y_{i-1}$
- Most common: Nth-order linear prediction -- $\tilde{X}_i = \sum_{j=1}^{N} a_j\, Y_{i-j}$,

  where N = *order* of predictor and $a_1,\ldots,a_N$ are the *prediction coefficients*.
- Ideally ... choose a_i's to minimize $E\, U_i^2 = E\,(X_i - \sum_{j=1}^{N} a_j Y_{i-j})^2$,

  But ... this requires knowing the correlations $E\, X_i Y_{i-1},\ldots,E\, X_i Y_{i-N}$, which in turn depend on the a_i's. Therefore, a_i's are usually chosen to minimize

  $$E\,(X_i - \sum_{j=1}^{N} a_j X_{i-j})^2 = \text{MSPE} = \text{mean square prediction error.}$$

  That is, to predict X_i from $Y_{i-1},\ldots,Y_{i-N}$, use the predictor that would minimize the MSPE if we were predicting X_i from $X_{i-1},\ldots,X_{i-N}$. This is a reasonable because if the DPCM system is working well, then $X_j \cong Y_j$, $j<i$, and so

  $$E\,(X_i - \sum_{j=1}^{N} a_j Y_{i-j})^2 \cong E\,(X_i - \sum_{j=1}^{N} a_j X_{i-j})^2$$

MINIMUM MEAN-SQUARED ERROR LINEAR PREDICTION

Assume X is a zero-mean, wide-sense stationary random process with autocorrelation function $R_X(k) \equiv E X_i X_{i+k}$.

Linear prediction theory shows that the Nth-order linear predictor of X_i based on $X_{i-1},\ldots,X_{i-N}$ that minimizes MSPE has coefficients $\underline{a} = (a_1,\ldots,a_N)^t$ given by

$$\underline{a} = K^{-1}\underline{r}$$

where K is the $N \times N$ correlation matrix of $X_1,\ldots,X_N$, i.e.

$$K_{ij} = E\, X_i X_j = R_X(|i-j|),$$

K^{-1} is its inverse,

$$\underline{r} = \big(R_X(1),\ldots,R_X(N)\big)^t.$$

It can be shown that if K is not invertible, then the random process X is *deterministic* in the sense that there are coefficients $b_1,\ldots,b_{N-1}$ such that

$$X_i = b_1 X_{i-1} + \ldots + b_{N-1} X_{i-N+1}.$$

The resulting minimum mean-square prediction error is

$$\text{MMSPE}_N = \sigma^2 - \underline{a}^t\underline{r} = \sigma^2 - \underline{r}^t K^{-1}\underline{r}\,,$$

and the *prediction gain* is

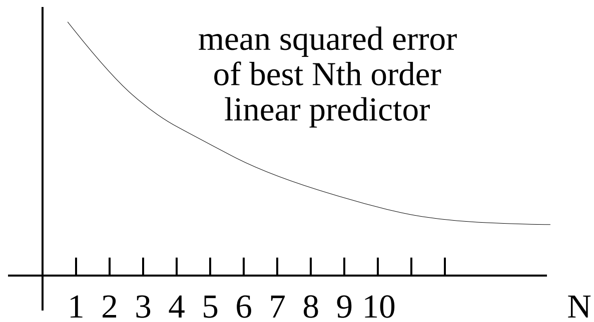
$$G_N = 10 \log_{10} \frac{\sigma^2}{\text{MMSPE}_N}$$

It can also be shown that

$$MMSPE_N = \frac{|K^{(N+1)}|}{|K^{(N)}|}$$

where $K^{(N)}$ and $K^{(N+1)}$ denote, respectively, the N×N and (N+1)×(N+1) covariance matrices of X, and |K| denotes the determinant of the matrix K.

