Goal: An effective low complexity quantizer for correlated sources based on scalar quantization.

Consider quantizing a highly correlated, i.e. slowly varying, stationary source, e.g. AR Gauss, \( \rho = 0.95 \), \( \mu = 0 \), \( \sigma^2 = 3.2 \), for which a portion of a typical sample function is shown to the right.

For future comparison, optimal scalar quantization with \( R = 2 \) yields \( D = 0.37 \).

Notice that most samples are similar to their predecessors. For example, successive sample differences, as shown to the right, are Gaussian with mean zero and very small variance.

\[
E(X_i - X_{i-1})^2 = EX_i^2 - 2EX_iX_{i-1} + EX_{i-1}^2 = 2\sigma^2 - 2\rho \sigma^2 = 0.1 \sigma^2 = 0.32 \ll \sigma^2 = 3.2
\]

This suggests trying to design an encoder that quantizes successive sample differences. In comparison, to direct scalar quantization of \( X \), scalar quantization of \( (X_i - X_{i-1}) \) with \( R = 2 \) yields \( D = 0.37 \times 0.32/3.2 = 0.037 \ll 0.37 \).

## CODING SAMPLE DIFFERENCES

### ATTEMPT 1: NAIVE DIFFERENTIAL CODING

- Assume \( X_0 = Y_0 = 0 \)
- Rate = rate of scalar quantizer.
- Key equation: \( Y_i = Y_{i-1} + Q(X_i - X_{i-1}) \)
- Consider the errors in the reproductions

\[
\begin{align*}
X_1 - Y_1 &= X_1 - Q(X_1) \\
X_2 - Y_2 &= X_1 + U_2 - (Y_1 + V_2) = (X_1 - Y_1) + (U_2 - V_2) \\
X_3 - Y_3 &= X_2 + U_3 - (Y_2 + V_3) = (X_2 - Y_2) + (U_3 - V_3) \\
&= (X_1 - Y_1) + (U_2 - V_2) + (U_3 - V_3) \\
X_i - Y_i &= X_{i-1} + U_i - (Y_{i-1} + V_i) = (X_{i-1} - Y_{i-1}) + (U_i - V_i) \\
&= (X_1 - Y_1) + (U_2 - V_2) + (U_3 - V_3) + \ldots + (U_i - V_i)
\end{align*}
\]

- Errors accumulate. This method does not work.
**ATTEMPT 2: DIFFERENTIAL PULSE CODE MODULATION (DPCM)**

Traditional block diagram:

\[
\begin{align*}
X_i & \quad U_i \\
\rightarrow & \quad \text{scalar} \\
\quad & \quad \text{quantizer} \quad V_i = Q(U_i) \\
\quad & \quad \text{binary} \\
\quad & \quad \text{encoder} \\
\quad & \quad \text{bits} \\
\quad & \quad \text{binary} \\
\quad & \quad \text{decoder} \\
\quad & \quad \text{bits} \\
\quad & \quad \text{Y_i} \\
\quad & \quad \text{Y_i-1} \\
\quad & \quad \text{delay} \\
\quad & \quad \text{Y_i} \\
\end{align*}
\]

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

\[
(X_i - Y_i = (Y_{i-1} + U_i) - (Y_{i-1} + Q(U_i)) = U_i - Q(U_i))
\]

Notes:

- Rate = rate of scalar quantizer.
- \( U_i \neq X_i-X_{i-1} \), as with naive encoder. However, \( U_i = X_i-Y_{i-1} \equiv X_i-X_{i-1} \).
  This implies \( \sigma_u^2 << \sigma_x^2 \), so quantizing \( U \) gives less MSE than quantizing \( X \).
  Since \( X_i-Y_i = U_i - Q(U_i) \), the small MSE for \( U \) carries over to \( X \).
- Therefore DPCM works well.
- The predictive/differential coding style in DPCM also appears in other lossy coding methods, e.g. interframe video coding.

