



第五章

预测编码

主要内容

■ 5.1 预测编码基本原理

- 5.1.1 DPCM的基本原理
- 5.1.2 ADPCM

■ 5.2 预测编码在语音编码中的应用

- 5.2.1 语音信号的特性
- 5.2.2 波形编码
- 5.2.3 声码编码/参数编码
- 5.2.4 混合编码

主要内容

- 5.3 预测编码在视频编码中的应用
 - 5.3.1 视频信号中的冗余
 - 5.3.2 帧内预测
 - 5.3.3 帧间预测：运动估计/运动补偿

5.1 预测编码的基本原理

- 信源的相邻符号通常是相关的
- 预测编码：利用信源相邻符号之间的相关性
 - 根据某一模型利用以往的样本值对新样本进行预测，然后将样本的实际值与其预测值相减得到一个误差值，最后对这一误差值进行编码。
 - 如果模型足够好，且样本序列在时间上相关性较强，则误差信号的幅度将远远小于原始信号，从而得到较大的数据压缩。

例：相邻像素之间的相关性



原始图像



水平差图像 $y_n = x_n - x_{n-1}$

值的范围比原图像小得多

例：相邻像素之间的相关性

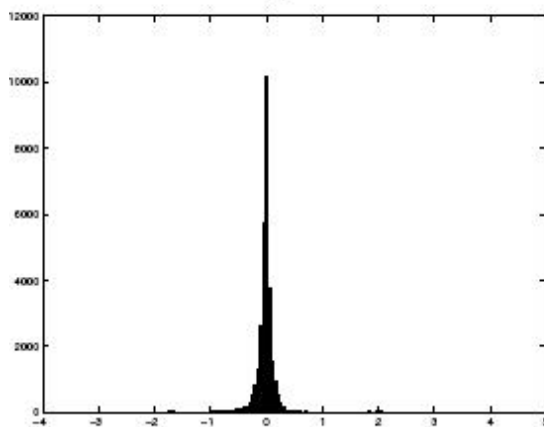
原始图像



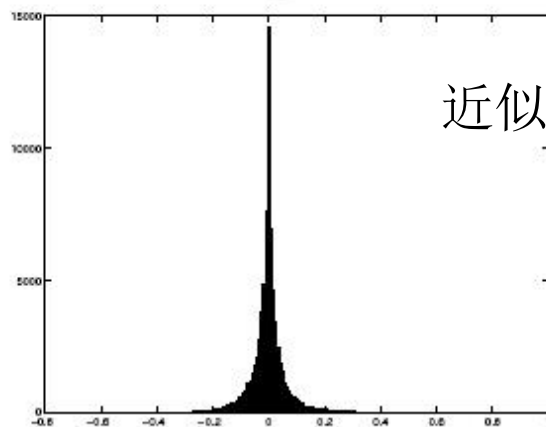
a



b



c

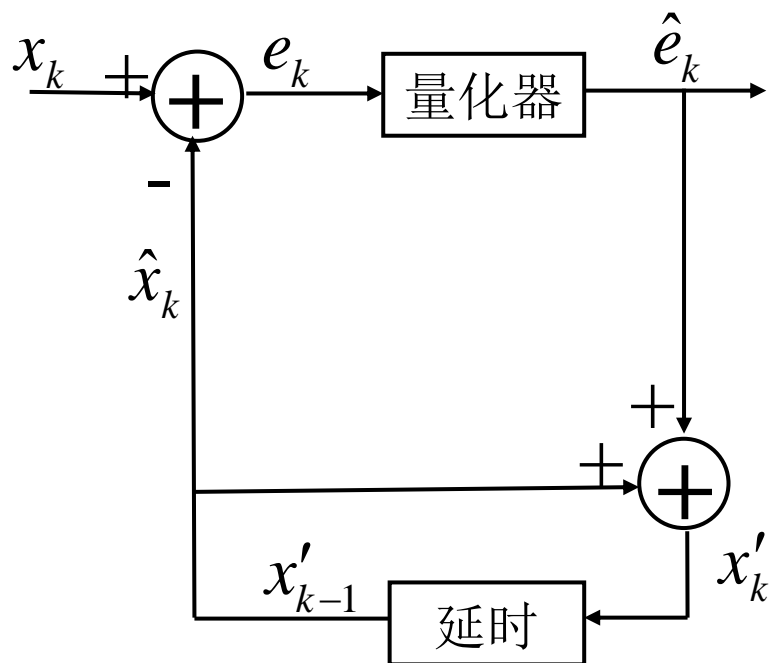


d

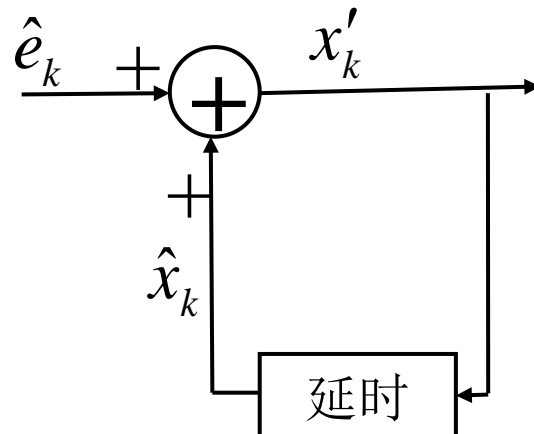
近似为Laplacian分布

5.1.1 DPCM的基本原理

(Differential Pluse Code Modulation)



编码器



解码器

\hat{x}_k 表示对 x_k 的预测，在此 $\hat{x}_k = x'_{k-1}$

$$x_k - x'_k = (\widehat{x_k} + e_k) - (\widehat{x_k} + \hat{e}_k) = e_k - \hat{e}_k$$

