Multimedia Communications

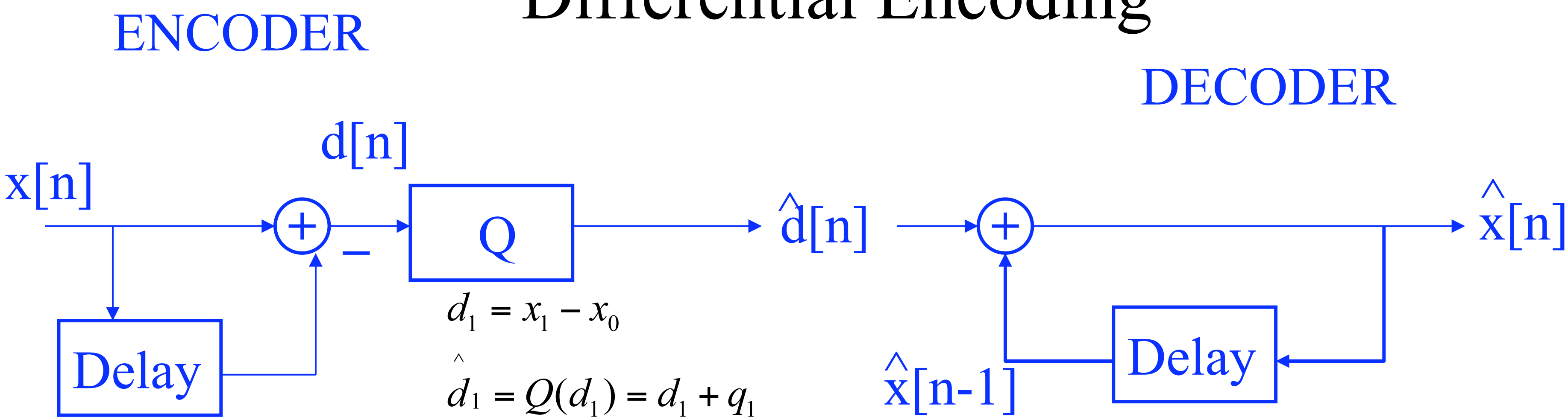
Differential Coding

McMaster University

Differential Coding

- In many sources, the source output does not change a great deal from one sample to the next.
- This means that both the dynamic range and the variance of the sequence of differences $\{d_n = x_n - x_{n-1}\}$ are significantly smaller than that of the source output sequence.
- For correlated sources the distribution of d_n is highly peaked at zero.
- Techniques that transmit information by encoding differences are called differential encoding

