More quantization noise

Book uses formulas containing "sinc ..." terms.

This is based on ideal sampling —
using sinc(---) waveforms to multiply the signal --
as opposed to impulses or square (zero-order) pulses.

Noise with other sampling similar.

$m(t)$ = the original signal $= \sum m(kT_s)\, s(t)$

$m(kT_s)$ = k^{th} sample of $m(t)$

$\hat{m}(kT_s)$ = quantized value of $m(kT_s)$

$\hat{m}(t)$ = signal reconstructed from quantized samples.
$= \sum \hat{m}(kT_s)\, s(t)$

where
$s(t)$ = the sample waveform

in book
$s(t) = \text{sinc}(2\pi Bt - k\pi)$

Quantization error:

$$q(t) = \hat{m}(t) - m(t)$$

$$\left(= \sum \underbrace{\left[\hat{m}(kT_s) - m(kT_s)\right]}_{= q(kT_s)} s(t)\right)$$

$q(kT_s)$: quantization error in kth sample.

To find S/N ratio ---
Find power in the noise signal
(derivation -- p. 266 omitted)

Leads to: $\dfrac{S_o}{N_o} = 3L^2\left(\dfrac{\widetilde{m^2(t)}}{m_p^2}\right) \rightarrow$ $\dfrac{\text{mean square value}}{\text{peak value}}$

= .5 for sine wave
= 1 for square wave

Where does this come from?

L = # of levels

L^2 = converts it to power

3 = coefficient indicating that statistically, the mean square value of the noise is $\frac{1}{3}$ of its peak value.

Emperical derivation.

$\dfrac{S}{N} = \dfrac{\#\text{ of levels}}{1\text{ level}} = L$

Voltage

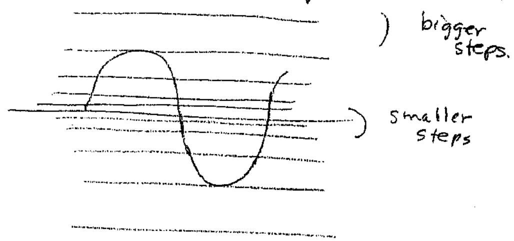
worst case -
mult. by $\sqrt{\frac{1}{3}}$ to get typical.

$\dfrac{S}{N} = L^2$

power

Non-uniform quantization

SNR can be improved by using smaller steps near 0.



Application ...

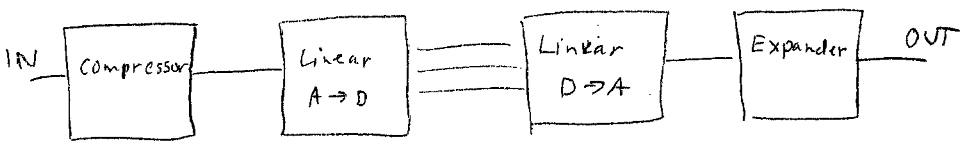
Normal speech levels are usually much lower than the peak value.

Less resolution is needed at higher levels — so the % resolution is similar.

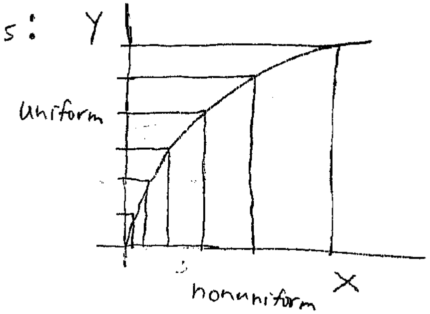
Noise is less important when signal is bigger.

How to:

Analog method.



Compressor and expander are nonlinear circuits:



Compressor: input is x
output is y

expander: input is y
output is x

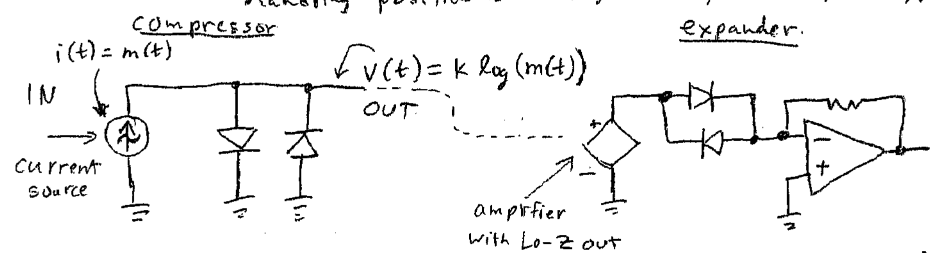
Common curves:

① Diode $V = \frac{KT}{q} \ln\left(1 + \frac{I}{I_s}\right)$

compressor: input is I
output is V

expander: input is V
output is I

Usually, the circuit would use a pair of diodes — handling positive and negative signals separately.



