Digital Speech Processing—Lecture 16

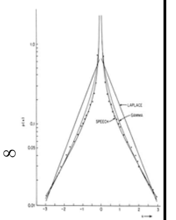
Speech Coding Methods Based on Speech Waveform Representations and Speech Models—Adaptive and Differential Coding

Speech Waveform Coding-Summary of Part 1

1. Probability density function for speech samples

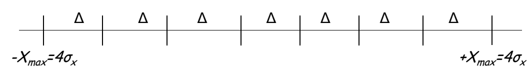
Gamma $p(x)=\frac{1}{\sqrt{2}\sigma_x}e^{-\frac{\sqrt{2}|x|}{\sigma_x}}$ $p(0)=\frac{1}{\sqrt{2}\sigma_x}$

Laplacian $p(x)=\left[\frac{\sqrt{3}}{8\pi\sigma_x|x|}\right]^{1/2}e^{-\frac{\sqrt{3}|x|}{2\sigma_x}}$ $p(0)=\infty$



2. Coding paradigms

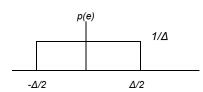
• **uniform** -- divide interval from $+X_{max}$ to $-X_{max}$ into 2^B intervals of length $\Delta = (2X_{max}/2^B)$ for a B-bit quantizer



Speech Waveform Coding-Summary of Part 1

$\hat{x}[n]=x[n]+e[n]$

$SNR=6B+4.77-20\log 10\left[\frac{X_{\max}}{\sigma_x}\right]$

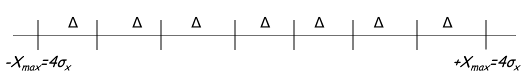


- sensitivity to $X_{\max}/\sigma_x$ (σ_x varies a lot!!!)
- not great use of bits for actual speech densities!

$\left[\frac{X_{\max}}{\sigma_x}\right]$	$20\log 10\left[\frac{X_{\max}}{\sigma_x}\right]$	SNR (uniform) (B=8)
2	6.02	46.75
4	12.04	40.73
8	18.06	34.71
16	24.08	28.69
32	30.10	22.67
64	36.12	16.65

30 dB loss as X_{max}/σ_x varies over a 32:1 range

- $X_{\max}$ (or equivalently σ_x) varies a lot across sounds, speakers, environments
- need to adapt coder ($\Delta[n]$) to time varying σ_x or $X_{\max}$
- key question is how to adapt



Speech Waveform Coding-Summary of Part 1

• **pseudo-logarithmic** (constant percentage error)

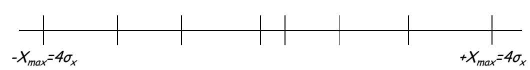
- compress $x[n]$ by pseudo-logarithmic compander
- quantize the companded $x[n]$ uniformly
- expand the quantized signal

$$y[n]=F[x[n]]$$

$$=X_{\max}\frac{\log\left[1+\mu\frac{|x[n]|}{X_{\max}}\right]}{\log(1+\mu)}\cdot sign[x[n]]$$

- large $|x[n]|$

$$|y[n]|\approx\frac{X_{\max}}{\log\mu}\cdot\log\left[\frac{\mu|x[n]|}{X_{\max}}\right]$$



Speech Waveform Coding-Summary of Part 1

$$SNR(dB)=6B+4.77-20\log_{10}[\ln(1+\mu)]-10\log_{10}\left[1+\left(\frac{X_{\max}}{\mu\sigma_x}\right)^2+\sqrt{2}\left(\frac{X_{\max}}{\mu\sigma_x}\right)\right]$$

- insensitive to $X_{\max}/\sigma_x$ over a wide range for large μ

• **maximum *SNR* coding** — match signal quantization intervals to model probability distribution (Gamma, Laplacian)

  • interesting—at least theoretically

Adaptive Quantization

- linear quantization => *SNR* depends on σ_x being constant (this is clearly not the case)
- instantaneous companding => *SNR* only weakly dependent on X_{max}/σ_x for large μ-law compression (100-500)
- optimum *SNR* => minimize σ_e^2 when σ_x^2 is known, non-uniform distribution of quantization levels

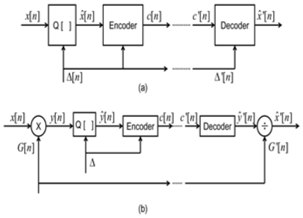
Quantization dilemma: want to choose quantization step size large enough to accomodate maximum peak-to-peak range of $x[n]$; at the same time need to make the quantization step size small so as to minimize the quantization error

– the non-stationary nature of speech (variability across sounds, speakers, backgrounds) compounds this problem greatly

Solutions to Quantization Dilemna

Adaptive Quantization:

- **Solution 1** - let Δ vary to match the variance of the input signal => $\Delta[n]$
- **Solution 2** - use a variable gain, $G[n]$, followed by a fixed quantizer step size, Δ => keep signal variance of $y[n]=G[n]\,x[n]$ constant



Case 1: $\Delta[n]$ proportional to σ_x => quantization levels and ranges would be linearly scaled to match σ_x^2 => need to reliably estimate σ_x^2

Case 2: $G[n]$ proportional to $1/\sigma_x$ to give $\sigma_y^2 \approx$ constant

- *need reliable estimate of σ_x^2 for both types of adaptive quantization*

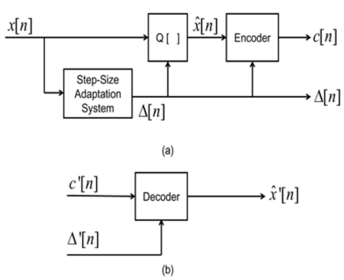
7

Types of Adaptive Quantization

- instantaneous-amplitude changes reflect sample-to-sample variations in $x[n]$ => rapid adaptation
- syllabic-amplitude changes reflect syllable-to-syllable variations in $x[n]$ => slow adaptation
- feed-forward-adaptive quantizers that estimate σ_x^2 from $x[n]$ itself
- feedback-adaptive quantizers that adapt the step size, Δ, on the basis of the quantized signal, $\hat{x}[n]$, (or equivalently the codewords, $c[n]$)

8

Feed Forward Adaptation



Variable step size

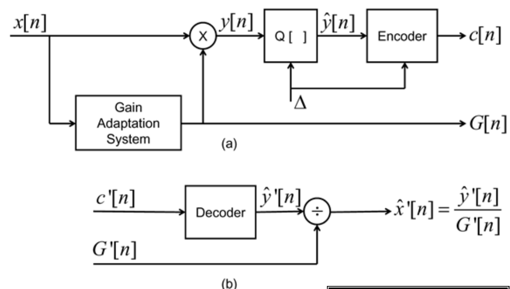
- assume uniform quantizer with step size $\Delta[n]$
- $x[n]$ is quantized using $\Delta[n]$ => $c[n]$ and $\Delta[n]$ need to be transmitted to the decoder
- if $c'[n]=c[n]$ and $\Delta'[n]=\Delta[n]$ => no errors in channel, and

$$\hat{x}'[n] = \hat{x}[n]$$

Don't have $x[n]$ at the decoder to estimate $\Delta[n]$ => need to transmit $\Delta[n]$; this is a major drawback of feed forward adaptation

9

Feed-Forward Quantizer



time varying gain, $G[n]$ => $c[n]$ and $G[n]$ need to be transmitted to the decoder

Can't estimate $G[n]$ at the decoder => it has to be transmitted

10

Feed Forward Quantizers

- feed forward systems make estimates of σ_x^2, then make Δ or the quantization levels proportional to σ_x, or the gain is inversely proportional to σ_x
  - assume $\sigma_x^2 \propto$ short-time energy

$$\sigma^2[n] = \sum_{m=-\infty}^{\infty} x^2[m]\,h[n-m] \Big/ \sum_{m=0}^{\infty} h[m] \quad \text{where } h[n] \text{ is a lowpass filter}$$

$$E\left[\sigma^2[n]\right] \propto \sigma_x^2 \quad \text{(this can be shown)}$$

  - consider $h[n] = \alpha^{n-1} \quad n \ge 1$; $= 0$ *otherwise*

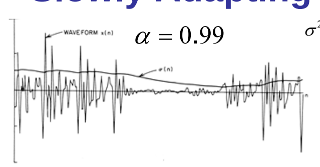
$$\sigma^2[n] = \sum_{m=-\infty}^{n-1} x^2[m]\,\alpha^{n-m-1}(1-\alpha) \quad (0<\alpha<1)$$

$$\sigma^2[n] = \alpha\,\sigma^2[n-1] + x^2[n-1](1-\alpha) \quad \text{(recursion)}$$

