Lecture 2

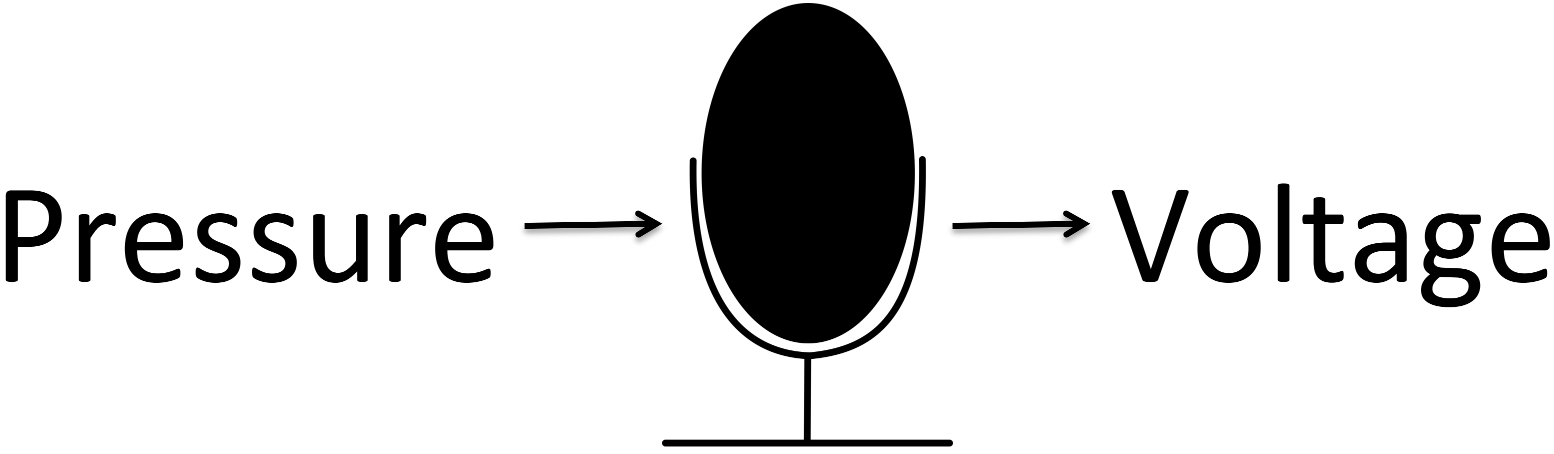
Audio Capture and Compression

Ref: Fundamentals of Multimedia

Audio is a wave phenomenon

- Speaker moves back and fourth and generates longitudinal pressure waves
- Ear perceives that as sound
- No air, no sound

Pressure to Voltage Conversion



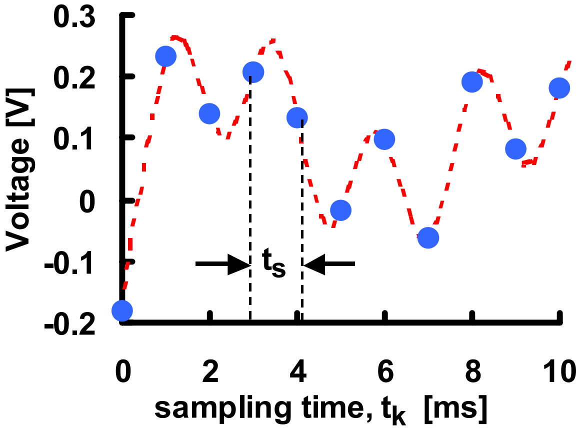
Pressure → Voltage

Analogue

Digital

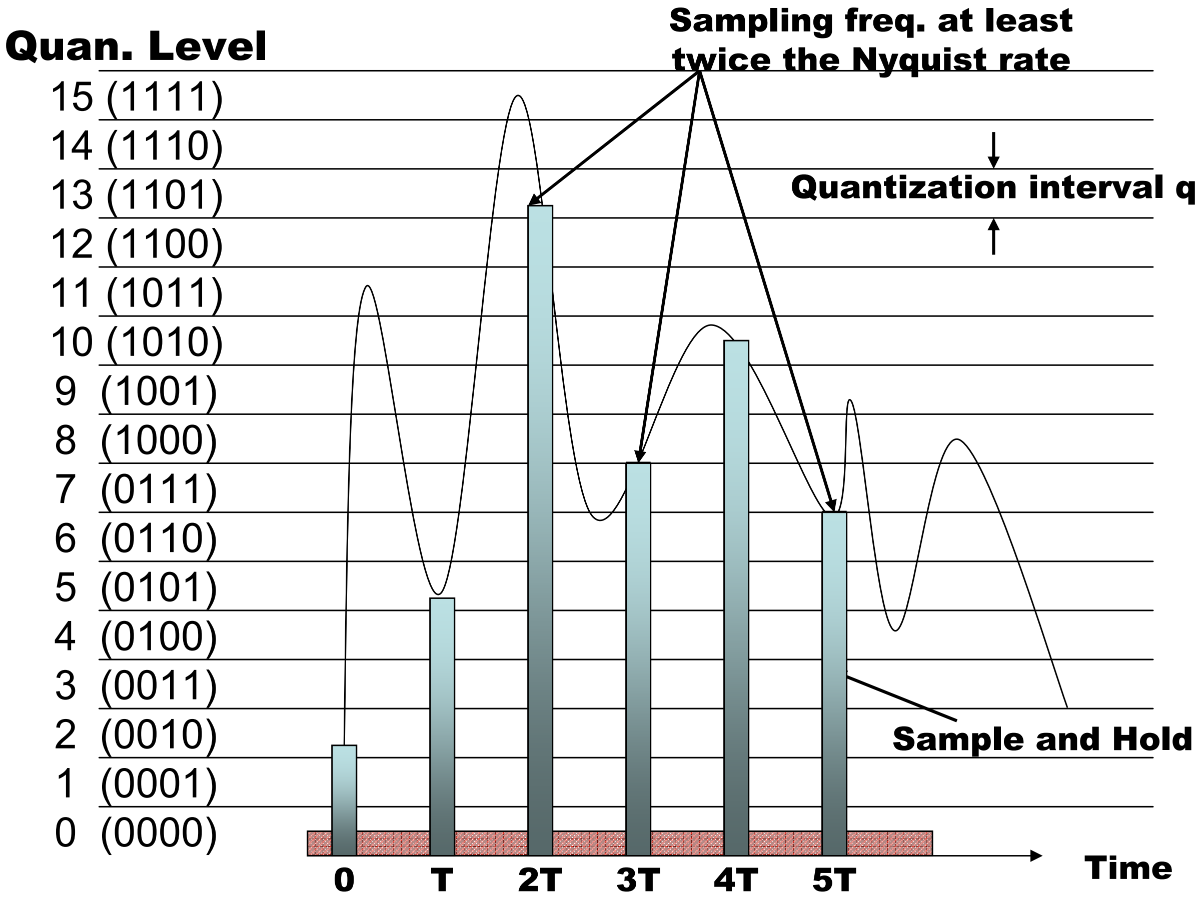
The image cannot be displayed. Your computer may not have enough memory to open the image, or the image may have been corrupted. Restart your computer, and then open the file again. If the red x still appears, you may have to delete the image and then insert it again.