Differential Encoding



$$d_1 = x_1 - x_0$$

$$\hat{d}_1 = Q(d_1) = d_1 + q_1$$

$$\hat{x}_1 = x_0 + \hat{d}_1 = x_0 + d_1 + q_1 = x_1 + q_1$$

$$d_2 = x_2 - x_1$$

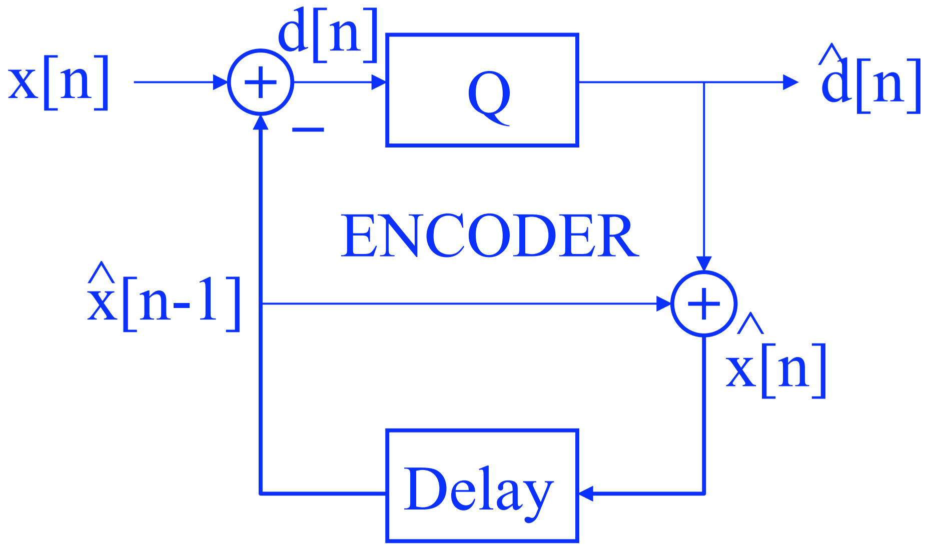
$$\hat{d}_2 = Q(d_2) = d_2 + q_2$$

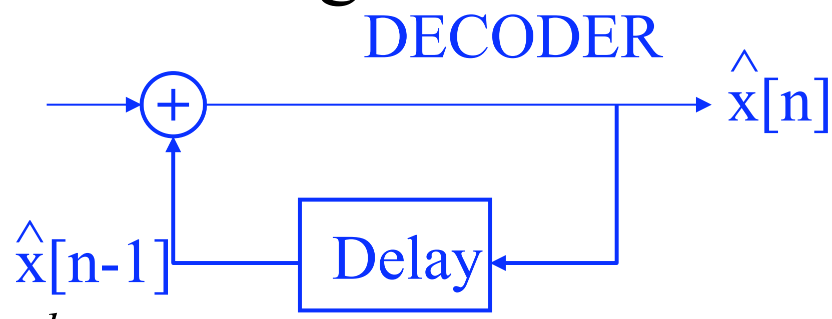
$$\hat{x}_2 = \hat{x}_1 + \hat{d}_2 = x_1 + q_1 + d_2 + q_2 = x_2 + q_1 + q_2$$

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

McMaster University

Differential Encoding





$$d_1 = x_1 - x_0$$

$$\hat{d}_1 = Q(d_1) = d_1 + q_1$$

$$\hat{x}_1 = x_0 + \hat{d}_1 = x_0 + d_1 + q_1 = x_1 + q_1$$

$$d_2 = x_2 - \hat{x}_1$$

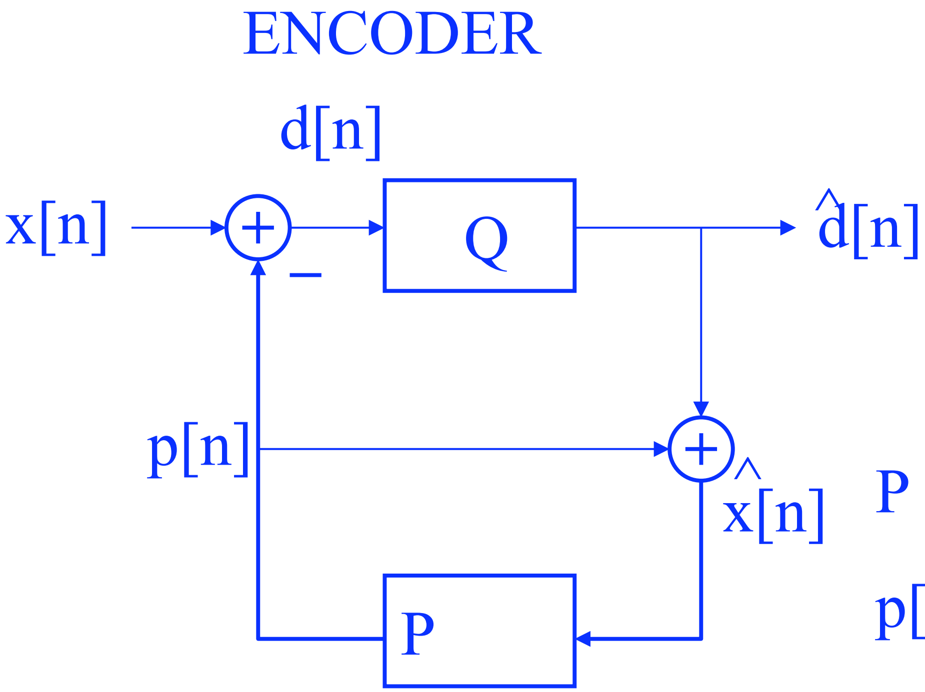
$$\hat{d}_2 = Q(d_2) = d_2 + q_2$$

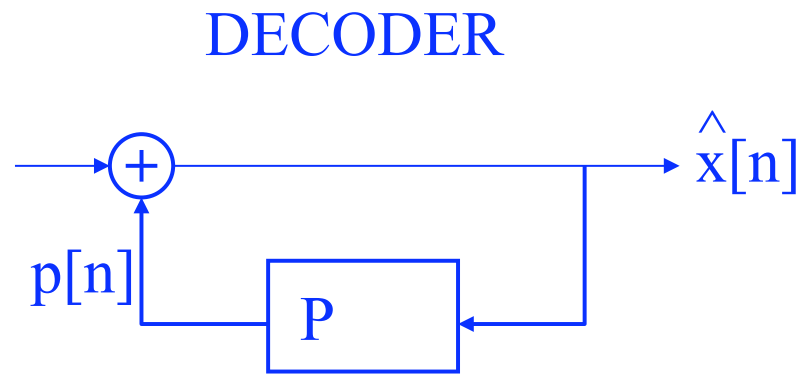
$$\hat{x}_2 = \hat{x}_1 + \hat{d}_2 = \hat{x}_1 + d_2 + q_2 = x_2 + q_2$$

$$\hat{x}_n = x_n + q_n$$

McMaster University

Differential Encoding





P = m-th order predictor

$$p[n] = f(\hat{x}[n-1], \hat{x}[n-2], \ldots, \hat{x}[n-m])$$

Differential Pulse Code Modulation (DPCM)