  - this gives $\Delta[n] = \Delta_0\,\sigma[n]$ and $G[n] = G_0/\sigma[n]$

11

Slowly Adapting Gain Control



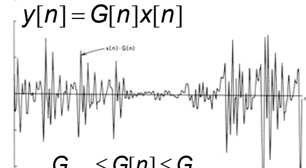
$\alpha = 0.99$

$$\sigma^2[n] = \alpha\sigma^2[n-1] + x^2[n-1](1-\alpha) = \sum_{m=-\infty}^{n-1} \alpha^{n-m-1} x^2[m](1-\alpha)$$

$$G[n] = \frac{G_0}{\sqrt{\sigma^2[n]}}$$

$$y[n] = G[n]x[n]$$

$$\hat{y}[n] = Q_\Delta\{G[n]x[n]\}$$



$$\Delta[n] = \Delta_0\sqrt{\sigma^2[n]}$$

$$\hat{x}[n] = Q_{\Delta[n]}\{x[n]\}$$

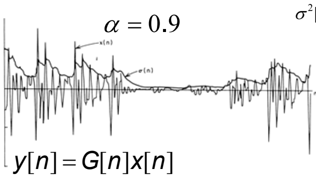
$$G_{\min} \le G[n] \le G_{\max}$$

$$\Delta_{\min} \le \Delta[n] \le \Delta_{\max}$$

$\alpha = 0.99$ => brings up level in low amplitude regions => time constant of 100 samples (12.5 msec for 8 kHz sampling rate) => syllabic rate

Rapidly Adapting Gain Control

$\alpha = 0.9$

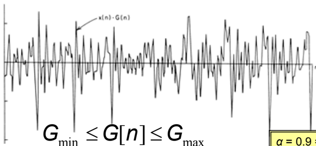


$$\sigma^2[n] = \alpha\sigma^2[n-1] + x^2[n-1](1-\alpha)$$
$$= \sum_{m=-\infty}^{n-1} \alpha^{n-m-1} x^2[m](1-\alpha)$$

$$G[n] = \frac{G_0}{\sqrt{\sigma^2[n]}}$$

$y[n] = G[n]x[n]$

$$\hat{y}[n] = Q_\Delta\{G[n]x[n]\}$$



$$\Delta[n] = \Delta_0\sqrt{\sigma^2[n]}$$

$$\hat{x}[n] = Q_{\Delta[n]}\{x[n]\}$$

$$G_{\min} \le G[n] \le G_{\max}$$
$$\Delta_{\min} \le \Delta[n] \le \Delta_{\max}$$

$\alpha = 0.9$ => system reacts to amplitude variations more rapidly => provides better approximation to σ_y^2 = constant => time constant of 9 samples (1 msec at 8 kHz) for change => instantaneous rate

Feed Forward Quantizers

- $\Delta[n]$ and $G[n]$ vary slowly compared to $x[n]$
  - they must be sampled and transmitted as part of the waveform coder parameters
  - rate of sampling depends on the bandwidth of the lowpass filter, $h[n]$—for $\alpha = 0.99$, the rate is about 13 Hz; for $\alpha = 0.9$, the rate is about 135 Hz
- it is reasonable to place limits on the variation of $\Delta[n]$ or $G[n]$, of the form

$$G_{\min} \le G[n] \le G_{\max}$$
$$\Delta_{\min} \le \Delta[n] \le \Delta_{\max}$$

- for obtaining $\sigma_y^2 \approx$ constant over a 40 dB range in signal levels =>

$$\frac{G_{\max}}{G_{\min}} = \frac{\Delta_{\max}}{\Delta_{\min}} = 100 \quad (40\,\text{dB range})$$

Feed Forward Adaptation Gain

$$\sigma^2[n] = \frac{1}{M}\sum_{m=n-M+1}^{n} x^2[m]$$

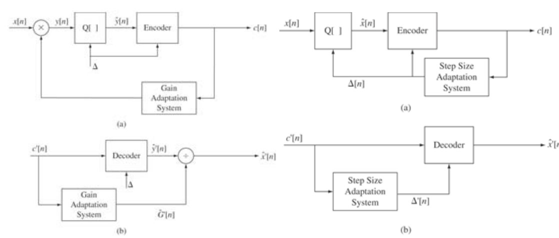
- $\Delta[n]$ or $G[n]$ evaluated every M samples
- used $M = 128$, 1024 samples for estimates
- adaptive quantizer achieves up to 5.6 dB better SNR than non-adaptive quantizers
- can achieve this SNR with low "idle channel noise" and wide speech dynamic range by suitable choice of $\Delta_{\min}$ and $\Delta_{\max}$

Table 5.4 Adaptive 3-bit Quantization with Feed-forward Adaptation. (After Noll [12].)

Nonuniform Quantizers	*Nonadaptive SNR (dB)*	*Adaptive (M=128) SNR (dB)*	*Adaptive (M=1024) SNR (dB)*
μ-law (μ=100, $X_{\max}=8\sigma_x$)	9.5	–	–
Gaussian	7.3	15.0	12.1
Laplace	9.9	13.3	12.8
Uniform Quantizers			
Gaussian	6.7	14.7	11.3
Laplace	7.4	13.4	11.5

feed-forward adaptation gain with B=3—less gain for M=1024 than M=128 by 3 dB => M=1024 is too long an interval

Feedback Adaptation



- $\sigma^2[n]$ estimated from quantizer output (or the code words)
- advantage of feedback adaptation is that neither $\Delta[n]$ nor $G[n]$ needs to be transmitted to the decoder since they can be derived from the code words
- disadvantage of feedback adaptation is increased sensitivity to errors in codewords, since such errors affect $\Delta[n]$ and $G[n]$

Feedback Adaptation

$$\sigma^2[n] = \sum_{m=-\infty}^{\infty} \hat{x}^2[m]\, h[n-m] \Big/ \sum_{m=0}^{\infty} h[m]$$

- $\sigma^2[n]$ based only on past values of $\hat{x}[n]$
- two typical windows/filters are

1. $h[n] = \alpha^{n-1} \quad n \ge 1$
   $\quad = 0 \quad$ *otherwise*
2. $h[n] = 1/M \quad 1 \le n \le M$
   $\quad = 0 \quad$ *otherwise*

$$\sigma^2[n] = \frac{1}{M}\sum_{m=n-M}^{n-1} \hat{x}^2[m]$$

- can use very short window lengths (e.g., $M = 2$) to achieve 12 dB SNR for a $B = 3$ bit quantizer

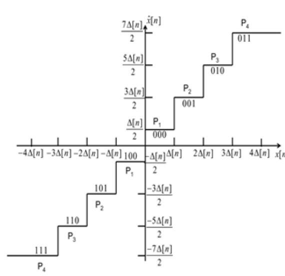
Alternative Approach to Adaptation

$$\Delta[n] = P \cdot \Delta[n-1]; \; P = \{P_1, P_2, P_3, P_4\}$$
$$P \propto |c[n-1]|$$
$$\hat{x}(n) = \frac{\Delta[n]\, sign[c[n]]}{2} + \Delta[n]\, c[n]$$

- $\Delta[n]$ only depends on $\Delta[n-1]$ and $c[n-1]$
- => only need to transmit codewords
- also necessary to impose the limits

$$\Delta_{\min} \le \Delta[n] \le \Delta_{\max}$$

- the ratio $\Delta_{\max} / \Delta_{\min}$ controls the dynamic range of the quantizer

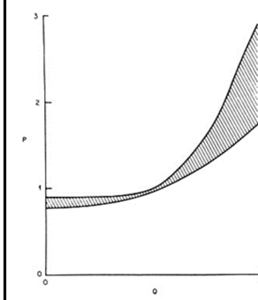


Adaptation Gain

- key issue is how should P vary with $|c[n-1]|$
  - If c[n-1] is either largest positive or largest negative codeword, then quantizer is overloaded and the quantizer step size is too small => $P_4 > 1$
  - if $c[n-1]$ is either smallest positive or negative codeword, then quantization error is too large => $P_1 < 1$
  - need choices for P_2 and P_3

19

Adaptation Gain



$$Q = \frac{1+2|c[n-1]|}{2^B - 1}$$

- shaded area is variation in range of P values due to different speech sounds or different B values

Can see that step size increases ($P>1$) are more vigorous than step size decreases ($P<1$) since signal growth needs to be kept within quantizer range to avoid 'overloads'

20

Optimal Step Size Multipliers

Table 5.5 Step Size Multipliers For Adaptive Quantization Methods. (After Jayant [15].)

