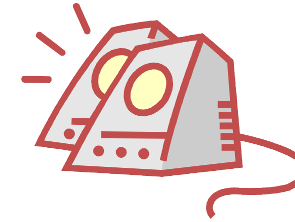
Multimedia Computing

Basics of Digital Audio

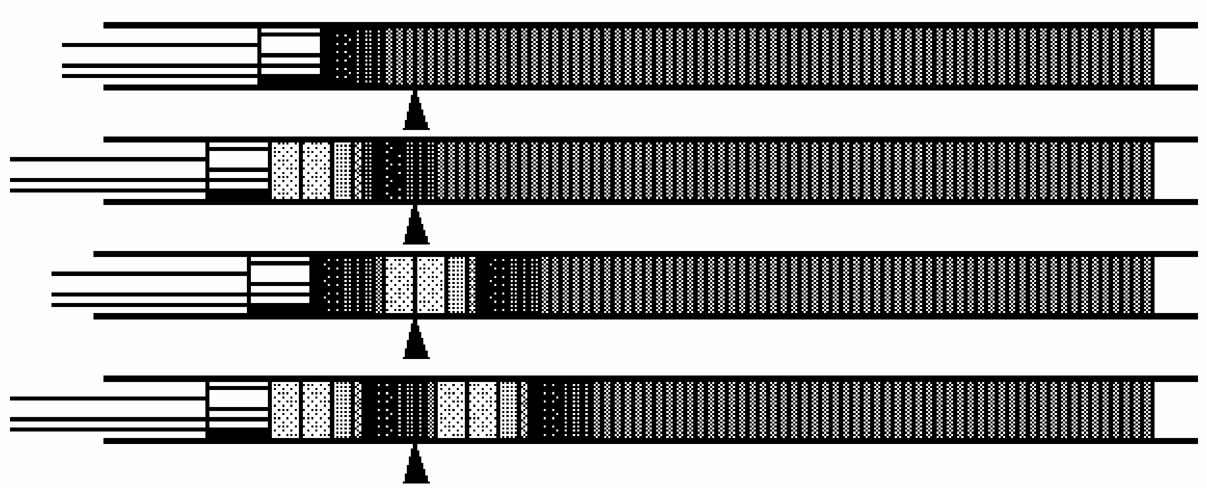
What is Sound?



- Sound is a wave phenomenon like light, and it involves molecules of air being compressed and expanded under the action of some physical device.
  - E.g., a speaker in an audio system vibrates back and forth and produces a pressure wave that we perceive as sound.
  - Since sound is a pressure wave, it takes on continuous values, as opposed to digitized ones.
  - If we wish to use a digital version of sound waves we must form digitized representations of audio information.

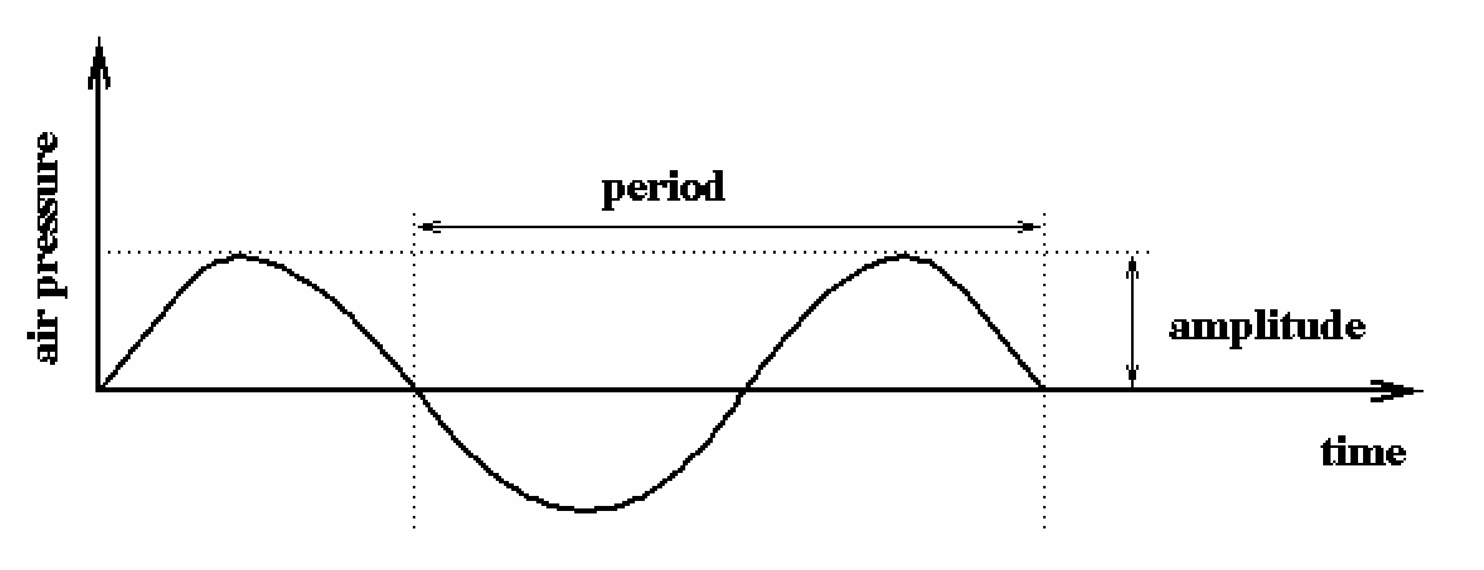
What is Sound?

- Sound is a continuous longitudinal wave that travels through the air. The wave is made up of pressure differences.
- Sound waves have usual wave properties (reflection, refraction, diffraction, etc.) and it is detected by measuring the pressure level at a point.



What is Sound?

- The amplitude of a sound is the measure of displacement of the air pressure wave from its mean.

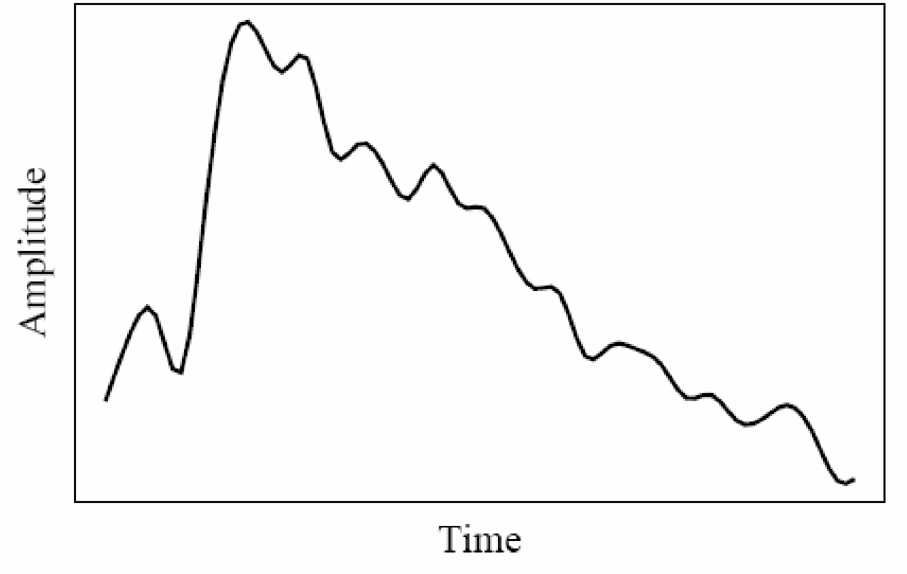


Outline

- Digitization of Sound
- Quantization
- Coding of Audio

Digitization

- The 1-dimensional nature of sound: amplitude values depend on a 1D variable, time.
- To digitize, the signal must be **sampled** in each dimension: in time and in amplitude.