**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 \( Q \):

[Diagram showing quantizer]

Shifted quantizer \( b + Q(x-b) \):

[Diagram showing shifted quantizer]

DPCM-3
BACKWARD ADAPTIVE QUANTIZATION

- DPCM is a kind of backward adaptive quantizer. The quantizer used on $X_i$ is adapted based on the reconstructions of past $X$'s.
- It is critical that the rule for adapting 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.
  $$Y_i = Y_{i-1} \cdot Q(X_i/Y_{i-1}).$$

FORWARD ADAPTATION (A DIGRESSION)

Example:

Encoding: Given $X_1, \ldots, X_N$ (e.g. $N = 100$ or $1000$) and scalar quantizers $\{Q_1, Q_2, \ldots, Q_S\}$ each with rate $R_0$, choose $Q_s$ that gives least distortion on $X_1, \ldots, X_N$, i.e. that minimizes $\sum_{i=1}^{N} (X_i - Q_s(X_i))^2$ (This is not expected distortion!).

Encoder output: two components
  a. $\lceil \log_2 S \rceil$ bits describing index $s$ of the chosen quantizer.
  b. $e_s(X_1), \ldots, e_s(X_N)$, i.e., binary output from applying $Q_s$ to each $X_i$

Rate: $R = R_0 + \frac{\lceil \log_2 S \rceil}{N}$ bits/samples

Implementation approaches:
  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 $\frac{1}{N} \sum_{i=1}^{N} X_i$.

Design:

Many possibilities for the $Q_s$'s. Different shifts, scales, point densities, ... . One might even design a quantizer from scratch based on $X_1, \ldots, X_N$, and then use the quantizer and lossy encode the quantizer parameters.

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

Which is better, forward or backward adaption? No conclusive answer.
DPCM: THIRD VIEWPOINT -- ENCODER CONTROLS DECODER

- Recall: \( Y_i = Y_{i-1} + V_j \) where \( V_j \in C = \{w_1, \ldots, w_M\} \) 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

- Example of predictor: \( \tilde{X}_i = a Y_{i-1} \)
- Most common: Nth-order linear prediction -- \( \tilde{X}_i = \sum_{j=1}^{N} a_j Y_{i-j} \), where \( N = \text{order} \) of predictor and \( a_1, \ldots, a_N \) are the prediction coefficients.
- Assume \( Y_0 = 0 \)
- Key Equations:
  - \( Y_i = Y_{i-1} + Q(X_i - \tilde{X}_{i-1}) \) 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

The system shown above, with finite-order linear prediction, is the usual kind of DPCM. We will presume this in the subsequent discussion.
PERFORMANCE

Rate:
The rate of DPCM is the rate of the scalar quantizer.
The scalar quantizer could be of the fixed-rate or variable-rate type.
DPCM is fixed rate or variable rate, respectively.
If the scalar quantizer is fixed rate with \( M \) levels, then the rate of DPCM is
\[
R = \log_2 M
\]
If the scalar quantizer is variable-rate, then we need to discuss the nonstationarity of DPCM before giving a formula for rate.

Distortion:
Since \((X_i - Y_i) = (U_i - Q(U_i))\), the distortion of DPCM is the distortion of the scalar quantizer operating on \( U \). However, before giving a formula for the distortion of DPCM, we need to discuss its nonstationarity.

The nonstationarity of DPCM:
As will be illustrated below, the \( U_i \)'s are not identical random variables, even when the source to which DPCM is applied is stationary.
Therefore, the mean squared error, \( E((X_i - Y_i)^2) = E((U_i - Q(U_i))^2) \), changes with \( i \).
Moreover, when variable-rate coding is used, the average length of the codewords, \( E(l(U_i)) \), changes with \( i \).

Example illustrating nonstationarity:
If the predictor produces \( \tilde{X}_i = aY_{i-1} \), then
\[
\begin{align*}
U_1 &= X_1 - aY_1 = X_1, \\
Y_1 &= Q(U_1) + aY_0 = Q(X_1) \\
U_2 &= X_2 - aY_2 = X_2 - aQ(X_1), \\
Y_2 &= Q(U_2) + aY_1 = Q(X_2 - aQ(X_1)) + aQ(X_1) \\
U_3 &= X_3 - aY_3 = X_3 - aQ(X_2 - aQ(X_1)) + aQ(X_1) \\
U_4 &= \ldots
\end{align*}
\]
We can see from the above that the probability distribution of \( U_i \) changes with \( i \).