	Coder Type	
B	*PCM*	*DPCM*
2	0.6, 2.2	0.8, 1.6
3	0.85, 1, 1, 1.5	0.9, 0.9, 1.25 1.75
4	0.8, 0.8, 0.8, 0.8, 1.2, 1.6, 2.0, 2.4	0.9, 0.9, 0.9, 0.9, 1.2, 1.6, 2.0, 2.4
5	0.85, 0.85, 0.85, 0.85, 0.85, 0.85, 0.85, 0.85, 1.2, 1.4, 1.6, 1.8, 2.0, 2.2, 2.4, 2.6	0.9, 0.9, 0.9, 0.9, 0.95, 0.95, 0.95, 0.95, 1.2, 1.5, 1.8, 2.1, 2,4, 2.7, 3.0, 3.3

Table 5.6 Improvements in Signal-to-Noise Ratio Using Optimum Step Size Multipliers for Adaptive Quantization. (After Jayant [15].)

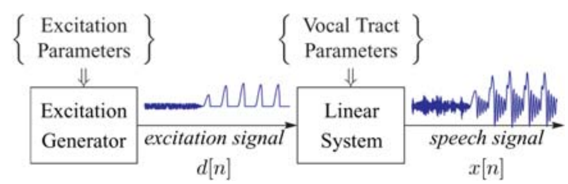
B	*Logarithmic PCM with μ-law (μ=100) Quantization*	*Adaptive PCM with Uniform Quantization*
2	3 db	9 db
3	8 db	15 db
4	15 db	19 db

- optimal values of P for B=2,3,4,5

- improvements in *SNR*

- 4-7 dB improvement over μ-law
- 2-4 dB improvement over non-adaptive optimum quantizers

Quantization of Speech Model Parameters



- Excitation and vocal tract (linear system) are characterized by sets of parameters which can be estimated from a speech signal by LP or cepstral processing.
- We can use the set of estimated parameters to synthesize an approximation to the speech signal whose quality depends of a range of factors.

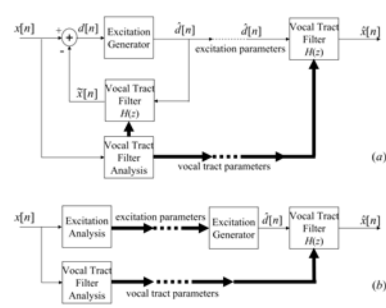
22

Quantization of Speech Model Parameters

- Quality and data rate of synthesis depends on:
  - the ability of the model to represent speech
  - the ability to reliably and accurately estimate the parameters of the model
  - the ability to quantize the parameters in order to obtain a low data rate digital representation that will yield a high quality reproduction of the speech signal

23

Closed-Loop and Open-Loop Speech Coders



Closed-loop – used in a feedback loop where the synthetic speech output is compared to the input signal, and the resulting difference used to determine the excitation for the vocal tract model.

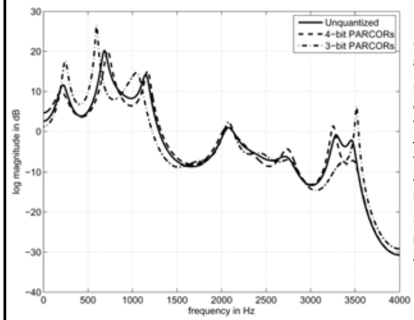
Open-loop – the parameters of the model are estimated directly from the speech signal with no feedback as to the quality of the resulting synthetic speech.

24

Scalar Quantization

- Scalar quantization – treat each model parameter separately and quantize using a fixed number of bits
  - need to measure (estimate) statistics of each parameter, i.e., mean, variance, minimum/maximum value, pdf, etc.
  - each parameter has a different quantizer with a different number of bits allocated
- Example of scalar quantization
  - pitch period typically ranges from 20-150 samples (at 8 kHz sampling rate) => need about 128 values (7-bits) uniformly over the range of pitch periods, including value of zero for unvoiced/background
  - amplitude parameter might be quantized with a μ-law quantizer using 4-5 bits per sample
  - using a frame rate of 100 frames/sec, you would need about 700 bps for pitch period and 400-500 bps for amplitude

Scalar Quantization



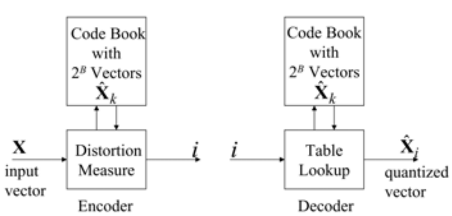
- 20-th order LPC analysis frame
- Each PARCOR coefficient transformed to range: $-\pi/2 < \sin^{-1}(k_i) < \pi/2$ and then quantized with both a 4-bit and a 3-bit uniform quantizer.
- Total rate of quantized representation of speech about 5000 bps.

Techniques of Vector Quantization

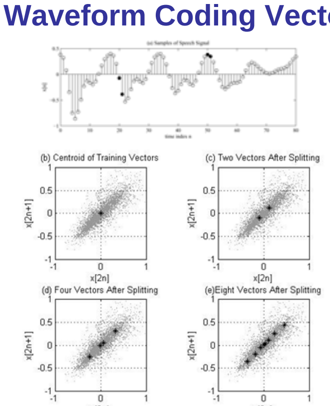
Vector Quantization

- code block of scalars as a vector, rather than individually
- design an optimal quantization method based on mean-squared distortion metric
- essential for model-based and hybrid coders

Vector Quantization



Waveform Coding Vector Quantizer



VQ code pairs of waveform samples,

X[n]=(x[2n],x[2n+1]);

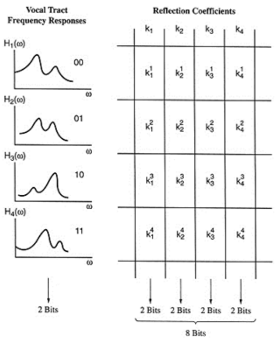
(b) Single element codebook with cluster centroid (0-bit codebook)

(c) Two element codebook with two cluster centers (1-bit codebook)

(d) Four element codebook with four cluster centers (2-bit codebook)

(e) Eight element codebook with eight cluster centers (3-bit codebook)

Toy Example of VQ Coding



• 2-pole model of the vocal tract => 4 reflection coefficients

• 4-possible vocal tract shapes => 4 sets of reflection coefficients

1. ***Scalar Quantization*** -assume 4 values for each reflection coefficient => 2-bits x 4 coefficients = 8 bits/frame

2. ***Vector Quantization*** -only 4 possible vectors => 2-bits to choose which of the 4 vectors to use for each frame (pointer into a codebook)

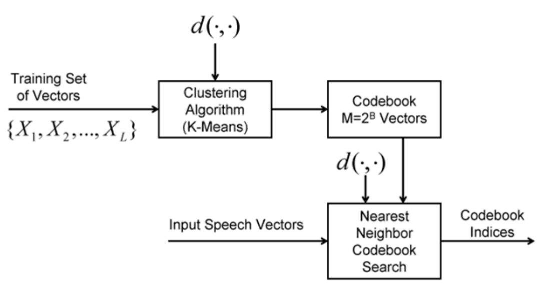
• this works because the scalar components of each vector are highly correlated

• if scalar components are independent => VQ offers no advantage over scalar quantization

Elements of a VQ Implementation

1. A large training set of analysis vectors; $X=\{X_1,X_2,\ldots,X_L\}$, L should be much larger than the size of the codebook, M, i.e., 10-100 times the size of M.
2. A measure of distance, $d_{ij}=d(X_i,X_j)$, between a pair of analysis vectors, both for clustering the training set as well as for classifying test set vectors into unique codebook entries.
3. A centroid computation procedure and a centroid splitting procedure.
4. A classification procedure for arbitrary analysis vectors that chooses the codebook vector closest in distance to the input vector, providing the codebook index of the resulting nearest codebook vector.

VQ Implementation



The VQ Training Set

- The VQ training set of $L \geq 10M$ vectors should span the anticipated range of:
  - talkers, ranging in age, accent, gender, speaking rate, speaking levels, etc.
  - speaking conditions, range from quiet rooms, to automobiles, to noisy work places
  - transducers and transmission systems, including a range of microphones, telephone handsets, cellphones, speakerphones, etc.
  - speech, including carefully recorded material, conversational speech, telephone queries, etc.