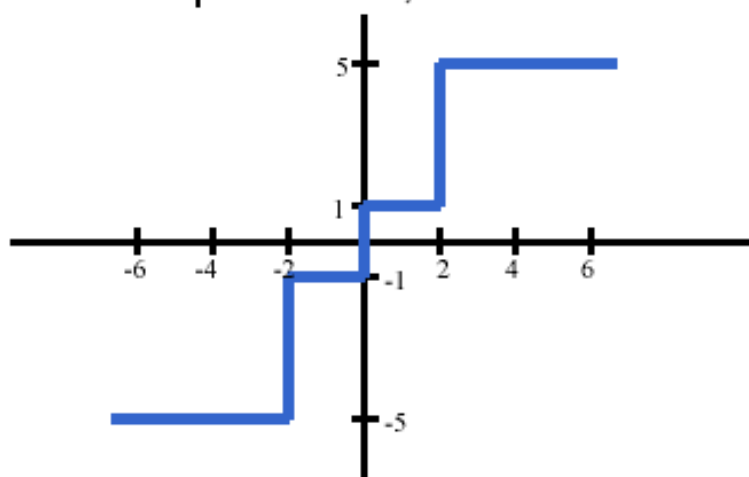
Example

The sequence 100, 102, 120, 120, 120, 118, 116, is to be predictively coded using the prediction rule:

$$\tilde{u}(n) = \tilde{u}(n-1) \quad \text{for DPCM}$$

$$\bar{u}(n) = u(n-1) \quad \text{for the feedforward predictive coder.}$$

Assume a 2-bit quantizer, as shown below, is used,



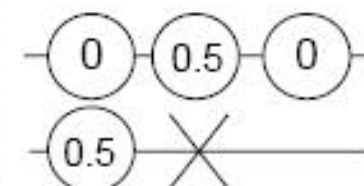
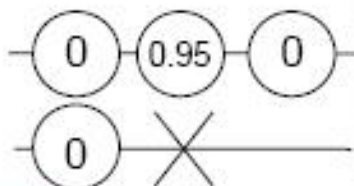
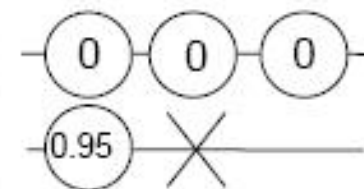
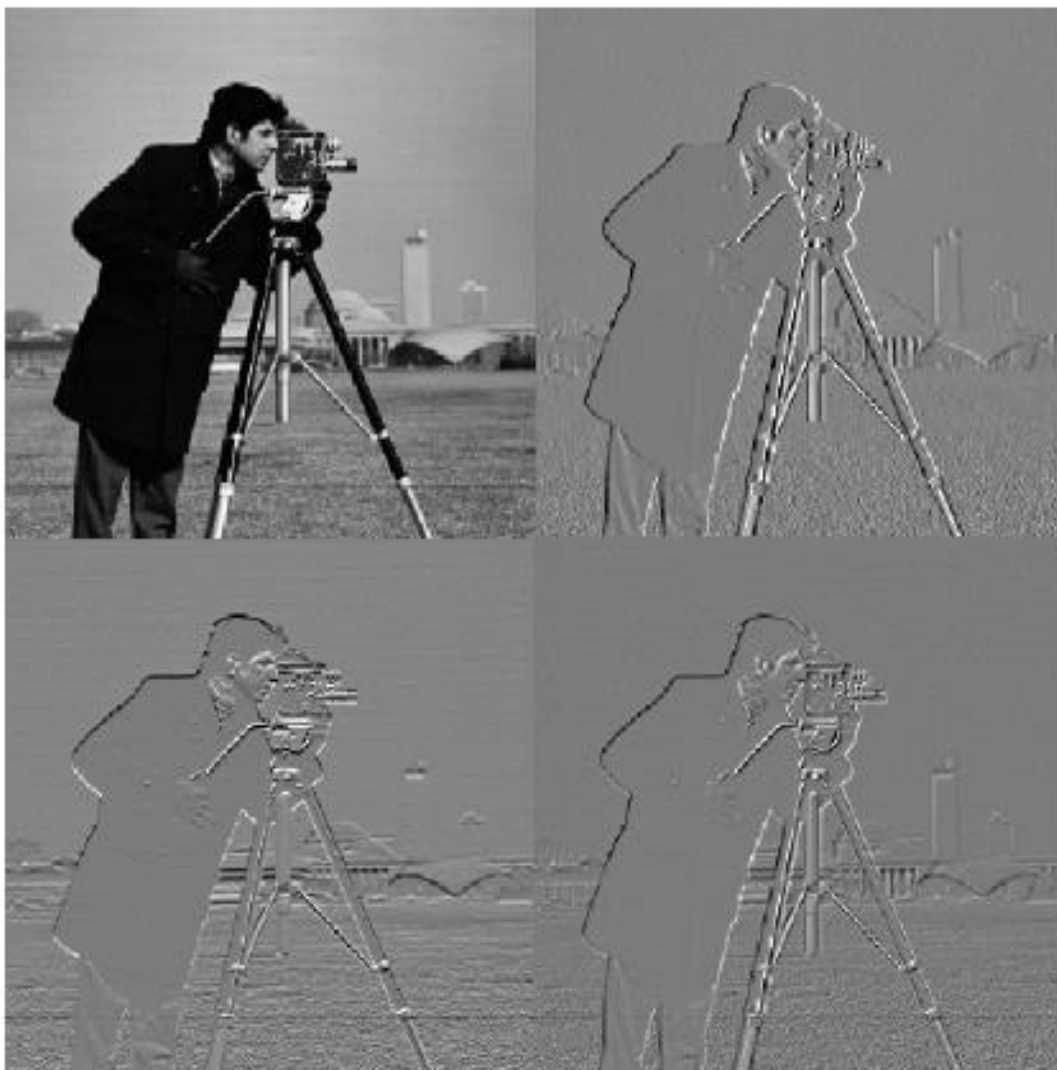
Except the first sample is quantized separately by a 7-bit uniform quantizer, given $\tilde{u}(0) = u(0) = 100$.