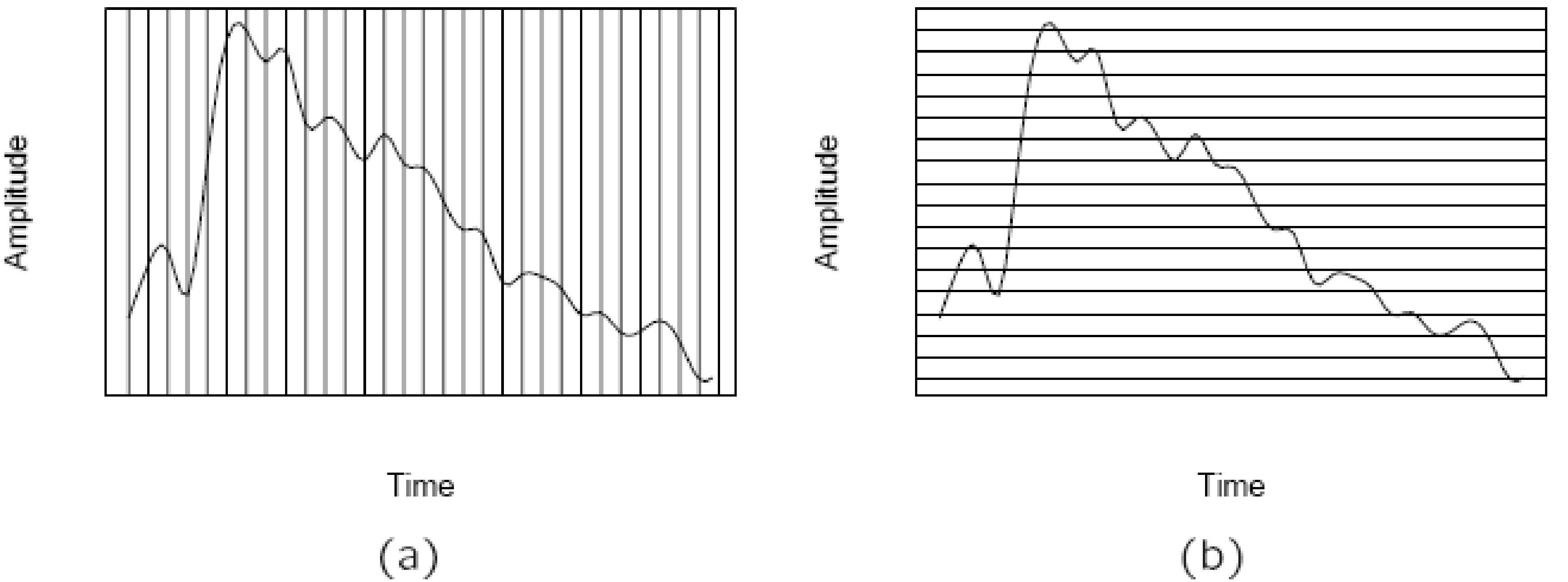


An analog signal: continuous measurement of pressure wave.

Digitization

- **Digitization** means sampling the analog signal to a stream of numbers, and preferably these numbers should be integers for efficiency.
  - Sampling means measuring the quantity we are interested in, usually at evenly-spaced intervals.
  - The sampling, using measurements only at evenly spaced time intervals, is simply called sampling. The rate at which it is performed is called the sampling frequency.
    - For audio, typical sampling rates are from 8 kHz (8,000 samples per second) to 48 kHz.
  - Sampling in the amplitude or voltage dimension is called **quantization**.

Digitization



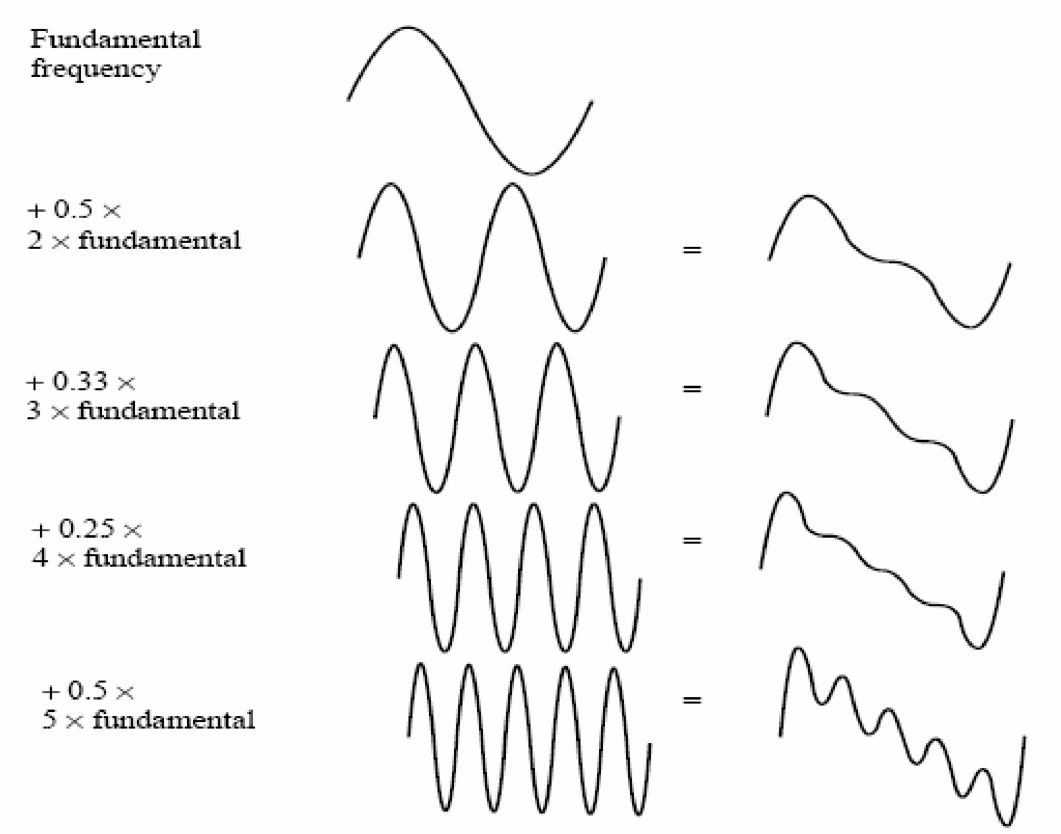
Sampling and Quantization. (a): Sampling the analog signal in the time dimension. (b): Quantization is sampling the analog signal in the amplitude dimension.

Two key factors in digitization

- To decide how to digitize audio data we need to answer the following questions:
  - What is the sampling rate?
    - This is about the sampling of time dimension.
  - How finely is the data to be quantized, and is quantization uniform?
    - This is about the sampling of amplitude dimension. We will discuss this later.

Nyquist Theorem

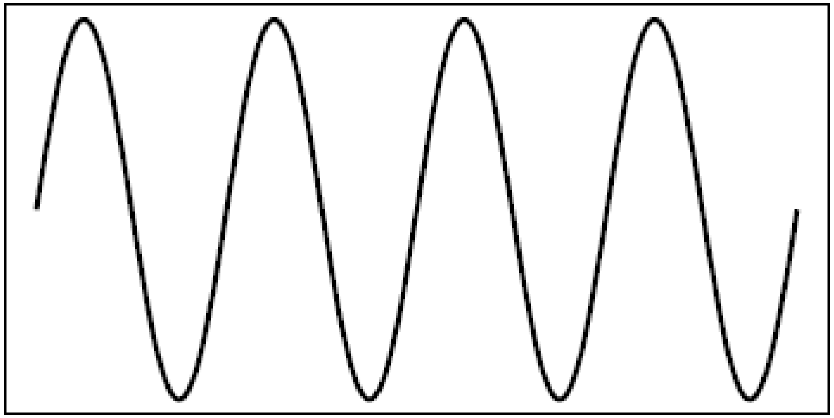
- Signals can be decomposed into a sum of sinusoids.