For a typical random process, $MMSEP_N$ decreases with N to some limit as illustrated below

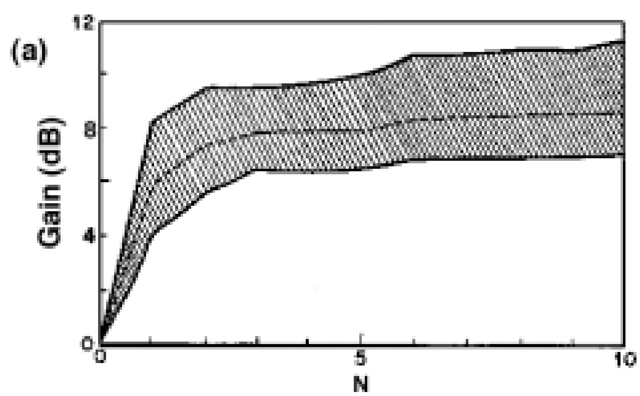


and the prediction gains G_N increase with N to some limit, as illustrated in the example below.

TYPICAL PREDICTION GAINS FOR SPEECH

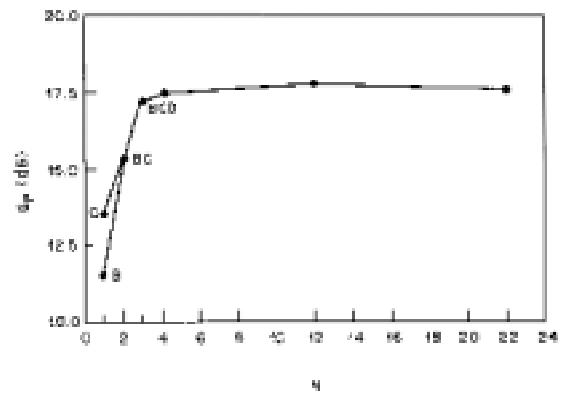
Jayant & Noll, p. 271

For each of several speakers, $R_X(k)$ was emprically estimated and the prediction gains were computed for various N's. The range of G_N values found is shown below. The speech was lowpass filtered, probably to 3kHz.



TYPICAL PREDICTION GAINS FOR IMAGES

Prediction gains in interframe image coding from Jayant and Noll, Fig 6.8, p. 272.



B -- prediction from pixel immediately to the left

$\tilde{X} = .965\, X_B$

C -- prediction from corresponding pixel in previous frame

$\tilde{X} = .977\, X_C$

BC -- prediction from both pixels

$\tilde{X} = .379\, X_B + .617\, X_C$

BCD -- prediction from the two aforementioned pixels plus the pixel to the left of the corresponding in the previous frame

$\tilde{X} = .746\, X_B + .825\, X_C - .594\, X_D$

For larger values of N, the predictor is based on N pixels from the present and the previous frame.

ASYMPTOTIC LIMIT OF $MMSPE_N$

For a wide sense stationary random process, it can be shown that as $N \to \infty$, $MMSPE_N$ decreases to

$MMSPE_\infty$ = "one-step prediction error"

$$= \exp\left\{ \frac{1}{2\pi} \int_{-\pi}^{\pi} \ln S_X(\omega)\, d\omega \right\}$$

where $S_X(\omega)$ is the power spectral density of X, i.e.

$S_X(\omega)$ = discrete-time Fourier transform of $R_X(k)$

$$= \sum_{k=-\infty}^{\infty} R_X(k)\, e^{-jk\omega}$$

For an Nth-order AR process, $MMSPE_\infty = MMSPE_N$.

COMPLEXITY OF DPCM

arithmetic complexity (storage is usually small)

	scalar quantization	prediction	
encoding	R	2N+1	ops/sample
decoding	0	2N+1	ops/sample

THE ORIGINS OF THE "DPCM" NAME

The letters "PCM" comes from "Pulse code modulation", which is a modulation technique for transmitting analog sources such as speech. In PCM, an analog source, such as speech, is sampled; the samples are quantized; fixed-length binary codewords are produced; and each bit of each binary codeword determines which of two specific pulses are sent (e.g. one pulse might be the negative of the other, or the pulse might be sinusoidal with different frequencies). Thus PCM is a modulation technique for transmitting analog sources that competes with AM, FM and various other forms of pulse modulation. Invented in the 1940's, "PCM" is now viewed as three systems: sampler, quantizer and digital modulator. However, the quantizer by itself is often referred to as PCM. DPCM was originally proposed as an improved DPCM type modulator consisting of a sampler, a DPCM quantizer as we know it, and a digital modulator. Nowadays DPCM usually refers just to the quantization part of the system.

