Differential Coding

- If the source $\{x_n\}$ is highly correlated from sample to sample
  - the dynamic range and the variance of $\{d_n\}$ are significantly smaller than that of $\{x_n\}$ $\quad d_n = x_n - x_{n-1}$
- Example
  - input sequence x_n : 6.2 9.7 13.2 5.9 8 7.4 4.2 1.8
  - difference sequence $d_n = x_n - x_{n-1}$ (assume $x_0 = 0$)

    6.2 3.5 3.5 -7.3 2.1 -0.6 -3.2 -2.4

  (in lossless compression mode)
  - the original sequence can be recovered exactly by $x_n = x_{n-1} + d_n$

  (in lossy compression mode)
  - Suppose a 7-level quantizer Q with output values { -6, -4, -2, 0, 2, 4, 6} is adopted

$$\hat{d}_n = Q[d_n] \;:\; 6 \quad 4 \quad 4 \quad -6 \quad 2 \quad 0 \quad -4 \quad -2$$

$$\hat{x}_n = \hat{x}_{n-1} + \hat{d}_n \;:\; 6 \quad 10 \quad 14 \quad 8 \quad 10 \quad 10 \quad 6 \quad 4$$

  - The error between the original sequence and the reconstructed sequence

$$x_n - \hat{x}_n : \; 0.2 \quad -0.3 \quad -0.8 \quad -2.1 \quad -2 \quad -2.6 \quad -1.8 \quad -2.2$$

Differential Coding – type I

- For a sequence $\{x_n\}$

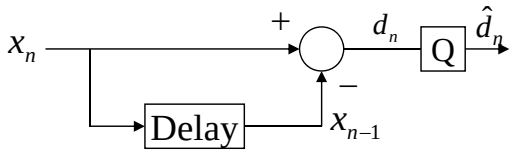
  - the difference sequence $\{d_n\}$ is generated by $d_n = x_n - x_{n-1}$

  - this difference sequence is then quantized to obtain the sequence $\{\hat{d}_n\}$

$$\hat{d}_n = Q[d_n] = d_n + q_n \qquad \text{where } q_n \text{ is the quantization error}$$

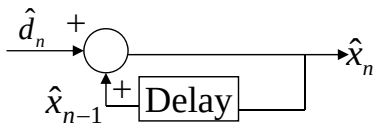
  - the reconstructed sequence $\{\hat{x}_n\}$ is obtained by

$$\hat{x}_n = \hat{x}_{n-1} + \hat{d}_n$$



Encoder

- If both encoder and decoder start with x_0, that is $\hat{x}_0 = x_0$ (=0)



Decoder

$$d_1 = x_1 - x_0$$

$$\hat{d}_1 = Q[d_1] = d_1 + q_1$$

$$\hat{x}_1 = \hat{x}_0 + \hat{d}_1 = x_0 + d_1 + q_1 = x_0 + (x_1 - x_0) + q_1 = x_1 + q_1$$

$$d_2 = x_2 - x_1$$

$$\hat{d}_2 = Q[d_2] = d_2 + q_2$$

$$\hat{x}_2 = \hat{x}_1 + \hat{d}_2 = x_1 + q_1 + d_2 + q_2 = x_1 + q_1 + (x_2 - x_1) + q_2 = x_2 + q_1 + q_2$$

$$\vdots$$

$$\hat{x}_n = x_n + \sum_{k=1}^{n} q_k$$

Quantization error accumulates as the process continues

Differential Coding – type II

- For a sequence $\{x_n\}$
  - the difference sequence $\{d_n\}$ is generated by $d_n = x_n - \hat{x}_{n-1}$
  - this difference sequence is then quantized to obtain the sequence $\{\hat{d}_n\}$

    $\hat{d}_n = Q[d_n] = d_n + q_n$

    where q_n is the quantization error
  - the reconstructed sequence $\{\hat{x}_n\}$ is obtained by

