BENC 4173 Multimedia Technology & Application

Chapter 4: Digital Audio

Low Yin Fen

BSc (UTM), MEngSc (MMU), Dr.rer.med (UdS, Germany)

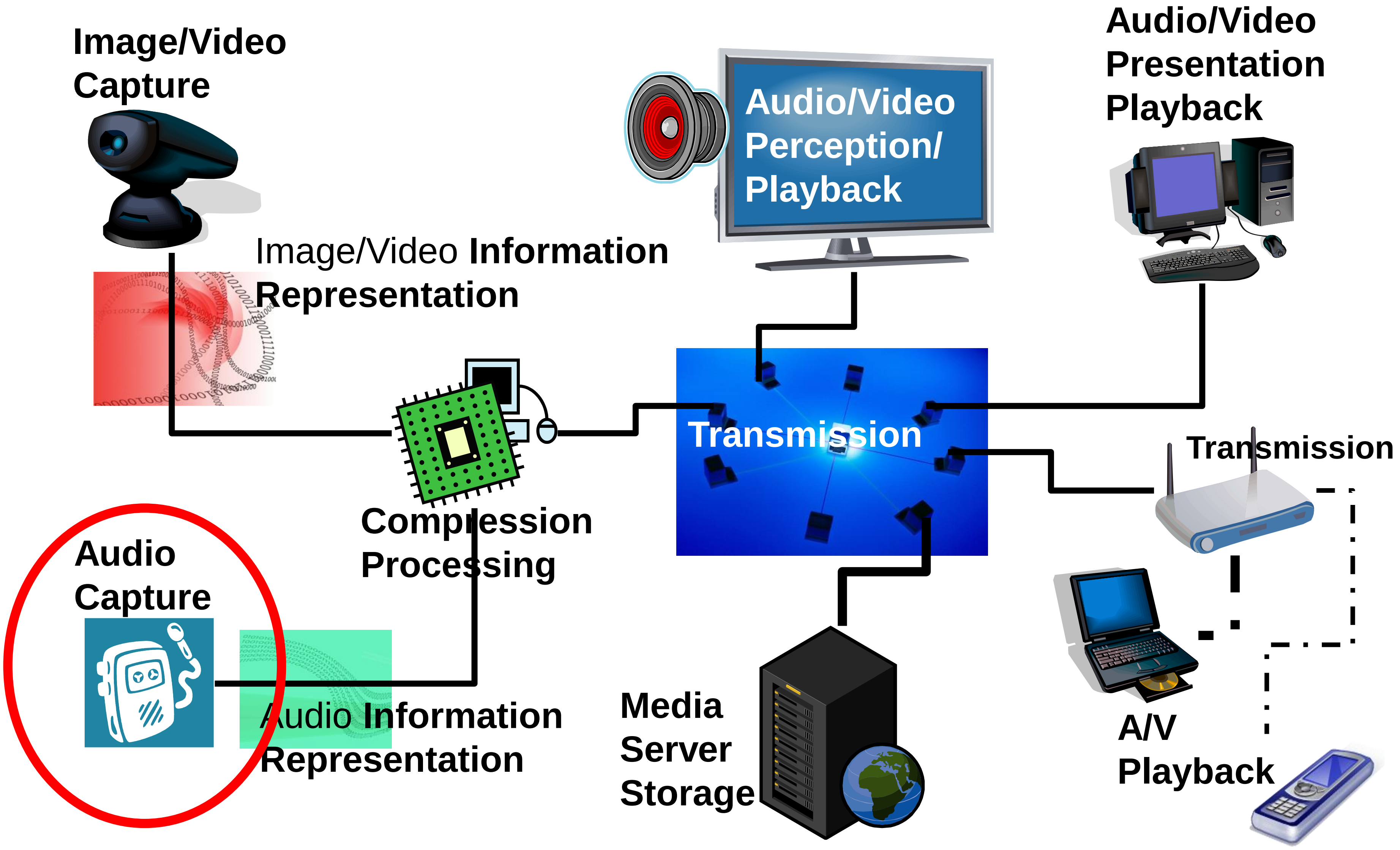
yinfen@utem.edu.my

Room: A3-34 (06-5552139)

Updated on March 2017

- Fundamentals of Digital Audio
  - The nature of sound waves
  - Amplitude measures of sound waves
  - digitizing sound
  - dynamic range, file size and file types
- Audio dithering & Noise shaping
- Non-linear Quantization
- Statistical analysis on an audio file
- Digital audio compression: MPEG Audio
- Surround sound standard: Dolby Digital