The asymptotic stationarity of DPCM:
It can be shown that, under ordinary conditions, when the source \( X \) to which DPCM is applied is stationary, the \( U \) random process is asymptotically stationary. This means, for example, that the density of \( U_i \) converges to some limiting steady-state density \( f_U(u) \) as \( i \to \infty \).
It can also be shown that \( Y_i \) and \( \tilde{X}_i \) are asymptotically stationary.
Moreover, taken jointly, \((X_i, Y_i, \tilde{X}_i, U_i, V_i)\) is asymptotically stationary.

We can now give formulas for the rate of DPCM with variable-rate coding and the distortion of DPCM.
Rate with variable-rate coding:

When variable-rate DPCM is applied to a stationary source, the rate is

\[
R = \lim_{n \to \infty} \frac{1}{n} \sum_{i=1}^{n} E l(Q(U_i)) = \lim_{n \to \infty} E l(Q(U_i)) = E l(Q(U))
\]

\[
= \int_{-\infty}^{\infty} l(Q(u)) f_U(u) \, du
\]

where \( l(v) \) denotes the length of the codeword assigned to quantization level \( v \), and \( f_U(u) \) is the asymptotic steady-state density of the \( U_i \)'s.

Distortion:

When DPCM is applied to a stationary source, the distortion is

\[
D = \lim_{n \to \infty} \frac{1}{n} \sum_{i=1}^{n} E(X_i - Y_i)^2 = \lim_{n \to \infty} \frac{1}{n} \sum_{i=1}^{n} E(U_i - Q(U_i))^2 = \lim_{n \to \infty} E(U_i - Q(U_i))^2
\]

\[
= \int_{-\infty}^{\infty} (u - Q(u))^2 f_U(u) \, du
\]

where \( f_U(u) \) is the steady state density of \( U \).

**Complexity of DPCM**

- **Storage:** usually quite small

- **Arithmetic complexity:**

<table>
<thead>
<tr>
<th></th>
<th>scalar quantization</th>
<th>prediction</th>
</tr>
</thead>
<tbody>
<tr>
<td>encoding</td>
<td>( R )</td>
<td>2N+1 ops/sample</td>
</tr>
<tr>
<td>decoding</td>
<td>0</td>
<td>2N+1 ops/sample</td>
</tr>
</tbody>
</table>

- The scalar quantization complexity assumed tree-structured search.
DESIGN OF DPCM

• Design involves the choice of the predictor and the quantizer

• Design of the predictor:

A first thought is to choose the $a_i$'s to minimize

$$E U_i^2 = E \left( X_i - \sum_{j=1}^{N} a_j Y_{i-j} \right)^2$$

(Here we assume that $i$ is so large that all random variables are characterized by their steady-state probability distributions.)

This minimization requires knowledge of the correlations $E X_i Y_{i-1}, \ldots, E X_i Y_{i-N}$, which in turn depend on the $a_i$'s. Thus, we have a chicken-and-egg problem. Therefore, the $a_i$'s are usually chosen to minimize

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

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 reasonable, because if the DPCM system is working well, then $X_j \cong Y_j$, $j<i$, and so

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

This is, essentially, a high-resolution approximation.

• Design of the quantizer:

A first thought is to design the quantizer to be optimal for the $U_i$'s, i.e. for the limiting density $f_U(u)$. However, we have another chicken-and-egg problem in that the density $f_U(u)$ depends on the quantizer. Therefore, the quantizer is usually designed to be optimal for the random variables

$$V_i = X_i - \sum_{j=1}^{N} a_{ij} X_{i-j}$$

(Again we assume that $i$ is so large that all random variables are characterized by their steady-state probability distributions.)

As before, this is a reasonable because if the DPCM system is working well, then $X_j \cong Y_j$, $j<i$, and so

$$V_i = X_i - \sum_{j=1}^{N} a_{ij} X_{i-j} \cong X_i - \sum_{j=1}^{N} a_{ij} Y_{i-j} = U_i$$

Again, this is essentially a high-resolution approximation.

• Designing the predictor and quantizer as stated above is not optimal. But it has stood the test of time, i.e. it seems to be close to optimal.
• One might try to iteratively design a better quantizer with the following algorithm
  a. Initialize the quantizer by designing it as described above
  b. Compute or measure the steady-state density $f_U(u)$ of $U$
  c. Redesign the quantizer to be optimal for $f_U(u)$.
  d. Go to b.