Optimal Prediction

- Choose prediction function f such that the prediction error is minimized.
- Minimize the expected square error: MMSE prediction.

$$\sigma_d^2 = E[(x[n] - p[n])^2] = E[d[n]^2]$$

- Difficult problem: jointly optimizing the predictor and the quantizer (closed-loop solution).
- Simpler problem: the open-loop solution. Assume fine quantization. Then

$$x[n] = f(x[n-1],x[n-2],\ldots,x[0]).$$

McMaster University

Optimal Prediction

- For stationary signals, the MMSE solution is given by the conditional expected value:

$$p[n] = E[x[n] \mid x[n-1], \ldots, x[0]]$$

- This solution is in general non-linear and requires knowledge of nth-order conditional probabilities which is generally not available.
- Most useful predictors are linear:

$$p[n] = \sum_{i=1}^{m} a_i x[n-i]$$

McMaster University

Optimal Linear Prediction

- Minimize

$$\sigma_d^{\ 2} = E[(x[n] - \sum_{i=1}^{m} a_i x[n-i])^2]$$

$$\frac{\partial \sigma_d^{\ 2}}{\partial a_j} = -2E[(x[n] - \sum_{i=1}^{m} a_i x[n-i])x[n-j]] = 0$$

$$\sum_{i=1}^{m} a_i E[x[n-i]x[n-j]] = E[x[n]x[n-j]]$$

$E[x[n]x[n-k]] = r[k]$ is the autocorrelation function.

$r[k] = r[-k]$

The Wiener-Hopf Equations

$$\sum_{i=1}^{m} a_i r[i-j] = r[j],\ j = 1,\ldots,m$$

$$\begin{bmatrix} r[0] & r[1] & \cdots & r[m-1] \\ r[1] & r[0] & & r[m-2] \\ \vdots & & & \vdots \\ r[m-1] & & \cdots & r[0] \end{bmatrix} \begin{bmatrix} a_1 \\ a_2 \\ \vdots \\ a_m \end{bmatrix} = \begin{bmatrix} r[1] \\ r[2] \\ \vdots \\ r[m] \end{bmatrix}$$

$$\underline{R}\,\underline{a} = \underline{r}$$

The Levinson-Durbin Algorithm

- $\underline{R}$ is a Toeplitz matrix, so an efficient algorithm exists for solving the W-H equations.
- The Levinson-Durbin algorithm iteratively computes the coefficients of the predictors of order 1,2,…,m.
- Advantage: we can increase the order until the prediction error is below some threshold.
- a_j^m are the predictor coefficients of order m
  $E_m = E[d[n]^2]$ is the prediction error for a predictor of order m

The Levinson-Durbin Algorithm

$$1.\ E_0 = r[0], i = 0$$

$$2.\ i = i + 1$$

$$3.\ k_i = \frac{1}{E_{i-1}}\left(\sum_{j=1}^{i-1} a_j^{\,i-1} r[i-j] - r[i]\right)$$

$$4.\ a_i^{\,i} = -k_i$$

$$5.\ a_j^{\,i} = a_j^{\,i-1} + k_i a_{i-j}^{\,i-1} \quad \text{for } j = 1,2,\ldots,i-1$$

$$6.\ E_i = (1 - k_i^{\,2})E_{i-1}$$

$$7.\ \text{If } i < m \text{ go to } (2)$$

McMaster University

The Levinson-Durbin Algorithm

$$E_0 = r[0], i = 0$$

$$i = 1$$

$$k_1 = -\frac{r[1]}{r[0]}$$

$$a_1^1 = -k_1 = \frac{r[1]}{r[0]} \qquad \text{(first-order predictor)}$$

$$E_1 = (1 - k_1^2)E_0 = \frac{r[0]^2 - r[1]^2}{r[0]}$$

McMaster University

The Levinson-Durbin Algorithm

$i = 2$

$$k_2 = \frac{(a_1^1 r[1] - r[2])}{E_1} = \frac{r[1]^2 - r[2]r[0]}{r[0]^2 - r[1]^2}$$

$$a_2^2 = \frac{r[2]r[0] - r[1]^2}{r[0]^2 - r[1]^2}$$

$$a_1^2 = \frac{r[1]r[0] - r[1]r[2]}{r[0]^2 - r[1]^2} \qquad \text{(second - order predictor)}$$

Adaptive DPCM

- DPCM consists of two main components: predictor, quantizer
- Making DPCM adaptive means making predictor and/or quantizer adaptive
- We can adapt a system based on its inputs (forward adaptation) or its outputs (backward adaptation)