→ ADC →



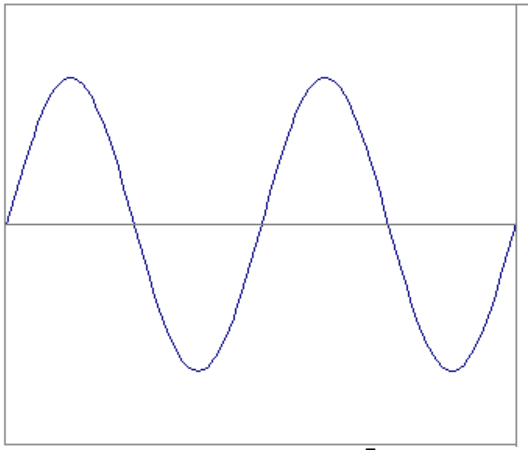
Discretization

- Time axis
  - sampling
- Amplitude axis
  - quantization

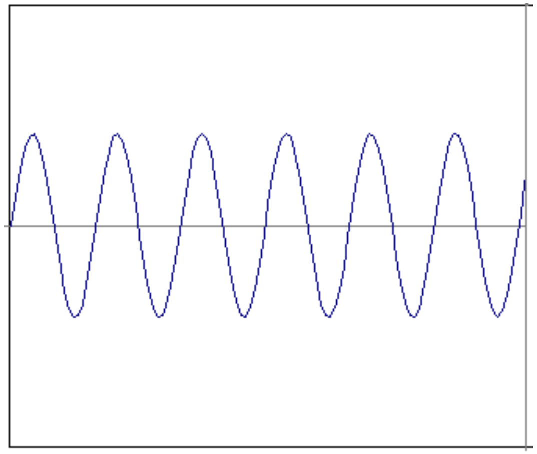


Ref: Dr. Wang Ye

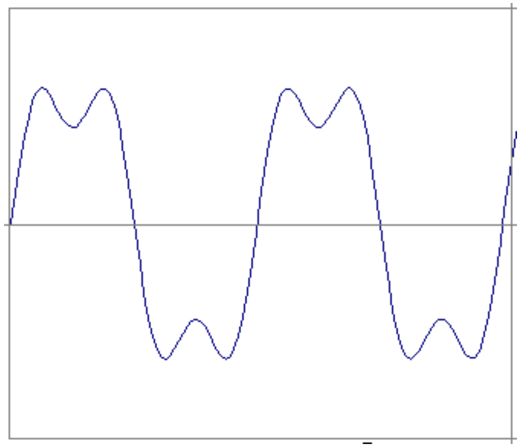
Frequency Components of Audio Signal

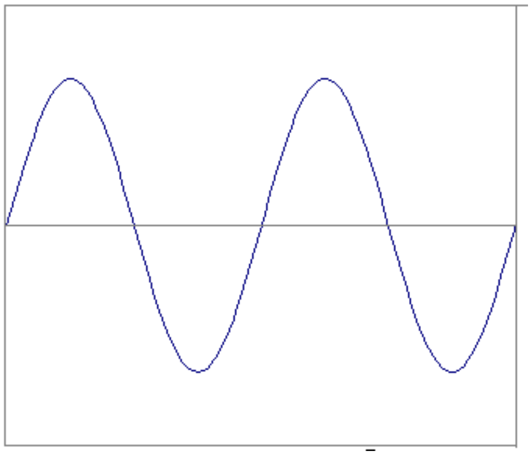


+

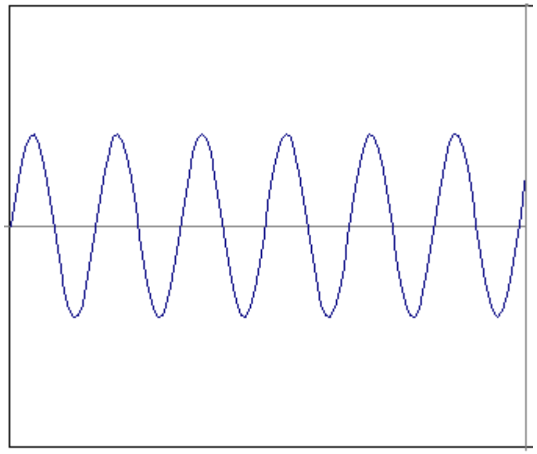


=

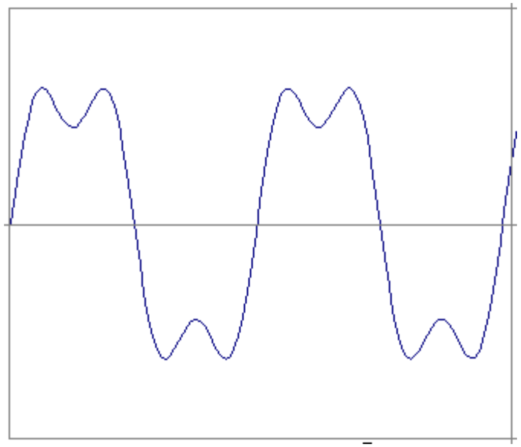


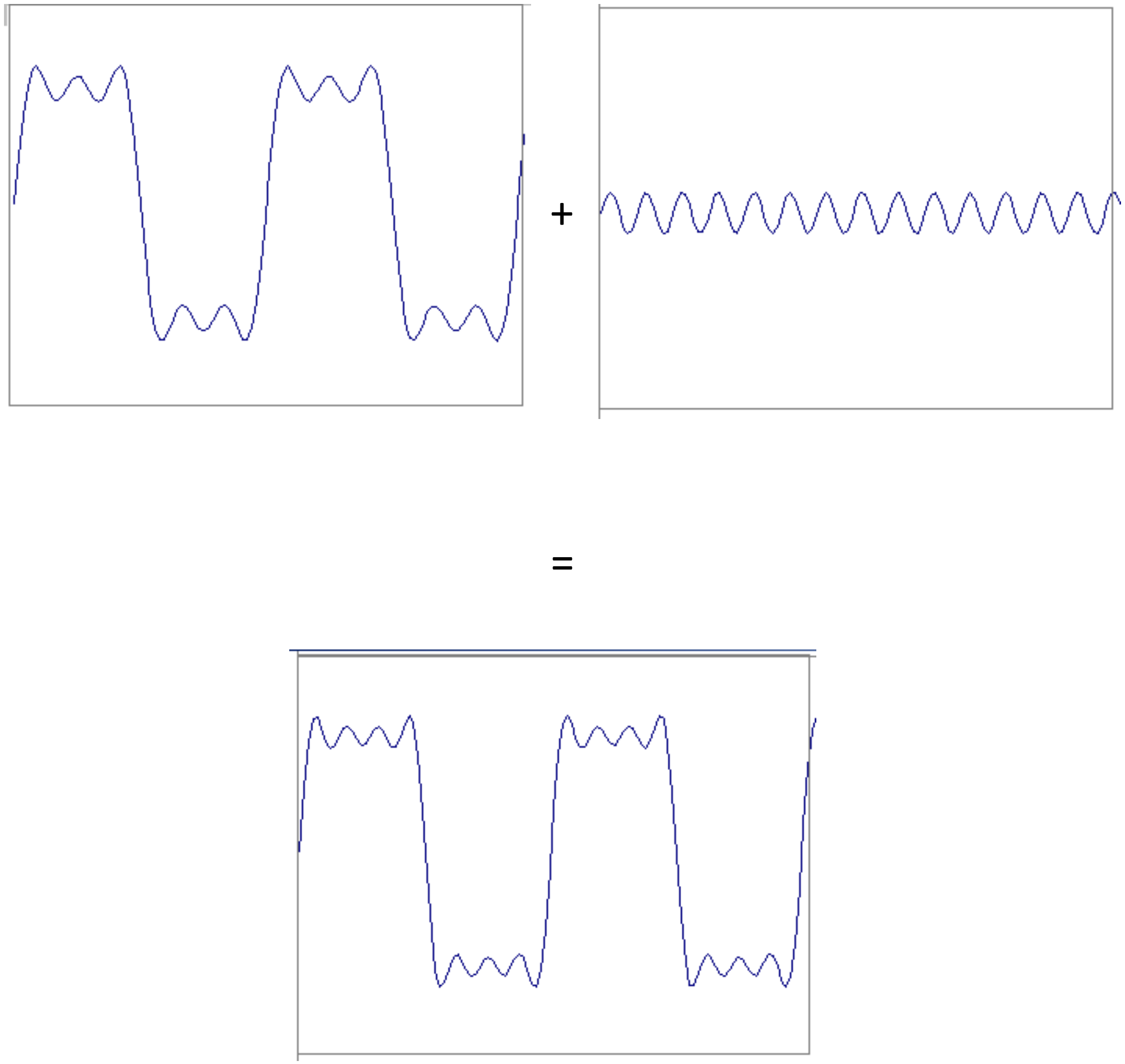


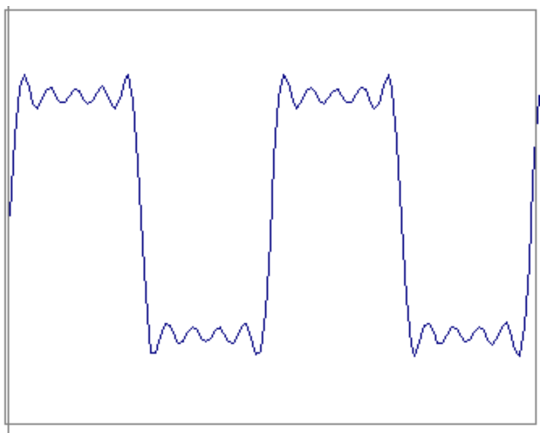
+



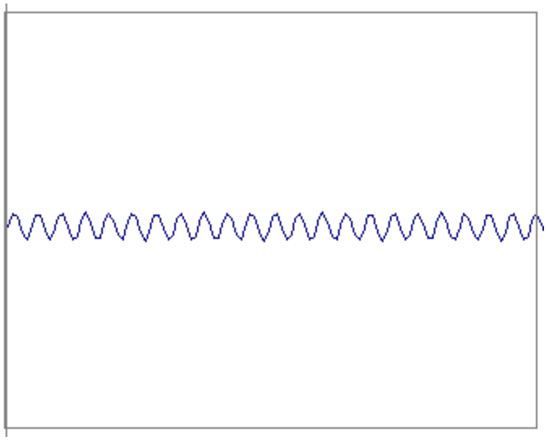
=



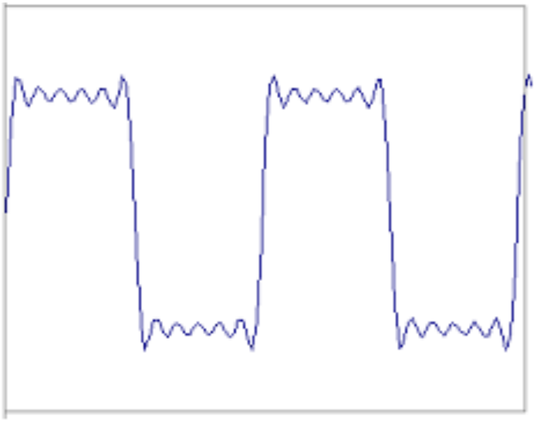