This seems to produce slightly better quantizers. However, it has not been proven that the distortion of DPCM decreases with each iteration.

• One could use a similar approach to iteratively design a better predictor than the one described previously.
  a. Initialize with the predictor described previously.
  b. Measure the relevant steady-state statistics and joint statistics of the $X_i$’s and $Y_i$’s.
  c. Redesign the predictor to be optimal for those statistics.
  d. Go to b.

This seems to produce slightly better predictors. However, it has not been proven that the distortion of DPCM decreases with each iteration.

FURTHER NOTES:

• It is not clear that for an optimal DPCM system, $Q$ is the optimal quantizer for the steady-state probability distribution of $U$. From now on, when we refer to $U$ or its probability distribution, we refer to its steady-state distribution.

One might think that if $Q$ weren’t optimal for $U$, then one could improve DPCM by making it optimal for $U$. However, changing $Q$, changes the pdf of $U$, so it’s not clear that changing $Q$ would help.

• Similarly, it is not clear that for an optimal DPCM system, $g$ is the optimal predictor for $X_i$ based on $Y_{i-N}, ..., Y_{i-1}$.

One might think that if $g$ weren’t optimal for $X$, then one could improve DPCM by making it optimal. However, changing $g$, changes the moments of the $Y$’s, so it’s not clear that changing $g$ would help.

• Because the distortion of DPCM is $E(U-Q(U))^2$ and because the steady-state density of $U$ is not easy to compute, there are iterative algorithms for computing actual distortion.

• Additional backward adaptivity can be added to DPCM in various ways, e.g. by adapting the quantizer or the predictor.

• The book by Jayant and Noll is an excellent reference on fancier versions of DPCM.
HIGH RESOLUTION ANALYSIS OF DPCM

- 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.
- As mentioned earlier, it can be shown that under ordinary conditions $(X_i,Y_i,\bar{X}_i,U_i,V_i)$ is asymptotically stationary. So we assume it is stationary. Therefore,

$$D = E(X_i-Y_i)^2 = E(U_i-Q(U_i))^2$$

(same for all $i$)

- Key assumption:

  When $R$ is large and quantizer is well designed

$$D \overset{\overset{\sim}{\sim}}{=} E(\tilde{U}_i-Q(\tilde{U}_i))^2$$

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

- Notes:

  $\tilde{U}$ is a stationary, rather than just an asymptotically stationary, random process.

  $\tilde{U}_i \overset{\overset{\sim}{\sim}}{=} U_i$ but $\tilde{U}_i \neq U_i$.

- The key assumption implies

  least possible distortion of DPCM with large rate $R$

  $\overset{\overset{\sim}{\sim}}{=} \text{least possible distortion of SQ with rate } R \text{ applied to } \tilde{U}$,

  minimized over choice of predictor $g$

It follows that for FLC or VLC, and large $R$

$$\delta_{\text{dpcm}}(R) \overset{\overset{\sim}{\sim}}{=} \min_g \delta_{\tilde{U},\text{sq}}(R) \overset{\overset{\sim}{\sim}}{=} \frac{1}{T^2} \left( \min_g \sigma_{\tilde{U}}^2 \alpha_{\tilde{U}} \right) 2^{-2R}$$

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 $\sigma_{\tilde{U}}^2 \alpha_{\tilde{U}}$

  (b) Choose $Q$ to achieve $\delta_{\tilde{U},\text{sq}}(R)$
• Common situation:
The pdf of $\tilde{U}$ is similar to a scaled version of that of $X$, which implies, $\beta \tilde{U} = \beta X$ and $\eta \tilde{U} = \eta X$. Indeed, if $X$ is Gaussian source, then so is $\tilde{U}$, and these approximations become exact. In this case (a) becomes

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

The resulting MSPE is called the N-step prediction error, and is denoted $\mathcal{M}_N = \min_g \sigma^2_{\tilde{U}}$

We now have the following.

