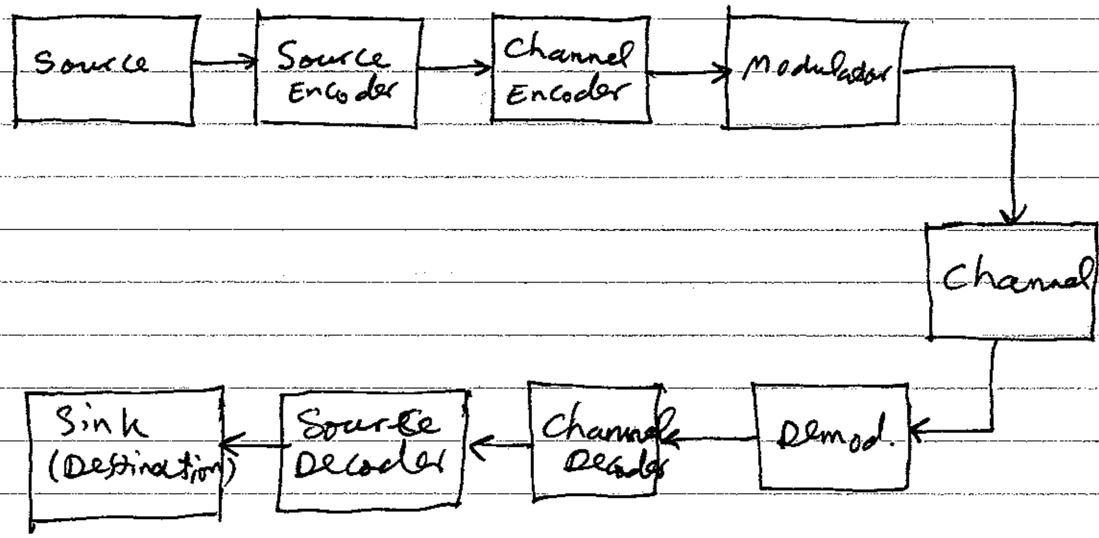
X Lecture 3, May 10, 2011

Sampling, Quantization, Line Coding, ...

A simplified block Diagram of a Communications Link:



Source Encoding (Formatting) Consists of:

1) Sampling,

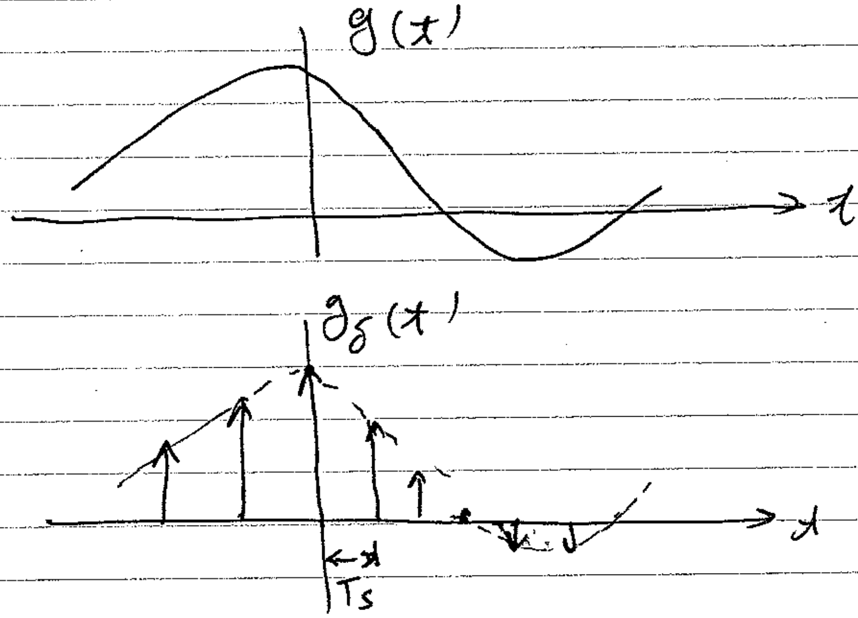
2) Quantization,

3) Coding into binary stream.

X Lecture 3, May 10, 2010.

Sampling

Sampling Process



$f_s = \frac{1}{T_s} \triangleq$ sampling rate

T_s = sample period (sample duration)

$$g_\delta(t) = \sum_{n=-\infty}^{\infty} g(nT_s)\,\delta(t - nT_s)$$

$g_\delta(t)$ is the ideal sampled signal

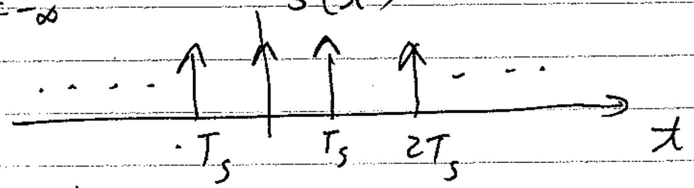
We have

$$G_\delta(f) = \sum_{n=-\infty}^{\infty} g(nT_s)\, e^{-j2\pi n f T_s}$$

We can write

$$g_\delta(t) = g(t)\, s(t)$$

where $s(t) = \sum_{n=-\infty}^{\infty} \delta(t - nT_s)$



using Modulation Theorem

$$G_\delta(f) = G(f) * S(f)$$

where

$$G(f) = \mathcal{F}[g(t)] \text{ and } S(f) = \mathcal{F}[s(t)]$$

$s(t)$ is a periodic function and has Fourier Series representation:

$$s(t) = \sum_{n=-\infty}^{\infty} c_n e^{-j\frac{2\pi n t}{T_s}}$$

where

$$c_n = \frac{1}{T_s} \int_{-\frac{T_s}{2}}^{\frac{T_s}{2}} \delta(t)\, dt = 1$$

So:

$$s(t) = \frac{1}{T_s} \sum_{n=-\infty}^{\infty} e^{-j\frac{2\pi n t}{T_s}}$$

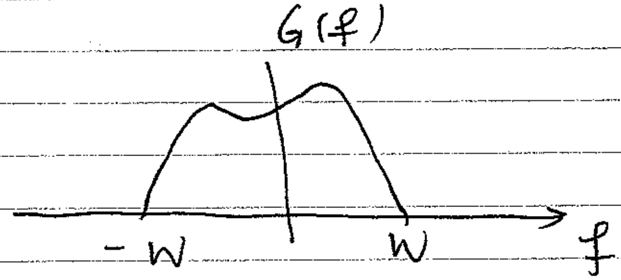
and

$$S(f) = \frac{1}{T_s} \sum_{n=-\infty}^{\infty} \delta\left(f - \frac{n}{T_s}\right) = f_s \sum_{n=-\infty}^{\infty} \delta(f - nf_s)$$

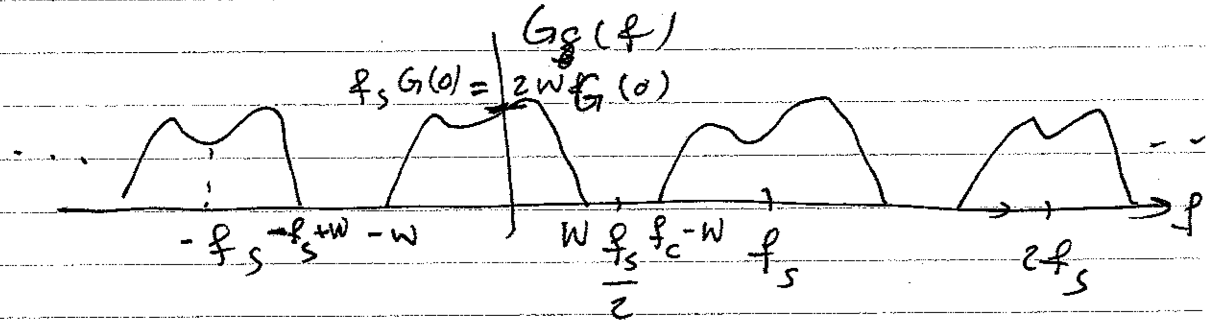
So:

$$G_\delta(f) = f_s \sum_{n=-\infty}^{\infty} G(f - n f_s)$$

Take

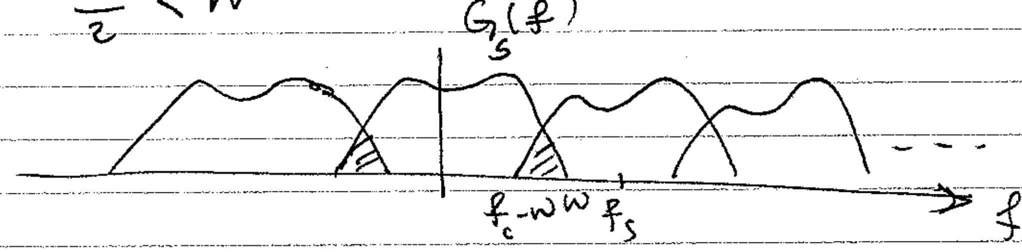


then for $\frac{f_s}{2} \geq W$ or $f_s \geq 2W$



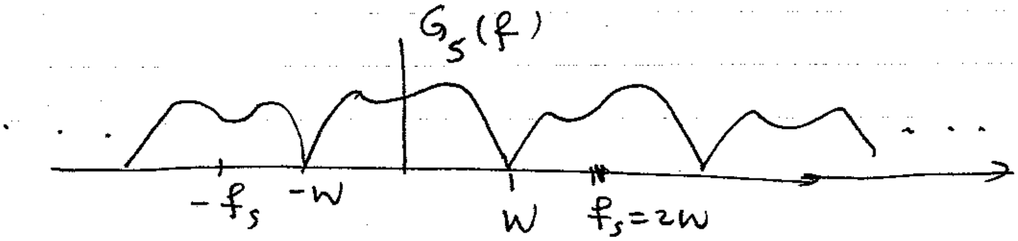
$G(f)$ can be recovered from $G_\delta(f)$ by low-pass filtering

if $\frac{f_s}{2} < W$ then



there is aliasing and the original signal cannot be fully recovered.

when $f_s = 2W$



$$G(f) = \frac{1}{2W} G_s(f) \qquad -W < f < W$$

But

$$G_s(f) = \sum_{n=-\infty}^{\infty} g\left(\frac{n}{2W}\right) \exp\left[-\frac{j\pi n f}{W}\right]$$