DPCM vs. Feedforward Predictive Coder

| Input | | DPCM | | | | | Feedforward Predictive Coder | | | | |
|-------|--------|----------------------|--------|----------------|----------------|---------------|------------------------------|------------------|--------------------------|----------------|---------------|
| N | $u(n)$ | $\bar{\tilde{u}}(n)$ | $e(n)$ | $\tilde{e}(n)$ | $\tilde{u}(n)$ | $\delta u(n)$ | $\bar{u}(n)$ | $\varepsilon(n)$ | $\tilde{\varepsilon}(n)$ | $\tilde{u}(n)$ | $\delta u(n)$ |
| 0 | 100 | — | — | — | 100 | 0 | — | — | — | 100 | 0 |
| 1 | 102 | 100 | 2 | 1 | 101 | 1 | 100 | 2 | 1 | 101 | 1 |
| 2 | 120 | 101 | 19 | 5 | 106 | 14 | 102 | 18 | 5 | 106 | 14 |
| 3 | 120 | 106 | 14 | 5 | 111 | 9 | 120 | 0 | -1 | 105 | 15 |
| 4 | 120 | 111 | 9 | 5 | 116 | 4 | 120 | 0 | -1 | 104 | 16 |
| 5 | 118 | 116 | 2 | 1 | 117 | 1 | 120 | -2 | -5 | 99 | 19 |

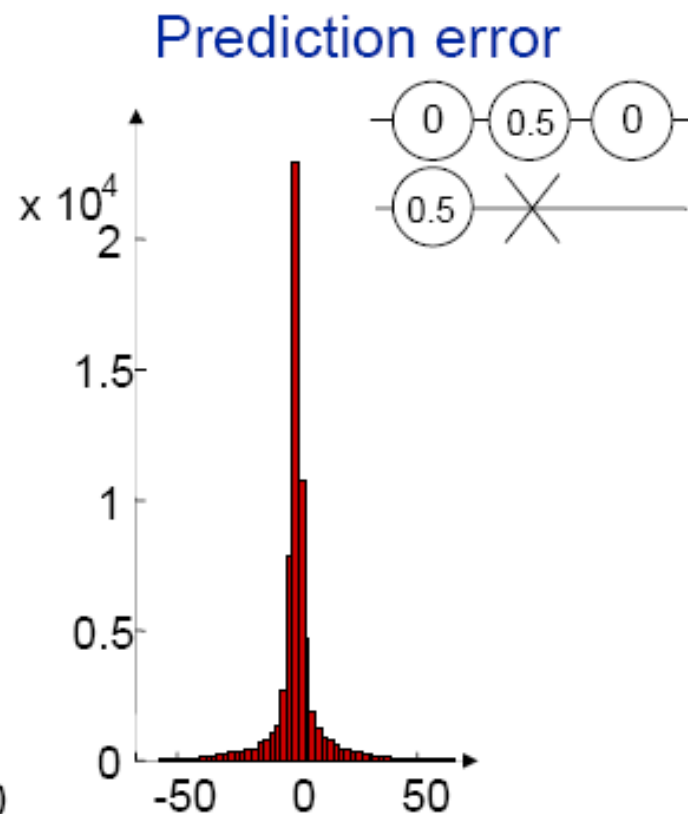
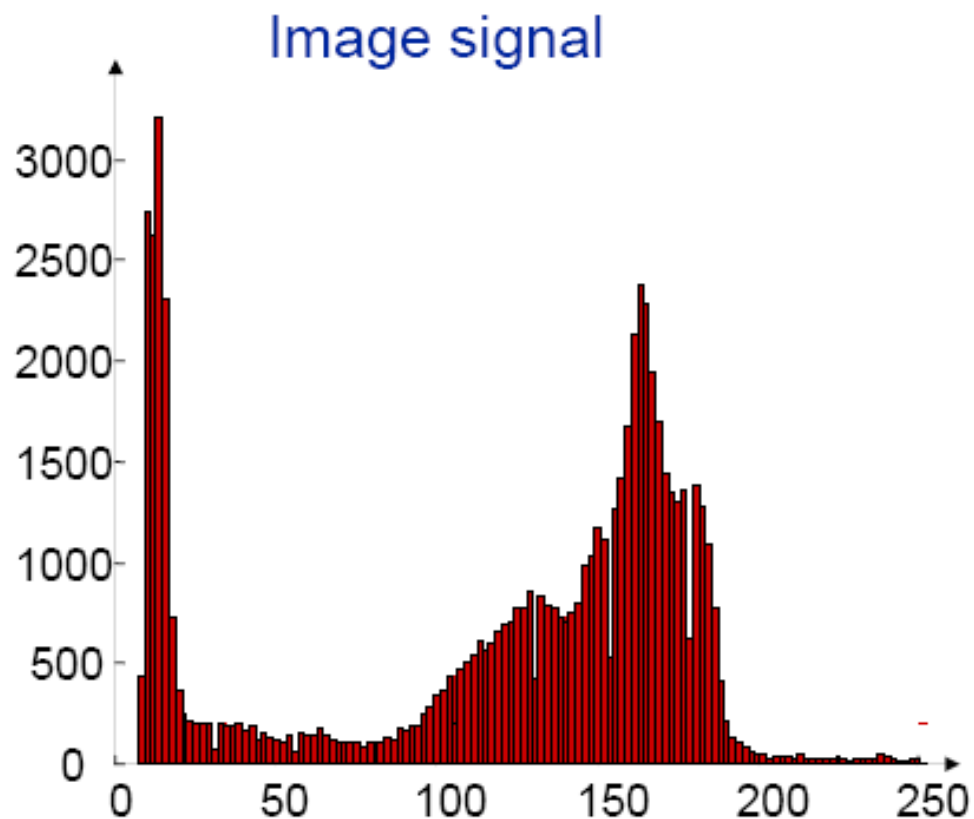
例：图像预测结果（Cameraman）

original



例：图像预测结果

直方图



5.1.1 DPCM的基本原理

- 预测误差为

$$e_k = x_k - \hat{x}_k = x_k - f(x'_{i-1}, x'_{i-2}, \dots, x'_{i-N})$$

- 预测误差的方差:

$$MSE = \sigma_e^2 = E[(x_k - \hat{x}_k)^2] = E[(x_k - f(x'_{i-1}, x'_{i-2}, \dots, x'_{i-N}))^2]$$

- 若用**MSE**作为失真度量，则最小化失真的最佳预测器为

$$\hat{x}_k = \hat{f}(x'_{i-1}, x'_{i-2}, \dots, x'_{i-N}) = E[x_k | x'_{i-1}, x'_{i-2}, \dots, x'_{i-N}]$$

- 若采用前**N**个样本预测，且每个样本为**B**比特，则上述条件概率需要一个有 2^{BN} 项的表

5.1.1 DPCM的基本原理

- 在一个DPCM系统中需要设计两个部分
 - 预测器
 - 量化器
- 理想情况下，应同时优化预测器和量化器，但实际应用中，采用一种次优化方法
 - 在这种方法中，**量化电平数必须足够大**