Building up a complex signal by superposing sinusoids

Nyquist Theorem

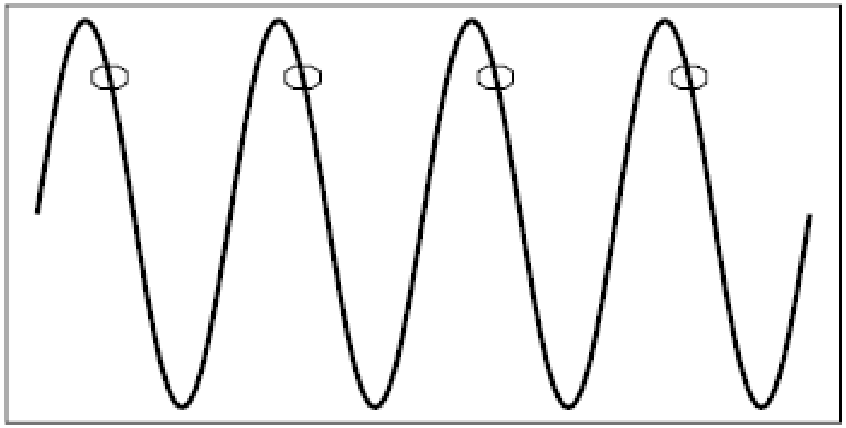
- The Nyquist theorem states how frequently we must sample in time to be able to recover the original sound.
  - This figure shows a single sinusoid: it is a single, pure, frequency (only electronic instruments can create such sounds).



(a)

Nyquist Theorem

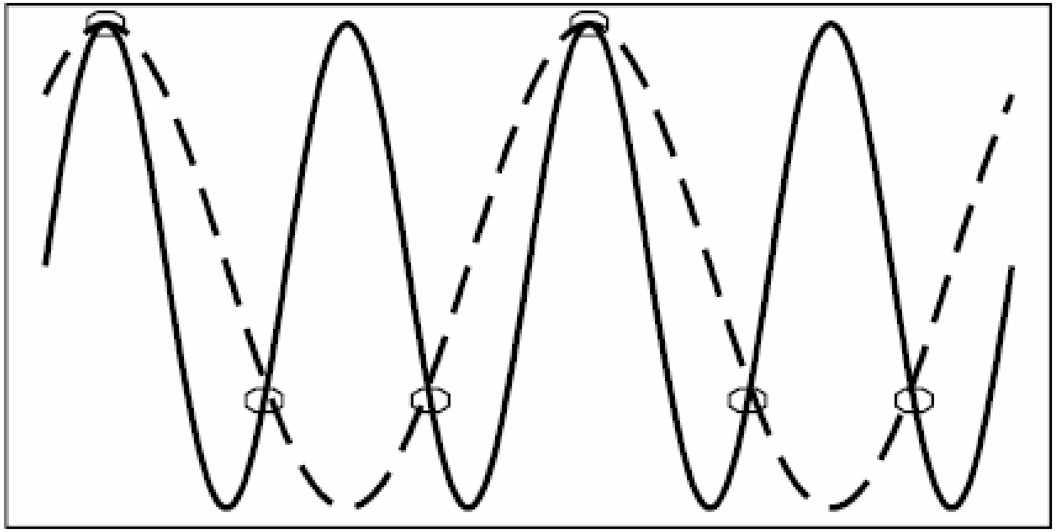
- If sampling rate just equals the actual frequency, a false signal will be detected: it is simply a constant, with zero frequency.



(b)

Nyquist Theorem

- Now if sample at 1.5 times the actual frequency, we will obtain an incorrect (**alias**) frequency that is lower than the correct one.



(c)

Nyquist Theorem

- For correct sampling we must use a sampling rate equal to at least twice the maximum frequency content in the signal. This rate is called the Nyquist rate.
- In practice, we usually consider band-limited signals, i.e., there is a lower frequency limit f_1 and an upper frequency limit f_2 of the components in the signal, the difference $\Delta f = f_2 - f_1$ is called bandwidth.
- **Nyquist Theorem**: If a signal is band-limited with bandwidth Δf, then the sampling rate (Nyquist rate) should be at least $2\Delta f$.

Nyquist Frequency

- **Nyquist frequency**: half of the Nyquist rate. (The two terms are somewhat confusing.)
  - Suppose we have a fixed sampling rate. Since it is impossible to recover frequencies higher than Nyquist frequency in any event, most systems use an **anti-aliasing filter** to restrict the frequency content in the input to the sampler to a range at or below Nyquist frequency.
- If the Sampling Frequency is higher than True Frequency but less than twice of it, the Alias Frequency is defined as follows:

$$f_{alias} = f_{sampling} - f_{true}; \text{ for } f_{true} < f_{sampling} < 2f_{alias}$$

Outline

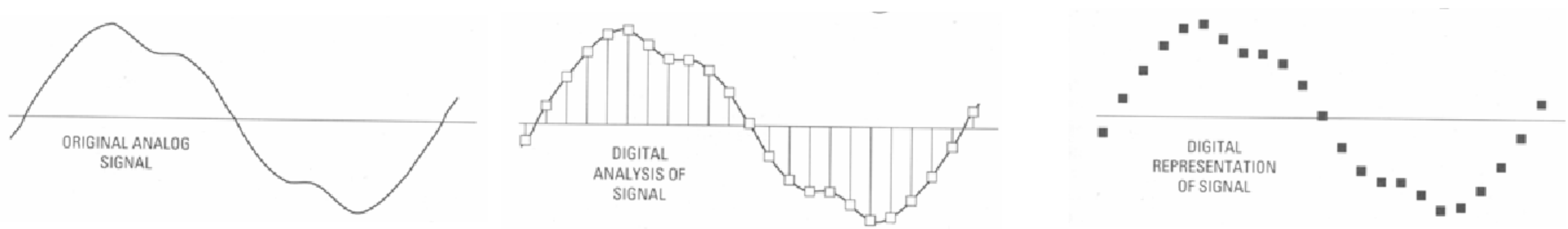
- Digitization of Sound
- Quantization
- Coding of Audio

Digitization of audio signal

$x_c(t)$ → Sampler → $x(n)=x_c(nT)$ → Quantizer → $x(n)$

continuous-time
audio signal

discrete sequence
audio samples



Quantization Examples

- Examples
  - Continuous to discrete
    - A quarter of milk, two gallons of gas, normal temperature is 20.5^{o}C, Yao Ming's height is 217cm
  - Discrete to discrete
    - Round your tax return to integers.
    - The temperature today is 5^{o}C (round from 5.4^{o}C).
- Play with bits
  - Precision is finite: the more precise, the more bits you need.
    - You can use 8bits to represent 252 but you need more bits to represent 252.56.
  - However, not every bit has the same impact
    - How much did you pay for your car?
      (100,000 HK$ vs. 100,001HK$)

Signal to Noise Ratio (SNR)

- The ratio of the power of the correct signal and the noise is called the *signal to noise ratio* (**SNR**) – a measure of the quality of the signal.
- The SNR is usually measured in decibels (dB). The SNR value, in units of dB, is defined in terms of base-10 logarithms of powers, as follows:

$$\boxed{SNR = 10\log_{10}\frac{V_{signal}^2}{V_{noise}^2} = 20\log_{10}\frac{V_{signal}}{V_{noise}}}$$

SNR

- The power in a signal is proportional to the square of the voltage. For example, if the signal voltage V_{signal} is 10 times the noise, then the SNR is 20 log10(10)=20dB.
- The usual levels of sound we hear around us are described in terms of decibels, as a ratio to the quietest sound we are capable of hearing.