So:

$$G(f) = \frac{1}{2W}\sum_{n=-\infty}^{\infty} g\left(\frac{n}{2W}\right) \exp\left[-\frac{j\pi n f}{W}\right] \qquad |f|<W$$

$$g(t) = \int_{-\infty}^{\infty} G(f) \exp(j2\pi f t)\, df$$

$$= \int_{-W}^{W} \left[\frac{1}{2W}\sum_{n=-\infty}^{\infty} g\left(\frac{n}{2W}\right) \exp\left(-\frac{j\pi n f}{2W}\right)\right] \exp(j2\pi f t)\, df$$

$$= \sum_{n=-\infty}^{\infty} g\left(\frac{n}{2W}\right) \frac{1}{2W} \int_{-W}^{W} \exp\left[j2\pi f\left(t - \frac{n}{2W}\right)\right] df$$

or

$$g(t) = \sum_{n=-\infty}^{\infty} g\left(\frac{n}{2W}\right) \frac{\sin(2\pi W t - n\pi)}{2\pi W t - n\pi}$$

$$\boxed{g(t) = \sum_{n=-\infty}^{\infty} g\left(\frac{n}{2W}\right) \operatorname{sinc}(2Wt - n) \qquad -\infty < t < \infty}$$

Quantization:

$$m \longrightarrow \boxed{Q(\cdot)} \longrightarrow v$$

m takes any value (in a given range), i.e., it can be any real number in that range. However v takes only a discrete set of values $v_1, v_2, \dots, v_L$

to represent L objects, we need: $\log_2 L$ bits

$R = \log_2 L$ in bits/Sample is the rate of quantizer:

$$L = 2^R.$$

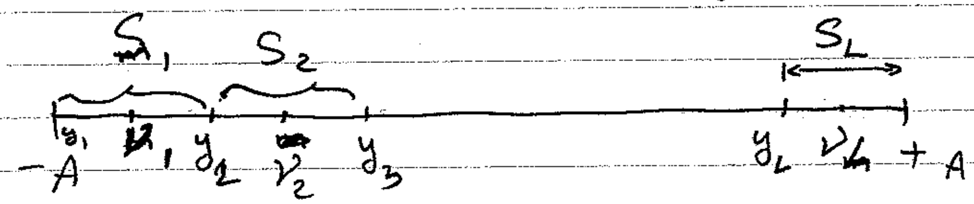
Optimum Quantizer:

Assume mean squared error being used as a measure of distortion then if m is encoded as v_k the distortion is

$$d(m, v_k) = (m - v_k)^2$$

Assume that the dynamic range of m is $-A$ to $+A$ and the real line is partitioned into $S_1, S_2, \dots S_L$ segments such that any value of m falling into S_k is encoded as ν_k.

$\{S_k\} \leftrightarrow$ encoder
$\{\nu_k\} \leftrightarrow$ decoder



The total Distortion will be:

$$D = E[(m-\nu)^2] = \int_{-\infty}^{\infty} (m-\nu)^2 f_M(m)\,dm$$

or

$$D = \sum_{k=1}^{L} \int_{S_k} (m-\nu_k)^2 f_M(m)\,dm$$

1) for a given encoder, i.e., $\{S_k\}$ to optimize (minimized D), we need to set $\frac{\partial D}{\partial \nu_k} = 0$

$$\frac{\partial D}{\partial \nu_k} = -2\int_{S_k} (m-\nu_k) f_M(m)\,dm = 0 \qquad \text{all } k \in [1, \dots$$

$$\Rightarrow \nu_k = E[m \mid S_k] \qquad k = 1, 2, \dots, L$$

or

$$\boxed{\nu_k = \int_{S_k} m\, f_M(m)\,dm = 0 \Big/ \int_{S_k} f_M(m)\,dm}$$

2) For a given decoder $\{\nu_k\}$, S_k are defined based on the following rule

if $(m-\nu_k)^2 \leq (m-\nu_j)^2$ all $j \neq k$

then $m \in S_k$ or m is encoded as ν_k.

this is based on the fact that we need to minimize each $(m-\nu_k)^2$ individually.

Then the thresholds will be:

$$\boxed{y_k = \frac{\nu_k + \nu_{k-1}}{2}}$$

- Llyod-Max Algorithm.

<u>Lloyd-Max Algorithm:</u>

iteratively use:

$$1) \quad \nu_k = \frac{\int_{S_k} m f_M(m)\,dm}{\int_{S_k} f_M(m)\,dm}$$

and

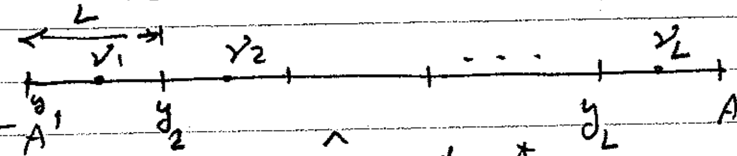
$$2) \quad y_k = \frac{\nu_k + \nu_{k-1}}{2}$$

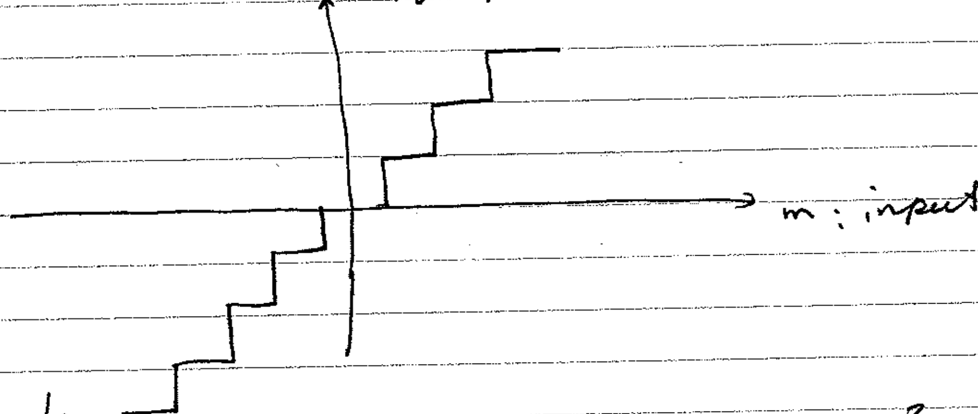
until the algorithm converges, i.e.,

$$\frac{D_{n-1} - D_n}{D_{n-1}} \leq \varepsilon$$

Uniform Quantizer:

For uniform quantizer: the steps will be equal: $\frac{2A}{L} = \Delta$





$$D = \sum_{k=1}^{L} \int_{S_k} (m - \nu_k)^2 f_M(m)\, dm = L \int_{S_k} (m - \nu_k)^2 f_M(m)\, dm$$

$$D = \frac{L}{2A} \int_{S_k} (m - \nu_k)^2 dm = \frac{1}{\Delta} \int_{\nu_k - \frac{\Delta}{2}}^{\nu_k + \Delta/2} (m - \nu_k)^2 dm$$

$$D = \frac{1}{\Delta} \int_{-\Delta/2}^{\Delta/2} e^2 de = \frac{\Delta^2}{12} = \left(\frac{2A}{L}\right)^2 / 12 = \frac{A^2}{3L^2} = \frac{m_{max}^2}{3L^2}$$

or

$$D = \frac{m_{max}^2}{3 \times 2^{2R}} = \frac{1}{3} m_{max}^2 \cdot 2^{-2R}$$

$$SQNR = \frac{P}{D} = \frac{3P}{m_{max}^2} 2^{2R}$$

in dB

$$SQNR = 10 \log \frac{3P}{m_{max}^2} + 20 R \log 2 \cong 6R + 10 \log \frac{3P}{m_{max}^2}$$

~~Example~~

let $P = m_{RMS}^2$

Then

$$SQNR = 6R + 10 \log \frac{3 m_{RMS}^2}{m_{max}^2}$$

$$= 6R + 10 \log 3 + 20 \log \frac{m_{RMS}}{m_{max}}$$

Example: A sinusoidal signal with amplitude A_m.

Then $P = \frac{A_m^2}{2}$

and

$$m_{max} = A_m$$

So:

$$SQNR = 6R + 10 \log \frac{3 A_m^2}{2 A_m^2} = 6R + 10 \log 3/2 = 6R + 1.8 \text{ dB}$$

example: For an 8-bit quantizer

$$L = 2^8 = 256$$

$$SQNR \simeq 6 \times 8 + 1.8 = 49.8 \text{ dB}.$$

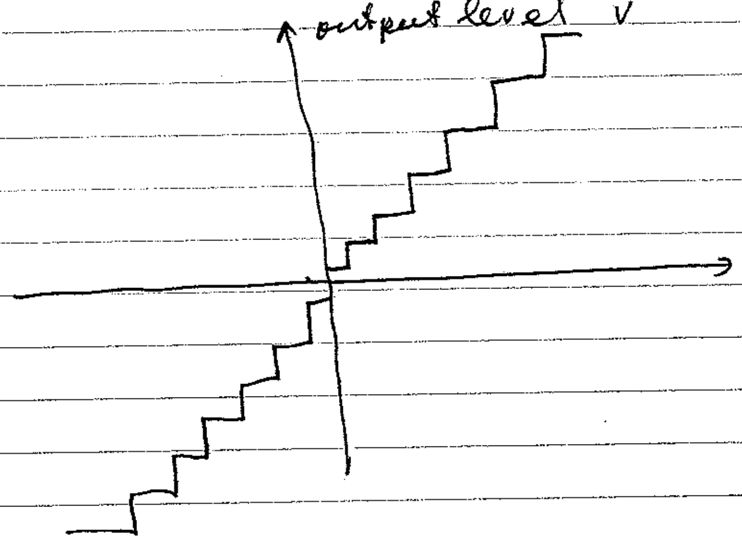
for 16-bit quantizer

$$L = 2^{16} = 65536$$

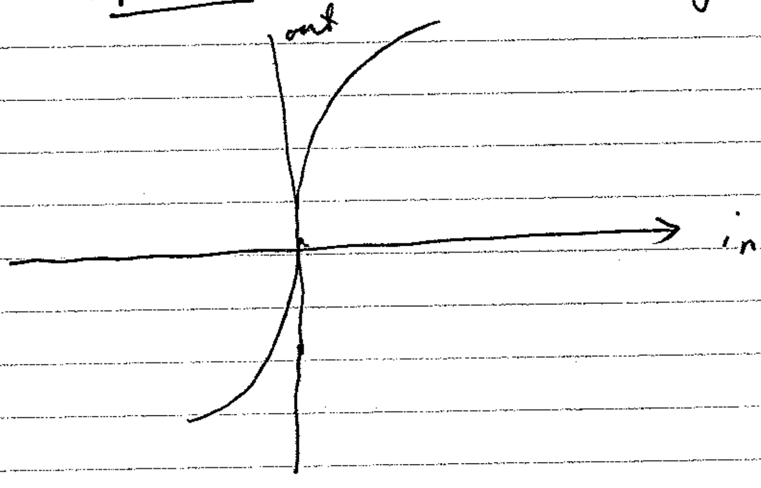
and $SQNR = 97.8$ dB.