(**M≥8**)才能获得好的性能

$$SNR = \frac{E[S^2(k)]}{E[e^2(k)]} = \frac{E[S^2(k)]}{E[d^2(k)]} \cdot \frac{E[d^2(k)]}{E[e^2(k)]} = G_p \cdot SNR_q$$

量化误差均方值

预测误差均方值

5.1.1 DPCM的基本原理

线性预测器的设计

- 具有最小均方预测误差的预测器 – 最大的预测增益
- 即在最小 $E[d^2]$ 的条件下，确定一组最佳预测系数（略去量化误差）

$$E[d^2] = E[(S(k) - S_e(k))^2] = E[(S(k) - \sum_{i=1}^N a_i S(k-i))^2]$$

最佳预测系数需满足： $\frac{\partial E[d^2]}{\partial a_i} = 0 \quad i = 1, 2, \dots, N$

描述随机过程在两个不同时刻取值之间的相关程度

一组线性方程：（矩阵形式）

$$\begin{bmatrix} R(1) \\ R(2) \\ \vdots \\ R(N) \end{bmatrix} = \begin{bmatrix} R(0) & R(1) & \dots & R(N-1) \\ R(1) & R(0) & \dots & R(N-2) \\ \vdots & \vdots & \ddots & \vdots \\ R(N-1) & R(N-2) & \dots & R(0) \end{bmatrix} \begin{bmatrix} a_1 \\ a_2 \\ \vdots \\ a_N \end{bmatrix}$$

自相关函数
 $R(i) = E[S(k)S(k-i)]$
 $i = 0, 1, 2, \dots, N-1$

$$\begin{bmatrix} a_1 \\ a_2 \\ \vdots \\ a_N \end{bmatrix} = \begin{bmatrix} \mathbf{R}(0) & \mathbf{R}(1) & \cdots & \mathbf{R}(N-1) \\ \mathbf{R}(1) & \mathbf{R}(0) & \cdots & \mathbf{R}(N-2) \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{R}(N-1) & \mathbf{R}(N-2) & \cdots & \mathbf{R}(0) \end{bmatrix}^{-1} \begin{bmatrix} \mathbf{R}(1) \\ \mathbf{R}(2) \\ \vdots \\ \mathbf{R}(N) \end{bmatrix}$$

最小均方误差为：

$$E[d^2]_{\min} = E[S^2(k)] - \sum_{i=1}^N a_i \mathbf{R}(i) = \mathbf{R}(0) - \sum_{i=1}^N a_i \mathbf{R}(i)$$

- ❑ 假定均值为0，误差信号的方差比原信号的方差小。和实际信号的分布相比，预测误差是关于0的高尖峰。因此，预测误差总是具有比原始密度更小的熵。这意味着预测的过程是把样值间的大部分冗余（统计冗余）去掉
- ❑ 预测器的复杂度取决于 n

例1：DPCM

- 例：令DPCM中的参数 $N=1, w_1=1$ ，即取前一个取样 x_{k-1} 值作为当前取样值 x_k 的预测值，则

$$\begin{aligned}\sigma_e^2 &= E\left[(x_k - \hat{x}_k)^2\right] = E\left[(x_k - x_{k-1})^2\right] = E\left(x_k^2 - 2x_k x_{k-1} + x_{k-1}^2\right) \\ &= 2\left[1 - \frac{R(1)}{R(0)}\right] R(0) = 2(1 - \rho) R(0)\end{aligned}$$

- 其中 $\rho = R(1)/R(0)$ 为信号的自相关系数，当 $0.5 < \rho \leq 1$ 时， $\sigma_e^2 < R(0)$ ，可达到压缩的效果
 - 对电视图像，通常 $\rho = 0.95 \sim 0.98$ ，此时 $\sigma_e^2/R(0) = 1/10 \sim 1/24$ 这意味着误差信号的功率比原始信号降低了 $10 \sim 14 \text{ dB}$
 - 降低 6 dB 约减少 1 bit

例2: DPCM

- 例：在图像中的例子。 $N=3$ ，像素X用其3个邻居A、B和C来预测，求MSE下的最佳预测系数

$$\sigma_e^2 = \frac{1}{K} \sum_{k=1}^K (x_k - \hat{x}_k)^2 = \frac{1}{K} \sum_{k=1}^K \left(x_k - \sum_{i=1}^N w_i x_{k-i} \right)^2$$

| | | |
|---|---|--|
| | | |
| B | C | |
| A | X | |
| | | |

| | | | | |
|--|-----|-----|-----|-----|
| | 101 | 128 | 108 | 110 |
| | 100 | 90 | 95 | 100 |
| | 102 | 80 | 90 | 85 |
| | 105 | 75 | 96 | 91 |

输入图像

A: -1
B: -2
C: -3

$$\mathbf{x} = \{90, 95, 100, 80, 90, 85, 75, 96, 91\}$$

$$\mathbf{x}_{-1} = \{100, 90, 95, 102, 80, 90, 105, 75, 96\}$$

$$\mathbf{x}_{-2} = \{101, 128, 108, 100, 90, 95, 102, 80, 90\}$$

$$\mathbf{x}_{-3} = \{128, 108, 110, 90, 95, 100, 80, 90, 85\}$$

$$\sigma_e^2 = \left(\mathbf{x} - \sum_{i=1}^3 w_i \mathbf{x}_{-i} \right)^2$$

用最小二乘法求解，得到 $w_1 = 0.1691$, $w_2 = 0.1988$, and $w_3 = 0.5382$

例3 预测阶数的选择

- 例：1维N阶高斯马尔科夫过程，自相关系数为

$$r(k) = \frac{R(k)}{R(0)} = \rho^{|k|} \quad 0 < \rho < 1$$

- 求解 a_i

$$\begin{bmatrix} 1 & \rho & \rho^2 & \dots & \rho^{N-1} \\ \rho & 1 & \rho & \dots & \rho^{N-2} \\ \vdots & & & & \\ \rho^{N-1} & \rho^{N-2} & \rho^{N-3} & \dots & 1 \end{bmatrix} \begin{bmatrix} a_1 \\ a_2 \\ \vdots \\ a_N \end{bmatrix} = \begin{bmatrix} \rho \\ \rho^2 \\ \vdots \\ \rho^N \end{bmatrix}$$