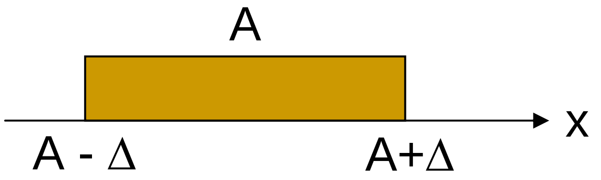
Threshold of hearing	0
Rustle of leaves	10
Very quiet room	20
Average room	40
Conversation	60
Busy street	70
Loud radio	80
Train through station	90
Riveter	100
Threshold of discomfort	120
Threshold of pain	140
Damage to ear drum	160

Signal to Quantization Noise Ratio (SQNR)

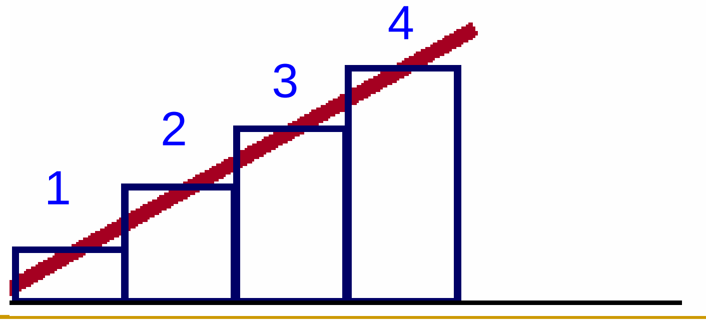
- Aside from any noise corrupted in the original analog signal, there is also an additional error that results from quantization.
  - If voltages are actually in 0 to 1 but we have only 8 bits in which to store values, then we have to force all continuous values of voltage into only 256 different values.
  - This introduces a round-off error. It is not really "noise". Nevertheless it is called **quantization noise** (or quantization error).
- The quality of the quantization is characterized by the Signal to Quantization Noise Ratio (**SQNR**).
  - **Quantization noise**: the difference between the actual value of the analog signal, for the particular sampling time, and the quantized value.

Uniform Quantization

Uniform quantization (UQ) divides the axis into uniformly distributed intervals, and approximates all the values fall into the interval with a single value.



Example



Example: SQNR

$$SQNR = 10\,log_{10}\frac{V_{signal}^{2}}{V_{quan\_noise}^{2}} = 20\,log_{10}\frac{V_{signal}}{V_{quan\_noise}}$$

- Suppose the original signal is X=[1.1, 2.2, 3.4, 3.8]. After uniform quantization, it is quantized into Y=[1, 2, 3, 4]. What is the SQNR of Y?

The quantization noise is: N=X-Y=[0.1 0.2 0.4 -0.2]

The powers of original signal and quantization noise are:

$$V_{signal}^{2} = 1.1^2 + 2.2^2 + 3.4^2 + 3.8^2 = 32.05$$

$$V_{quan\_noise}^{2} = 0.1^2 + 0.2^2 + 0.4^2 + (-0.2)^2 = 0.25$$

The SQNR is:

$$SQNR = 10\,log_{10}\frac{32.05}{0.25} = 21.0789dB$$

Peak SQNR (PSQNR)

- Suppose we choose a quantization accuracy of *N* bits per sample. One bit is used to indicate the sign of the sample. Then the maximum signal value is mapped to $2^{N-1}-1$ ($\approx 2^{N-1}$) and the most negative signal is mapped to -2^{N-1}.
- The uniform quantization interval is 1, so the quantization error is at most ½, the half of the interval.
- The SQNR can be simply expressed:

$$SQNR = 20\log_{10}\frac{V_{signal}}{V_{quan\_noise}} = 20\log_{10}\frac{2^{N-1}}{\frac{1}{2}}$$

$$= 20 \times N \times \log 2 = 6.02\,N(\text{dB})$$

- The above equation is the *Peak SQNR* (PSQNR): peak signal and peak noise.

The 6dB/bit rule

$$SQNR = 20 \log_{10} \frac{V_{signal}}{V_{quan\_noise}} = 20 \log_{10} \frac{2^{N-1}}{\frac{1}{2}}$$

$$= 20 \times N \times \log 2 = 6.02\, N (\text{dB})$$

- 6.02N is the worst case because we assume the noise is the maximum ½.
- We can see that for a uniformly quantized source, adding 1 bit/sample can improve the SNR by 6dB. This is called the 6dB rule.

Non-uniform quantization

- Non-uniform (non-linear) quantization: set up more finely-spaced levels where humans hear with the most acuity.
- Human auditory system exhibits a logarithmic sensitivity
    - More sensitive at small-amplitude range (e.g., 0 might sound different from 0.1)
    - Less sensitive at large-amplitude range (e.g., 0.7 might not sound different much from 0.8)
- Basic idea: assign smaller quantization step-size for small-amplitude regions and larger quantization step-size for large-amplitude regions.

Non-uniform quantization

- Nonlinear quantization works by first transforming an analog signal from the raw space *x* into another space *y*, and then uniformly quantizing the resulting values.
- Two types of nonlinear mapping functions
  - **Mu-law** adopted by North American telecommunications systems
  - **A-law** adopted by European telecommunications systems
  - The two laws are very similar.

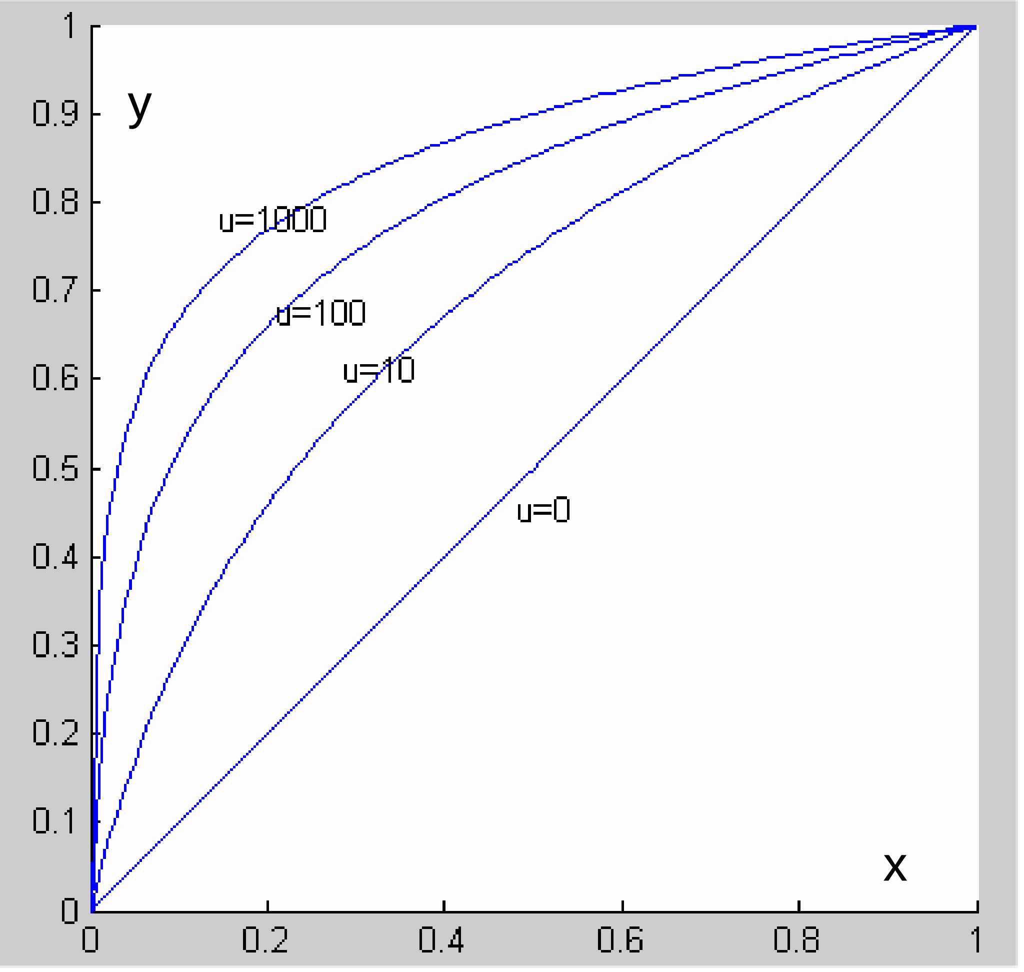
Mu-Law (μ-law)

$$y = F(x) = \operatorname{sgn}(x)\frac{\ln(1+\mu|x|)}{\ln(1+\mu)} \qquad -1 \le x \le 1$$

$$\Updownarrow$$

$$x = F^{-1}(y) = \operatorname{sgn}(y)\left(1/\mu\right)\left[(1+\mu)^{|y|} - 1\right] \qquad -1 \le y \le 1$$