The VQ Distance Measure

- The VQ distance measure depends critically on the nature of the analysis vector, X.
  - If X is a log spectral vector, then a possible distance measure would be an L_p log spectral distance, of the form:

$$d(X_i, X_j) = \left[\sum_{k=1}^{R} | x_i^k - x_j^k |^p \right]^{1/p}$$

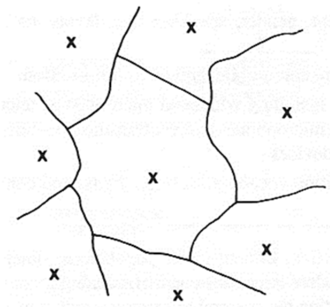
• If X is a cepstral vector, then the distance measure might well be a cepstral distance of the form:

$$d(X_i, X_j) = \left[\sum_{k=1}^{R} (x_i^k - x_j^k)^2 \right]^{1/2}$$

Clustering Training Vectors

- Goal is to cluster the set of *L* training vectors into a set of *M* codebook vectors using generalized Lloyd algorithm (also known as the K-means clustering algorithm) with the following steps:
  1. Initialization – arbitrarily choose *M* vectors (initially out of the training set of *L* vectors) as the initial set of codewords in the codebook
  2. Nearest Neighbor Search – for each training vector, find the codeword in the current codebook that is closest (in distance) and assign that vector to the corresponding cell
  3. Centroid Update – update the codeword in each cell to the centroid of all the training vectors assigned to that cell in the current iteration
  4. Iteration – repeat steps 2 and 3 until the average distance between centroids at successive iterations falls below a preset threshold

Clustering Training Vectors



Voronoi regions and centroids

Centroid Computation

 Assume we have a set of V vectors,

$$X^C = \{X_1^C, X_2^C, \ldots, X_V^C\}$$

where all V vectors are assigned to cluster C.

 The centroid of the set X^C is defined as the vector $\bar{Y}$ that minimizes the average distortion, i.e.,

$$\bar{Y} = \min_{Y} \frac{1}{V} \sum_{i=1}^{V} d(X_i^C, Y)$$

 The solution for the centroid is highly dependent on the choice of distance measure. When both X_i^C and Y are measured in a K-dimensional space with the L_2 norm, the centroid is the mean of the vector set

$$\bar{Y} = \frac{1}{V} \sum_{i=1}^{V} X_i^C$$

 When using an L_1 distance measure, the centroid is the median vector of the set of vectors assigned to the given class.

Vector Classification Procedure

 The classification procedure for arbitrary test set vectors is a full search through the codebook to find the "best" (minimum distance) match.

 If we denote the codebook vectors of an M-vector codebook as CB_i for $1 \le i \le M$, and we denote the vector to be classified (and vector quantized) as X, then the index, $i*$, of the best codebook entry is:

$$i* = \arg \min_{1 \le i \le M} d(X, CB_i)$$

Binary Split Codebook Design

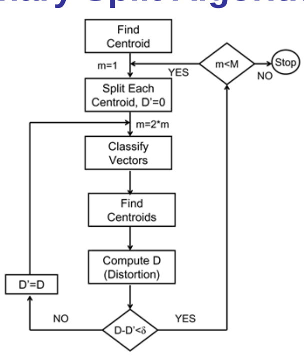
1. Design a 1-vector codebook; the single vector in the codebook is the centroid of the entire set of training vectors
2. Double the size of the codebook by splitting each current codebook vector, Y_m, according to the rule:

   $$Y_m^+ = Y_m(1+\varepsilon)$$

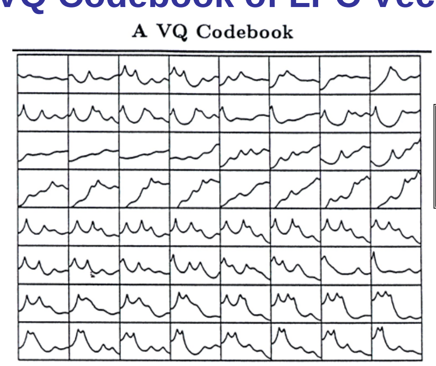
   $$Y_m^- = Y_m(1-\varepsilon)$$

   where *m* varies from 1 to the size of the current codebook, and epsilon is a splitting parameter (0.01 typically)
3. Use the *K*-means clustering algorithm to get the best set of centroids for the split codebook
4. Iterate steps 2 and 3 until a codebook of size *M* is designed.

Binary Split Algorithm

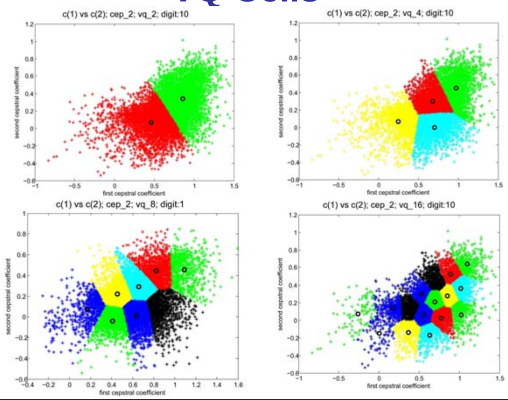


VQ Codebook of LPC Vectors

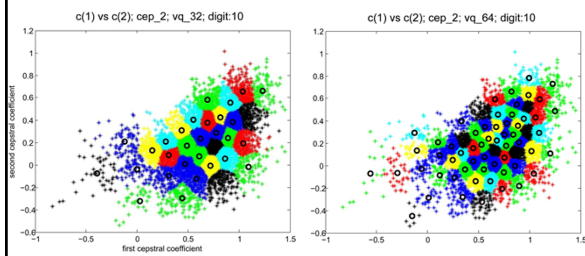


64 vectors in a codebook of spectral shapes

VQ Cells

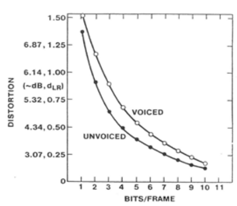


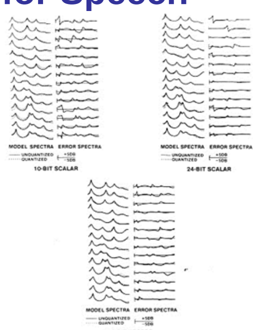
VQ Cells



44

VQ Coding for Speech





'distortion' in coding computed using a spectral distortion measure related to the difference in log spectra between the original and the codebook vectors

10-bit VQ comparable to 24-bit scalar quantization for these examples

Differential Quantization

Theory and Practice

46

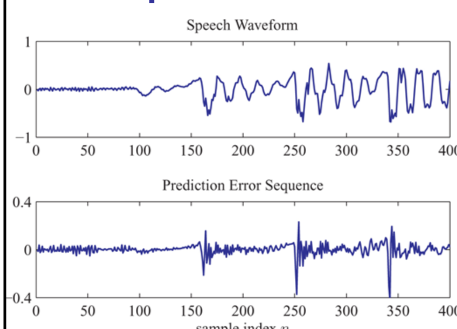
Differential Quantization

- we have carried instantaneous quantization of $x[n]$ as far as possible
- time to consider correlations between speech samples separated in time => differential quantization
- high correlation values => signal does not change rapidly in time => difference between adjacent samples should have lower variance than the signal itself

differential quantization can increase *SNR* at a given bit rate, or lower bit rate for a given *SNR*

47

Example of Difference Signal

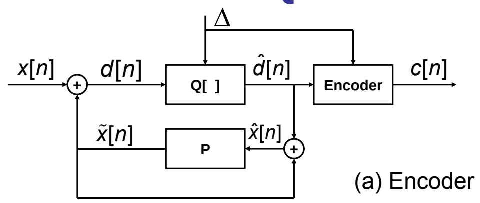


Prediction error from LPC analysis using p=12

Prediction error is about 2.5 times smaller than signal

48

Differential Quantization



$d[n] = x[n] - \tilde{x}[n]$

- where $x[n]$ = unquantized input sample

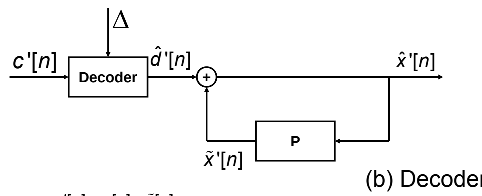
  $\tilde{x}[n]$ = estimate or prediction of $x[n]$