Pulse Code Modulation (PCM):

Non-uniform quantizer:



instead of having variable step sizes, one can compress the samples and then use a uniform quantiz and then expand (COMPANDing):



Compressor.

μ-Law (North America)

$$|v| = V_{max} \frac{\log\left(1+\mu \frac{|u|}{u_{max}}\right)}{\log(1+\mu)}$$

where $\mu = 0$ results in uniform quantizer

A-law (Europe/Asia)

$$|v| = \begin{cases} v_{max} \dfrac{A|u|/u_{max}}{1+\log A} & 0 \le \dfrac{|u|}{u_{max}} \le \dfrac{1}{A} \\ v_{max} \dfrac{1+\log \frac{A|u|}{u_{max}}}{1+\log A} & \end{cases}$$

A=1 corresponds to the Uniform Quantizer.

Overall PCM Standard:

Sampling 8 kHz → 8 k Samples/Sec.

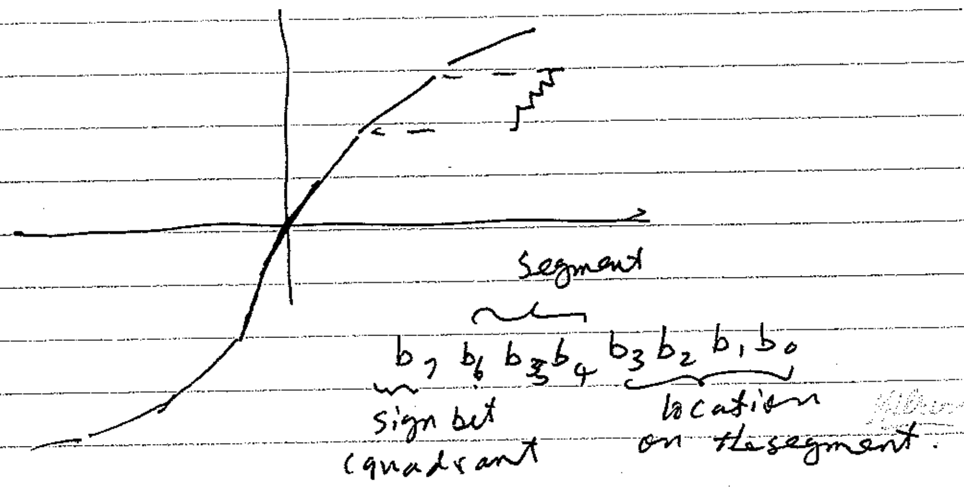
8 bits/Sample Quantization

So:

$8 \times 8 = 64$ kbps.

64 kbps is called DS0 or T0 or just one Digital Voice channel.

$\mu = 255$, 15 Segment Companding



3-14

Linear Segment	step size	value (on x-axis) of the segment end points
0	2	± 31
1a, 1b	4	± 95
2a, 2b	8	± 223
3a, 3b	16	± 479
4a, 4b	32	± 991
5a, 5b	64	± 2015
6a, 6b	128	± 4063
7a, 7b	256	± 8159

Line Codes

1) Unipolar NRZ

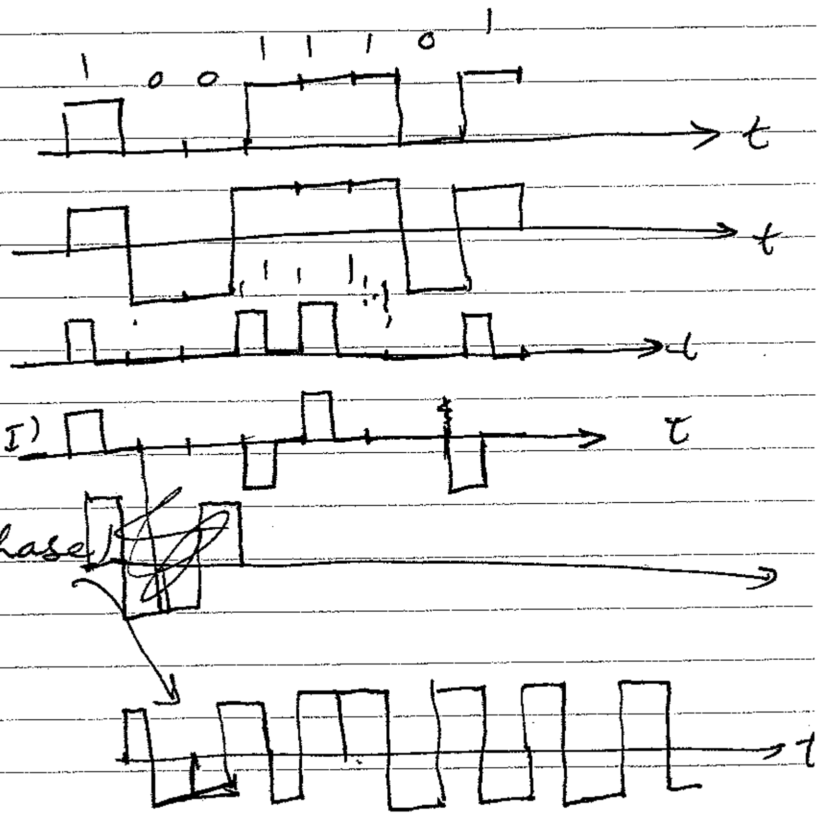
2) Bipolar NRZ

3) Unipolar RZ

4) Bipolar RZ (AMI)

5) Manchester (Bi-phase)

6) Miller Code.



3 - 15

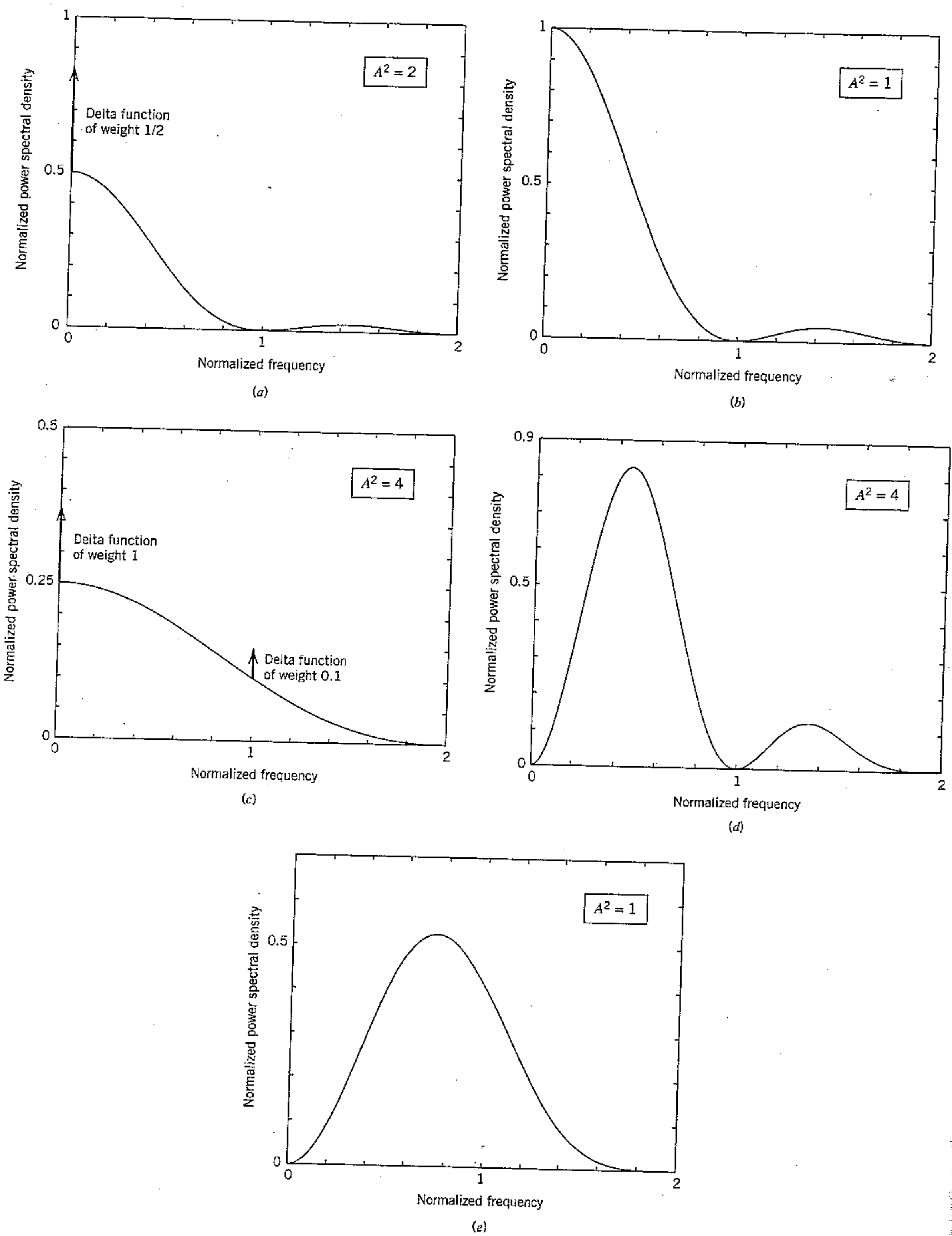


FIGURE 3.16 Power spectra of line codes: (*a*) Unipolar NRZ signal. (*b*) Polar NRZ signal. (*c*) Unipolar RZ signal. (*d*) Bipolar RZ signal. (*e*) Manchester-encoded signal. The frequency is normalized with respect to the bit rate $1/T_b$, and the average power is normalized to unity.