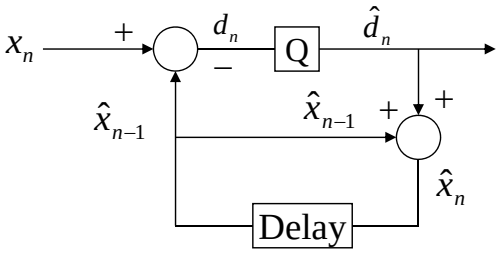
    $\hat{x}_n = \hat{x}_{n-1} + \hat{d}_n$
- If both encoder and decoder start with x_0, that is $\hat{x}_0 = x_0 = 0$

$$d_1 = x_1 - x_0$$
$$\hat{d}_1 = Q[d_1] = d_1 + q_1$$
$$\hat{x}_1 = \hat{x}_0 + \hat{d}_1 = x_0 + d_1 + q_1 = x_0 + (x_1 - x_0) + q_1 = x_1 + q_1$$
$$d_2 = x_2 - \hat{x}_1$$
$$\hat{d}_2 = Q[d_2] = d_2 + q_2$$
$$\hat{x}_2 = \hat{x}_1 + \hat{d}_2 = (x_2 - d_2) + (d_2 + q_2) = x_2 + q_2$$
$$\vdots$$
$$\hat{x}_n = x_n + q_n$$

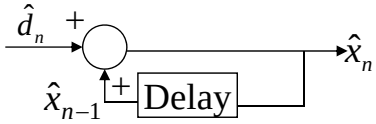
No accumulation of the quantization error

Differential Coding – type II (cont.)

- Input sequence: 6.2 9.7 13.2 5.9 8 7.4 4.2 1.8



Encoder



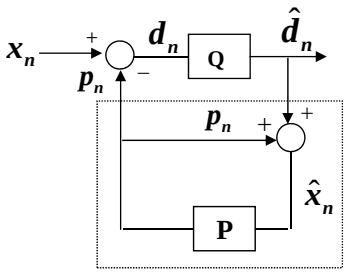
Decoder

- $d_n = x_n - \hat{x}_{n-1}$:		6.2	3.7	3.2	-8.1	0	-0.6	-3.8	-2.2
- $\hat{d}_n = Q[d_n]$:		6	4	4	-6	0	0	-4	-2
- $\hat{x}_n = \hat{x}_{n-1} + \hat{d}_n$:	0	6	10	14	8	8	8	4	2
- the error between the original sequence and the reconstructed sequence									
$x_n - \hat{x}_n$:		0.2	-0.3	-0.8	-2.1	0	-0.6	0.2	-0.2

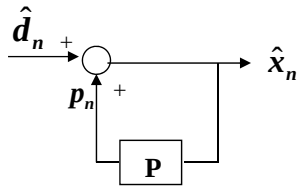
Q: 1. d_n 之 dynamic range type II 比 type I 大, ∴quantization error ↑

2. Type II 所獲得gain, 跟 1. 所付出代價熟大?

DPCM



Encoder



Decoder

- Differential pulse code modulation
  - Predictor P

$$p_n = f(\hat{x}_{n-1}, \hat{x}_{n-2}, \ldots, \hat{x}_0)$$

  - p_n should be as close to x_n as possible
  - Difference $d_n = x_n - p_n$
  - Quantizer Q

$$\hat{d}_n = Q[d_n] = d_n + q_n$$

$$\Rightarrow \quad \hat{x}_n = p_n + \hat{d}_n = (x_n - d_n) + (d_n + q_n) = x_n + q_n$$

If $p_n \equiv \hat{x}_{n-1}$ and $p_1 \equiv 0$ then
$d_1 = x_1$
$d_2 = x_2 - p_2 = x_2 - \hat{x}_1$
$d_3 = x_3 - \hat{x}_2$

Design of the DPCM Predictor (1)

- Linear Predictor
  - assume Δ is very small

$$p_n = a_1\hat{x}_{n-1} + a_2\hat{x}_{n-2} + \cdots + a_N\hat{x}_{n-N} = a_1 x_{n-1} + a_2 x_{n-2} + \cdots + a_N x_{n-N} = \sum_{i=1}^{N} a_i x_{n-i}$$

$$d_n = x_n - p_n$$

  - mean squared error σ_d^2

$$\sigma_d^2 = E\left[\left(x_n - \sum_{i=1}^{N} a_i x_{n-i}\right)^2\right]$$

$$P_n \rightarrow d_n \rightarrow q_n \rightarrow \hat{x}_n \quad (\hat{x}_n = x_n + q_n)$$