Introduction



Image/Video Capture

Image/Video **Information Representation**

Audio/Video Perception/ Playback

Audio/Video Presentation Playback

Compression Processing

Transmission

Transmission

Audio Capture

Audio **Information Representation**

Media Server Storage

A/V Playback

UNIVERSITI TEKNIKAL MALAYSIA MELAKA

Introduction

- digitally record and edit music
- combining instruments and voices
- edit the sound track for a digital video
- doing game programming and want to create sound effects, voice, and musical accompaniment for the game
- designing the sound to be used in the performance of a play or dance in the theatre

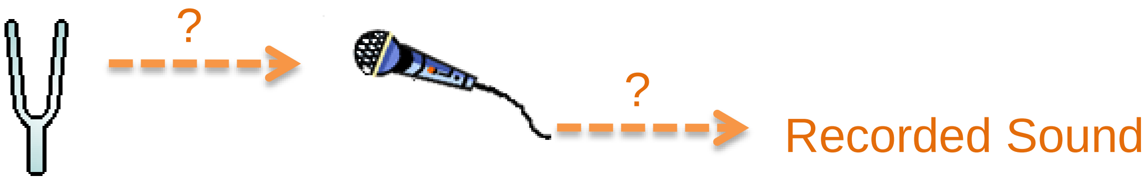
recording sound, choosing the appropriate sampling rate and sample size for a recording, knowing the microphones to use for your conditions, choosing a sound card and editing software for recording and editing, taking out the imperfections in recorded audio, processing with special effects, compressing, and selecting the right file type for storage.

Sound

- A wave that is generated by vibrating objects in a medium such as air

- Examples of vibrating objects:
  - vocal cords of a person
  - guitar strings
  - tunning fork

So how is vibration turned into sound we can hear or record with a microphone?

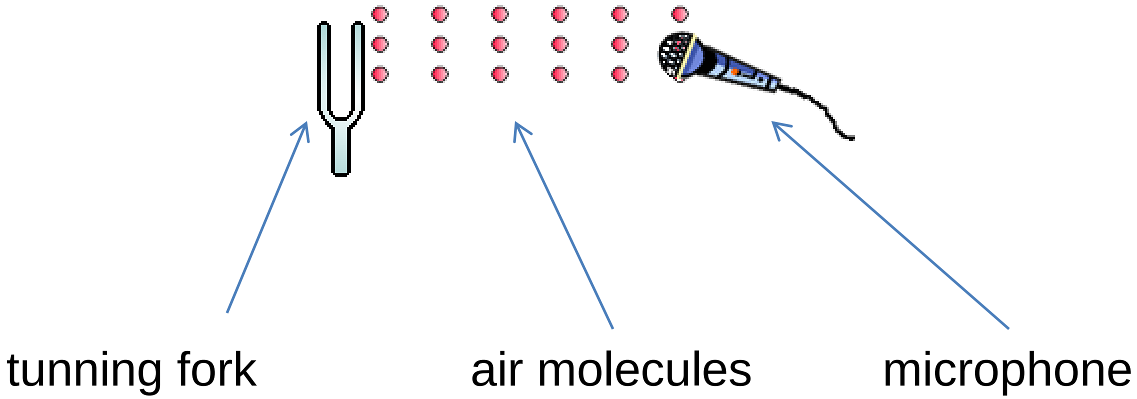


?

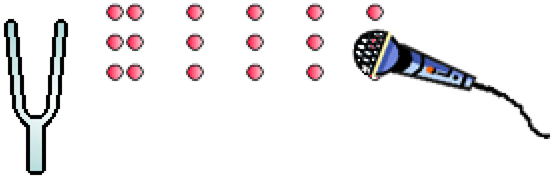
?