Predictive Coding

DPCM is also considered to be a kind of *predictive* coding.

Delta Modulation

The special case of DPCM where the quantizer has just two levels: $+\Delta$, $-\Delta$.

Delta modulation is used when the source is highly correlated, for example speech coding with a high sampling rate.

Speech Coding

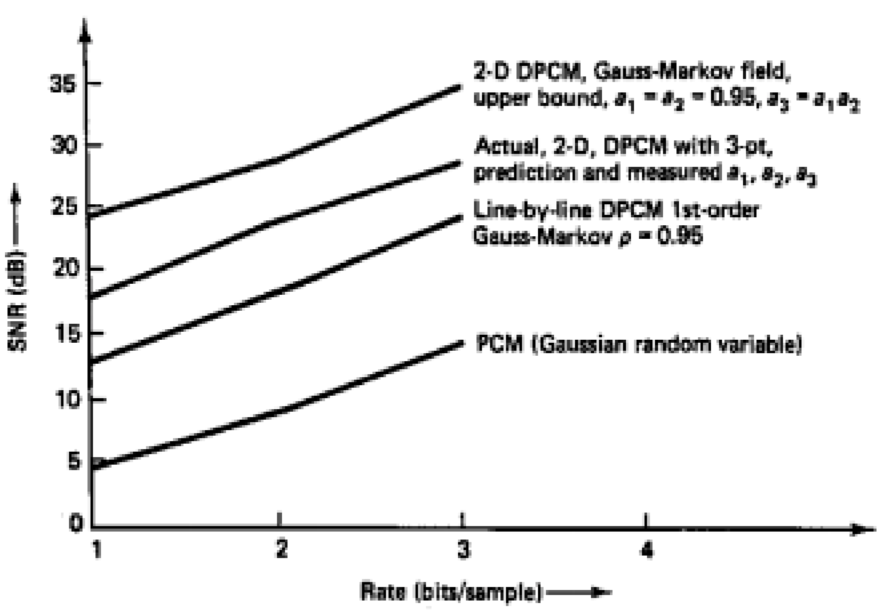
Adaptive versions of DPCM and Delta-Modulation have been standardized for use in the telephone industry for digitizing telephone speech. The speech is bandlimited to 3 kHz before sampling at 8 khz. However, this kind of speech coding is no longer considered state-of-the-art.

Examples of DPCM Image Coding

Example: Prediction for the present pixel X_{ij}

$$\tilde{X}_{ij} = a_1 Y_{i,j-1} + a_2 Y_{i-1,j} + a_3 Y_{i-1,j-1}$$

$Y_{i-1,j-1}$	$Y_{i-1,j}$
$Y_{i,j-1}$	$X_{i,j}$



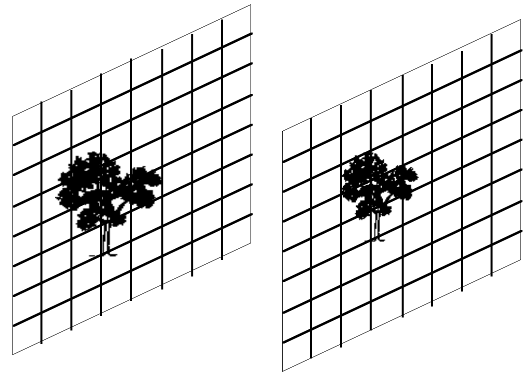
(from R. Jain, Fund'ls of Image Proc, p. 493)

All but second curve from top represents performance predicted with predictor matched to the stated source. For the second curve from the top, the values for a_1, a_2 and a_3 were designed for the actual measured correlations for a test set of images.