  $\tilde{x}[n]$ is the output of a predictor system, P, whose input is $\hat{x}[n]$, a quantized version of $x[n]$

  $d[n]$ = prediction error signal

  $\hat{d}[n]$ = quantized difference (prediction error) signal

Differential Quantization



$d[n] = x[n] - \tilde{x}[n]$

- where $x[n]$ = unquantized input sample

  $\tilde{x}[n]$ = estimate or prediction of $x[n]$

  $\tilde{x}[n]$ is the output of a predictor system, P, whose input is $\hat{x}[n]$, a quantized version of $x[n]$

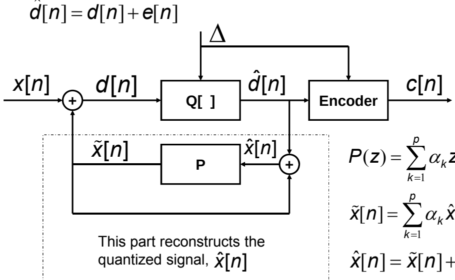
  $d[n]$ = prediction error signal

  $\hat{d}[n]$ = quantized difference (prediction error) signal

Differential Quantization

$d[n] = x[n] - \tilde{x}[n]$

$\hat{d}[n] = d[n] + e[n]$



$$P(z) = \sum_{k=1}^{p} \alpha_k z^{-k}$$

$$\tilde{x}[n] = \sum_{k=1}^{p} \alpha_k \hat{x}[n-k]$$

$$\hat{x}[n] = \tilde{x}[n] + \hat{d}[n]$$

$$\Rightarrow \hat{x}[n] = x[n] + e[n]$$

Differential Quantization

- difference signal, $d[n]$, is quantized - not $x[n]$
- quantizer can be fixed, or adaptive, uniform or non-uniform
- quantizer parameters are adjusted to match the variance of $d[n]$

$\hat{d}[n] = d[n] + e[n]$ $\quad - e[n]$ quantization error

$\hat{x}[n] = \tilde{x}[n] + \hat{d}[n]$ $\quad$ – predicted x plus quantized d

$\hat{x}[n] = x[n] + e[n]$ $\quad$ – quantized input has same quantization error as the difference signal => if $\sigma_d^2 < \sigma_x^2$, error is smaller

- independent of predictor, P, quantized $x[n]$ differs from unquantized $x[n]$ by $e[n]$, the quantization error of the difference signal!

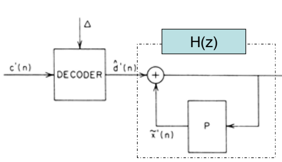
=> good prediction gives lower quantization error than quantizing input directly

Differential Quantization

- quantized difference signal is encoded into *c(n)* for transmission

$$\hat{X}'(z) = \hat{D}'(z) + P(z)\hat{X}'(z)$$

$$\hat{X}'(z) = \frac{1}{1 - P(z)}\hat{D}'(z) = H(z)\hat{D}'(z)$$



$$H(z) = \frac{1}{1 - \sum_{k=1}^{p} \alpha_k z^{-k}}$$

- first reconstruct the quantized difference signal from the decoder codeword, $c'[n]$ and the step size Δ
- next reconstruct the quantized input signal using the same predictor, P, as used in the encoder

SNR for Differential Quantization

- the SNR of the differential coding system is

$$SNR = \frac{E\left[x^2[n]\right]}{E\left[e^2[n]\right]} = \frac{\sigma_x^2}{\sigma_e^2}$$

$$\boxed{SNR = \frac{\sigma_x^2}{\sigma_d^2} \cdot \frac{\sigma_d^2}{\sigma_e^2} = G_P \cdot SNR_Q}$$

- where

$$SNR_Q = \frac{\sigma_d^2}{\sigma_e^2} = \text{signal-to-quantizing-noise ratio of the quantizer}$$

$$G_P = \frac{\sigma_x^2}{\sigma_d^2} = \text{gain due to differential quantization}$$

SNR for Differential Quantization

- SNR_Q depends on chosen quantizer and can be maximized using all of the previous quantization methods (uniform, non-uniform, optimal)
- G_P, hopefully, >1, is the gain in SNR due to differential coding
- want to choose the predictor, P, to maximize G_P $\Rightarrow$ since σ_x^2 is fixed, then we need to minimize σ_d^2, i.e., design the best predictor P

55

Predictor for Differential Quantization

- consider class of linear predictors, P

$$\tilde{x}[n] = \sum_{k=1}^{p} \alpha_k \hat{x}[n-k]$$

- $\tilde{x}[n]$ is a linear combination of previous quantized values of $x[n]$
- the predictor z-transform is

$$P(z) = \sum_{k=1}^{p} \alpha_k z^{-k} = 1 - A(z) \quad \text{-- predictor system function}$$

- with predictor impulse response coefficients (FIR filter)

$$p[n] = \alpha_k \quad 1 \le k \le p$$
$$= 0 \quad \textit{otherwise}$$

56

Predictor for Differential Quantization

- the reconstructed signal is the output, $\hat{x}[n]$, of a system with system function

$$H(z) = \frac{1}{1 - \sum_{k=1}^{p} \alpha_k z^{-k}} = \frac{\hat{X}(z)}{\hat{D}(z)} = \frac{1}{1 - P(z)} = \frac{1}{A(z)}$$

- where the input to the system is the quantized difference signal, $\hat{d}[n]$

$$\hat{d}[n] = \hat{x}[n] - \sum_{k=1}^{p} \alpha_k \hat{x}[n-k]$$

- where

$$\sum_{k=1}^{p} \alpha_k \hat{x}[n-k] = \tilde{x}[n]$$

57

Predictor for Differential Quantization

- to solve for optimum predictor, need expression for σ_d^2

$$\sigma_d^2 = E\left[d^2[n]\right] = E\left[\left(x[n] - \tilde{x}[n]\right)^2\right]$$

$$= E\left[\left[x[n] - \sum_{k=1}^{p} \alpha_k \hat{x}[n-k]\right]^2\right]$$

$$= E\left[\left(x[n] - \sum_{k=1}^{p} \alpha_k x[n-k] - \sum_{k=1}^{p} \alpha_k e[n-k]\right)^2\right]$$

$$(\hat{x}[n] = x[n] + e[n])$$

58

Solution for Optimum Predictor

- want to choose $\{\alpha_j\}$, $1 \le j \le p$, to minimize σ_d^2 =>

differentiate σ_d^2 wrt α_j, set derivatives to zero, giving

$$\frac{\partial \sigma_d^2}{\partial \alpha_j} = -2E\left[\left(x[n] - \sum_{k=1}^{p} \alpha_k \left(x[n-k] + e[n-k]\right) \cdot \left(x[n-j] + e[n-j]\right)\right)\right]$$
$$= 0 \qquad 1 \le j \le p$$

- which can be written in the more compact form

$$E\left[\left(x[n] - \tilde{x}[n]\right)\hat{x}[n-j]\right] = E\left[d[n] \cdot \hat{x}[n-j]\right] = 0, \quad 1 \le j \le p$$

- the predictor coefficients that minimize σ_d^2 are the ones that make the difference signal, $d[n]$, be uncorrelated with past values of the predictor input, $\hat{x}[n-j]$, $1 \le j \le p$

59

Solution for Alphas

$$E\left[\left(x[n] - \tilde{x}[n]\right)\hat{x}[n-j]\right] = E\left[d[n] \cdot \hat{x}[n-j]\right] = 0, \quad 1 \le j \le p$$

- basic equations of differential coding

$\hat{d}[n] = d[n] + e[n]$ — quantization of difference signal

$\hat{x}[n] = x[n] + e[n]$ — error same for original signal

$\hat{x}[n] = \tilde{x}[n] + \hat{d}[n]$ — feedback loop for signal

$\tilde{x}[n] = \sum_{k=1}^{p} \alpha_k \hat{x}[n-k]$ — prediction loop based on quantized input

$\hat{d}[n] = \hat{x}[n] - \sum_{k=1}^{p} \alpha_k \hat{x}[n-k]$ — direct substitution

$$\hat{x}[n-j] = x[n-j] + e[n-j]$$

$$\tilde{x}[n] = \sum_{k=1}^{p} \alpha_k \hat{x}[n-k] = \sum_{k=1}^{p} \alpha_k \left[x[n-k] + e[n-k]\right]$$