Channel Noise Consideration in PCM:

Assume that the bit error rate (BER) is P_B

that is, with probability P_b, 1 bit out of R (say 8) bits will be in error. ~~That yet~~

The erroneous bit can be at location 0 (LSB), location 1, ..., or 8 (MSB). The probability of each individual bit being in error is $\frac{1}{8} P_B$

The effect is a shift of $\Delta, 2\Delta, \ldots, 2^7\Delta$

and the noise power resulting will be $\Delta^2, (2\Delta)^2, \ldots (2^7\Delta)^2$ (due to channel noise)

So, the average distortion is:

$$D_e = E[D] = \frac{1}{8}P_b(\Delta)^2 + \frac{1}{8}P_b(2\Delta)^2 + \cdots + \frac{1}{8}P_b(2^7\Delta)^2$$

or

$$D_e = E[D] = \frac{P_b}{8}\Delta^2\left[1 + 2^2 + 4^2 + 8^2 + \cdots + (2^7)^2\right]$$

$$D_e = E[D] = \frac{P_b}{8}\Delta^2\left[1 + 4 + 4^2 + 4^3 + \cdots + 4^7\right]$$

So:

$$D_e = E[D] = \frac{P_b}{8}\Delta^2 \frac{4^8 - 1}{4 - 1}$$

Since $\Delta = \frac{2 U_{max}}{L}$

$$D_e = E[D] = \frac{P_b}{8} \times \frac{4 U_{max}^2}{L^2} \times \frac{4^8 - 1}{3} = P_b \frac{U_{max}^2}{6L^2} \times \frac{(4^8 - 1)}{1}$$

Total Distortion is:

$$D_T = D_q + D_e = \frac{u_{max}^2}{3L^2} + P_b \frac{u_{max}^2}{6L^2}(4^8-1)$$

$$D_T = \frac{u_{max}^2}{3L^2}\left[1 + \frac{P_b(4^8-1)}{2}\right]$$

if $\frac{P_b(4^8-1)}{2} \ll 1$ then the effect of channel noise is negligible.

That is if $P_b \ll \frac{2}{4^8-1} \cong 2^{-15} \cong 3\times10^{-5}$

for BPSK modulation (or any other binary anti-podal signaling) we have

E_b/N_0	P_b	time between errors (for 100 kbps transmission)
4.3 dB	10^{-2}	10^{-3} seconds
8.4	10^{-4}	10^{-1} "
10.6	10^{-6}	10 "
12	10^{-8}	20 minutes
13	10^{-10}	1 day
14	10^{-12}	3 month

Comparison with Analog Transmissio

8 bits $\rightarrow \approx 48$ dB $\Rightarrow 10^{4.8}$

With Digital we need 10.6 dB of $\frac{S}{N}$

$\frac{S}{N} = 10^{1.1} \approx 12.6$

with Analog (FM):

$$\frac{64}{2} = 32\text{ kHz} = \underbrace{8}_{2\beta} \times 4 \Rightarrow \beta = 4$$

$$\left(\frac{S}{N}\right)_{FM} = 3\beta^2 \left(\frac{S_c}{N_c}\right)$$

$$48\text{ dB} = 10\log 3\times16 + \left(\frac{S_c}{N_c}\right)\text{dB}$$

$$\left(\frac{S_c}{N_c}\right)_{dB} = 31.2\text{ dB} \Rightarrow \frac{S_c}{N_c} = 10^{3.12} = 1320$$

More than hundred times more power is required.

For CD quality 16 bits $\rightarrow$ 96 dB

$$E[D] = P_b[1 + 2^2 + \cdots\ 2^{30}]\frac{\Delta^2}{16} = P_b\Delta^2\frac{4^{16}-1}{3\times16}$$

$$\sigma_q^2 = \frac{\Delta^2}{12}$$

$$\sigma = \sigma_q^2 + E(D) = \frac{\Delta^2}{12} + P_b\Delta^2\frac{4^{16}-1}{3\times16}$$

$$P_b\frac{4^{16}-1}{3\times16} \ll \frac{1}{12} \Rightarrow P_b \lesssim 9.3\times10^{-10}$$

$\Big\} \Rightarrow BW = \frac{16\times44}{2} = 352k$

$\Rightarrow 13$ dB $\Rightarrow$ 20 rati

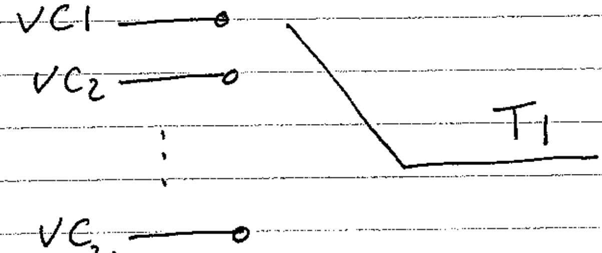
FM: $BW = 8\times44 = 2\beta\times22 \Rightarrow \beta = 8$

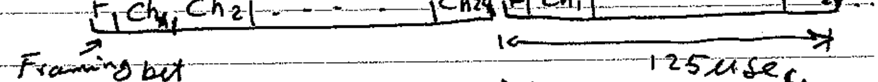
$\left(\frac{S}{N}\right)_{FM} = 10\log 3\times64 + \left(\frac{S}{N}\right)_{c,dB} \Rightarrow \frac{S}{N} = 96 - 22.8 \approx 73\text{ dB} \Rightarrow 10^{7.3} \approx 2\times10^{7}$

X

Digital Multiplexing

T_1 or DS1 :





each voice channel generates 8-bit every

$\frac{1}{8000} = 125\ \mu sec.$ (one PCM sample).

So, one frame of duration 125 µsec.

has $24 \times 8 = 192$ speech bits

in addition there is 1 synchronization bit

so the total number of bits in 1 frame

is 193.

The rate of a T_1 (DS1) system is:

$$\frac{193}{125 \times 10^{-6}} = 1.544 \text{ Mbps}$$

out of which 1.536 is information and

the balance 8 kbps is signalling.

$$\text{efficiency} = \frac{1.536}{1.544} \approx 99.5\% \quad * \frac{7\frac{5}{6}}{8} \approx 97.41\%$$

$\boxed{\text{every 6th sample the LSB is used for signalling}}$

The synchronization pattern is 100011011100

In T_1 synchronization is slow. The Demultiplexer has to look (in the worst case) to 193×12 frams in order to synchronize (on the average $\frac{193 \times 12}{2}$)

this is 0.2895 Sec.

The European standard E_1

32 time slots : 30 voice + 1 signalling + 1 framing

| S | F | Ch_1 | - - - - - | Ch_{30} | - - - -

$\longleftarrow$ 125 μsec. $\longrightarrow$

Total bit rate of E_1

$$\frac{32 \times 8}{125 \times 10^{-6}} = 2.048 \text{ Mbps}$$

usefual rate

$$\frac{30 \times 8}{125 \times 10^{-6}} = 1.92 \text{ Mbps}$$

efficiency = 0.9375

The advantage ; ease of Synchronization.

other members of Digital Hierarchy:

DS2 : $4 \times DS_1 = 6.312$ Mb/s

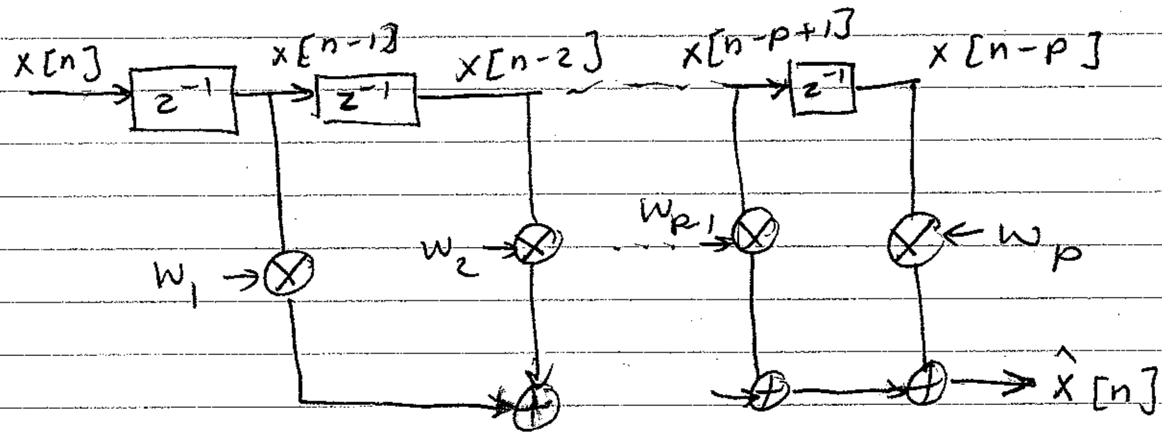
T_3: DS_3 : $7 \times DS2 = 44.736$ Mb/s

$DS4 = 6 \times DS3 = 274.176$ Mb/s

$DS5 = 2 \times DS4 = 560.160$ Mbps.

Optical Hierarchy OC1, OC3, OC12, ..., OC19

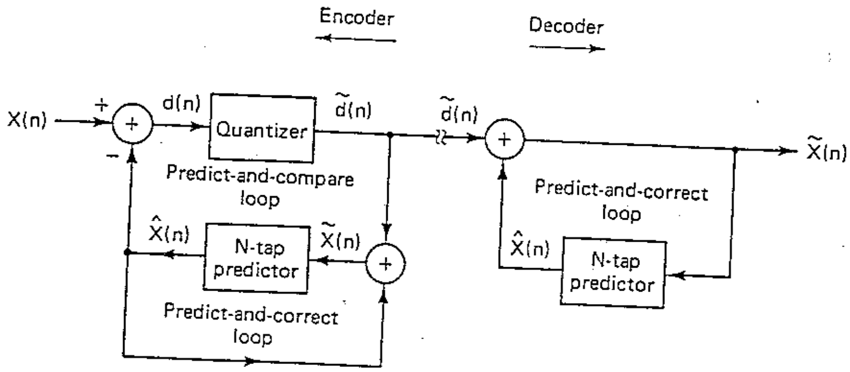
Linear Prediction



Block Diagram of an p-tap prediction filter

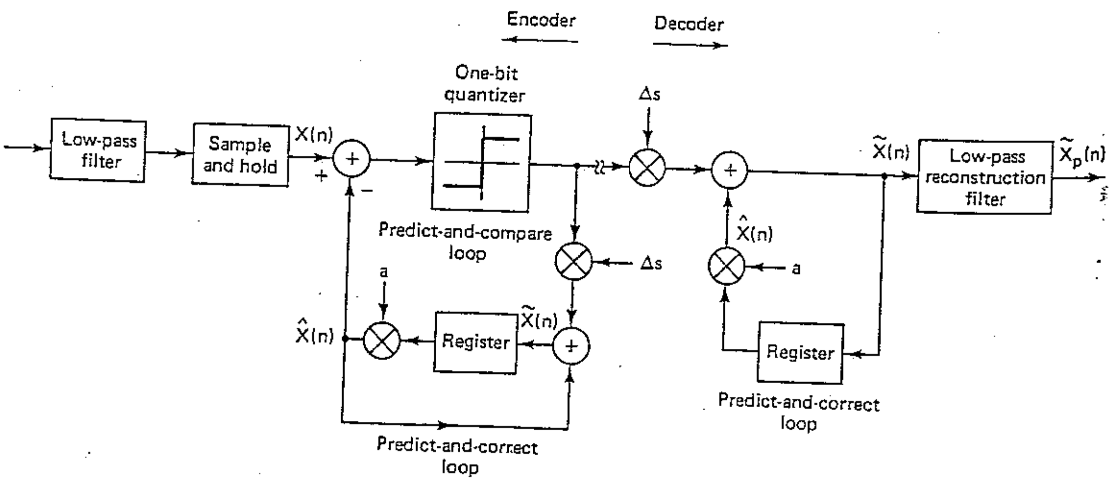
Differential Pulse Code Modulation (DPCM)