若 $N=1$, 则 $a_1 = \rho \hat{x}_k = \rho x_{k-1}$

$$\sigma_{\text{emin}}^2 = R(0) - a_1 R(1) = \left[1 - a_1 \frac{R(1)}{R(0)} \right] R(0) = (1 - \rho^2) R(0) < R(0)$$

$$\begin{aligned} E\{e_k e_{k+j}\} &= E\{[x_k - \rho x_{k-1}][x_{k+j} - \rho x_{k+j-1}]\} \\ &= R(j) - \rho R(j+1) - \rho R(j-1) + \rho^2 R(j) \\ &= R(0)[\rho^j - \rho \cdot \rho^{j+1} - \rho \cdot \rho^{j-1} + \rho^{j+2}] = 0 \end{aligned}$$

例3 预测阶数的选择

- 例：1维N阶高斯马尔科夫过程，自相关系数为

$$r(k) = \frac{R(k)}{R(0)} = \rho^{|k|} \quad 0 < \rho < 1$$

若 $N = 2$ ，即求 $\hat{x}_k = a_1 x_{k-1} + a_2 x_{k-2}$

$$\begin{bmatrix} 1 & \rho \\ \rho & 1 \end{bmatrix} \begin{bmatrix} a_1 \\ a_2 \end{bmatrix} = \begin{bmatrix} \rho \\ \rho^2 \end{bmatrix}$$

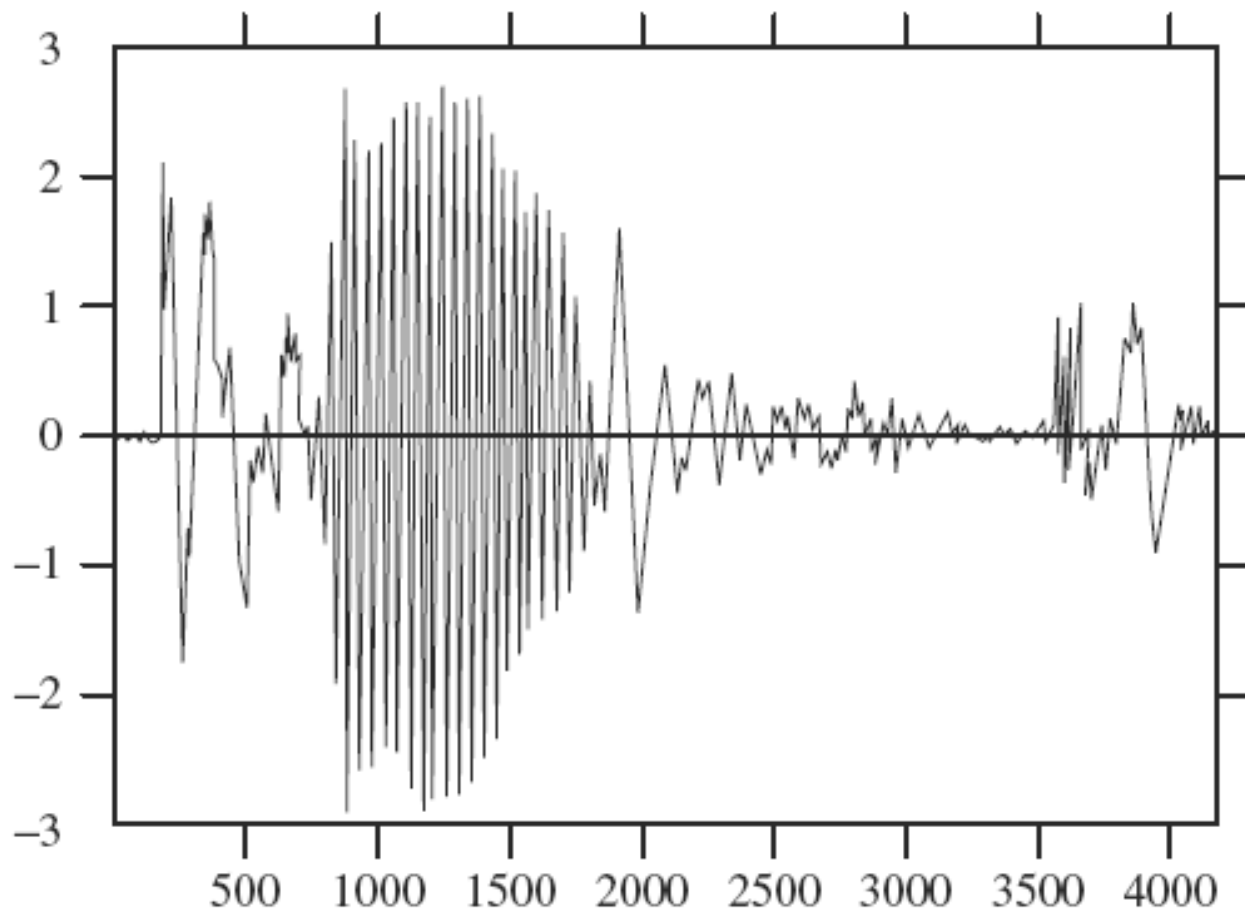
$$\begin{bmatrix} a_1 \\ a_2 \end{bmatrix} = \frac{1}{1-\rho^2} \begin{bmatrix} 1 & -\rho \\ -\rho & 1 \end{bmatrix} \begin{bmatrix} \rho \\ \rho^2 \end{bmatrix} = \begin{bmatrix} \rho \\ 0 \end{bmatrix}$$

可见：N=2时解得的预测方程与N=1时的完全相同

注意：所谓的最佳预测是在MMSE准则下。当考虑到人的主观感觉时，并不一定最佳。

例4：线性预测器

■ “test” 的语音片段



例4：线性预测器 (2)