60

Solution for Alphas

$$E\left[\left(x[n]-\tilde{x}[n]\right)\hat{x}[n-j]\right]=E\left[d[n]\cdot\hat{x}[n-j]\right]=0,\quad 1\le j\le p$$

$$\hat{x}[n-j]=x[n-j]+e[n-j]$$

$$\tilde{x}[n]=\sum_{k=1}^{p}\alpha_k\hat{x}[n-k]=\sum_{k=1}^{p}\alpha_k\left[x[n-k]+e[n-k]\right]$$

$$E\left[x[n]x[n-j]\right]+E\left[x[n]e[n-j]\right]-E\left[\tilde{x}[n]x[n-j]\right]-E\left[\tilde{x}[n]e[n-j]\right]$$

$$=E\left[\sum_{k=1}^{p}\alpha_k\left[x[n-k]+e[n-k]\right]x[n-j]\right]+$$

$$E\left[\sum_{k=1}^{p}\alpha_k\left[x[n-k]+e[n-k]\right]e[n-j]\right]$$

$$=\sum_{k=1}^{p}\alpha_k E\left[x[n-k]x[n-j]\right]+\sum_{k=1}^{p}\alpha_k E\left[e[n-k]x[n-j]\right]+$$

$$\sum_{k=1}^{p}\alpha_k E\left[x[n-k]e[n-j]\right]+\sum_{k=1}^{p}\alpha_k E\left[e[n-k]e[n-j]\right]$$

Solution for Optimum Predictor

- solution for α_k - first expand terms to give

$$E\left[x[n-j]\cdot x[n]\right]+E\left[e[n-j]\cdot x[n]\right]=\sum_{k=1}^{p}\alpha_k E\left[x[n-j]\cdot x[n-k]\right]$$

$$+\sum_{k=1}^{p}\alpha_k E\left[e[n-j]\cdot x[n-k]\right]+\sum_{k=1}^{p}\alpha_k E\left[x[n-j]\cdot e[n-k]\right]$$

$$+\sum_{k=1}^{p}\alpha_k E\left[e[n-j]\cdot e[n-k]\right],\qquad 1\le j\le p$$

- assume fine quantization so that $e[n]$ is uncorrelated with $x[n]$, and $e[n]$ is stationary white noise (zero mean), giving

$$E\left[x[n-j]\cdot e[n-k]\right]=0\qquad\forall n,j,k$$

$$E\left[e[n-j]\cdot e[n-k]\right]=\sigma_e^2\cdot\delta[j-k]$$

Solution for Optimum Predictor

- we can now simplify solution to form

$$\phi[j]=\sum_{k=1}^{p}\alpha_k\left[\phi[j-k]+\sigma_e^2\delta[j-k]\right],\quad 1\le j\le p$$

- where $\phi[j]$ is the autocorrelation of $x[n]$. Defining terms

$$\rho[j]=\frac{\phi[j]}{\sigma_x^2}=\sum_{k=1}^{p}\alpha_k\left[\rho[j-k]+\frac{\sigma_e^2}{\sigma_x^2}\delta[j-k]\right],\quad 1\le j\le p$$

- or in matrix form

$$\rho=C\alpha$$

$$\rho=\begin{bmatrix}\rho[1]\\ \rho[2]\\ .\\ .\\ \rho[p]\end{bmatrix},\quad \alpha=\begin{bmatrix}\alpha_1\\ \alpha_2\\ .\\ .\\ \alpha_p\end{bmatrix},\quad C=\begin{bmatrix}1+1/SNR & \rho[1] & \rho[p-1]\\ \rho[1] & 1+1/SNR & \rho[p-2]\\ & & \\ \rho[p-1] & \rho[p-2] & 1+1/SNR\end{bmatrix}$$

Solution for Optimum Predictor

$$\rho=C\alpha$$

$$\rho=\begin{bmatrix}\rho[1]\\ \rho[2]\\ .\\ .\\ \rho[p]\end{bmatrix},\quad \alpha=\begin{bmatrix}\alpha_1\\ \alpha_2\\ .\\ .\\ \alpha_p\end{bmatrix},\quad C=\begin{bmatrix}1+1/SNR & \rho[1] & \rho[p-1]\\ \rho[1] & 1+1/SNR & \rho[p-2]\\ & & \\ \rho[p-1] & \rho[p-2] & 1+1/SNR\end{bmatrix}$$

- with matrix solution

$\alpha=C^{-1}\rho$ (defining $SNR=\sigma_x^2/\sigma_e^2$)

- where C is a Toeplitz matrix => C^{-1} can be computed via well understood numerical methods
- the problem here is that C depends on $SNR=\sigma_x^2/\sigma_e^2$, but SNR depends on α_k coefficients of the predictor, which depend on SNR => bit of a dilemma

Solution for Optimum Predictor

- special case of $p=1$, where we can solve directly for α_1 of this first order linear predictor, as

$$\alpha_1=\frac{\rho[1]}{1+1/SNR}$$

- can see that $\alpha_1<\rho[1]<1$
- we will look further at this special case later

Solution for Optimum Predictor

- in spite of problems in solving for optimum predictor coefficients, we can solve for the prediction gain, G_P, in terms of the α_j coefficients, as

$$\sigma_d^2=E\left[\left(x[n]-\tilde{x}[n]\right)\cdot\left(x[n]-\tilde{x}[n]\right)\right]$$

$$=E\left[\left(x[n]-\tilde{x}[n]\right)\cdot x[n]\right]-E\left[\left(x[n]-\tilde{x}[n]\right)\cdot\tilde{x}[n]\right]$$

- where the term $\left(x[n]-\tilde{x}[n]\right)$ is the prediction error; we can show that the second term in the expression above is zero, i.e., the prediction error is uncorrelated with the prediction value; thus

$$\sigma_d^2=E\left[\left(x[n]-\tilde{x}[n]\right)\cdot x[n]\right]$$

$$=E\left[x^2[n]\right]-E\left[\sum_{k=1}^{p}\alpha_k\left(x[n-k]+e[n-k]\right)\cdot x[n]\right]$$

- assuming uncorrelated signal and noise, we get

$$\sigma_d^2=\sigma_x^2-\sum_{k=1}^{p}\alpha_k\phi[k]=\sigma_x^2\left[1-\sum_{k=1}^{p}\alpha_k\rho[k]\right]$$

$$(G_P)_{opt}=\frac{1}{1-\sum_{k=1}^{p}\alpha_k\rho[k]}\quad\text{for optimum values of }\alpha_k$$

First Order Predictor Solution

- For the case $p=1$ we can examine effects of sub-optimum value of α_1 on the quantity $G_P = \sigma_x^2 / \sigma_d^2$.
- The optimum solution is:

$$(G_P)_{opt} = \frac{1}{1-\alpha_1 \rho[1]}$$

- Consider choosing an arbitrary value for α_1; then we get

$$\sigma_d^2 = \sigma_x^2\left[1-2\alpha_1\rho[1]+\alpha_1^2\right]+\alpha_1^2\sigma_e^2$$

- Giving the sub-optimum result

$$(G_P)_{arb} = \frac{1}{1-2\alpha_1\rho[1]+\alpha_1^2(1+1/SNR)}$$

- Where the term α_1^2 / SNR represents the increase in variance of $d[n]$ due to the feedback of the error signal $e[n]$.

67

First Order Predictor Solution

- Can reformulate $(G_P)_{arb}$ as

$$(G_P)_{arb} = \frac{1-\dfrac{\alpha_1^2}{SNR_Q}}{1-2\alpha_1\rho[1]+\alpha_1^2}$$

- for any value of α_1 (including the optimum value).
- Consider the case of $\alpha_1 = \rho(1)$

$$(G_P)_{subopt} = \frac{1-\dfrac{\rho^2[1]}{SNR_Q}}{1-\rho^2[1]} = \left[\frac{1}{1-\rho^2[1]}\right]\cdot\left[1-\frac{\rho^2[1)}{SNR_Q}\right]$$

- the gain in prediction is a product of the prediction gain without the quantizer, reduced by the loss due to feedback of the error signal.