In order to take advantage of the correlation between the consecutive samples of the signal, one could, instead of quantizing the samples themselves, quantize the difference between the consecutive samples.



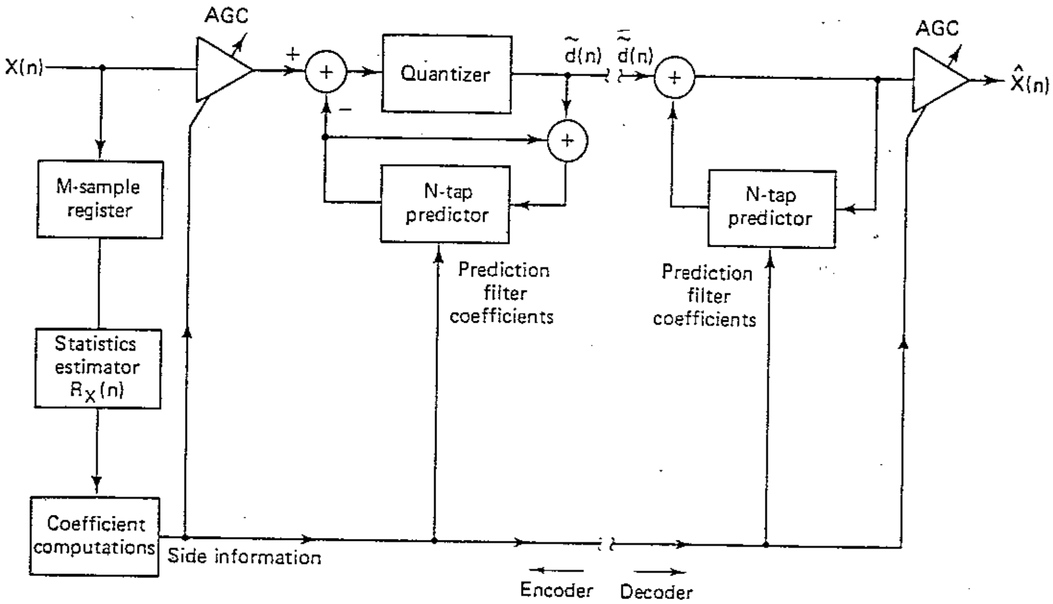
$$\tilde{X}(n) = \hat{X}(n) + \tilde{d}(n)$$

Delta Modulation: The implementation can be drastically simplified by using a single bit quantizer (just a comparator),

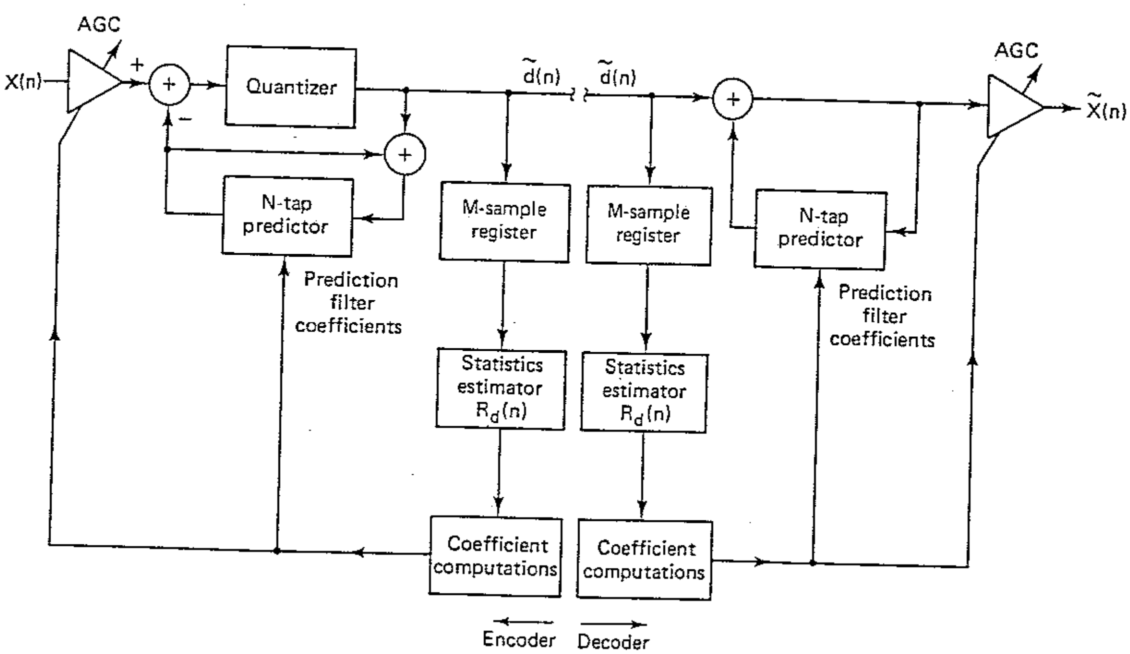


Adaptive Differential Pulse Code Modulation (ADPCM)

Extra improvement in performance or, equivalently, reduction in rate for the same performance could be achieved by making the prediction filter adaptive, i.e., changing the filter taps according to the source statistics.



Forward adaptive prediction and quantization coding.



Backward adaptive prediction and quantization coding.

3+24

MC14LC5540

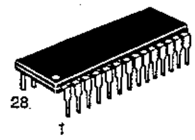
Technical Summary
ADPCM Codec

This technical summary provides a brief description of the MC14LC5540 ADPCM Codec. A complete data book for the MC14LC5540 is available and can be ordered from your local Motorola sales office. The data book number is MC145540/D.

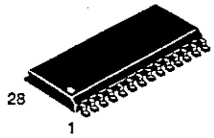
The MC14LC5540 ADPCM Codec is a single chip implementation of a PCM Codec–Filter and an ADPCM encoder/decoder, and therefore provides an efficient solution for applications requiring the digitization and compression of voiceband signals. This device is designed to operate over a wide voltage range, 2.7 to 5.25 V and, as such, is ideal for battery powered as well as ac powered applications. The MC14LC5540 ADPCM Codec also includes a serial control port and internal control and status registers that permit a microcomputer to exercise many built–in features.

The ADPCM Codec is designed to meet the 32 kbps ADPCM conformance requirements of CCITT Recommendation G.721–1988 and ANSI T1.301. It also meets ANSI T1.303 and CCITT Recommendation G.723–1988 for 24 kbps ADPCM operation, and the 16 kbps ADPCM standard, CCITT Recommendation G.726. This device also meets the PCM conformance specification of the CCITT G.714 Recommendation.

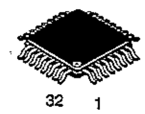
- Single 2.7 to 5.25 V Power Supply
- Typical 2.7 V Power Dissipation of 43 mW, Power–Down of 15 µW
- Differential Analog Circuit Design for Lowest Noise
- Complete Mu–Law and A–Law Companding PCM Codec–Filter
- ADPCM Transcoder for 64, 32, 24, and 16 kbps Data Rates
- Universal Programmable Dual Tone Generator
- Programmable Transmit Gain, Receive Gain, and Sidetone Gain
- Low Noise, High Gain, Three Terminal Input Operational Amplifier for Microphone Interface
- Push–Pull, 300 Ω Power Drivers with External Gain Adjust for Receiver Interface
- Push–Pull, 300 Ω Auxiliary Output Drivers for Ringer Interface
- Voltage Regulated Charge Pump to Power the Analog Circuitry in Low Voltage Applications
- Receive Noise Burst Detect Algorithm
- Order Complete Document as MC145540/D
- Device Supported by MC145537EVK ADPCM Codec Evaluation Kit



P SUFFIX
PLASTIC DIP
CASE 710



DW SUFFIX
SOG PACKAGE
CASE 751F



FU SUFFIX
TQFP
CASE 873A

ORDERING INFORMATION

MC14LC5540P	Plastic DIP
MC14LC5540DW	SOG Package
MC14LC5540FU	TQFP

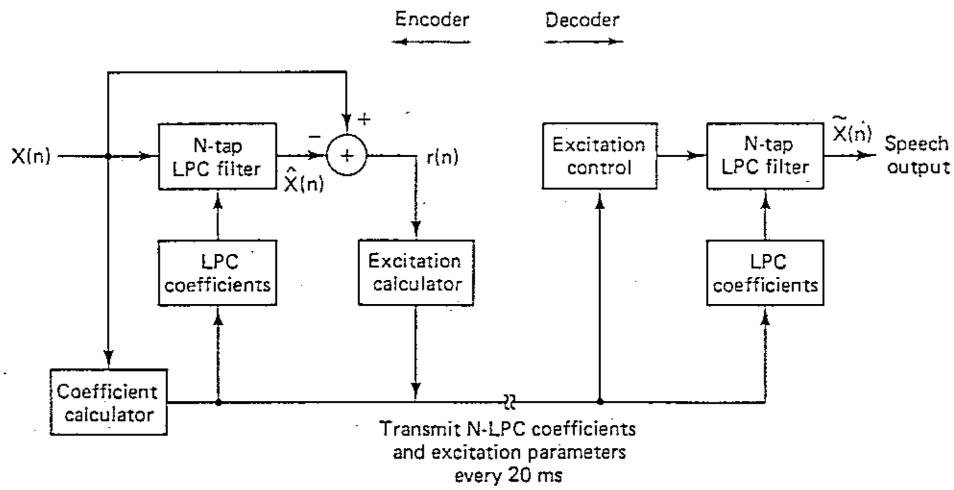
This document contains information on a product under development. Motorola reserves the right to change or discontinue this product without notice.