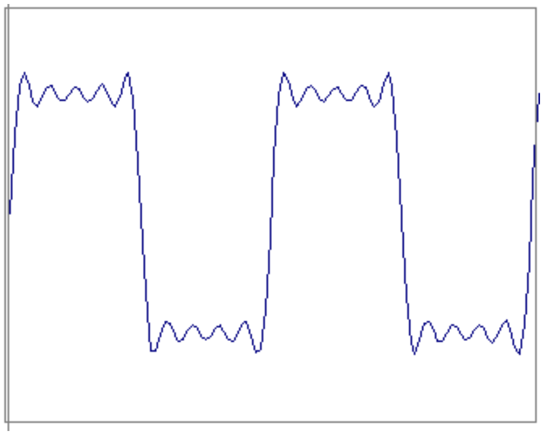


+

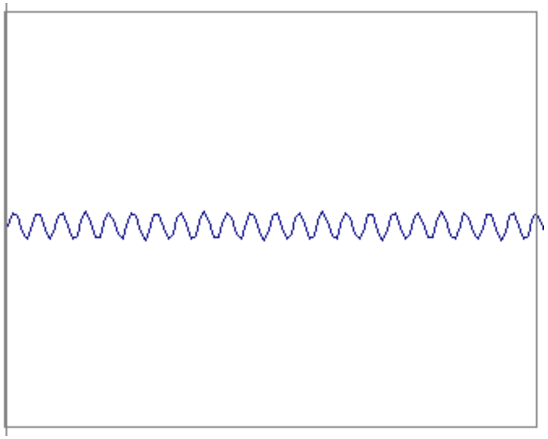


=

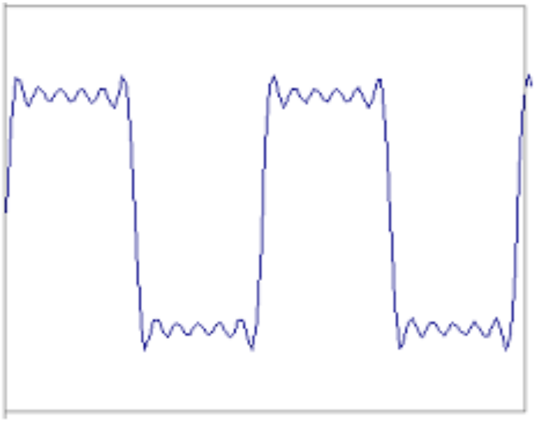


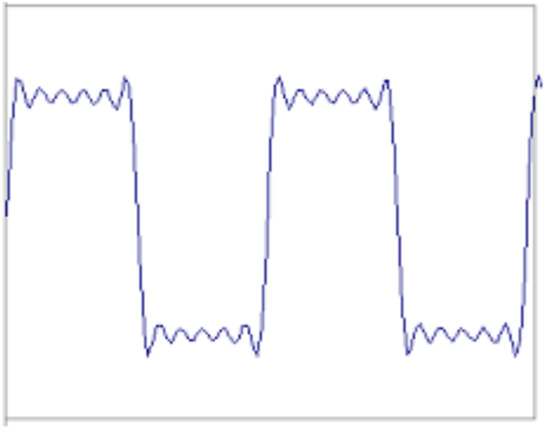


+

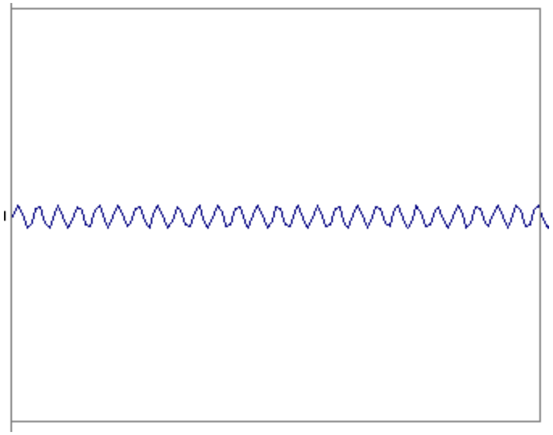


=

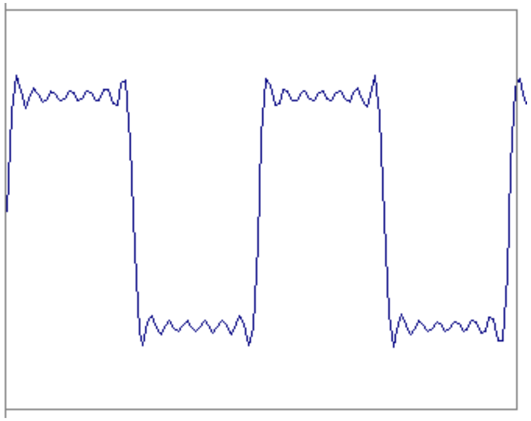


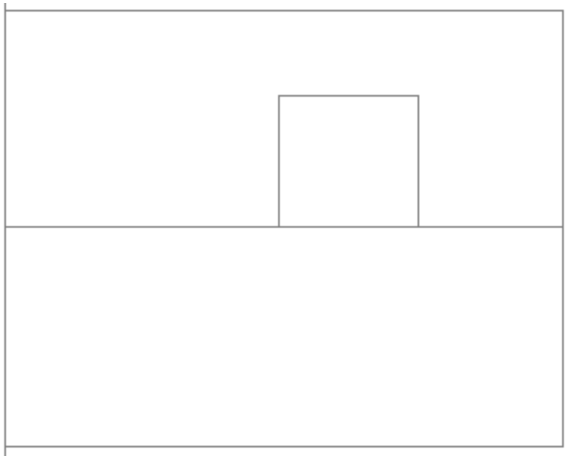


+

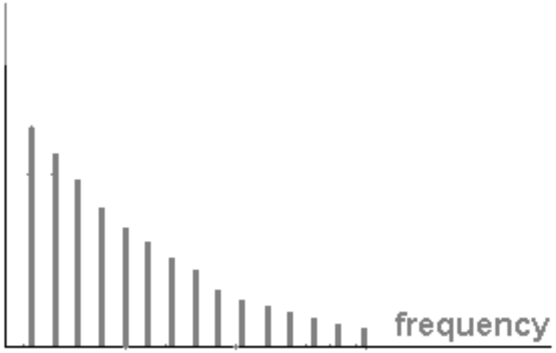


=

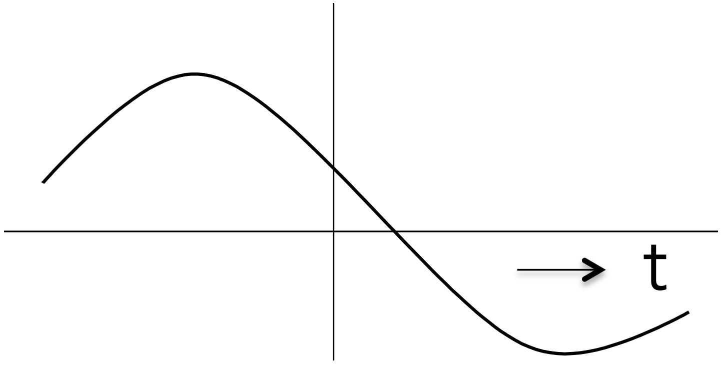


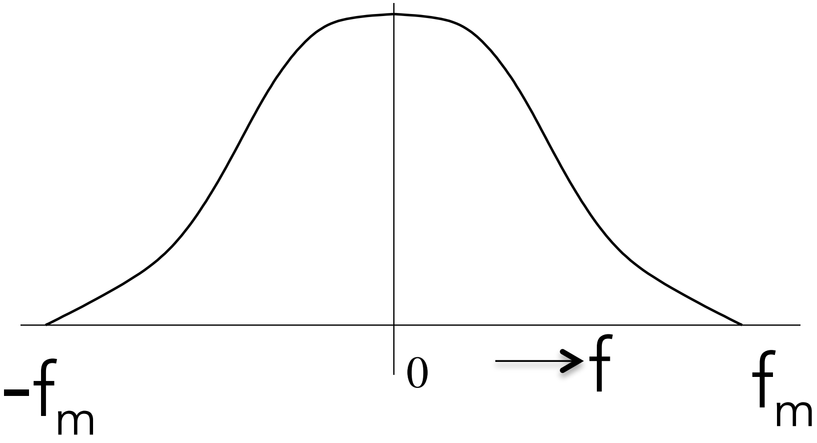


~


frequency

Time Domain Vs Frequency Domain





Sampling Theorem

A signal can be reconstructed exactly if it is sampled at a rate at least twice the maximum frequency component in it!

Nyquist rate = $2f_m$

What should be sampling rate for audio?

Audible Range: 20 Hz to 20 kHz

Sample (music) >40k samples

Voice Range: 0 to 4 kHz!

Sample(speech) > 8k samples

How do you ensure no aliasing?

Low pass filter the signal before passing to ADC!

Quantization

- Representing large set of values with a smaller number of values.
- The large set may have continuous values also.

Signal to Quantization Noise Ratio

SQNR = 6.02NDB

Exercise: Quantize the following 5 bit signals into 2 bit signals!

{23, 12, 9, 5, 31, 16, 19, 4, 13, 22}

- There are four code-words: 0, 1, 2, 3
- Obtain interval each code-word represents to obtain codes
- Obtain representative value corresponding to each code-word to decode
- Put all this information in a table.

Linear Vs Non-linear Quantization

- Linear Quantization: equal step
- Non-linear Quantization: unequal steps

Pulse Code Modulation

- Each audio sample is represented by an integer code-word.
- Linear PCM uses linear quantization and non-linear PCM uses non-linear quantization
- Non-linear PCM is also called companding