Recorded Sound

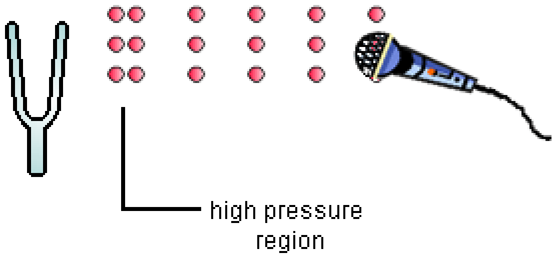
An illustration of how the propagating sound wave formed by changes of the air pressure reaches the microphone



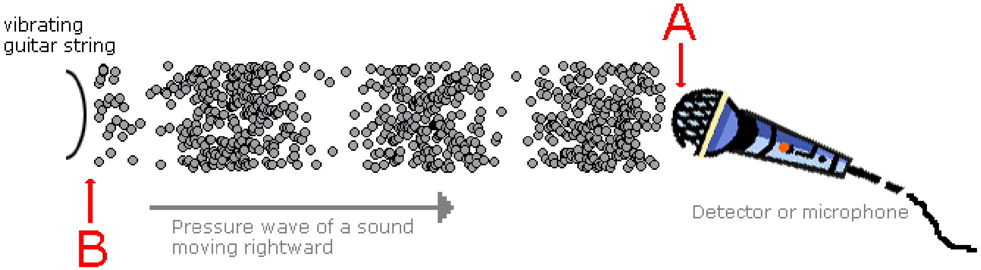
An illustration of how the propagating sound wave formed by changes of the air pressure reaches the microphone



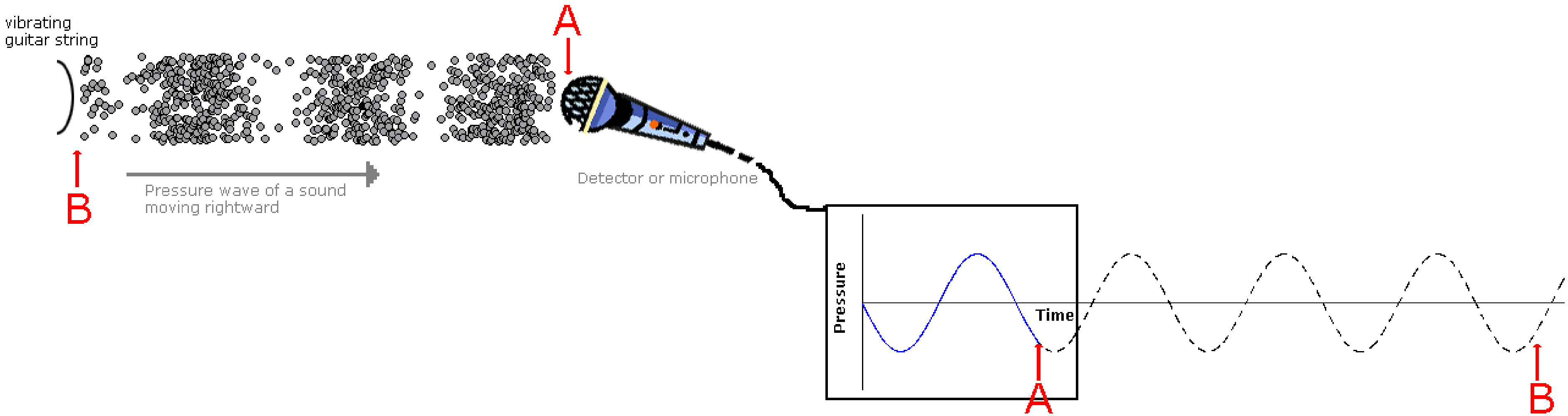
Let's step through the process slowly



The changes of pressure in the propagating sound wave reaching the recorder are captured as changes of electrical signals over time.



The sound wave can be represented graphically with the changes in air pressure or electrical signals plotted over time—a *waveform*.