When $R$ is large and the pdf of $\tilde{U}$ is similar to a scaled version of that of $X$, e.g. when $X$ is Gaussian, the OPTA function of DPCM with Nth-order prediction is

$$\delta_{\text{dpcm}}(N, R) \equiv \frac{1}{12} \mathcal{M}_N \alpha X 2^{-2R} \equiv \frac{\mathcal{M}_N}{\sigma^2} \delta_{\text{sq}}(R) = \frac{1}{G_N} \delta_{\text{sq}}(R)$$

$$S_{\text{dpcm}}(N, R) \equiv S_{\text{sq}}(R) + G_N \text{ dB},$$

where

$$G_N = 10 \log_{10} \frac{\sigma^2}{\mathcal{M}_N} \text{ dB} = \text{pred'n gain}$$

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

---

**REVIEW OF MINIMUM MEAN-SQUARED ERROR LINEAR ESTIMATION**

• Consider the task of linearly estimating random variable $Y$ from random variables $X_1,...,X_N$ with minimum mean squared error. That is, we wish to choose coefficients $a_1,...,a_N$ such that the estimate

$$\hat{Y} = \sum_{i=1}^{N} a_i X_i$$

has the smallest possible mean squared estimation error

$$E(Y - \hat{Y})^2 = E \left( Y - \sum_{i=1}^{N} a_i X_i \right)^2$$

• Optimal linear estimators.

We will now show how to choose the $a_i$'s to minimize MSPE.

Let us assume that $Y$ and the $X_i$'s have zero mean. If not, we could subtract their means to obtain zero mean random variables, apply optimal linear estimation, and then add the mean of $Y$ to the estimate of $Y-EY$. Equivalently, we could optimize an estimator of the form $\sum_{i=1}^{N} a_i X_i$, which is called affine.

We consider $\mathbf{X} = (X_1,...,X_N)^t$ and $\mathbf{a} = (a_1,...,a_N)^t$ to be column vectors. In this vector notation,

$$\hat{Y} = \mathbf{a}^t \mathbf{X}$$

We assume that the $N \times N$ covariance matrix $\mathbf{K} = [EX_iX_j]$ of $\mathbf{X}$ is known, as is the vector of covariances

$$\mathbf{r} = (EX_1Y, ..., EX_NY)^t$$
ORTHOGONALITY CRITERIA FOR OPTIMALITY

To find the optimal choice of \( a \) we will prove the following.

• Orthogonality Criteria for Linear Estimators

\( a \) is the optimal linear predictor coefficients iff

\[
E (Y-a'\mathbf{X})X_j = 0, \ j = 1, \ldots, N
\]

i.e. iff the estimation error is orthogonal to all observations. The above equation is called the orthogonality criteria for optimality.

Proof:

If statement: Suppose \( a \) satisfies the above, and \( b \) is an arbitrary set of coefficients. Then,

\[
E (Y-b'\mathbf{X})^2 = E \left( (Y-a'\mathbf{X}) + (a'\mathbf{X}-b'\mathbf{X}) \right)^2
\]

\[
= E (Y-a'\mathbf{X})^2 + 2 E (Y-a'\mathbf{X})(a'\mathbf{X}-b'\mathbf{X}) + E (a'\mathbf{X}-b'\mathbf{X})^2
\]

\[
= E (Y-a'\mathbf{X})^2 + 2 \sum_{j=1}^{N} (a_j-b_j)X_j + E (a'\mathbf{X}-b'\mathbf{X})^2
\]

\[
= E (Y-a'\mathbf{X})^2 + 2 \sum_{j=1}^{N} (a_j-b_j)E (Y-a'\mathbf{X})X_j + E (a'\mathbf{X}-b'\mathbf{X})^2
\]

\[
= E (Y-a'\mathbf{X})^2 + E (a'\mathbf{X}-b'\mathbf{X})^2
\]

\[
\geq E (Y-a'\mathbf{X})^2
\]

Since this is true for every choice of \( b \), it must be that \( a \) is optimal. In other words, if \( a \) satisfies the orthogonality criteria, it is optimal.