Weber's Law

$$\Delta\text{Response} \propto \frac{\Delta\text{Stimulus}}{\text{Stimulus}}$$

🔊 100

🔊 200

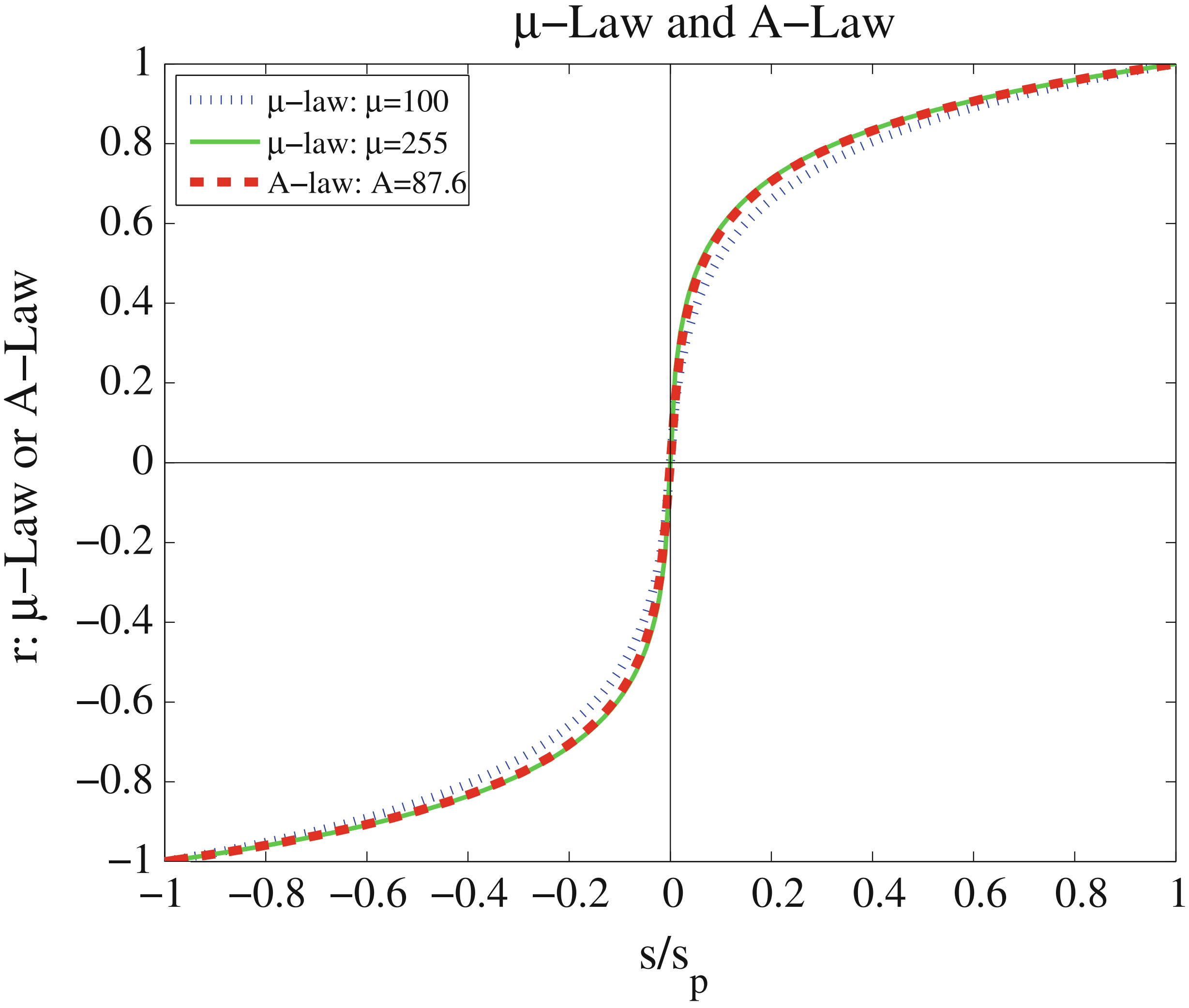
🔊 500

🔊 600

🔊 1200

🔊 1300

Companding


μ–Law and A–Law
r: μ–Law or A–Law
s/s_p
μ–law: μ=100
μ–law: μ=255
A–law: A=87.6
1
0.8
0.6
0.4
0.2
0
−0.2
−0.4
−0.6
−0.8
−1
−1
−0.8
−0.6
−0.4
−0.2
0
0.2
0.4
0.6
0.8
1

μ-law Companding

$$r = \frac{\text{sign}(s)}{\ln(1+\mu)} \ln \left\{ 1 + \mu \left| \frac{s}{s_\text{p}} \right| \right\}$$

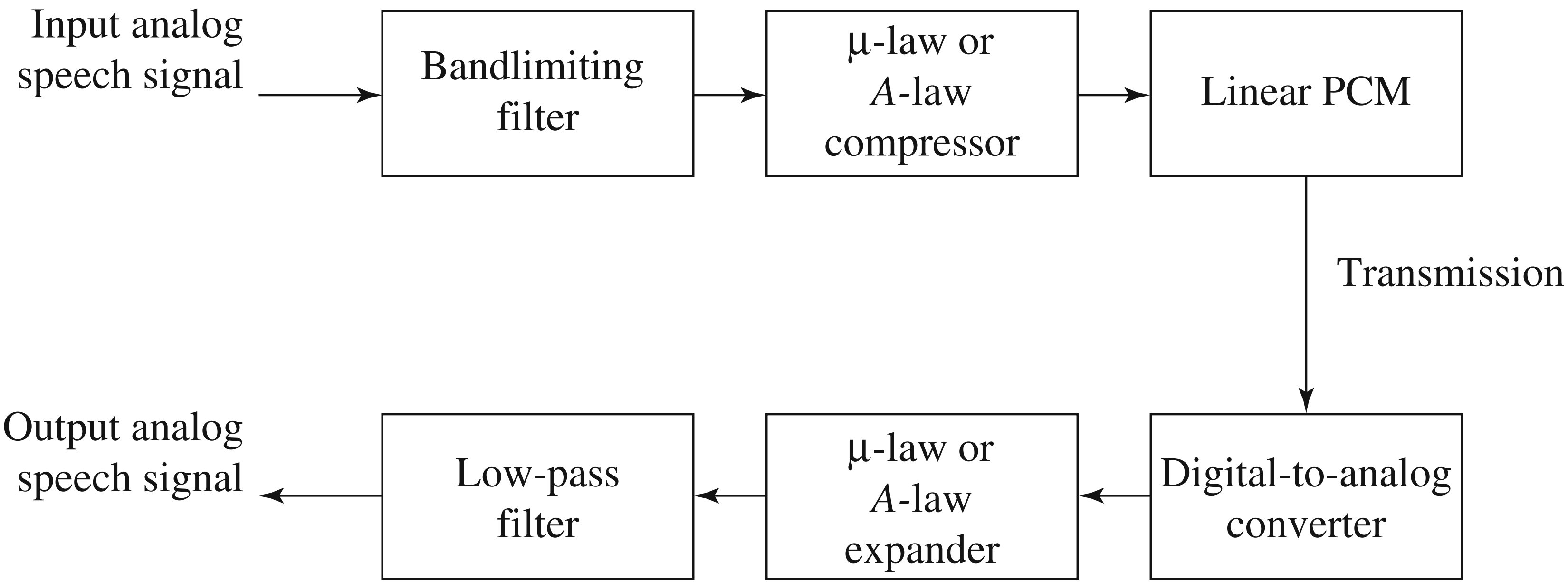
A-law Companding!

$$r = \begin{cases} \frac{A}{1+\ln A} \left(\frac{s}{s_\text{p}} \right), & \left| \frac{s}{s_\text{p}} \right| \leq \frac{1}{A} \\ \frac{\text{sign}(s)}{1+\ln A} \left[1 + \ln A \left| \frac{s}{s_\text{p}} \right| \right], & \frac{1}{A} \leq \left| \frac{s}{s_\text{p}} \right| \leq 1 \end{cases}$$

$$\text{where } \ \text{sign}(s) = \begin{cases} 1 & \text{if } \ s > 0, \\ -1 & \text{otherwise} \end{cases}$$

How to Implement Companding?