68

First Order Predictor Solution

- We showed before that the optimum value of α_1 was $\rho[1]/[1+1/SNR]$
- If we neglect the term in 1/SNR (usually very small), then

$$\alpha_1 = \rho[1]$$

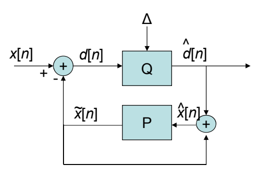
and the gain due to prediction is

$$(G_P)_{opt} = \frac{1}{1-\rho^2[1]}$$

- Thus there is a prediction gain so long as $\rho[1] \neq 0$
- It is reasonable to assume that for speech, $\rho[1] > 0.8$, giving $(G_P)_{opt} > 2.77$ (or 4.43 dB)

69

Differential Quantization



$$\tilde{x}[n] = \sum_{k=1}^{p} \alpha_k \hat{x}[n-k]$$

$$d[n] = x[n] - \tilde{x}[n]$$

$$\hat{d}[n] = d[n] + e[n]$$

$$\hat{x}[n] = \tilde{x}[n] + \hat{d}[n]$$

$$\hat{x}[n] = x[n] + e[n]$$

$$SNR = \frac{\sigma_x^2}{\sigma_e^2} = \frac{\sigma_x^2}{\sigma_d^2}\frac{\sigma_d^2}{\sigma_e^2} = G_p \cdot SNR_Q$$

First Order Predictor:

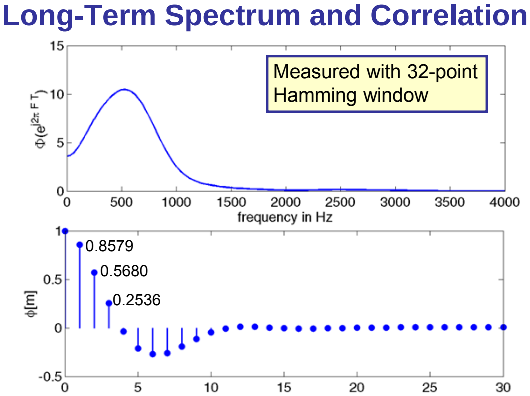
$$\alpha_1 = \frac{\rho[1]}{1+1/SNR}$$

$$G_p = \left(\frac{1}{1-\rho^2[1]}\right)\left(1-\frac{\rho^2[1]}{SNR_Q}\right) \approx \left(\frac{1}{1-\rho^2[1]}\right)$$

The error, $e[n]$, in quantizing $d[n]$ is the same as the error in representing $x[n]$

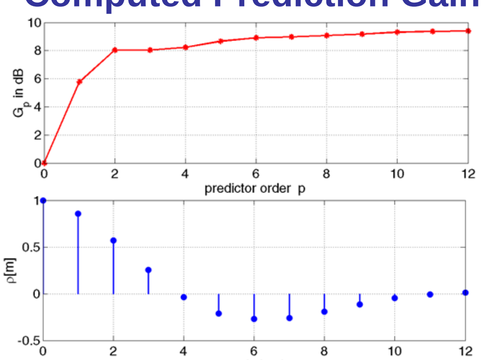
Prediction gain dependent on ρ[1], the first correlation coefficient

Long-Term Spectrum and Correlation



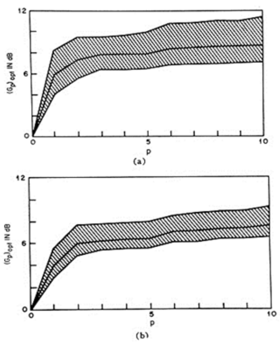
71

Computed Prediction Gain



72

Actual Prediction Gains for Speech



- variation in gain across 4 speakers
- can get about 6 dB improvement in *SNR* => 1 extra bit equivalent in quantization—but at a price of increased complexity in quantization

- differential quantization works!!
- gain in *SNR* depends on signal correlations
- fixed predictor <u>cannot</u> be optimum for all speakers and for all speech material

73

Delta Modulation

Linear and Adaptive

74

Delta Modulation

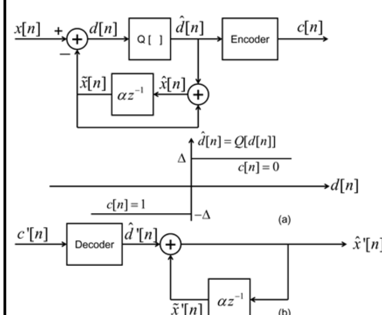
- simplest form of differential quantization is in delta modulation (DM)
- sampling rate chosen to be <u>many times the Nyquist rate</u> for the input signal => adjacent samples are highly correlated
- in the limit as $T \rightarrow 0$, we expect

$$\phi[1] \rightarrow \sigma_x^2 \quad \text{as} \quad T \rightarrow 0$$

- this leads to a high ability to predict *x*[*n*] from past samples, with the variance of the prediction error being very low, leading to a high prediction gain => can use simple 1-bit (2-level) quantizer => the bit rate for DM systems is just the (high) sampling rate of the signal

75

Linear Delta Modulation



- 2-level quantizer with fixed step size, Δ, with quantizer form

$$\hat{d}[n] = \Delta \quad \text{if } d[n] > 0 \quad (c[n]=0)$$
$$= -\Delta \text{ if } d[n] < 0 \quad (c[n]=1)$$

- using simple first order predictor with optimum prediction gain

$$(G_P)_{opt} = \frac{1}{1-\rho^2[1]}$$

- as $\rho[1] \rightarrow 1$, $(G_P)_{opt} \rightarrow \infty$ (qualitatively only since the assumptions under which the equation was derived break down as $\rho[1] \rightarrow 1$)

76

Illustration of DM

- basic equations of DM are

$$\hat{x}[n] = \alpha\ \hat{x}[n-1] + \hat{d}[n]$$

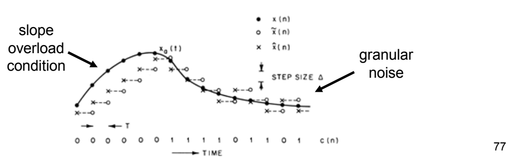
- when $\alpha \approx 1$ (essentially digital integration or accumulation of increments of $\pm\Delta$)

$$d[n] = x[n] - \hat{x}[n-1] = x[n] - x[n-1] - e[n-1]$$

- $d[n]$ is a first backward difference of $x[n]$, or an approximation to the derivative of the input
- how big do we make Δ--at maximum slope of $x_a(t)$ we need

$$\frac{\Delta}{T} \geq \max\left|\frac{dx_a(t)}{dt}\right|$$

- or else the reconstructed signal will lag the actual signal => called 'slope overload' condition--resulting in quantization error called 'slope overload distortion'
- since $\hat{x}[n]$ can only increase by fixed increments of Δ, fixed DM is called linear DM or LDM



77

DM Granular Noise

- when $x_a(t)$ has small slope, Δ determines the peak error => when $x_a(t) = 0$, quantizer will be alternating sequence of 0's and 1's, and $\hat{x}[n]$ will alternate around zero with peak variation of Δ => this condition is called "granular noise"

- need **<u>large</u>** step size to handle wide dynamic range
- need **<u>small</u>** step size to accurately represent low level signals

- with LDM we need to worry about dynamic range and amplitude of the difference signal => choose *Δ* to minimize mean-squared quantization error (a compromise between slope overload and granular noise)

78

Performance of DM Systems

- normalized step size defined as

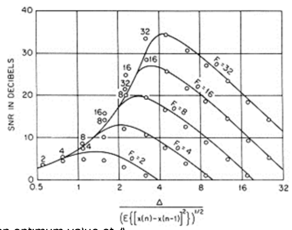
$$\frac{\Delta}{E\left[\left(x[n]-x[n-1]\right)^2\right]^{1/2}}$$

- oversampling index defined as

$$F_0 = F_S/(2F_N)$$

- where F_S is the sampling rate of the DM and F_N is the Nyquist frequency of the signal
- the total bit rate of the DM is

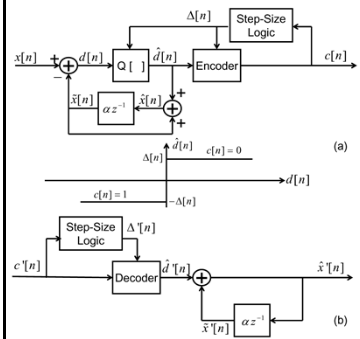
$$BR = F_S = 2F_N \cdot F_0$$



- can see that for given value of $\mathbf{F_0}$, there is an optimum value of Δ
- optimum *SNR* increases by 9 dB for each doubling of F_0 => this is better than the 6 dB obtained by increasing the number of bits/sample by 1 bit
- curves are very sharp around optimum value of Δ => *SNR* is very sensitive to input level
  - for *SNR*=35 dB, for F_N=3 kHz => 200 Kbps rate
  - for toll quality need much higher rates