Only if statement: Suppose \( b \) is optimal and \( a \) satisfies the orthogonality criteria. Then as shown above,

\[
E (Y-b'\mathbf{X})^2 = E (Y-a'\mathbf{X})^2 + E (a'\mathbf{X}-b'\mathbf{X})^2
\]

Since \( b \) is optimal

\[
E (Y-b'\mathbf{X})^2 \leq E (Y-a'\mathbf{X})^2.
\]

If follows from the above two equations that

\[
E (a'\mathbf{X}-b'\mathbf{X})^2 = 0
\]

which implies \( b = a \). That is, if \( b \) is optimal, it satisfies the orthogonality criteria.
From the orthogonality principle we know that if $\mathbf{a}$ are the coefficients of the optimal linear estimator, then

$$E(Y-a^t\mathbf{X})\mathbf{X}_j = 0, \; j = 1,...,N$$

This implies

$$EYY_j = a^tEXX_j, \; j = 1,...,N$$

or equivalently

$$EYY_j = (EX_1X_j,...,X_NX_j)^t \mathbf{a}, \; j = 1,...,N$$

or

$$
\begin{bmatrix}
EYX_1 \\
. \\
. \\
EYX_N
\end{bmatrix} = 
\begin{bmatrix}
EX_1X_1 & ... & EX_1X_N \\
. & ... & . \\
. & ... & . \\
EX_NX_1 & ... & EX_NX_N
\end{bmatrix}
\begin{bmatrix}
a_1 \\
. \\
. \\
a_N
\end{bmatrix}
$$

or

$$\mathbf{r} = \mathbf{K}\mathbf{a}$$

Where $\mathbf{K}$ is the $N\times N$ covariance matrix of the $\mathbf{X}$ and $\mathbf{r} = (EYX_1,...,EYX_N)^t$.

If $\mathbf{K}$ is invertible, then

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

is the vector of optimal linear prediction coefficients.

If $\mathbf{K}$ is not invertible, then it can be shown that at least one of the $X_i$'s is linear combination of the others. Such $X_i$'s can be removed from $\mathbf{X}$ because any estimator that used them could also be computed from the others. With these $X_i$'s reduced, the covariance matrix $\mathbf{K}$ becomes invertible and we use the previous approach.

The resulting Mean Squared Estimation Error is

$$\text{MSE} = EY^2 - \mathbf{r}^t\mathbf{K}^{-1}\mathbf{r}$$

Derivation:

$$\text{MSE} = E(Y-a^tX)^2 = E(Y-a^tX)(Y-a^tX) = E(Y-a^tX)Y + E(Y-a^tX)(-a^tX)$$

$$= E(Y-a^tX)Y \text{ by the orthogonality criteria}$$

$$= EY^2 - a^tEXY$$

$$= EY^2 - a^t\mathbf{r}$$

$$= EY^2 - \mathbf{r}^t\mathbf{K}^{-1}\mathbf{r} \quad \text{since for the optimal estimator } \mathbf{r} = \mathbf{K}\mathbf{a}, \text{ and so } a^t = (\mathbf{K}^{-1})^t = \mathbf{r}^t\mathbf{K}^{-1} \text{ since } \mathbf{K} \text{ and } \mathbf{K}^t \text{ are symmetric}$$
Let us now apply what we've learned to the task of designing the optimal predictor of $X_{N+1}$ from $X_1,\ldots,X_N$.

Let us assume that $X$ is a zero-mean, wide-sense stationary random process with autocorrelation function $R_X(k) \equiv \mathbb{E}[X_iX_{i+k}]$.

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

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

where $\mathbf{K}$ is the $N \times N$ correlation/covariance matrix of $X_1,\ldots,X_N$, i.e.

$$K_{ij} = \mathbb{E}[X_iX_j] = R_X(|i-j|), \quad \mathbf{K}^{-1} \text{ is its inverse, } \mathbf{r} = (R_X(1),\ldots,R_X(N))^t.$$ 

It can be shown that if $\mathbf{K}$ 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 $X_N = b_1X_1 + \ldots + b_{N-1}X_{N-1}$.

When $\mathbf{K}$ is invertible, the resulting minimum mean square prediction error is $\mathcal{M}_N = \sigma^2 - \mathbf{a}^t\mathbf{r} = \sigma^2 - \mathbf{r}^t\mathbf{K}^{-1}\mathbf{r} = \text{"N-step prediction error"}$

and the prediction gain is

$$G_N = 10 \log_{10}\frac{\sigma^2}{\mathcal{M}_N}$$

**Behavior of the N-step Prediction Errors**

- Clearly, $\mathcal{M}_N$ decreases with $N$ to some limit, as illustrated below, and the prediction gains $G_N$ increase with $N$ to some limit.