REV 2
6/97 TN97060200

Linear Predictive Coding (LPC)

Taken from Digital Communications by B. Sklar

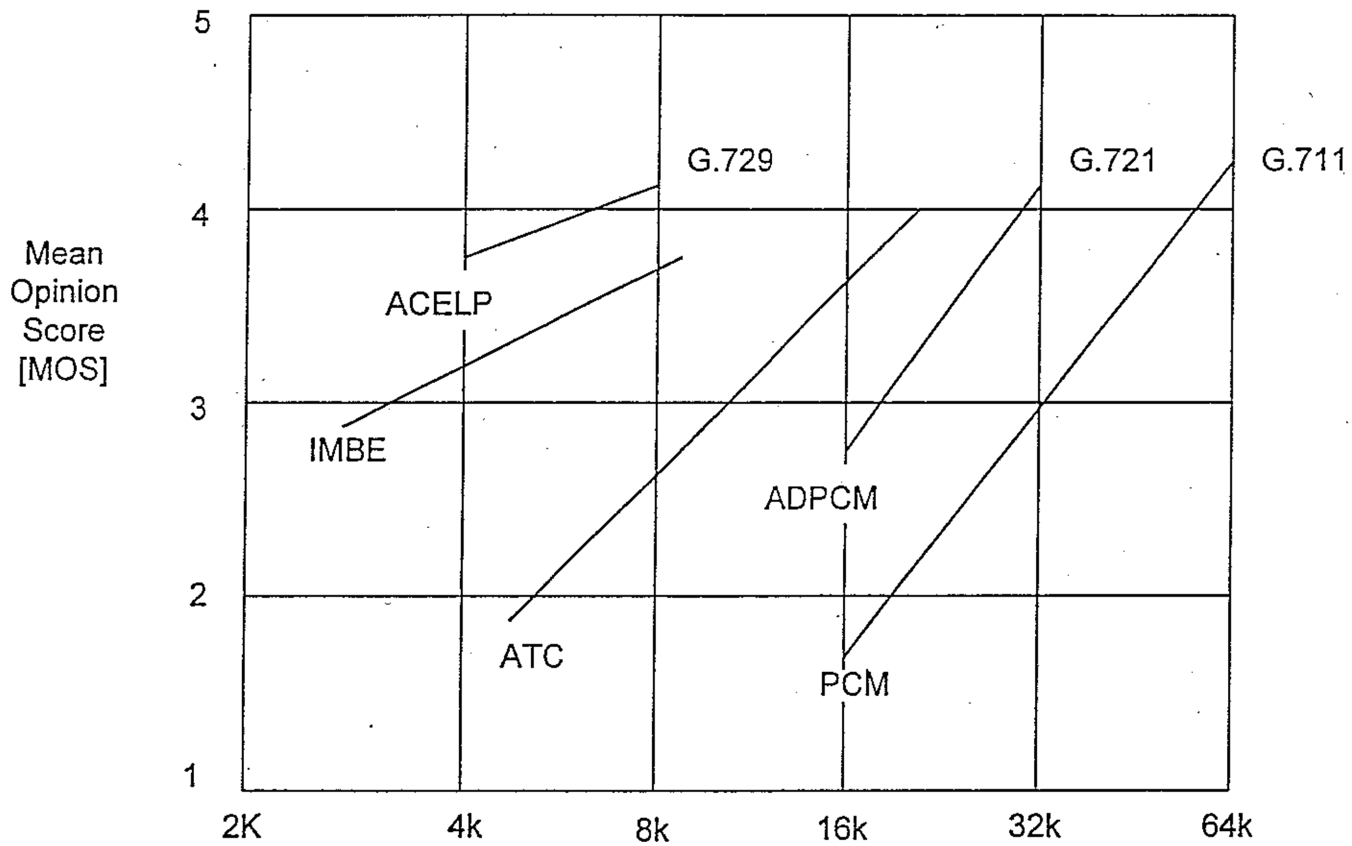
The adaptive predictors, described in Section 11.3.4, were designed to predict or form good estimates of an input speech signal. In the adaptive form, the prediction coefficients are recomputed as side information from periodic examination of the input data. Then the difference between the input and the prediction is transmitted to the receiver to resolve the prediction error. *Linear predictive coders* (LPCs) are the natural extension of N-tap predictive coders. When the filter coefficients are periodically computed with an optimal algorithm, the prediction is so good that there is (essentially) no prediction error information worth transmitting to the receiver. Rather than transmit these low-level prediction errors, the LPC system transmits the filter coefficients and the voiced/unvoiced excitation decision for the model. Thus the only data sent in LPC is the high-quality side information of the classic adaptive algorithm. An LPC model for voice synthesis is shown in Figure 11.34. The Texas Instruments Speak and Spell learning games use a 12-tap LPC speech synthesizer implemented by a single microchip.



Linear predictive coefficient (LPC) speech modeling.

Comparison of Different Speech CODECs

COMPRESSED VOICE QUALITY



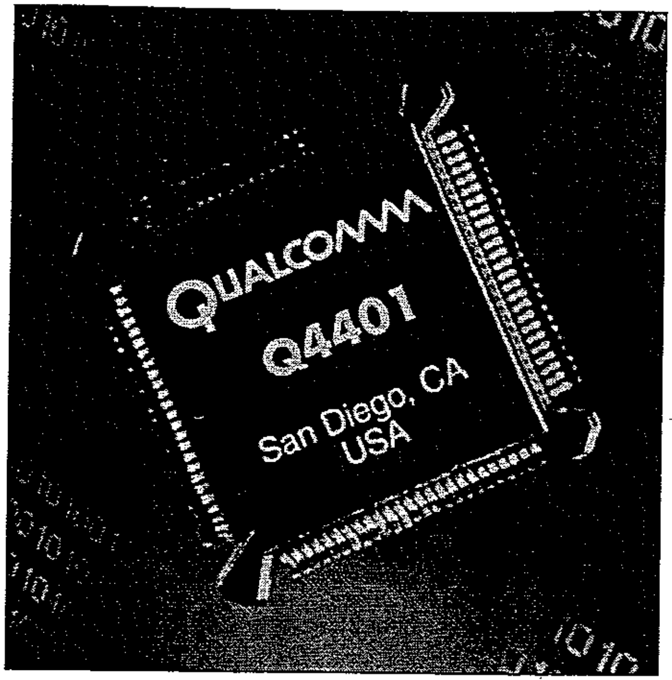
CODEC bit rate [bps]

- Definition of MOS
- Difference between objective & Subjective Distortion measures.

Q4401

3-27

VARIABLE RATE VOCODER



- Brief mention of the advantages of variable rate vocoders, i.e., adaptability ~~of~~ to channel quality, particularly for CDMA.

GENERAL DESCRIPTION

The QUALCOMM Q4401 Variable Rate Vocoder is a full-duplex speech Encoder and Decoder that produces near toll-quality speech at compressed data rates of under 9.6 kilobits per second (kbps). The Q4401 provides a single-chip solution to the speech compression requirements for digital telephone, wireless communications, voice storage, and speech synthesis systems. The Q4401 uses the proprietary QUALCOMM Codebook Excited Linear Predictive (QCELP) speech coding algorithm to achieve high speech quality at low data rates.

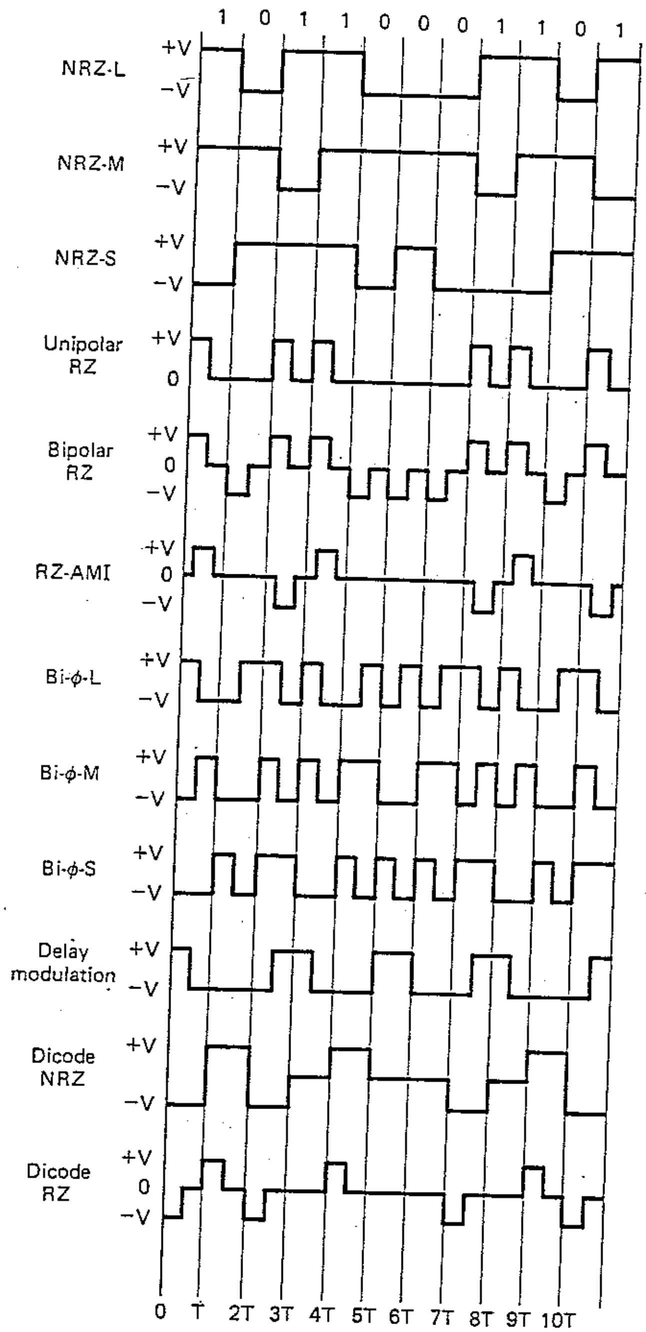
The Q4401 can encode speech at fixed or variable data rates. In Fixed Rate Mode, the Q4401 can code speech at rates of 4 kbps, 4.8 kbps, 8 kbps or 9.6 kbps. In Variable Rate Mode, the Q4401 automatically adjusts the data rate from 800 bps to 8 kbps (Normal Variable Rate Mode) or from 800 bps to 9.6 kbps (Enhanced Variable Rate Mode) every 20 milliseconds (ms). When in Variable Rate Mode, the Q4401 codes speech at under 7 kbps in continuous speech applications and at under 3.5 kbps in typical two-way telephone conversations, without degrading the speech quality.