79

Adaptive Delta Mod



- step size adaptation for DM (from codewords)

$$\Delta[n] = M \cdot \Delta[n-1]$$
$$\Delta_{min} \le \Delta[n] \le \Delta_{max}$$

- M is a function of $c[n]$ and $c[n-1]$, since $c[n]$ depends only on the sign of $d[n]$

$$d[n] = x[n] - \alpha\,\hat{x}[n-1]$$

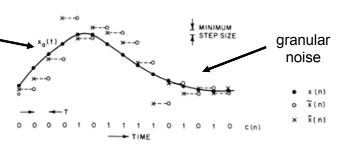
- the sign of $d[n]$ can be determined before the actual quantized value $\hat{d}[n]$ which needs the new value of $\Delta[n]$ for evaluation
- the algorithm for choosing the step size multiplier is

$$M = P > 1 \quad if \quad c[n] = c[n-1]$$
$$M = Q < 1 \quad if \quad c[n] \ne c[n-1]$$

80

Adaptive DM Performance

slope overload condition



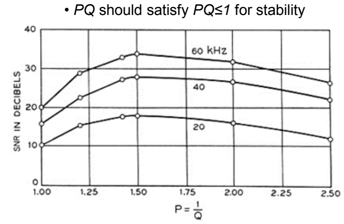
granular noise

- slope overload in LDM causes runs of 0's or 1's
- granularity causes runs of alternating 0's and 1's
- figure above shows how adaptive DM performs with P=2, Q=1/2, α=1
  - during slope overload, step size increases exponentially to follow increase in waveform slope
  - during granularity, step size decreases exponentially to Δ_{min} and stays there as long as slope remains small

81

ADM Parameter Behavior

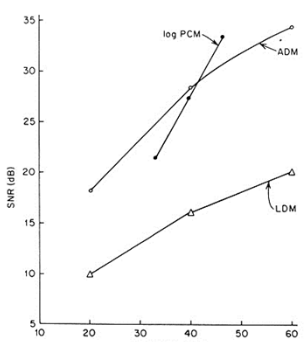
- ADM parameters are P, Q, Δ_{min} and Δ_{max}
  - choose Δ_{min} and Δ_{max} to provide desired dynamic range
  - choose $\Delta_{max}/\Delta_{min}$ to maintain high SNR over range of input signal levels
  - Δ_{min} should be chosen to minimize idle channel noise
  - PQ should satisfy $PQ \le 1$ for stability



- PQ chosen to be 1
- peak of SNR at P=1.5, but for range 1.25<P<2, the *SNR* varies only slightly

82

Comparison of LDM, ADM and log PCM



- ADM is 8 dB better *SNR* at 20 Kbps than LDM, and 14 dB better *SNR* at 60 Kbps than LDM
- ADM gives a 10 dB increase in *SNR* for each doubling of the bit rate; LDM gives about 6 dB
- for bit rate below 40 Kbps, ADM has higher *SNR* than μ-law PCM; for higher bit rates log PCM has higher *SNR*

83

Higher Order Prediction in DM

- first order predictor gave

$$\tilde{x}[n] = \alpha\,\hat{x}[n-1]$$

- with reconstructed signal satisfying

$$\hat{x}[n] = \alpha\hat{x}[n-1] + \hat{d}[n]$$

- with system function

$$H_1(z) = \frac{\hat{X}(z)}{\hat{D}(z)} = \frac{1}{1-\alpha z^{-1}}$$

- digital equivalent of a leaky integrator.
- consider a second order predictor with

$$\tilde{x}[n] = \alpha_1\hat{x}[n-1] + \alpha_2\hat{x}[n-2]$$

- with reconstructed signal

$$\hat{x}[n] = \alpha_1\hat{x}[n-1] + \alpha_2\hat{x}[n-2] + \hat{d}[n]$$

- with system function

$$H_2(z) = \frac{\hat{X}(z)}{\hat{D}(z)} = \frac{1}{\left(1-\alpha_1 z^{-1} - \alpha_2 z^{-2}\right)}$$

- assuming two real poles, we can write $H_2(z)$ as

$$H_2(z) = \frac{\hat{X}(z)}{\hat{D}(z)} = \frac{1}{\left(1-az^{-1}\right)\left(1-bz^{-1}\right)}, \quad 0 < a, b < 1$$

- better prediction is achieved using this "double integration" system with up to 4 dB better SNR
- there are issues of interaction between the adaptive quantizer and the predictor

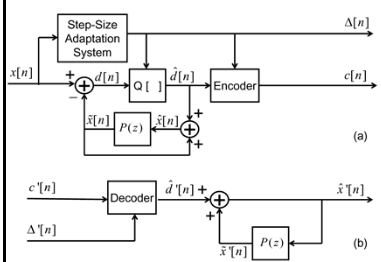
84

Differential PCM (DPCM)

- fixed predictors can give from 4-11 dB *SNR* improvement over direct quantization (PCM)
- most of the gain occurs with first order predictor
- prediction up to 4^{th} or 5^{th} order helps

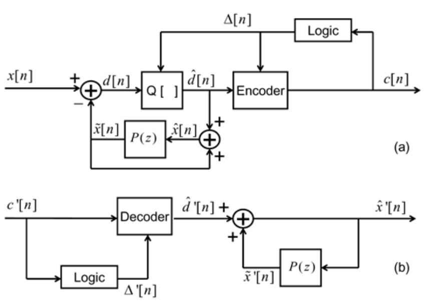
85

DPCM with Adaptive Quantization



- quantizer step size proportional to variance at quantizer input
- can use $d[n]$ or $x[n]$ to control step size
- get 5 dB improvement in *SNR* over μ-law non-adaptive PCM
- get 6 dB improvement in *SNR* using differential configuration with fixed prediction => ADPCM is about 10-11 dB *SNR* better than from a fixed quantizer

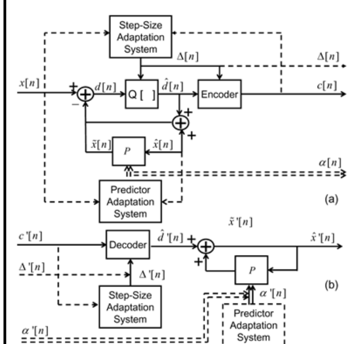
Feedback ADPCM



can achieve same improvement in *SNR* as feed forward system

87

DPCM with Adaptive Prediction



need adaptive prediction to handle non-stationarity of speech

88

DPCM with Adaptive Prediction

- prediction coefficients assumed to be time-dependent of the form

$$\tilde{x}[n] = \sum_{k=1}^{p} \alpha_k[n]\hat{x}[n-k]$$

- assume speech properties remain fixed over short time intervals
- choose $\alpha_k[n]$ to minimize the average squared prediction error over short intervals
- the optimum predictor coefficients satisfy the relationships

$$R_n[j] = \sum_{k=1}^{p} \alpha_k[n]R_n[j-k], \quad j = 1, 2, \ldots, p$$

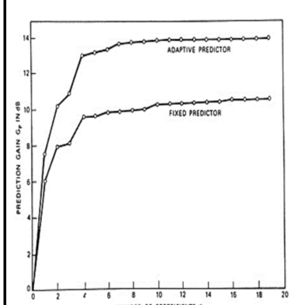
- where $R_n[j]$ is the short-time autocorrelation function of the form

$$R_n[j] = \sum_{m=-\infty}^{\infty} x[m]w[n-m]x[j+m]w[n-m-j], \quad 0 \le j \le p$$

- $w[n-m]$ is window positioned at sample n of input
- update α's every 10-20 msec

89

Prediction Gain for DPCM with Adaptive Prediction

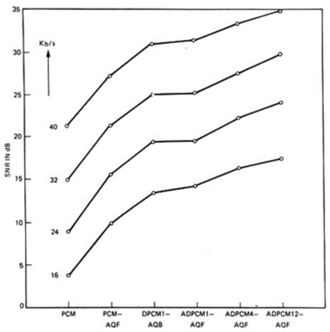


$$10\log_{10}\left[G_P\right] = 10\log_{10}\left[\frac{E\left[x^2[n]\right]}{E\left[d^2[n]\right]}\right]$$

- fixed prediction $\rightarrow$ 10.5 dB prediction gain for large p
- adaptive prediction $\rightarrow$ 14 dB gain for large p
- adaptive prediction more robust to speaker, speech material

90

Comparison of Coders



- 6 dB between curves
- sharp increase in *SNR* with both fixed prediction and adaptive quantization
- almost no gain for adapting first order predictor

91