(2) "μ-law" or "A-law"

A standard — to specify exact curve --

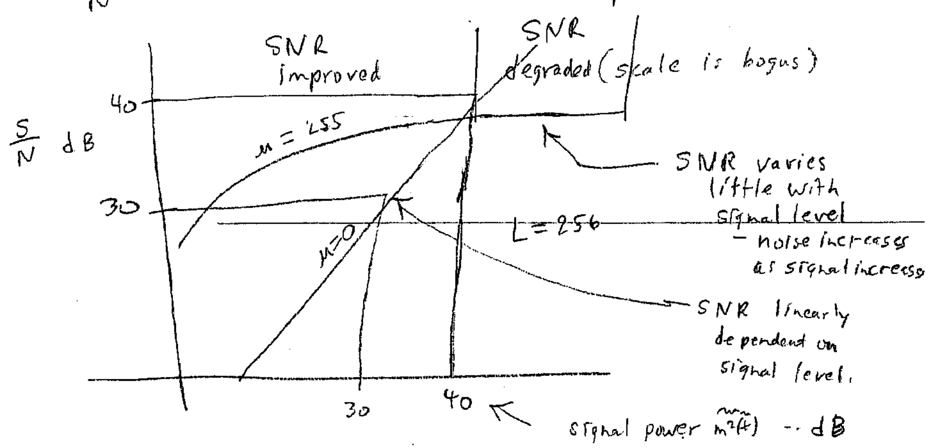
μ-law
$$y = \frac{1}{\ln(1+\mu)} \ln\left(1 + \frac{\mu m}{m_p}\right) \qquad 0 \le \frac{m}{m_p} \le 1$$

m = signal sample
m_p = peak value.

A-law
$$y = \begin{cases} \dfrac{A}{1+\ln A}\left(\dfrac{m}{m_p}\right) & 0 \le \dfrac{m}{m_p} \le \dfrac{1}{A} \\[2ex] \dfrac{1}{1+\ln A}\left(1 + \ln \dfrac{Am}{m_p}\right) & \dfrac{1}{A} \le \dfrac{m}{m_p} \le 1 \end{cases}$$

These are two curves commonly used—
pick the value of μ or A
for the amount of compression you want.
$\mu = 0$ or $A = 1$
means no compression.

$\frac{S}{N}$ ratio with + without compression --



Digital method ---
When you have a good A-D converter, (lots of levels)
and fewer levels in the channel.
Do it in software -- Lookup table.

<u>Transmission bandwidth.</u>
If we transmit bits serially ---
suppose we have bandwidth of $m(t)$ is B.
Encode it with <u>n</u> bits.
<u>2B</u> samples per second -
we need <u>2nB</u> bits per second
A cycle can transmit 2 bits,
so we need a bandwidth of <u>nB</u>
to transmit a bit stream.

Output SNR – improves.

$$\frac{S_o}{N_o} = c\,(2^{2n})\,L^2 = c\,2^{\frac{2B_T}{B}}$$

$$c = \frac{3\widetilde{m^2(t)}}{m_p^2} \quad \text{– uncompressed.}$$

Why? – only need decision – $\frac{1}{2}$ highest level.

SNR increases exponentially with B_T

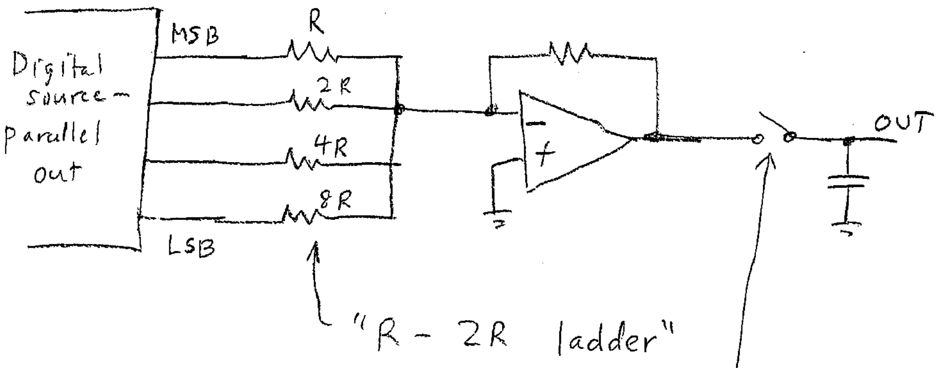
$$\frac{S}{N}_{DB} = 10\log_{10}\left(\frac{S}{N}\right) = \underbrace{10\log_{10} c}_{\text{constant}} + \underbrace{6n}$$

↑ increase n by 1 results in 6 dB improvement

not crossed out.