1. Use non-uniform quantization steps in the ADC internally
2. Use additional nonlinear analog circuit before linear ADC
3. Use a 12 bit ADC and a software lookup table to get the codes


Input analog speech signal
Bandlimiting filter
μ-law or *A*-law compressor
Linear PCM
Transmission
Digital-to-analog converter
μ-law or *A*-law expander
Low-pass filter
Output analog speech signal

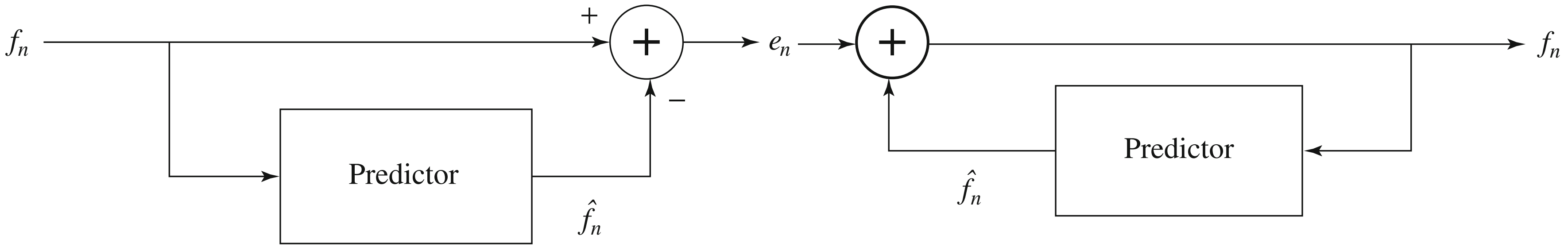
Companding in mainly used in Telephony!

Quality	Sampling rate (kHz)	Bits per sample	Mono/ Stereo	Bitrate (if uncompressed) (kB/s)	Signal bandwidth (Hz)
Telephone	8	8	Mono	8	200–3,400
AM radio	11.025	8	Mono	11.0	100–5,500
FM radio	22.05	16	Stereo	88.2	20–11,000
CD	44.1	16	Stereo	176.4	5–20,000
DVD audio	192 (max)	24 (max)	Up to 6 channels	1,200.0 (max)	0–96,000 (max)

Differential Coding

- In real life, things vary smoothly, hence the differences are small.
- Encode the current sample based on the value of previously encoded sample!

Lossless Predictive Coding



What is the goal of predictor?

Based on the current and past samples, predict a value as close to the next sample value as possible!

Would taking difference always compress the signal?

Case 1: differences are limited to a small range

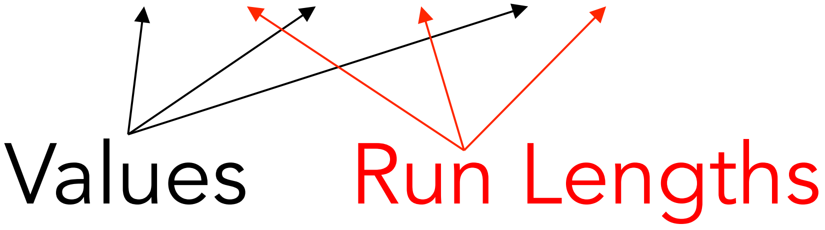
- E.g. -14, -12, -10, -5, 4, 11, 15, -3
- Range is -16 to + 15
- Need 5 bits to encode each sample

Case 2: differences consists of runs of numbers

- This will happen when input mostly varies linearly
  - e.g. 1,1,1,0,0,-1,-1,-1,-1
- Count the run length, and just send the count for each symbol
  - 1(3),0(2),-1(4)

Run Length Encoding

1110011111 → 1 3 0 2 1 5



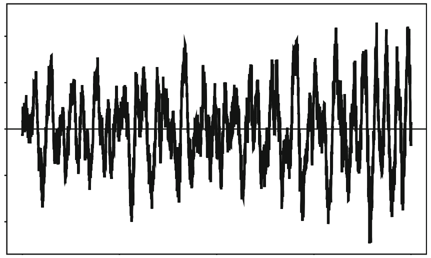
Values

Run Lengths

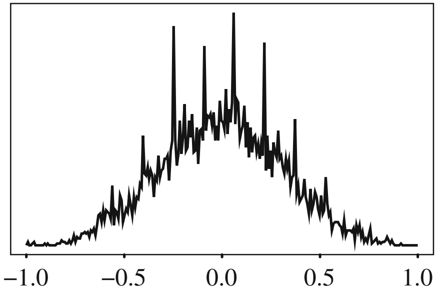
Case 3: differences are not limited to a small range

- The maximum possible range is -512 to +512
- It is actually double of the original signal range
- In this case, observe the histogram of the differences

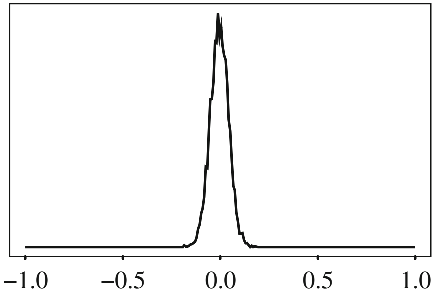
Histogram of differences is more peaked!