The Q4401 is a masked ROM version of a digital signal processor (DSP) device. Digitized speech is transferred to and from the Q4401 via a digital serial interface that connects to a 64 kbps μ-law or A-law speech codec. Compressed speech packets are transferred to and from the Q4401 via an 8-bit parallel data bus interface that connects to standard microprocessor buses. The Q4401 is also controlled via this processor interface.

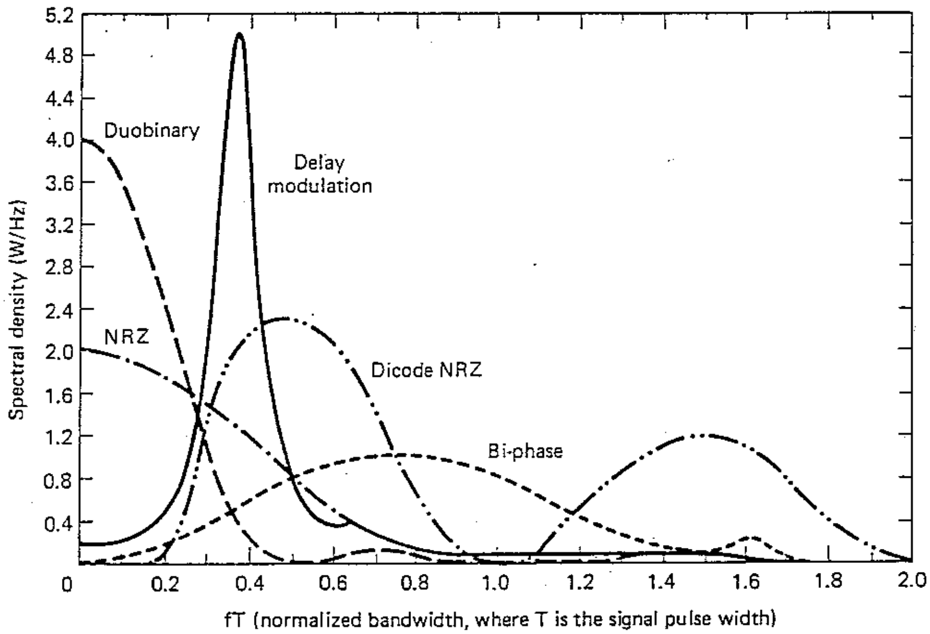
QUALCOMM Incorporated, ASIC Products
6455 Lusk Boulevard, San Diego, CA 92121-2779, USA

http://www.qualcomm.com/ProdTech/asic
E-mail: asic-products@qualcomm.com
Telephone: (619) 658-5005
Fax: (619) 658-1556

Various PCM Waveforms



Spectral Densities ofVarious PCM Waveforms

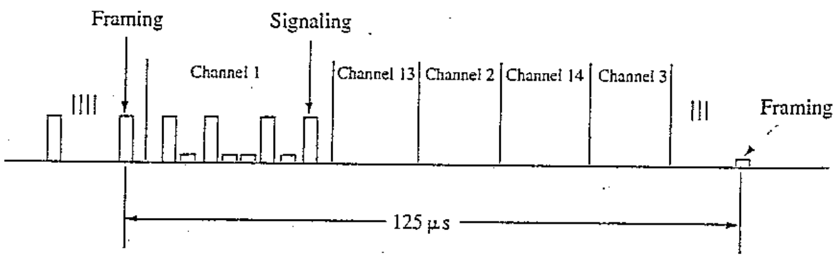


Digital Hierarchy

DS0: Single Voice Channel

Sampling speech at a rate of 8000 samples/sec. and quantizing each sample with an 8 bit PCM -> 64 kbps.

DS1 or T1: Equivalent of 24 PCM (DS0) channels.



There are 8x24+1 bits in a 125 usec frame -> 1.544 Mbps. One bit out of every 193 bit is spent for synchronization. So, the actual rate is 1.536 Mbps. Furthermore, every sixth frame, the least significant bit of each channel is used for signaling. That is each sample is actually encoded with 7 5/6 bits instead of 8 bits.

E1 (European Standard): Equivalent of 32 PCM channels: 31 information channels+1 signaling channels. The total rate is 64 kbpsx32 = 2.048 Mbps.

DS2 or T2: Equivalent of four T1 or 96 PCM channels -> 6.312Mbps.

DS3 or T3: Equivalent of 28 T1 or 672 PCM channels -> 44.736 Mbps.

DS4 or T4: Equivalent of 4032 PCM channels -> 274.176 Mbps.

Linear Prediction:

The objective here is to predict the <u>next</u> sample of a waveform based on the <u>previous</u> samples.

This will be useful in discussion of both <u>Adaptive</u> DPCM (ADPCM) and Vocoders such as CELP.

denote the samples of the signal as $x[k]$ i.e., $x[k] = x(kT_s)$.

Then a p step predictor for $\hat{x}[n]$ is:

$$\hat{x}[n] = w_1 x[n-1] + w_2 x[n-2] + \cdots + w_p x[n-p]$$

or

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

The error in prediction is the difference between the actual value $x[n]$ and $\hat{x}[n]$,

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

The objective is to <u>optimize</u> the predictor by minimizing ~~making~~ the mean-squared-value of the prediction error (MMSE: Min. mean Squared Error predictor.)

That is, to minimize:

$$J = E[e^2(t)]$$

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

or

$$J = E[x^2[n]] - 2\sum_{k=1}^{p} E[x[n]\,x[n-k]]\, w_k$$

$$+ \sum_{j=1}^{p}\sum_{k=1}^{p} w_j\, w_k\, E[x[n-j]\,x[n-k]]$$

$E[x^2[n]] = \sigma_x^2$ if $x[n]$ is zero-mean & stationary.

$$E[x[n]\,x[n-k]] = R_x(k) \triangleq R_x(kT_s)$$

so:

$$J = \sigma_x^2 - 2\sum_{k=1}^{p} w_k R_x[k] + \sum_{j=1}^{p}\sum_{k=1}^{p} w_j w_k R_x[k-j]$$

let $\dfrac{\partial J}{\partial w_k} = 0$ all k

we get

$$\sum_{j=1}^{p} w_j R_x(k-j) = R_x(k) = R_x(-k)$$

$$k = 1, 2, \ldots, p$$

this means:

$$R_x(1) = w_1 R_x(0) + w_2 R_x(1) + \cdots + w_p R_x(p-1)$$

$$R_x(2) = w_1 R_x(1) + w_2 R_x(0) + w_3 R_x(1) \cdots + w_p R_x(p-2)$$

$$\vdots$$

$$R_x(p) = w_1 R_x(p-1) + \cdots \qquad + w_p R_x(0)$$

let $\underline{W}_o = [w_1, w_2, \cdots w_p]^T = \begin{bmatrix} w_1 \\ w_2 \\ \vdots \\ w_p \end{bmatrix}$

$$\underline{r}_x = \begin{bmatrix} R_x(1) \\ R_x(2) \\ \vdots \\ R_x(p) \end{bmatrix}$$

and

$$\underline{R}_x = \begin{bmatrix} R_x(0) & R_x(1) & \cdots & R_x(p-1) \\ R_x(1) & R_x(0) & & R_x(p-2) \\ \vdots & \vdots & & \\ R_x(p-1) & R_x(p-2) & \cdots & R_x(0) \end{bmatrix}$$

then

$$\underline{R}_x \underline{W}_o = \underline{r}_x$$

or

$$\boxed{\underline{W}_o = \underline{R}_x^{-1} \underline{r}_x}$$

and $J_{min} = \sigma_x^2 - r_x^T R_x^{-1} r_x$

Adaptive Linear Prediction

Take

$$g_k = \frac{\partial J}{\partial w_k} \qquad k = 1, 2, \ldots, p$$

note that a positive g_k means that increasing w_k results in higher distortion, so, w_k should be reduced and vice versa, so, we use the following rule for <u>updating</u> the taps:

$$w_k[n+1] = w_k[n] - \frac{1}{2} \mu g_k \qquad k = 1, 2, \ldots, p$$

where μ specifies the step size.

larger $\mu \Rightarrow$ faster convergence but higher jitter

smaller $\mu \Rightarrow$ slow convergence but, lower jitter.