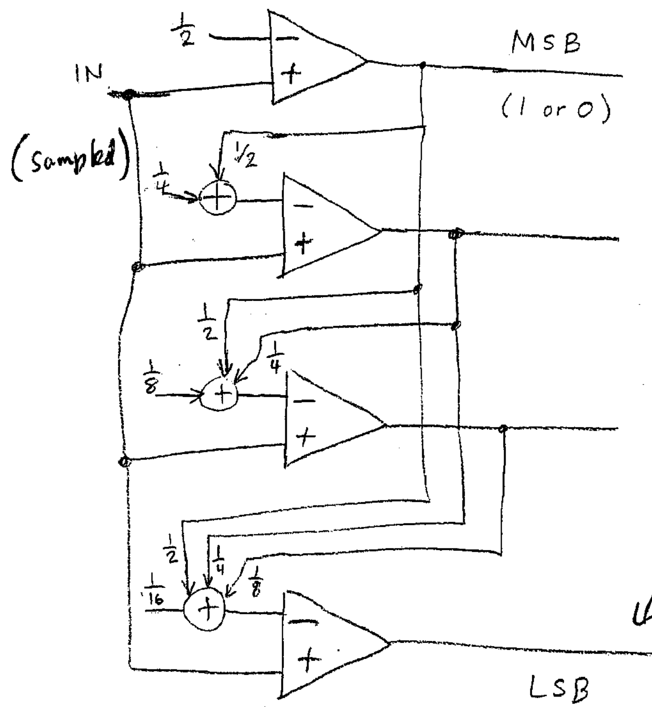
This is only quantization error ~~[illegible]~~. no— don't
Not pulse detection error.

A simple D–A converter 10A-7



Switch is closed when signal is stable. Open during switching to avoid glitches.

A simple A – D converter 10A-8



takes a while to settle down on each sample

"Successive approximation" method.

Differentrial Pulse code modulation

Instead of sending the code directly,

Try to predict the value, then send the difference between predicted and actual.

Advantage:
Usually difference is small, so fewer bits needed.

Prediction methods used:

Simple: same value.

Curve fit: Use the most recent n samples.

Do a polynomial curve fit.

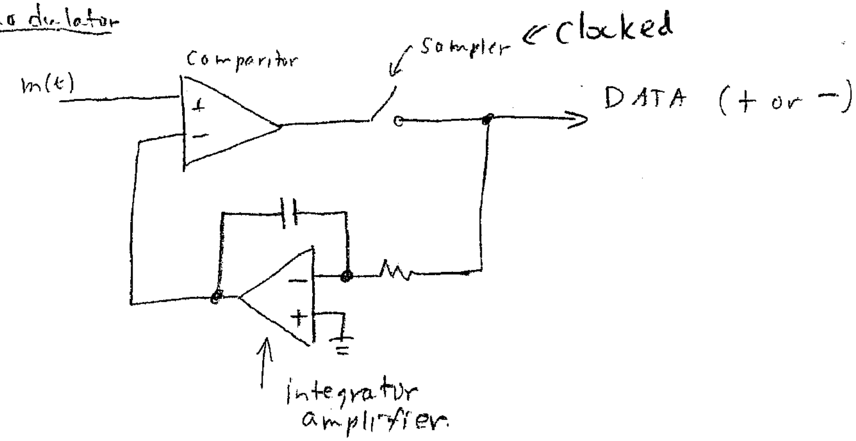
Use this to predict next value.

Delta-modulation

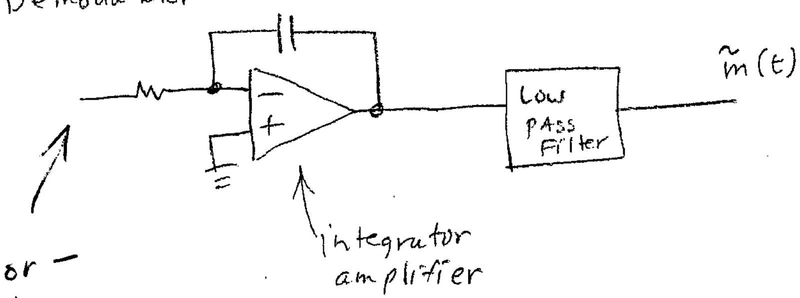
One-bit variant of DPCM.

Just say whether it is up or down.

Modulator



Demodulator:



+ or − pulses.
fixed amplitude.

(show waveforms)

Sending it –
just send the stream –
No synchronization needed.
But it does need "DC restore"