Mu-Law Examples



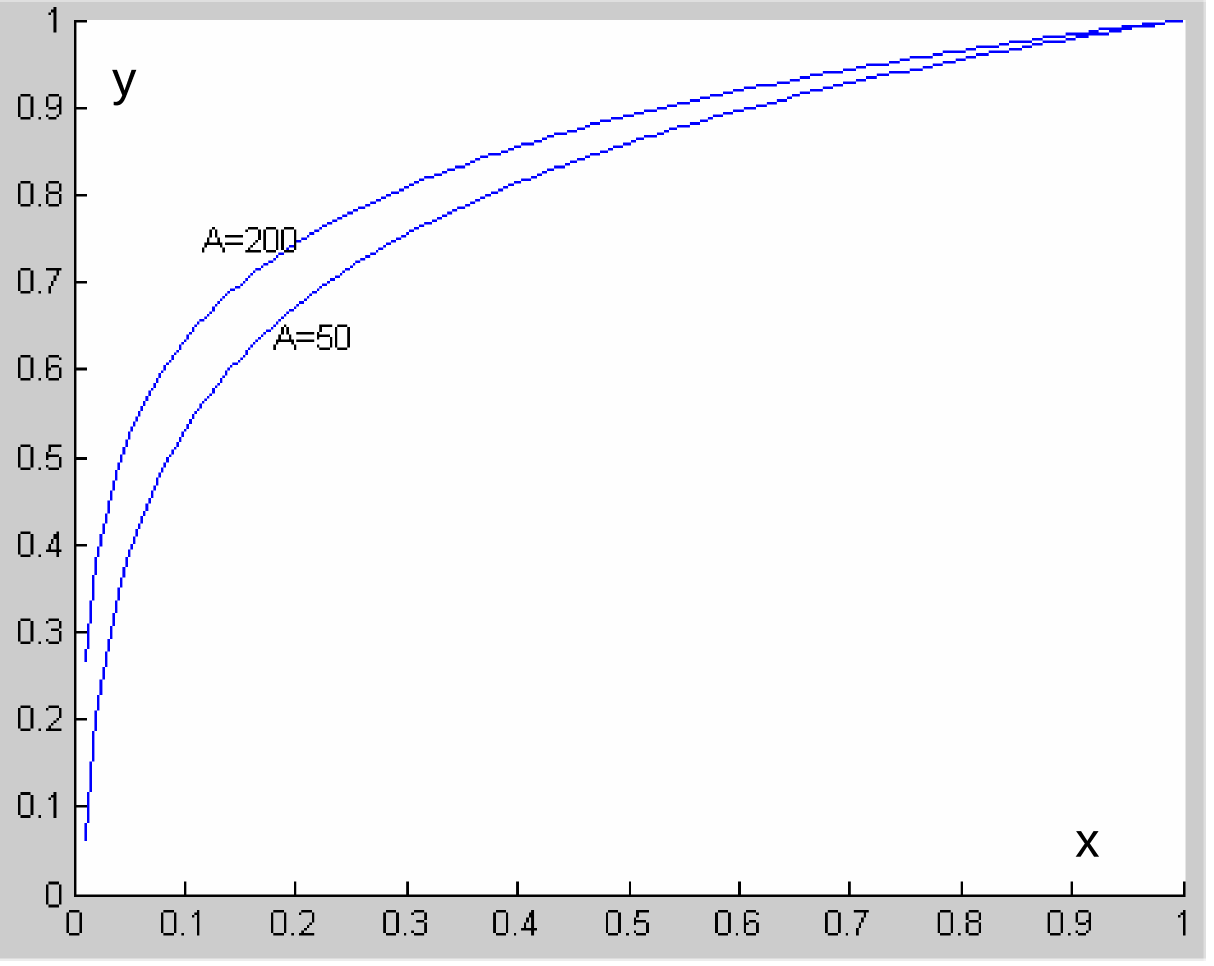
A-Law

$$y = F(x) = \operatorname{sgn}(x)\begin{cases} \dfrac{A|x|}{1+\ln(A)} & 0 \le |x| \le \dfrac{1}{A} \\[2ex] \dfrac{1+\ln(A|x|)}{1+\ln(A)} & \dfrac{1}{A} \le |x| \le 1 \end{cases}$$