![Graph showing the decrease of mean squared error with N](image)

- As mentioned in the transform coding section, it can also be shown that

$$\mathcal{M}_N = \frac{|K^{(N+1)}|}{|K^{(N)}|}$$

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

This is derived in supplementary notes.
EXAMPLE: FIRST-ORDER AUTOREGRESSIVE (AR) GAUSSIAN SOURCE.

Assume $X$ is stationary and

$$X_i = \rho X_{i-1} + Z_i \quad (\cdot)$$

where $-1 \leq \rho < 1$, $Z_i$'s are IID Gaussian with zero mean and variances $\sigma_Z^2$. $Z_i$ is independent of $X_j$, $j < i$. ($\rho = .9$ is a typical value.) Then

1) $E X_i = 0$

2) $\sigma_X^2 = E X_i^2 = \frac{\sigma_Z^2}{1-\rho^2} \quad (= 5.26 \sigma_Z^2 \text{ if } \rho = .9)$

3) $R_X(k) = \text{autocorrelation function} = E X_i X_{i+k} = \sigma_X^2 \rho^{|k|}$

$\Rightarrow \rho = \text{correlation coefficient} = \frac{\text{cov}(X_i X_{i+1})}{\sigma_X^2} = \frac{R_X(1)}{\sigma_X^2}$

4) Best predictor for $X_i$ based on $X_{i-1}, X_{i-2}, \ldots, X_{i-N}$ is $\tilde{X}_i = \rho X_{i-1}$ with $\mathcal{M}_N = \sigma_Z^2$

Derivation: It's not easy to use $a = K^{-1}$. However, we may directly use the orthogonality principle, which indicates that $\tilde{X}_i$ is optimal if and only if

$$E (X_i - \tilde{X}_i) X_{i-j}, \text{ for } j = 1, \ldots, N.$$  

We find $E (X_i - \rho X_{i-1}) X_{i-j} = E (\rho X_{i-1} + Z_i - \rho X_{i-1}) X_{i-j} = E Z_i X_{i-j}$

$= E Z_i E X_{i-j} = 0, \text{ since } Z_i \text{ indep. of past } X$'s

It follows from orthog. principal that $\tilde{X}_i = \rho X_{i-1}$ is optimal.

The MSPE is

$$\mathcal{M}_N = E(X_i - \rho X_{i-1})^2 = E(\rho X_{i-1} + Z_i - \rho X_{i-1})^2 = E Z_i^2 = \sigma_Z^2$$

Therefore, for $N \geq 1$, the prediction gain (which is the gain of DPCM over SQ) is

$$G_N = 10 \log_{10} \frac{\sigma_X^2}{\mathcal{M}_N} = 10 \log_{10} \frac{1}{1-\rho^2} \quad (= 7.2 \text{ dB if } \rho = .9)$$
For each of several speakers, $R_X(k)$ was empirically 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
\[ \tilde{X} = 0.965 X_B \]

C -- prediction from corresponding pixel in previous frame
\[ \tilde{X} = 0.977 X_C \]

BC -- prediction from both pixels
\[ \tilde{X} = 0.379 X_B + 0.617 X_C \]

BCD -- prediction from the two aforementioned pixels plus the pixel to the left of the corresponding in the previous frame
\[ \tilde{X} = 0.746 X_B + 0.825 X_C - 0.594 X_D \]

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

---

**ASYMPTOTIC LIMIT OF \( M_N \)**

For a wide sense stationary random process, it can be shown (see Transform Coding Lectures) that as \( N \to \infty \), \( M_N \) decreases to

\[ Q = \lim_{N \to \infty} M_N = "one-step prediction error" \]

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

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

\[ S_X(\omega) = \text{discrete-time Fourier transform of } R_X(k) \]

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

For an Nth-order AR process, \( Q = M_N \) for all \( N \).
**ASYMPTOTIC OPTA OF DPCM**

In view of the fact that the asymptotic limit of $M_N$ is $Q$, we have the following.