Input Signal



Sample values



Sample differences

Assign short codes to prevalent values!

For exceptionally large differences, use SU (+32) and SD (-32) symbols!

Huffman Codes for the Following Source

A:(15), B:(7), C:(6), D:(6), E:(5)

Huffman coding is also called

- Entropy coding
- Variable length coding
- Prefix coding
- Source coding

Limitation: Bits per symbol can never be less than entropy of the source!

This is also limit of any lossless compression method!

Let us represent the differences in a fewer number of levels!

Quantization!

Example

For 8 bit integer values, error is in the range -255 to 255

- 511 levels
- quantization step =16
- code-words = 32
- representative value is midway of step
- the lookup table has steps, code (0 to 31), and quantized error

How to choose quantization step?

- Bigger step needs less bits but produces more distortion
- Smaller step will increase the number of bits and reduce compression
- Optimal solution will need non-linear quantization

Goal: for a fixed number of levels, design best non-linear quantization scheme!

Minimize Quantization Error on a Patch

$$\min \sum_{n=i}^{i+N-1} (f_{\mathrm{n}} - Q[f_{\mathrm{n}}])^2$$

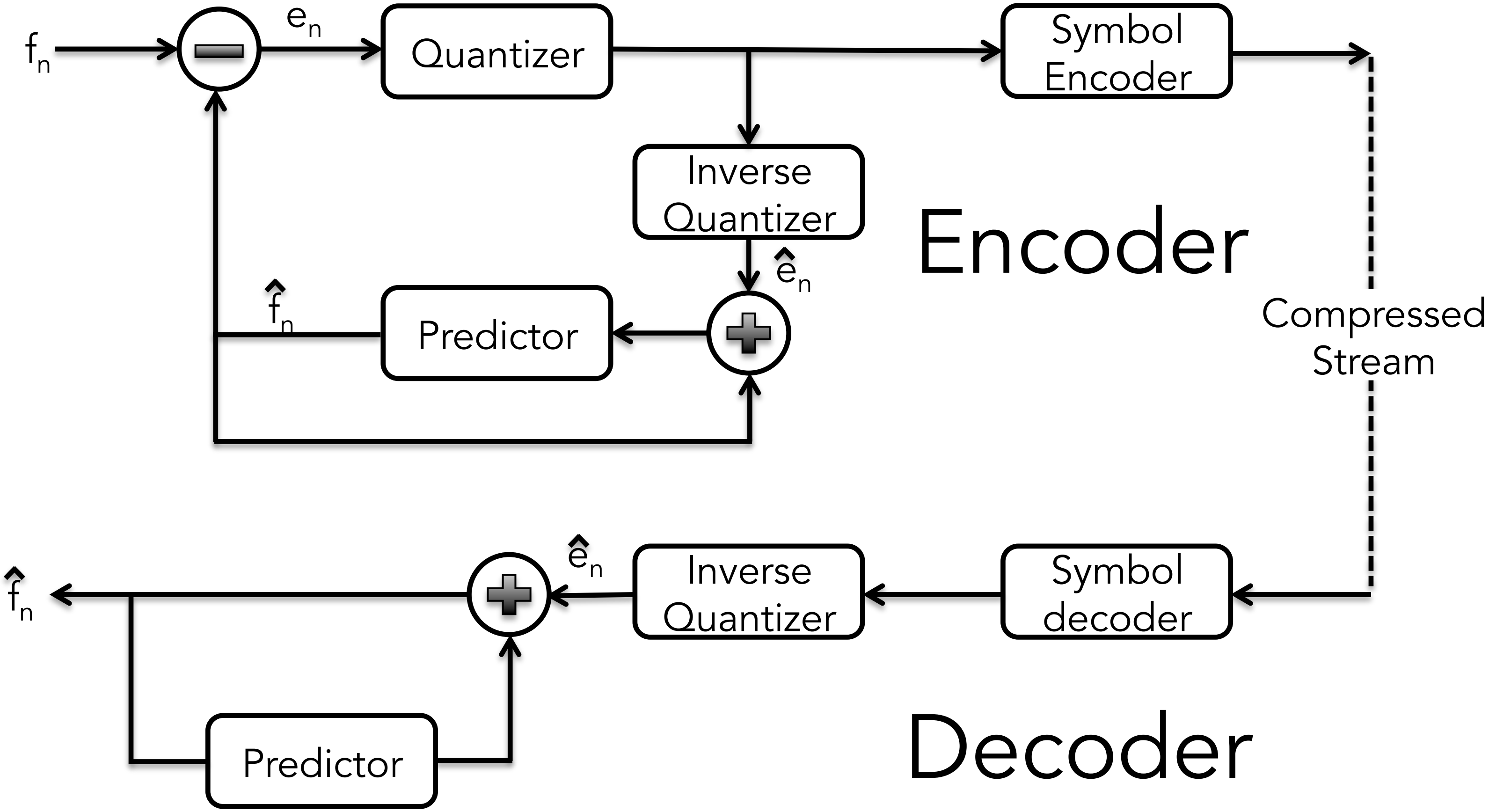
Symbol (Sample Value) Distribution

- Gaussian
- Laplacian
- Non-parameterized

Lioyd-Max Quantizer

$$\min \sum_{n=i}^{i+N-1} (d_{\mathrm{n}} - Q[d_{\mathrm{n}}])^2\, l(d_{\mathrm{n}})$$

Differential PCM (DPCM)



Example

$$\begin{array}{ccccc} f_1 & f_2 & f_3 & f_4 & f_5 \\ 130 & 150 & 140 & 200 & 230 \end{array}$$

Input Frames

$$\hat{f} = \boxed{130}, \ 130, \ 142, \ 144, \ 167$$

$$e = \boxed{0}, \ 20, \ -2, \ 56, \ 63$$

$$\tilde{e} = \boxed{0}, \ 24, \ -8, \ 56, \ 56$$

$$\tilde{f} = \boxed{130}, \ 154, \ 134, \ 200, \ 223$$

Initialization