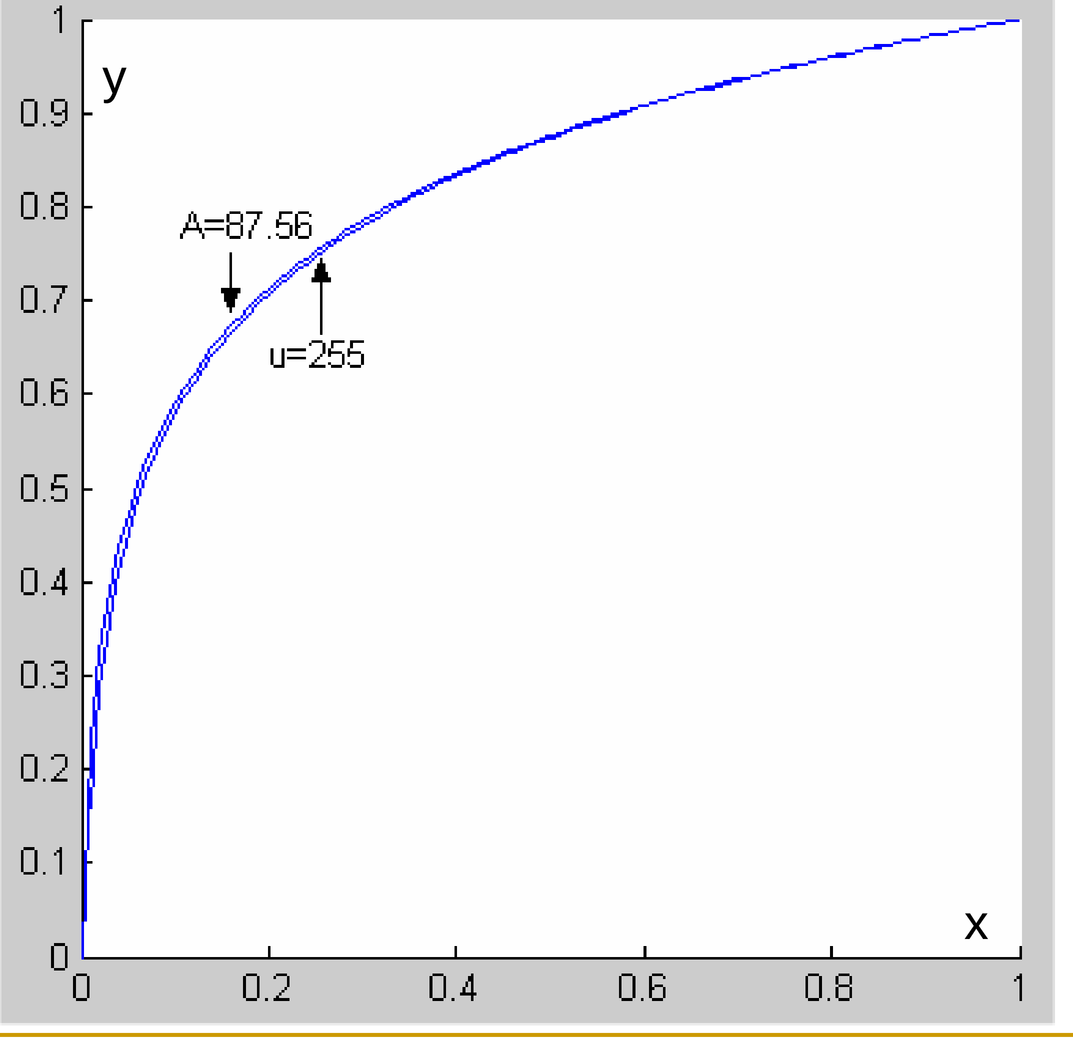
$$\Updownarrow$$

$$x = F^{-1}(y) = \operatorname{sgn}(y)\begin{cases} \dfrac{|y|(1+\ln(A))}{A} & 0 \le |y| < \dfrac{1}{1+\ln(A)} \\[2ex] \dfrac{\exp(|y|(1+\ln(A))-1)}{A(1+\ln(A))} & \dfrac{1}{1+\ln(A)} \le |y| < 1 \end{cases}$$

A-Law Examples



Comparison



Audio Filtering

- Prior to sampling and A/D (analogy to digital) conversion, the audio signal is also usually *filtered* to remove unwanted frequencies. The frequencies kept depend on the application:
  - For speech, typically from 50Hz to 10kHz is retained, and other frequencies are blocked.
  - An audio music signal will typically contain from about 20Hz up to 20kHz.

Outline

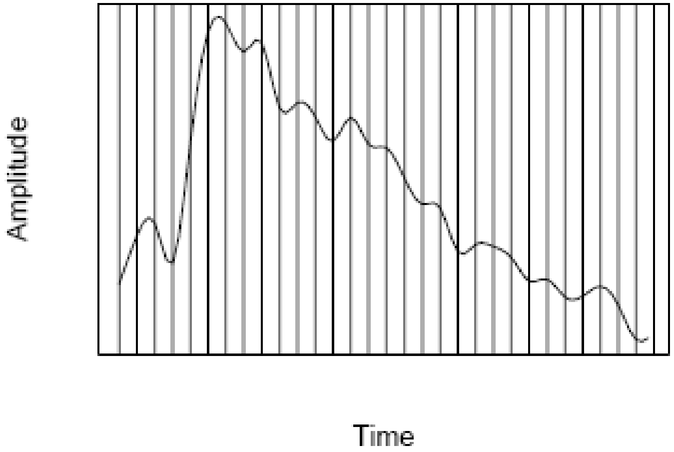
- Digitization of Sound
- Quantization
- Coding of Audio

Coding of Audio

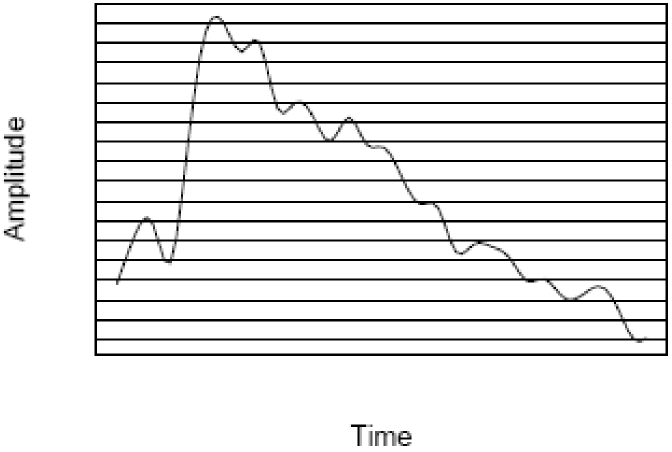
- Quantization and transformation of data are collectively known as **coding** of the data.
  - For audio, the Mu-law technique is usually combined with an algorithm that exploits the temporal redundancy in audio signals.
  - Differences in signals between the present and a past time can reduce the size of signal values and also concentrate the pixel values into a much smaller range.
  - The entropy coding (Huffman coding, arithmetic coding, etc.) technique can then be used to produce a bit-stream of the signal.
- In general, producing quantized output for audio
  - is called **PCM** (Pulse Code Modulation).
  - The differences version is called **DPCM**.
  - The adaptive version is called **ADPCM**.

Pulse Code Modulation (PCM)

- The basic techniques for creating digital signals from analog signals are **sampling** and **quantization**.
- Quantization consists of selecting breakpoints in magnitude, and then remapping any value within an interval to one of the representative output levels.



Sampling



Quantization

PCM

- After sampling and quantization, we may wish to compress the data, by assigning a bit stream that uses fewer bits for the most prevalent signal values.
- Every compression scheme has three stages:
  A. The input data is **transformed** to a new representation that is easier or more efficient to compress.
  B. We may introduce **loss** of information. Quantization is the main lossy step → we use a limited number of reconstruction levels, fewer than in the original signal.
  C. Entropy Coding. Assign a codeword (thus forming a binary bit-stream) to each output level or symbol. E.g. Huffman coding.

Differential Coding of Audio

- Audio is often stored not in simple PCM but instead in a form that exploits differences -- which are generally smaller numbers, so offer the possibility of using fewer bits to store.
  - If a has some consistency over time, the difference signal, subtracting the current sample from the previous one, will have a more peaked PDF with the peak being around zero.
  - For example, as an extreme case the PDF for a linear ramp signal that has constant slope is flat, whereas the PDF for the difference signal consists of a spike at the slope value.
  - So if we then go on to assign bit-string codewords to differences, we can assign short codes to prevalent values and long codewords to rarely occurring ones.

Lossless Predictive Coding (LPC)

- Predictive coding: transmitting differences – predict the next sample as being equal to the current sample; send not the sample itself but the difference between previous and next.
  - Predictive coding consists of finding differences, and transmitting these using a PCM system.
  - Note that differences of integers will be integers. Denote the integer input signal as f_n. Then we predict values $\hat{f}_n$ as simply the previous value, and define the error e_n as the difference between them:

$$\hat{f}_n = f_{n-1}$$

$$e_n = f_n - \hat{f}_n$$

LPC

- But it is often the case to use a function of a few of the previous values, f_{n-1}, f_{n-2}, f_{n-3}, etc., to provide a better prediction.
- Typically, a linear predictor function is used:

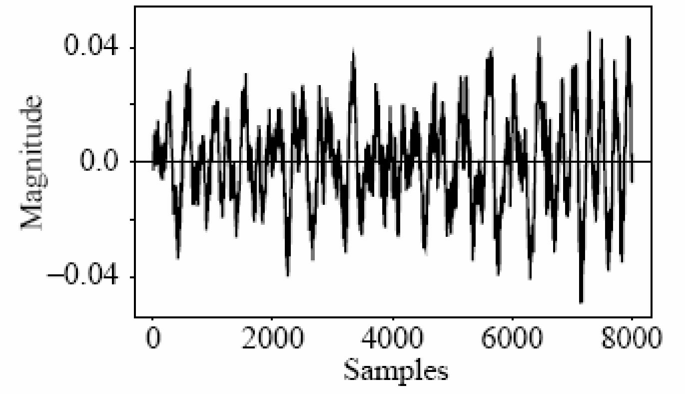
$$\hat{f}_n = \sum_{k=1}^{m} a_k f_{n-k}$$

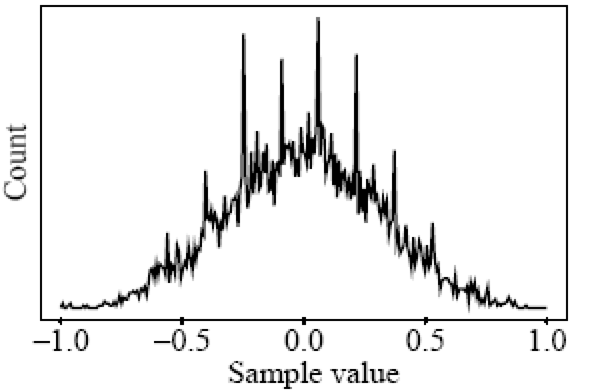
Usually *m* can be set between 2 to 4.