DPCM has been extensively studied for image coding. But it is not often used today. Transform coding is more common.

DPCM IN VIDEO CODING

Though not usually called DPCM, the most commonly used methods of video coding use a form of DPCM, e.g. MPEG, H.26X, HDTV, satellite "Direct TV", DVD.



Prediction is done on frame-by-frame basis, with prediction of a frame being the decoded reconstruction of previous frame, or a "motion-compensated" version thereof. In latter case, coder has a forward-adaptive component in addition to the backward adaption.

Example: (Netravali & Haskell, Digital Pictures, p. 331)

Prediction Coefficients				
left pixel, curr. frame	left pixel prev. frame	same pixel prev frame	MSPE	Pred. gain[1]
1			53.1	12.7 dB
1	-1/2	1/2	29.8	15.3 dB
3/4	-1/2	3/4	27.9	15.5 dB
7/8	-5/8	3/4	26.3	15.8 dB

[1]Based on educated guess that $\sigma^2 = 1000$.

DESIGN AND PERFORMANCE -- HIGH RESOLUTION ANALYSIS

Warning: Though this analysis is almost universally accepted, it has never been satisfactorily proven to be correct.

Assume source is stationary, zero-mean random process.

It can be shown that under ordinary conditions $\{(X_i,U_i,V_i,Y_i)\}$ is asymptotically stationary. So we assume it is stationary. Therefore,

$$D = E(X_i-Y_i)^2 = E(U_i-Q(U_i))^2 \qquad \text{(same for all i)}$$

Key assumption:

When R is large and quantizer is well designed

$$D \cong E(\tilde{U}_i-Q(\tilde{U}_i))^2$$

where $\tilde{U}_i = X_i - g(X_{i-1},X_{i-2}, \ldots)$. $\tilde{U}_i \neq U_i$ but $\tilde{U}_i \cong U_i$

Note: $\tilde{U}$ is a stationary random process.

Key assumption implies

least possible distortion of DPCM with rate R

$\cong$ least possible distortion of SQ with rate R encoding $\tilde{U}$, minimized over choice of predictor g

That is,

$$\delta_{dpcm}(R) \cong \min_g \delta_{\tilde{U},sq}(R) \quad \text{(true for FLC or VLC)}$$

$$\cong \min_g \frac{1}{12} \sigma^2_{\tilde{U}} \alpha_{\tilde{U}} 2^{-2R}$$

where $\alpha_{\tilde{U}} = \begin{cases} \beta_{\tilde{U}}, & \text{for FLC} \\ \eta_{\tilde{U}}, & \text{for VLC} \end{cases}$

Conclusions:

A good (but not necessarily optimal) way to design DPCM is to

(a) Choose g to minimize $\sigma^2_{\tilde{U}} \alpha_{\tilde{U}}$

(b) Choose q to achieve $\delta_{\tilde{U},sq}(R)$

Common situation:

The pdf of $\tilde{U}$ is similar to that of X, which implies, $\beta_{\tilde{U}} \cong \beta_X$ and $\eta_{\tilde{U}} \cong \eta_X$. Example: exact equalities hold if X is Gaussian random process. In such cases, one need only

(a') Choose g to minimize $\sigma^2_{\tilde{U}}$

(b) Choose q to achieve $\delta_{\tilde{U},sq}(R)$

It follows that

$$\delta_{dpcm}(R) \cong \min_g \frac{1}{12} \sigma^2_{\tilde{U}} \alpha_X 2^{-2R} = \min_g \sigma^2_{\tilde{U}} \frac{1}{12} \alpha_X 2^{-2R} \cong \frac{MMSPE}{\sigma^2} \delta_{sq}(R)$$

$$\Rightarrow S_{dpcm}(R) \cong S_{sq}(R) + G_N \text{ dB}$$

where $G_N = 10 \log_{10} \frac{\sigma^2}{MMSPE}$ dB = prediction gain

That is, the gain of DPCM over SQ approximately equals the prediction gain.