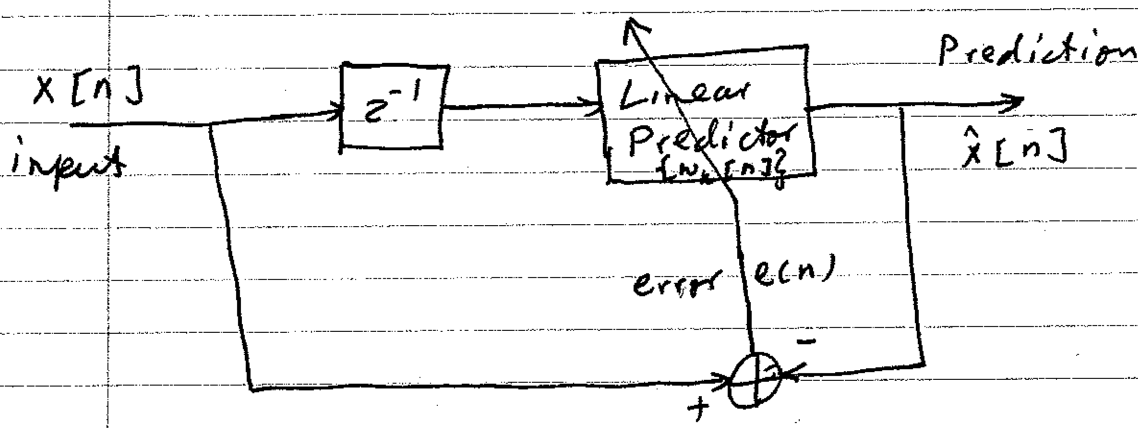
with some minor mathematical manipulation, we get

$$\hat{w}_k[n+1] = w_k[n] + \mu x[n-k] e(n)$$

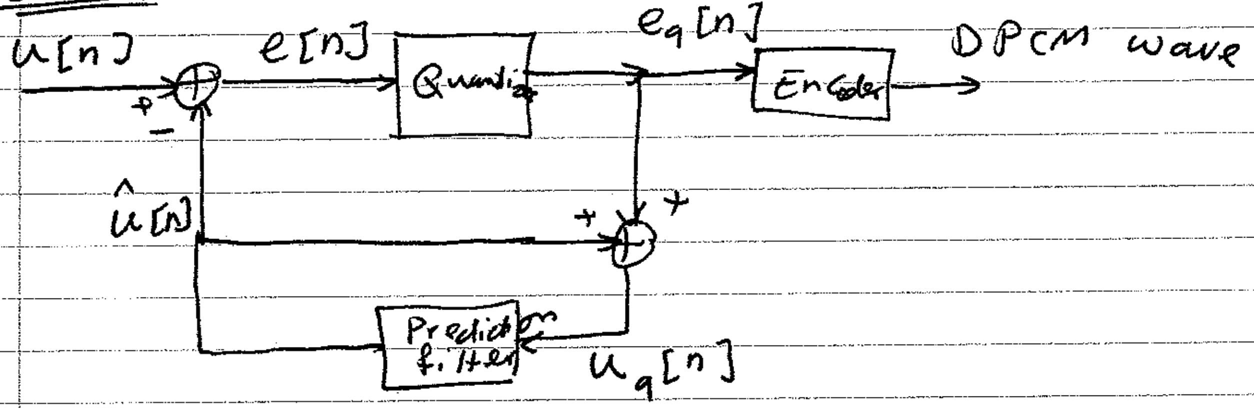
where

$$e(n) = x[n] - \sum_{j=1}^{p} \hat{w}_j[n] x[n-j]$$

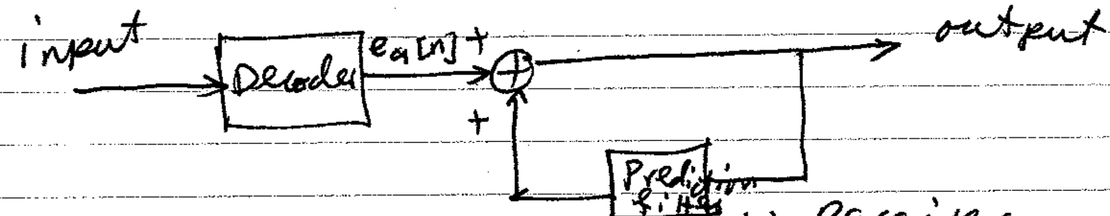
So, we have the following block Diagram:



DPCM



a) Transmitter



b) Receiver

$e[n] = u[n] - \hat{u}[n]$ & $e_q[n] = e[n] + q[n]$

But $e_q[n] = u[n] - \hat{u}[n] + q[n]$

$$u_q[n] = \hat{u}[n] + e_q[n] = u[n] + q[n]$$

$(SNR)_o = \dfrac{\sigma_M^2}{\sigma_Q^2}$ The overall SNR

$$(SNR)_0 = \frac{\sigma_M^2}{\sigma_Q^2}$$

σ_M^2 = Power of the signal (message)

σ_Q^2 = variance of quantization error

we can write

$$(SNR)_0 = \frac{\sigma_M^2}{\sigma_Q^2} = \left(\frac{\sigma_M^2}{\sigma_E^2}\right)\left(\frac{\sigma_E^2}{\sigma_Q^2}\right)$$

$$= G_p (SNR)_Q$$

wher G_p is the processing gain

$$G_p = \frac{\sigma_M^2}{\sigma_E^2}$$

(it is the ratio of the message power to error signal power.) and

$$(SNR)_Q = \frac{\sigma_E^2}{\sigma_Q^2}$$

is the SNR of the [inner] quantizer.

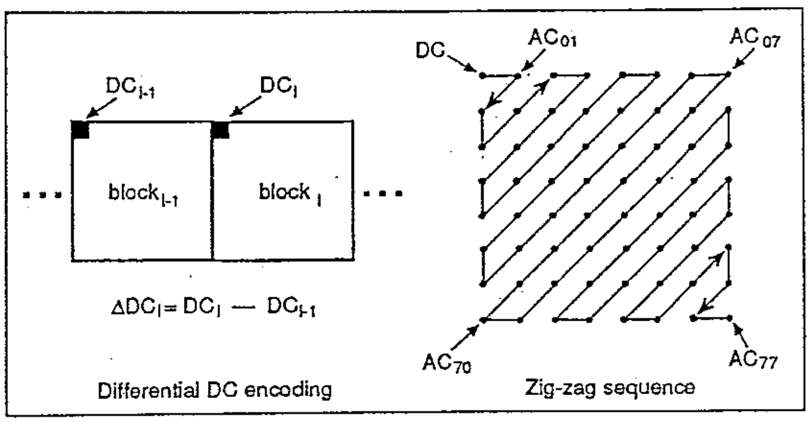
In order to improve the overall performance, ~~of the~~ we need to improve $\frac{\sigma_M^2}{\sigma_E^2}$, i.e., to reduce σ_E^2.

JPEG Image Compression Algorithm

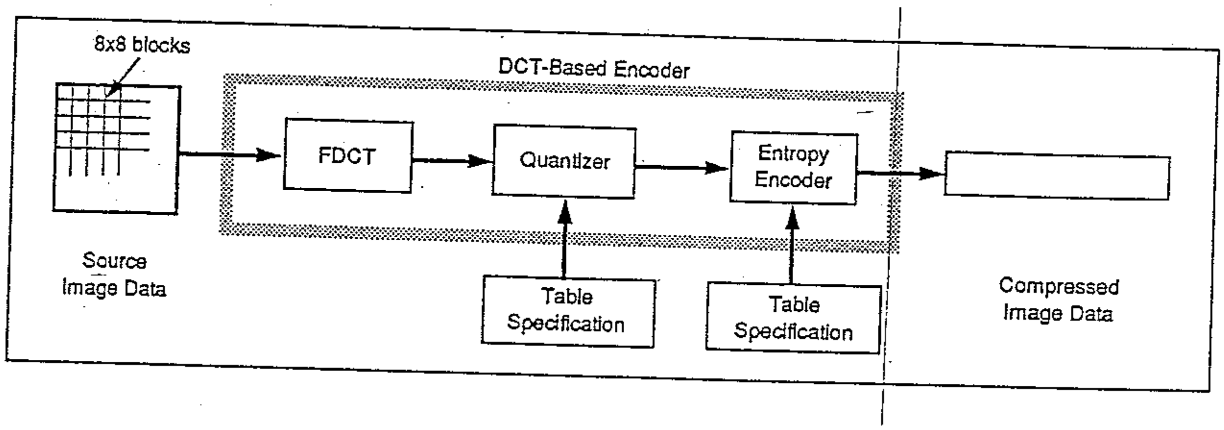
JPEG Developed by Joint Photographic Expert Group is the result of collaboration between CCITT and ISO.

TJPEG is a Transform Coding Technique consisting of the following steps:

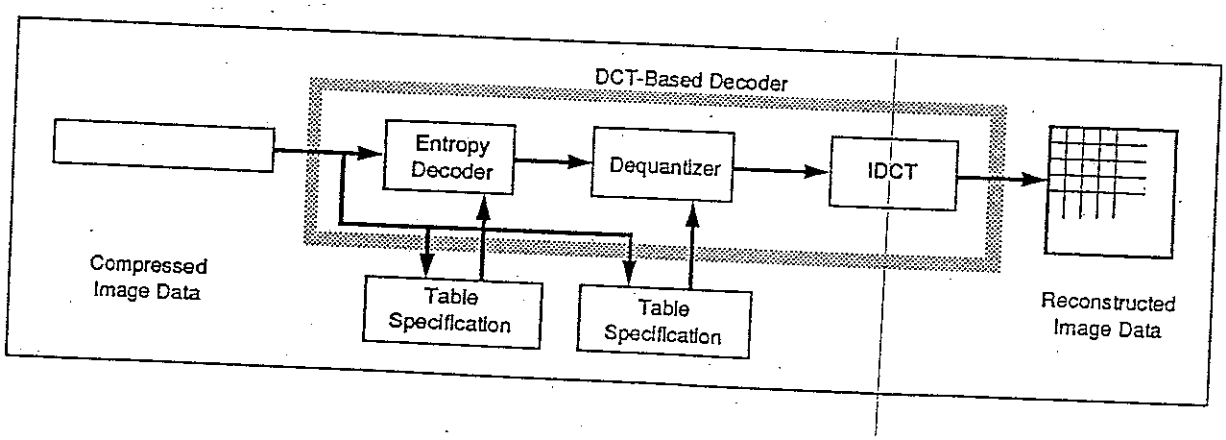
1- Segmenting the image into 8x8 blocks,
2- Taking Discrete Cosine Transform (DCT) of each block resulting in 63 AC and 1 DC coefficient,
3- Quantizing the DCT coefficients using step sizes that depend on the required performance, to utilize the inter-block correlation, the DC (mean) coefficients are quantized using DPCM,
4- Compressing the output of quantizer using entropy coding (Hufmann Coding). That is assigning smaller number of bits to more likely (usually small magnitude) values.



JPEG Encoder



JPEG Decoder



MPEG Video Compression

In Video Compression in addition to removing spatial (intra-frame) redundancy, one tries to also remove temporal (inter-frame) redundancy using motion compensated interpolation.

In MPEG, temporal redundancy removal is achieved using three types of pictures:

I or Intra-pictures: these are quite high resolution pictures (low compression) and are used for random access and reference for predicted pictures,

P or Predicted pictures: these are encoded with reference to a previous picture (either Intra- or Predicted) and are used as reference for future Predicted pictures.

B or Bidirectional predicted pictures: these are most compressed images and require both a past and a future picture for prediction. The B pictures are never used as reference for other pictures.