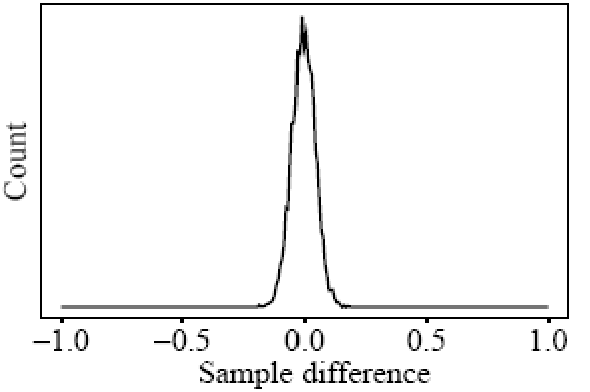
- The idea of forming differences is to make the PDF of sample values more peaked, i.e. to reduce the entropy of the sample signal.

PDFs of the sample signal and its difference

- The top figure plots 1 second of sampled speech at 8 kHz, with 8 bits per sample.
- The middle figure plots the PDF of these values.
- The bottom figure plots the PDF of the speech signal differences, whose values are much more clustered around zero than the sample values themselves.
- As a result, the entropy of the difference signal will be smaller than that of the original signal. We can use fewer bits to code the difference signal and obtain greater compression ratio.



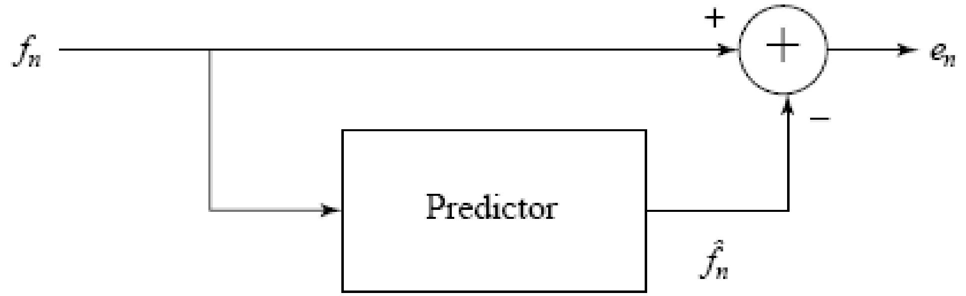




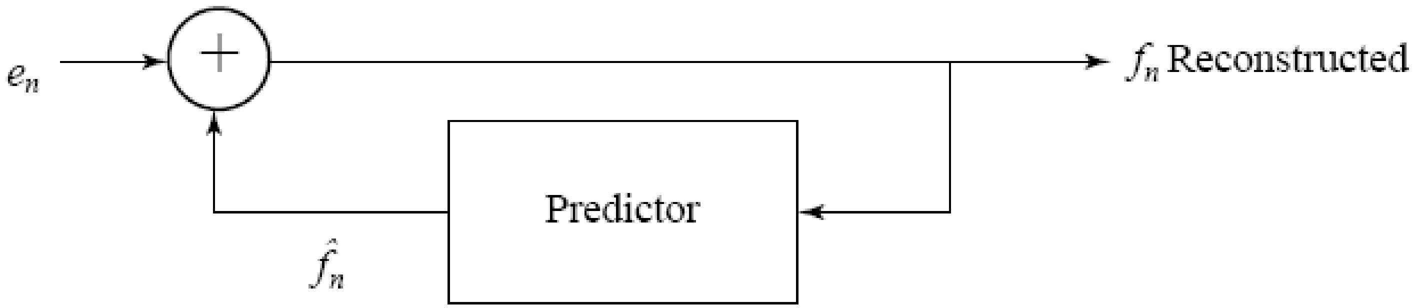
LPC

- Lossless predictive coding -- the decoder produces the same signals as the original.

Encoder



Decoder



An example

- As a simple example, suppose we devise a predictor as follows:

$$\hat{f}_n = \left\lfloor \tfrac{1}{2} f_{n-1} + \tfrac{1}{2} f_{n-2} \right\rfloor$$

$$e_n = f_n - \hat{f}_n$$

where symbol $\lfloor \bullet \rfloor$ means biggest integer less than $\bullet$, e.g. $\lfloor 1.8 \rfloor = 1$.

- Suppose we wish to code the sequence $[f_1, f_2, f_3, f_4, f_5] = [21, 22, 27, 25, 22]$. Since we use 2 past values in the predictor, we invent an extra signal value f_0, and let it equal to $f_1 = 21$. The initial value f_1 is transmitted without coding.

An example

$$\hat{f}_2 = \left\lfloor \tfrac{1}{2} f_1 + \tfrac{1}{2} f_0 \right\rfloor = 21; e_2 = 22 - 21 = 1$$

$$\hat{f}_3 = \left\lfloor \tfrac{1}{2} f_2 + \tfrac{1}{2} f_1 \right\rfloor = \lfloor 21.5 \rfloor = 21; e_3 = 27 - 21 = 6$$

$$\hat{f}_4 = \left\lfloor \tfrac{1}{2} f_3 + \tfrac{1}{2} f_2 \right\rfloor = \lfloor 24.5 \rfloor = 24; e_4 = 25 - 24 = 1$$

$$\hat{f}_5 = \left\lfloor \tfrac{1}{2} f_4 + \tfrac{1}{2} f_3 \right\rfloor = \lfloor 26 \rfloor = 26; e_5 = 22 - 26 = -4$$

- Now we only need to transmit $[f_1, e_2, e_3, e_4, e_5] = [21, 1, 6, 1, -4]$.
- Exercise
  - Please reconstruct the original signal from $[f_1, e_2, e_3, e_4, e_5]$.

Differential PCM (DPCM)

- DPCM is very similar to Predictive Coding, except that it incorporates a **quantization** step.
- Form the prediction, form an error; then quantize the error to a quantized version.

$$\hat{f}_n = function\_of(\tilde{f}_{n-1},, \tilde{f}_{n-2}, \tilde{f}_{n-3}, \ldots)$$