coupling effect

  - find $\{a_k\}_{k=1\sim N}$ so as to minimize σ_d^2

$$\frac{\partial \sigma_d^2}{\partial a_k} = \frac{\partial E\left[\left(x_n - \sum_{i=1}^{N} a_i x_{n-i}\right)^2\right]}{\partial a_k} = -2E\left[\left(x_n - \sum_{i=1}^{N} a_i x_{n-i}\right)x_{n-k}\right] = 0$$

$$\Rightarrow E\left[\left(x_n - \sum_{i=1}^{N} a_i x_{n-i}\right)x_{n-k}\right] = 0$$

$$\Rightarrow E\left[x_n x_{n-k}\right] = \sum_{i=1}^{N} a_i E\left[x_{n-i} x_{n-k}\right]$$

$$\Rightarrow R_{xx}(k) = \sum_{i=1}^{N} a_i R_{xx}(i-k)$$

$\Rightarrow RA = P$
$\therefore A = R^{-1}P$

Design of the DPCM Predictor (2)

$$R = \begin{bmatrix} R_{xx}(0) & R_{xx}(1) & R_{xx}(2) & \dots & R_{xx}(N-1) \\ R_{xx}(1) & R_{xx}(0) & R_{xx}(1) & \dots & R_{xx}(N-2) \\ R_{xx}(2) & R_{xx}(1) & R_{xx}(0) & \dots & R_{xx}(N-3) \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ R_{xx}(N-1) & R_{xx}(N-2) & R_{xx}(N-3) & \dots & R_{xx}(0) \end{bmatrix}$$

$$A = \begin{bmatrix} a_1 \\ a_2 \\ \vdots \\ a_N \end{bmatrix} \quad \mathrm{P} = \begin{bmatrix} R_{xx}(1) \\ R_{xx}(2) \\ \vdots \\ R_{xx}(N) \end{bmatrix}, \text{where } R_{xx}(-k) \triangleq R_{xx}(k).$$

Example:

- N = 1

$$\boldsymbol{R}_{xx}(\mathbf{1}) = \boldsymbol{a}_1 \boldsymbol{R}_{xx}(\mathbf{0}) = \boldsymbol{a}_1 \sigma_x^2$$

$$\Rightarrow \boldsymbol{a}_1 = \frac{\boldsymbol{R}_{xx}(\mathbf{1})}{\sigma_x^2}$$

- N = 2

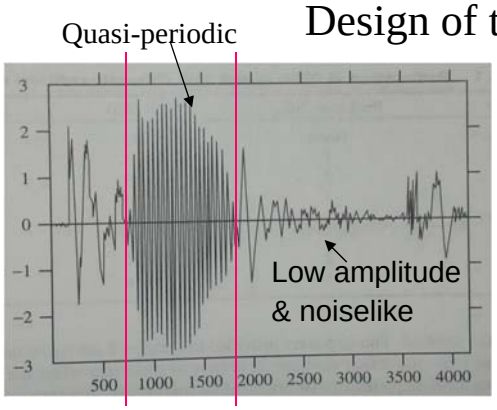
$$\boldsymbol{R}_{xx}(\mathbf{1}) = \boldsymbol{a}_1 \boldsymbol{R}_{xx}(\mathbf{0}) + \boldsymbol{a}_2 \boldsymbol{R}_{xx}(\mathbf{1})$$

$$\boldsymbol{R}_{xx}(\mathbf{2}) = \boldsymbol{a}_1 \boldsymbol{R}_{xx}(\mathbf{1}) + \boldsymbol{a}_2 \boldsymbol{R}_{xx}(\mathbf{0})$$

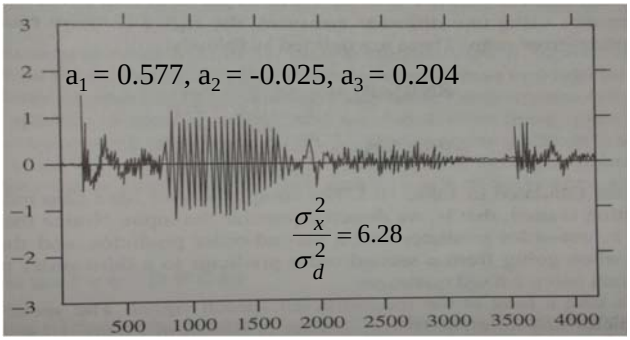
Example: Given M speech data points x_n

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

Design of the DPCM Predictor (3)