■ 计算

$$R_{xx}(k) = \frac{1}{M-k} \sum_{i=1}^{M-k} x_i x_{i+k}$$

给定M个数据点
，用平均值估计
自相关值

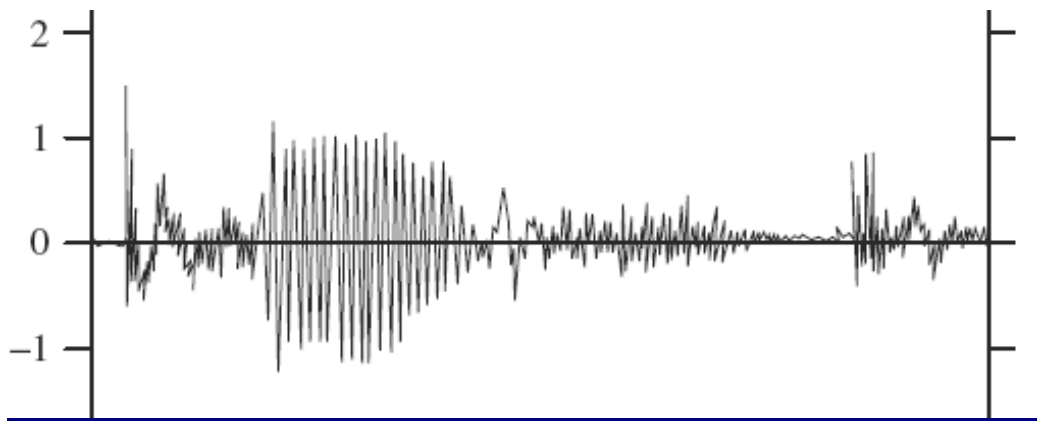
■ 分别用1阶、2阶和3阶预测器

➤ $N = 1 \rightarrow a_1 = 0.66$

➤ $N = 2 \rightarrow a_1 = 0.596, a_2 = 0.096$

➤ $N = 3 \rightarrow a_1 = 0.577, a_2 = -0.025, a_3 = 0.204$

残差信号：



例4：线性预测器 (3)

■ 采用最简单的均匀量化器

- 差值集中在0附近，用Laplacian近似
- 给定方差的最优均匀量化器

■ 步长：

- 4-水平：1阶：0.75；2阶：0.59；3阶：0.43
- 8-水平：1阶：0.3；2阶：0.4；3阶：0.5

信噪比

$$\text{SNR(dB)} = \frac{\sum_{i=1}^M x_i^2}{\sum_{i=1}^M (x_i - \hat{x}_i)^2}$$

信号预测误差比

$$\text{SPER(dB)} = \frac{\sum_{i=1}^M x_i^2}{\sum_{i=1}^M (x_i - p_i)^2}$$

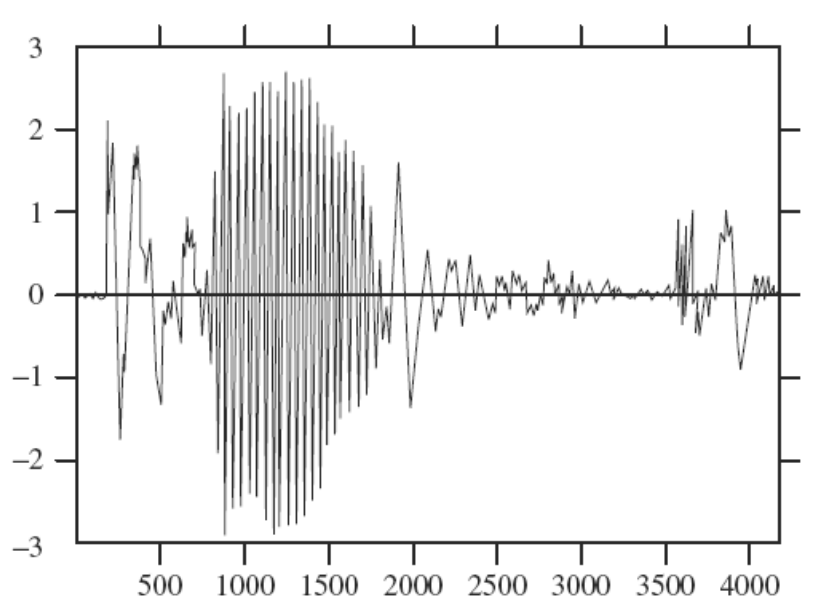
Prediction Error

例4：线性预测器 (4)

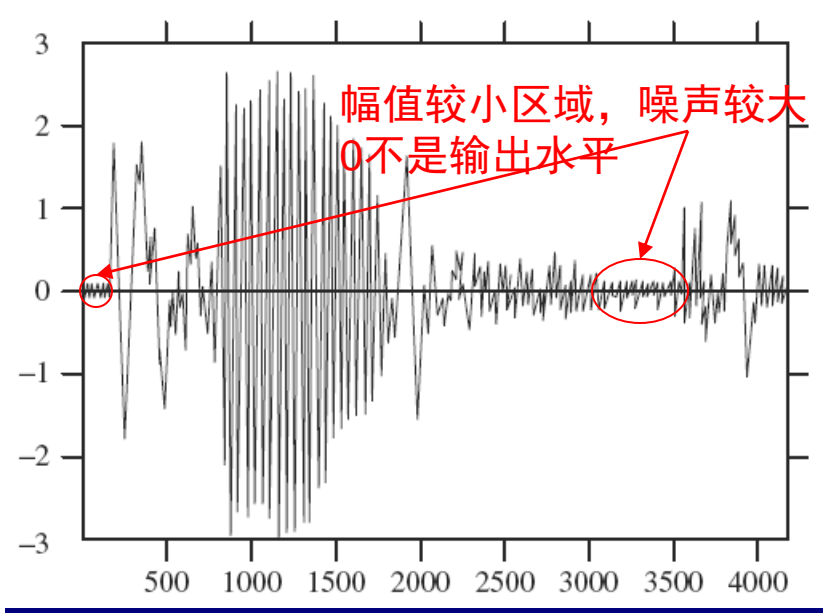
| Quantizer | Predictor Order | SNR (dB) | SPER (dB) |
|-------------|-----------------|----------|-----------|
| Four-level | None | 2.43 | 0 |
| | 1 | 3.37 | 2.65 |
| | 2 | 8.35 | 5.9 |
| | 3 | 8.74 | 6.1 |
| Eight-level | None | 3.65 | 0 |
| | 1 | 3.87 | 2.74 |
| | 2 | 9.81 | 6.37 |
| | 3 | 10.16 | 6.71 |

从一阶预测到二阶预测有很大的差异，从二阶预测转为三阶预测，增大值相对减小。

例4：线性预测器 (5)



原始信号



重构信号：

Laplacian 8水平量化器
3阶预测器

例5: DPCM+量化



1 bit/pixel
prediction error coding

slope overload



2 bit/pixel

edge busyness

granular noise



3 bit/pixel

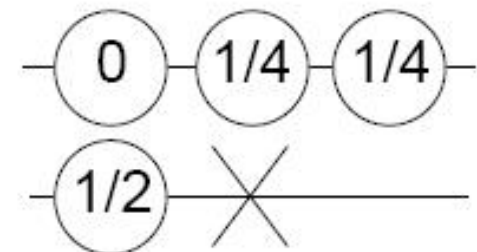


4 bit/pixel



original

■ Linear predictor:



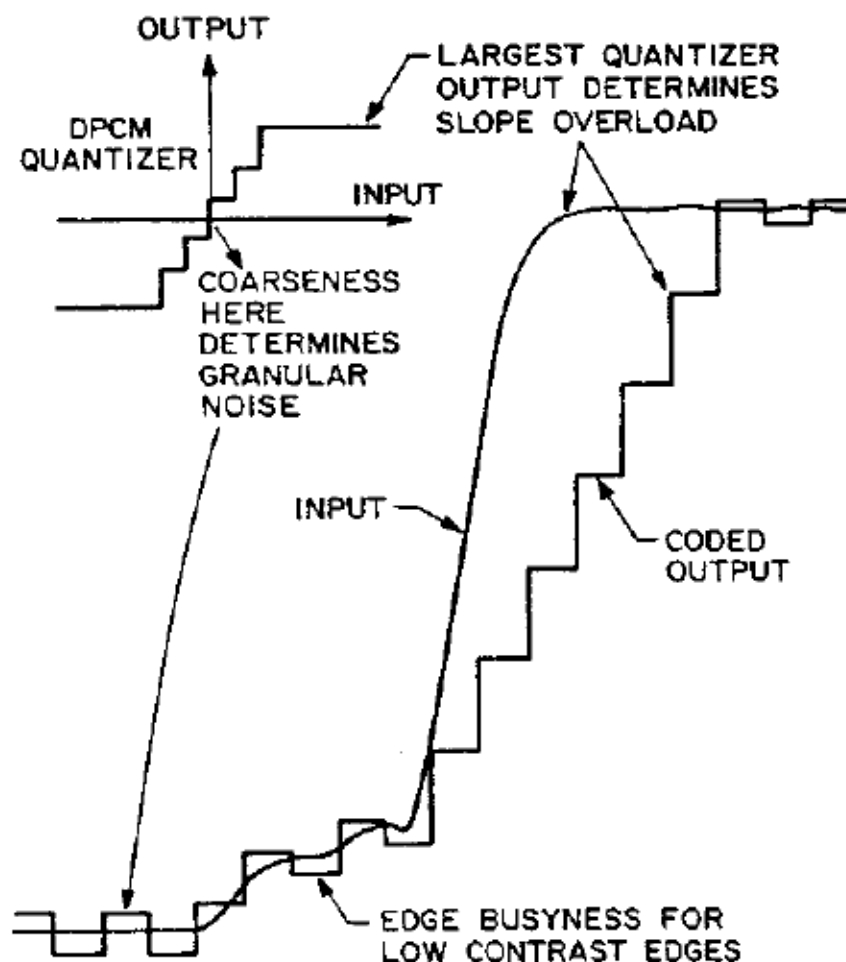
■ Lloyd-Max quantizers

■ Fixed-length coding



例5: DPCM+量化: 信号失真

- 粒状噪声 (Granular noise)
: 平坦区域的随机噪声
- Edge busyness: 抖动的边缘
- 斜率过载 (Slope overload)
: 边缘模糊



例6：预测+熵约束量化



$M: 511, H: 4.79$

$M: 15, H: 1.98$

$M: 3, H: 0.68$

M : 重构水平的数目

H : 预测误差的熵

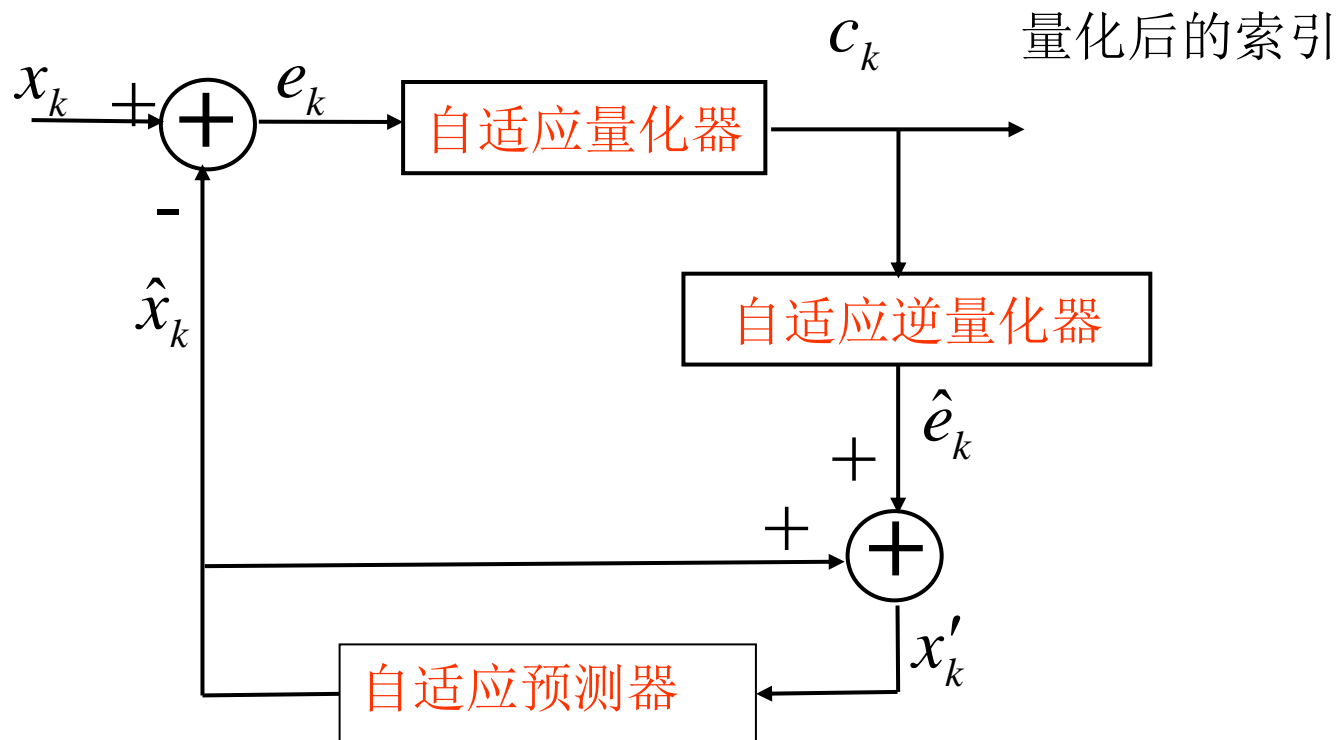
5.1.2 ADPCM

(Adaptive Differential Pluse Code Modulation)