$$e_n = f_n - \hat{f}_n ,$$

$$\tilde{e}_n = Q[e_n] ,$$

$$\text{reconstruct: } \tilde{f}_n = \hat{f}_n + \tilde{e}_n$$

f_n – the original signal

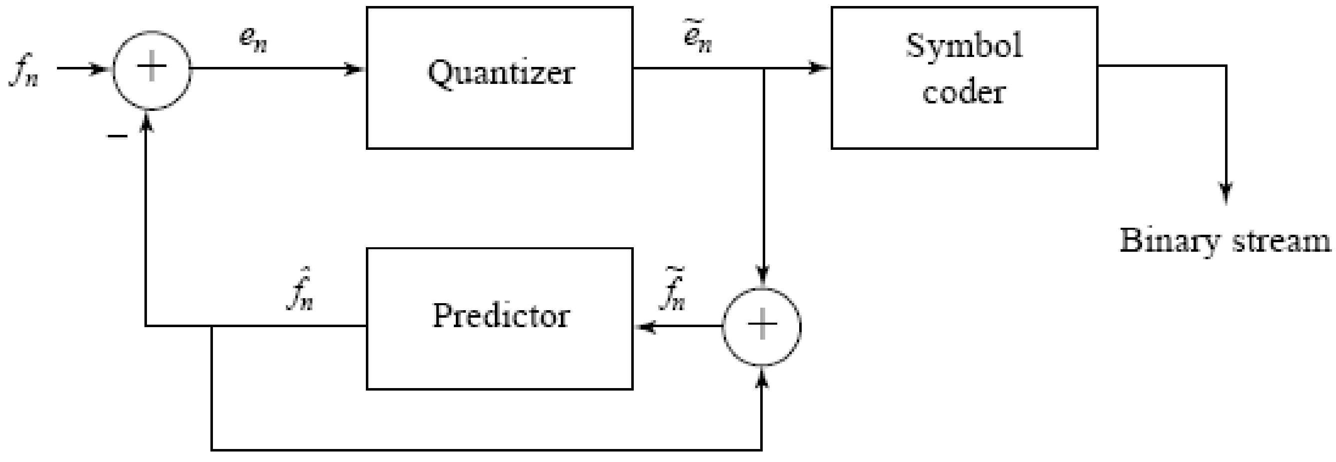
$\hat{f}_n$ – the predicated signal

$\tilde{f}_n$ – the reconstructed signal from the quantized error

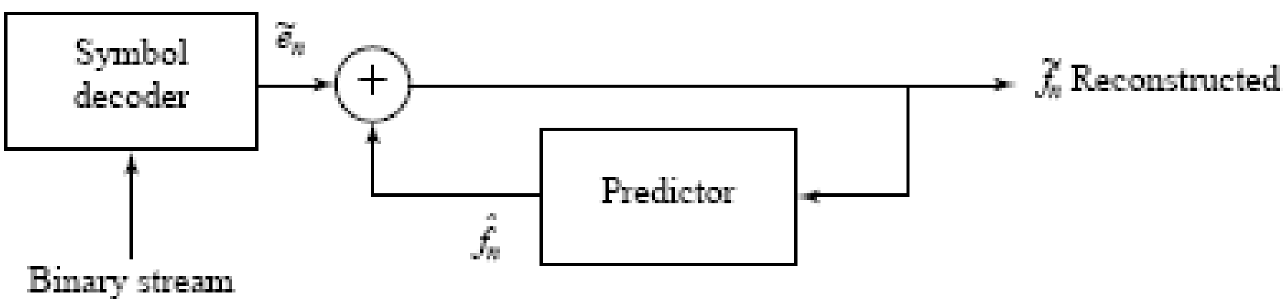
- We will code and transmit the quantized error $\tilde{e}_n$.

Diagram for DPCM encoder and decoder

Encoder



Decoder



Comments on DPCM

- Since the predicted error will be quantized, distortion (quantization noise) is introduced. Thus the reconstructed signal will not be equal to the original signal, i.e. DPCM is a lossy compression scheme.
- The degree of distortion depends on the used quantizer.
- One optimal non-uniform quantizer is the **Lloyd-Max** quantizer, which is based on a least-squares minimization of the error term.

Adaptive DPCM (ADPCM)

- ADPCM takes the idea of adapting the coder to suit the input signal.
- The key is how to determine the prediction coefficients a_k. Usually, for the input signal f_n we will adaptively find the predictor

$$\hat{f}_n = \sum_{k=1}^{M} a_k \tilde{f}_{n-k}$$

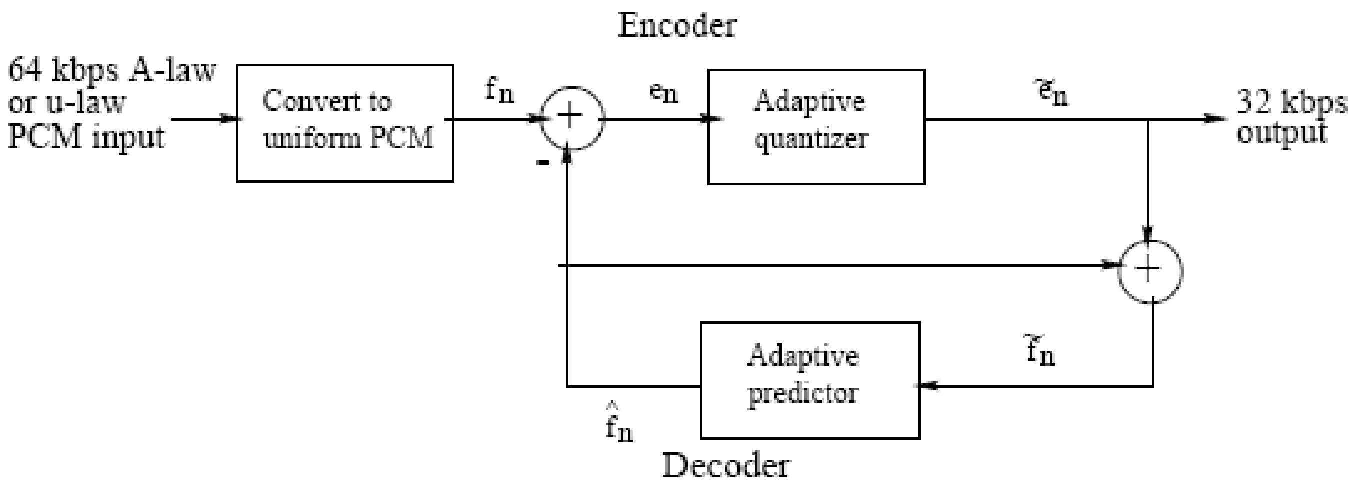
to minimize the error

$$\min \sum_{n=1}^{N} \left(f_n - \hat{f}_n \right)^2$$

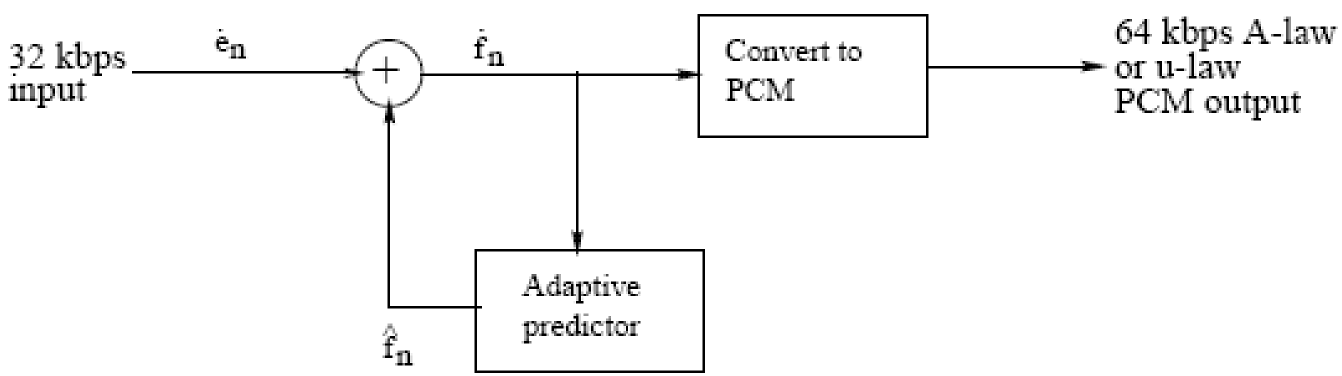
Note that the optimal prediction coefficients a_k depend on the input signal.

ADPCM encoder and decoder

Encoder



Decoder



Lab exercise

- In Matlab, you can use function "wavread" to read, use "wavplay" to play and use "wavwrite" to write a "*.wav" audio file. Please refer to the help file for more details.
- Write a Matlab program to read a "*.wav" sound file (e.g. a 10 second music or speech). Then re-sample it with a lower rate and/or quantize it to a lower precision. Play it to hear the output. Then you may try to code it by LPC or DPCM.

References

- Ze-Nian Li, M. S. Drew, *Fundamentals of Multimedia*, Prentice Hall Inc., 2004. Chapter 6.