A segment of speech x_n

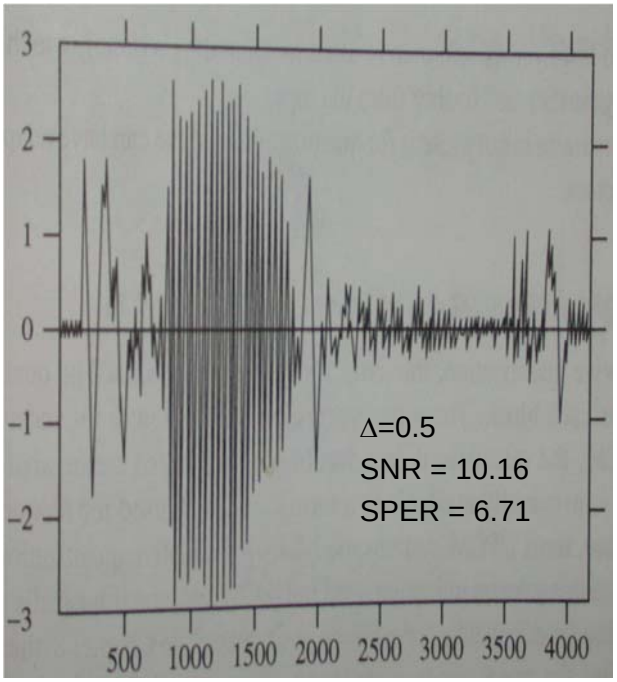


The residual sequence d_n using a 3-order predictor

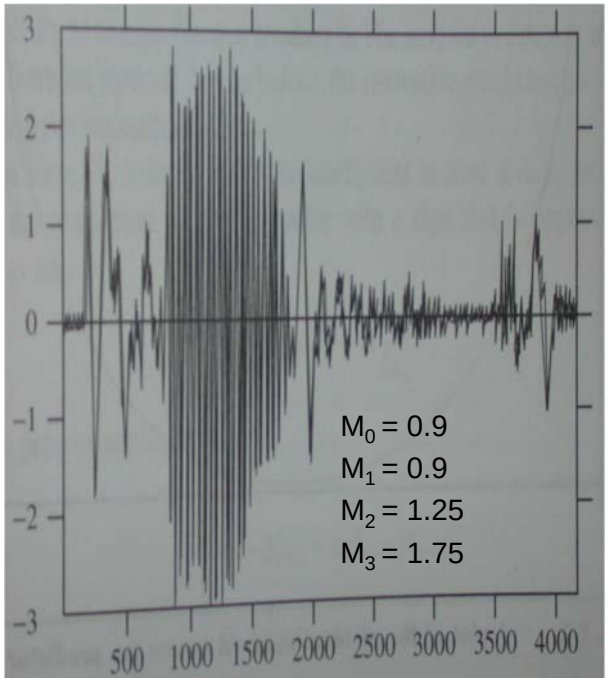
Quantizer	Predictor Order	SNR (dB)	SPER (dB)
Four-level	None	2.43	0
	1	3.37	2.65
	2	8.35	5.9
	3	8.74	6.1
Eight-level	None	3.65	0
	1	3.87	2.74
	2	9.81	6.37
	3	10.16	6.71

Performance of DPCM system with different predictors and quantizers

Design of the DPCM Predictor (4)



The reconstructed sequence using a 3-order predictor and an 8-level uniform quantizer



The reconstructed sequence using a 3-order predictor and an 8-level Jayant quantizer

Adaptive DPCM – ADPCM

- **In DPCM**
  - the input data is assumed to be stationary
    - i.e. the statistical characteristics do not change with time

⇒ the predictor and quantizer are designed with fixed parameters

- **Adaptive design**
  - an backword adaptive predictor usually has a adaptive quantizer