When $R$ is large and the pdf of $\tilde{U}$ is similar to a scaled version of that of $X$, e.g. when $X$ is Gaussian, the asymptotic OPTA function of DPCM optimized over $N$ is

$$\delta_{dpcm}(R) \equiv \frac{1}{2} Q \alpha X 2^{-2R} \equiv \frac{Q}{\sigma^2} \delta_{sq}(R) = \frac{1}{G_{\infty}} \delta_{sq}(R)$$

$$S_{dpcm}(R) \equiv S_{sq}(R) + G_{\infty} \text{ dB}$$

where

$$G_{\infty} = 10 \log_{10} \frac{\sigma^2}{Q} \text{ dB} = \text{prediction gain}$$

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

---

**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 is the special case of DPCM where the quantizer has just two levels: $\pm \Delta$.
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 $X_{ij}$

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

(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.
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)

<table>
<thead>
<tr>
<th>Prediction Coefficients</th>
<th>MSPE</th>
<th>Pred. gain$^1$</th>
</tr>
</thead>
<tbody>
<tr>
<td>left pixel, curr. frame</td>
<td>53.1</td>
<td>12.7 dB</td>
</tr>
<tr>
<td>left pixel, prev. frame</td>
<td>29.8</td>
<td>15.3 dB</td>
</tr>
<tr>
<td>same pixel, prev frame</td>
<td>27.9</td>
<td>15.5 dB</td>
</tr>
<tr>
<td>MSPE Pred.</td>
<td>26.3</td>
<td>15.8 dB</td>
</tr>
</tbody>
</table>

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

**Comparison of DPCM and Transform Coding**

Consider a stationary, Gaussian source and large $R$.

For $k$-dimensional transform coding:

$$\delta_{tr}(k,R) \equiv \frac{1}{T_2} |K^{(k)}|^{1/k} \alpha_G 2^{-2R}$$

For DPCM with $k$th-order linear prediction

$$\delta_{dpcm}(k,R) \equiv \frac{1}{T_2} M_k \alpha_G 2^{-2R}$$

where $M_k$ is MSPE of optimal $k$th-order linear prediction for $X_i$ from $X_{i-k}, \ldots, X_{i-1}$.

Fact A: $|K^{(k)}|^{1/k} \geq M_k$.

Proof: $|K^{(k)}|^{1/k} = \left( \sigma_X^2 \prod_{i=1}^{k-1} M_i \right)^{1/k}$ this was proved in the transform coding notes

$$\geq M_k$$ because all terms being averaged are $\geq M_k$

It follows from this fact that DPCM with $k$th-order prediction is at least as good as $k$-dimensional transform coding.
Fact B: \( \lim_{k \to \infty} \mathcal{M}_k = \lim_{k \to \infty} |K^{(k)}|^{1/k} = \exp\left\{ \frac{1}{2\pi} \int_{-\pi}^{\pi} \ln S(\omega) \, d\omega \right\} \)

Proof: This was proved in the transform coding notes.

It follows from this fact that for Gaussian sources, large \( k \) and large \( R \), DPCM and Transform Coding have the same performance, i.e.

\[ \delta_{\text{dpcm}}(\infty, R) = \delta_{\text{tr}}(\infty, R) \]

However, consider the example of a first-order AR Gaussian source. Then, since \( \mathcal{M}_1 = \lim_{k \to \infty} \mathcal{M}_k = Q = \text{"one-step prediction error"} \),

\[ \delta_{\text{dpcm}}(1, R) \cong \delta_{\text{tr}}(\infty, R) \]

Thus, in this case a simple form of DPCM attains performance as good as high dimensional transform coding.

In the Gaussian case, the SNR gain in dB of DPCM with first-order linear prediction over \( k \)-dimensional transform coding is

\[ 10 \log_{10} \frac{\delta_{\text{tr}}(k, R)}{\delta_{\text{dpcm}}(1, R)} = 10 \log_{10} \frac{(1-\rho^2)^{(k-1)/k}}{1-\rho^2} = \frac{10}{k} \log_{10} (1-\rho^2) \]

which is plotted below.

Similarly, for an \( N \)-th-order AR Gaussian source,

\[ \delta_{\text{dpcm}}(N, R) \cong \delta_{\text{tr}}(\infty, R) \]
• Given that DPCM is more efficient for the AR Gauss source, why is
transform coding more commonly used nowadays than DPCM, for example
for image coding?

My feeling is that it because transform coding can more easily be made to
take perceptual criteria into account. For example, with transform image
coding, it is easy to exploit the fact that the eye is more sensitive to errors in
low spatial frequencies than in high spatial frequencies. For example,
JPEG does this by using coarser scalar quantizers for high frequency
coefficients than for low frequency coefficients.