Adaptive Quantization in DPCM

- Forward: the input is divided into blocks, quantizer parameters are estimated for each block
- In DPCM the quantizer is in a feedback loop, which means that the input to the quantizer is not conveniently available
- Backward: a variation of the backward adaptive Jayant quantizer

DPCM with adaptive Prediction

- Forward:
- The input is divided into segments or blocks, autocorrelation coefficients are computed for each block, the predictor coefficients are obtained from autocorrelation coefficients and quantized using a high-rate quantizer
- Estimation of autocorrelation for each block:

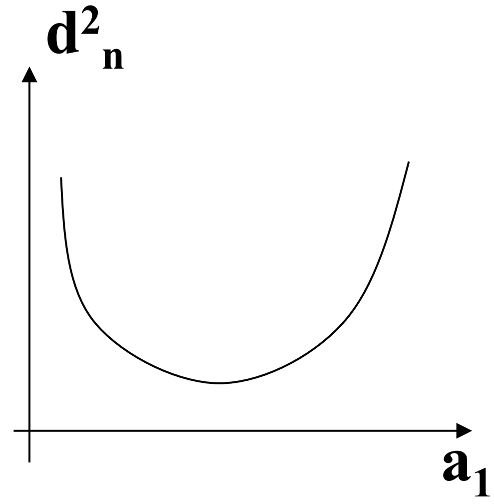
$$R_{xx}^{(l)}(k) = \frac{1}{M-k} \sum_{i=(l-1)M+1}^{lM-k} x_i x_{i+k}$$

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

McMaster University

DPCM with adaptive Prediction

- Backward:
- For a first order predictor $d_n^2 = (x_n - a_1 \hat{x}_{n-1})^2$
- When a_1 is to the left of the optimal value, the derivative (of distortion) is negative
  - To adapt a_1 we should add some to it
- When a_1 is to the right of the optimal value, the derivative is positive
  - To adapt a_1 we should subtract some from it



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

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

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

DPCM with adaptive Prediction

$$A^{(n+1)} = A^{(n)} + \alpha \hat{d}_n \hat{X}_{n-1}$$

$$X_n = \begin{bmatrix} \hat{x}_n \\ \hat{x}_{n-1} \\ \vdots \\ \hat{x}_{n-N+1} \end{bmatrix}$$

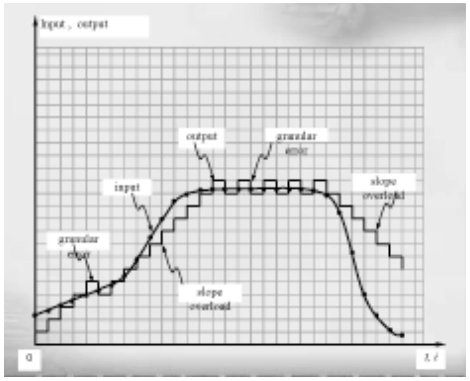
Delta modulator

- Delta modulator: a very simple DPCM system with a 1-bit (2 level) quantizer
- We can only represent sample to sample difference of Δ.
- Substantial distortion if sample to sample difference is very different from Δ
- Solution: sample the signal at a very high rate (several times of the Nyquist rate)

McMaster University

Delta modulator

- In a delta modulator with fixed step size, the reconstructed signal shows one of two behaviors:
  1. in the regions where source is relatively constant, the reconstructed signal alternates up or down by Δ (granular region)
  2. in regions where the source rises or falls fast, the reconstructed output cannot keep up (slope overload regions)
- To reduce the granular error step size should be small but this makes it difficult for reconstructed value to follow rapid changes
- In order to avoid overload error step size should be large but this increases granular error
- Solution: adapt step size



Delta modulator

- Idea of adaptive delta modulator: increase step size in overload regions and decrease it in granular regions
- How to know in which region the system is?
- Granular region: output of quantizer changes sign with almost every input sample
- Overload region: sign of quantizer output is the same for a string of input samples
- Use a history of one sample to decide whether the system is in the overload or granular region

- First approach:

$$s_n = \begin{cases} 1 & if \quad \hat{d}_n > 0 \\ -1 & if \quad \hat{d}_n < 0 \end{cases}$$

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

$$M_1 = \frac{1}{M_2} > 1$$

- Second approach:

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

- β is less than one, α_n is one if J of the last K quantizer outputs had the same sign, otherwise zero

McMaster University

Speech coding

- Differential coding schemes are popular for speech coding.
- G.726 recommendation for adaptive DPCM systems are rates 40,32,24, and 16 kbit.
- Speech is sampled 8000 samples per second.
- Number of levels in quantizer 2^{nb}-1
- Quantizer is a backward adaptive quantizer with an adaptation algorithm that is similar to the Jayant quantizer.