1. If the predictor does not adapt properly at some point, the large residues will tend to fall in the overload regions with unbounded quantization errors.
2. The reconstructed values with these large errors will then be used to adapt the predictor, which will result in the predictor moving further and further from the input

- but, an adaptive quantizer can have a fixed predictor

ADPCM with backword adaptive quantization

- **Adaptive quantization**
  - adapt the quantizer to the local behavior of nonstationary inputs
  - Jayant quantizer

$$\Delta_n = M_{l(n-1)}\Delta_{n-1}$$

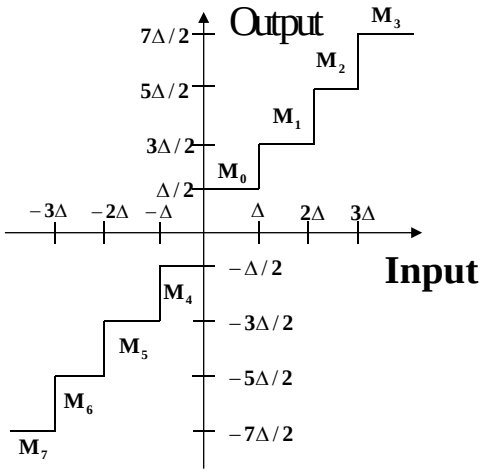
where $M_{l(n-1)}$ is the multiplier for quantization interval *l(n-1)* at time *n-1*

- e.g.

$\Delta_0 = 0.5$

$1 > M_0 \le M_1 \le M_2 \le M_3 > 1$

0.9 0.9 1.25 1.75



ADPCM with forward adaptive prediction

- **Forward Adaptive Prediction**
  - recomputed the covariance matrix and the corresponding weighting factors periodically
  - e.g.
    - for speech signal of sampling rate = 8000 samples/sec
      Each block consists of about 16 ms of speech

    ⇒ 128 samples/book
- **For a block length of *M***

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

or

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

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

ADPCM with backward adaptive prediction

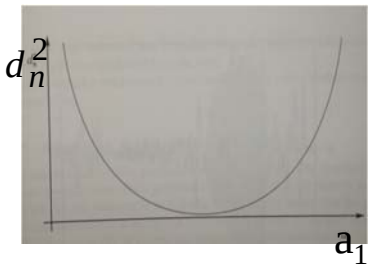
$$d_n^2 = (x_n - a_1 \hat{x}_{n-1})^2$$

$$a_1^{(n+1)} = a_1^{(n)} - \alpha \frac{\delta d_n^2}{\delta a_1}, \text{ where } \alpha \text{ is some proportionality constant}$$

$$\because \quad \frac{\delta d_n^2}{\delta a_1} = -2(x_n - a_1 \hat{x}_{n-1}) x_{n-1} = -2 d_n \hat{x}_{n-1}$$

$$\because \quad a_1^{(n+1)} = a_1^{(n)} + \alpha d_n \hat{x}_{n-1} \quad \text{(absorb the 2 into } \alpha\text{)}$$

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



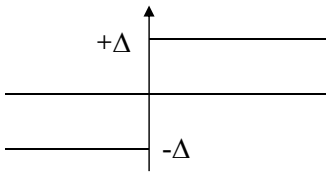
$$d_n^2 = (x_n - \sum_{i=1}^{N} a_i \hat{x}_{n-i})^2$$

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

Delta Modulation (DM)

- **Delta Modulation (DPCM system with two-level quantizer)**
  - **one-bit quantizer**

$$\hat{d}_n = Q[d_n] = \begin{cases} +\Delta, & \text{if } d_n > 0 \\ -\Delta, & \text{if } d_n \le 0 \end{cases}$$