- DPCM是以输入数据为平稳随机过程为依据。预测系数和量化器参数一次设计好后即不再改变。
- 但在一些应用中，如在图像平坦区和边缘处要求量化器的输出差别很大。否则会导致图像出现噪声。
- **ADPCM**工作原理：预测器的预测系数和量化器的量化参数能够根据信源符合的局部区域分布特点而自动调整。
 - 实践证明：不仅能改善语音/图像的评测质量和视觉效果，还能进一步压缩数据
- 自适应系统：
 - 自适应预测（预测系数的自适应调整）
 - 自适应量化（量化器步长的自适应调整）

5.1.2 ADPCM

(Adaptive Differential Pulse Code Modulation)



ADPCM编码结构

前向自适应预测器：DPCM-APF

- 将输入分成块
 - 语音编码：8000个样本/秒时，128个样本/块 (16ms)
 - 图像编码：8x8 块
- 自相关系数
 - 假设块外样本为0

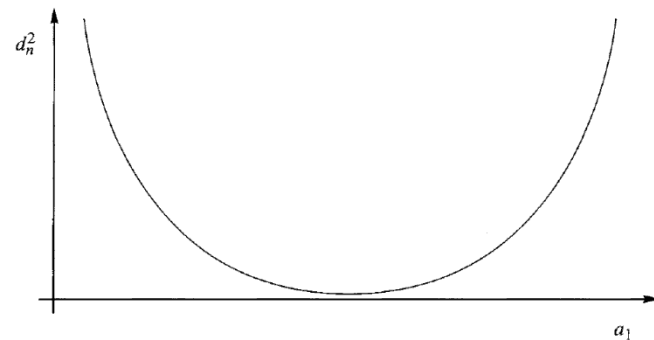
$$R_{xx}^{(l)}(k) = \frac{1}{M-k} \sum_{i=(l-1)M-k}^{lM-k} x_i x_{i+k}, \text{ for } k > 0$$

$$R_{xx}^{(l)}(k) = \frac{1}{M-k} \sum_{i=(l-1)M+1}^{lM} x_i x_{i+k} \text{ for } k < 0$$

$$R_{xx}^{(l)}(k) = R_{xx}^{(l)}(-k)$$

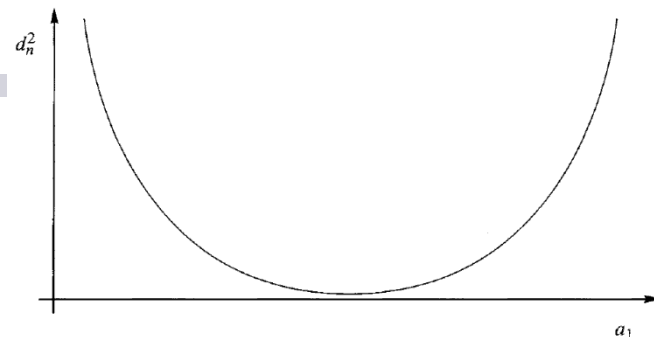
后向自适应预测器：DPCM-APB

去掉输入平稳的假设，用代数方法调整预测器
(梯度下降法)



均方误差是权系数向量的二次函数，它是一个中间向上凹的抛物形曲面，是具有唯一最小值的函数。调节权系数使均方误差为最小，相当于沿抛物形曲面下降找最小值。可以用梯度法来求该最小值

后向自适应预测器：DPCM-APB



■ 1阶预测器

- 比最佳值小, $\frac{\partial d_n^2}{\partial a_1} < 0$; 且 a_1 离最佳值越远, $\left| \frac{\partial d_n^2}{\partial a_1} \right|$ 越大
- 比最佳值大, $\frac{\partial d_n^2}{\partial a_1} > 0$; 且 a_1 离最佳值越远, $\left| \frac{\partial d_n^2}{\partial a_1} \right|$ 越大

- 在任意给定时刻, 为调整时刻n+1时刻的系数, 向时刻n的系数增加一个量, 这个量的大小与时刻n的导数大小成正比, 符号与导数相反。

$$\Rightarrow a_1^{(n+1)} = a_1^{(n)} - \alpha \frac{\partial d_n^2}{\partial a_1}$$

$$\Rightarrow a_1^{(n+1)} = a_1^{(n)} + \alpha d_n \hat{x}_{n-1}$$

$$\Rightarrow a_1^{(n+1)} = a_1^{(n)} + \alpha \hat{d}_n \hat{x}_{n-1}$$

$$\frac{\partial d_n^2}{\partial a_1} = -2(x_n - a_1 \hat{x}_{n-1}) \hat{x}_{n-1} = -2d_n \hat{x}_{n-1}$$

为使编码器和解码器使用同一算法, 用反量化后的预测误差替代预测误差

后向自适应预测器：DPCM-APB

■ 推广到N阶预测器

$$d_n^2 = \left(x_n - \sum_{i=1}^N a_i \hat{x}_{n-i} \right)^2$$

a 在这里是正比例常数。
在梯度下降法中称为学习率参数。

$$\begin{aligned} a_j^{(n+1)} &= a_j^{(n)} - a \hat{d}_n \hat{x}_{n-j} \\ \mathbf{A}^{(n+1)} &= \mathbf{A}^{(n)} - a \hat{d}_n \hat{\mathbf{X}}_{n-j} \end{aligned}$$

把所有N个方程合并为矢量形式：

最小均方（LMS）算法

Readings

- S. K. Goyal, J. B. O'Neal, "Entropy Coded Differential Pulse-Code Modulation Systems for Television Systems," IEEE Trans. Communications, pp. 660-666, June 1975.
- N. Farvadin, J. W. Modestino, "Rate-distortion performance of DPCM schemes for autoregressive sources," IEEE Trans. Information Theory, vol. 31, no. 3, pp. 402-418, May 1985.
- 作业：总结最小二乘——梯度下降、牛顿法、高斯牛顿法