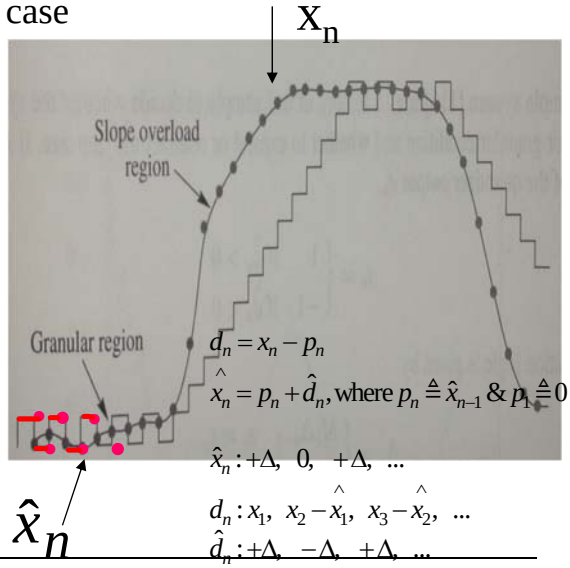


  - **slope overload noise**
    - caused by the inherent limitation of the stair case

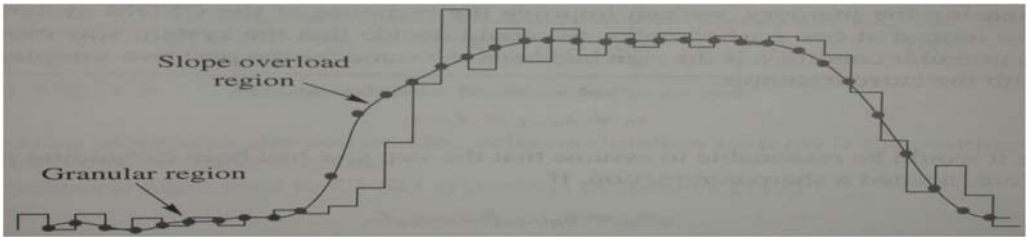
$$\text{maximum slope} = \frac{\text{step size}}{\text{sampling period}} = \frac{\Delta}{T}$$

  - **granular noise**
    - variation around the zero-difference region

- $\Delta \uparrow \Rightarrow$ slope overload $\downarrow$, but granular noise $\uparrow$
- $\Delta \downarrow \Rightarrow$ granular noise $\downarrow$, but slope overload $\uparrow$



Adaptive DM (1)



A source output sampled and coded using adaptive delta modulation

- **Constant Factor Adaptive Delta Modulation (CFDM)**
  - **by monitoring the slope of the input sample**

    · the sign of quantization output does not change $\Rightarrow$ in slope overload region $\Rightarrow$ increase Δ
    · the sign of quantization output changes $\Rightarrow$ in granular region $\Rightarrow$ decrease Δ

  - **e.g. (1-sample memory)**

– $s_n = \begin{cases} 1 & \text{if } \hat{d}_n > 0 \text{ (i.e. } +\Delta_n) \\ -1 & \text{if } \hat{d}_n < 0 \text{ (i.e. } -\Delta_n) \end{cases}$

– Adaptation

$$\Delta_n = \begin{cases} M_1\Delta_{n-1} & s_n = s_{n-1} \\ M_2\Delta_{n-1} & s_n \neq s_{n-1} \end{cases}$$

where $M_1 > 1$, $M_2 < 1$, and $M_1 M_2 = 1$ (In general $M_1 < 2$)

Adaptive DM (2)

e.g. (2-sample memory)

- adaptation

$$\Delta_n = \begin{cases} M_1\Delta_{n-1} = 0.4\Delta_{n-1} & s_n \neq s_{n-1} = s_{n-2} \\ M_2\Delta_{n-1} = 0.9\Delta_{n-1} & s_n \neq s_{n-1} \neq s_{n-2} \\ M_3\Delta_{n-1} = 1.5\Delta_{n-1} & s_n = s_{n-1} \neq s_{n-2} \\ M_4\Delta_{n-1} = 2.0\Delta_{n-1} & s_n = s_{n-1} = s_{n-2} \end{cases}$$

overload $\rightarrow$ granular
還在granular, but Δ 仍太大
granular $\rightarrow$ overload
還在overload, but Δ 仍太小

- Continuously Variable Slope Delta Modulation (CVSD)
  - adaptation

$$\Delta_n = \beta\Delta_{n-1} + \alpha_n\Delta_0$$

where $0 < \beta < 1$ and

$$\alpha_n = \begin{cases} 1 & s_n = s_{n-1} = \ldots = s_{n-j-1} \\ 0 & \text{otherwise} \end{cases}$$