Be careful…

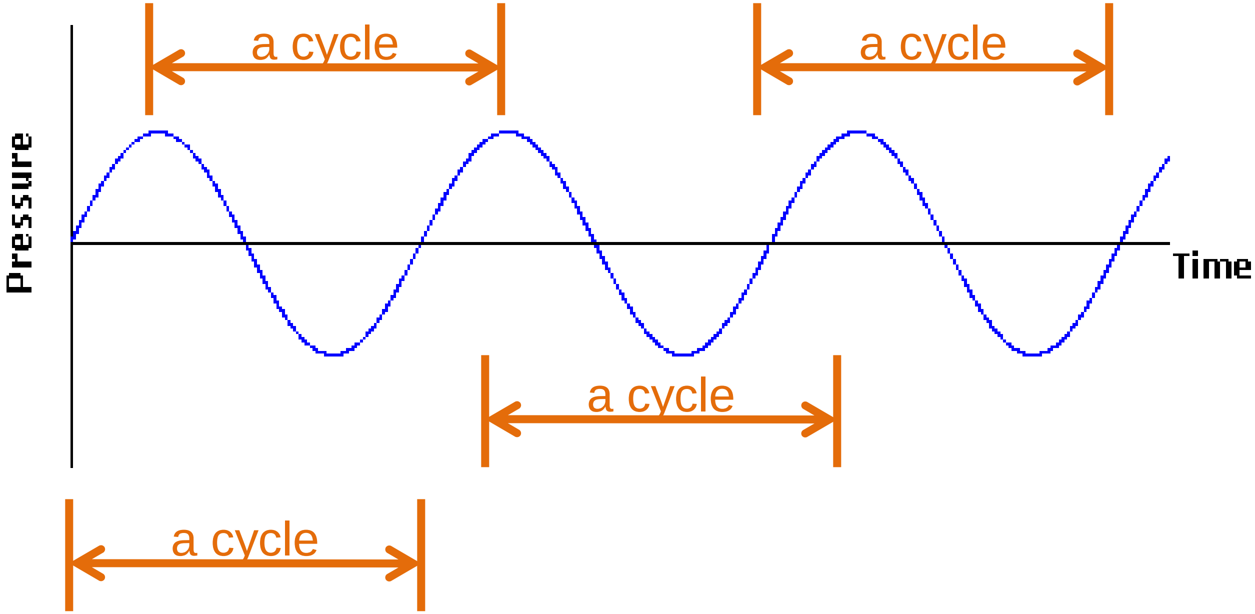
- NOT to interpret sound as a wave that has crests and troughs
  - – a longitudinal wave – air molecules are not going up and down, but **back and forth**, in the direction of the wave propagation

- NOT to interpret the waveform as a representation of the sound wave in space
  - – instead, the waveform graph represents the **pressure changes over time**

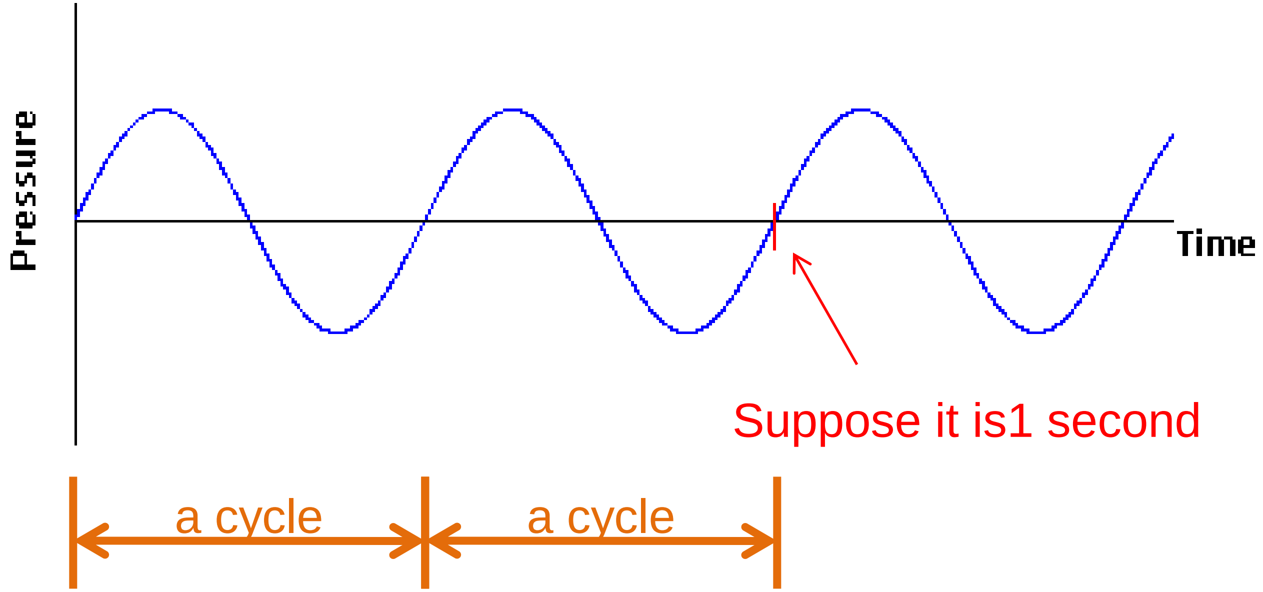
Frequency of Sound Wave

- Refers to the number of complete back-and-forth cycles of vibrational motion of the medium particles per unit of time
- Unit for frequency: Hz (Hertz)
- 1 Hz = 1 cycle/second

A Cycle



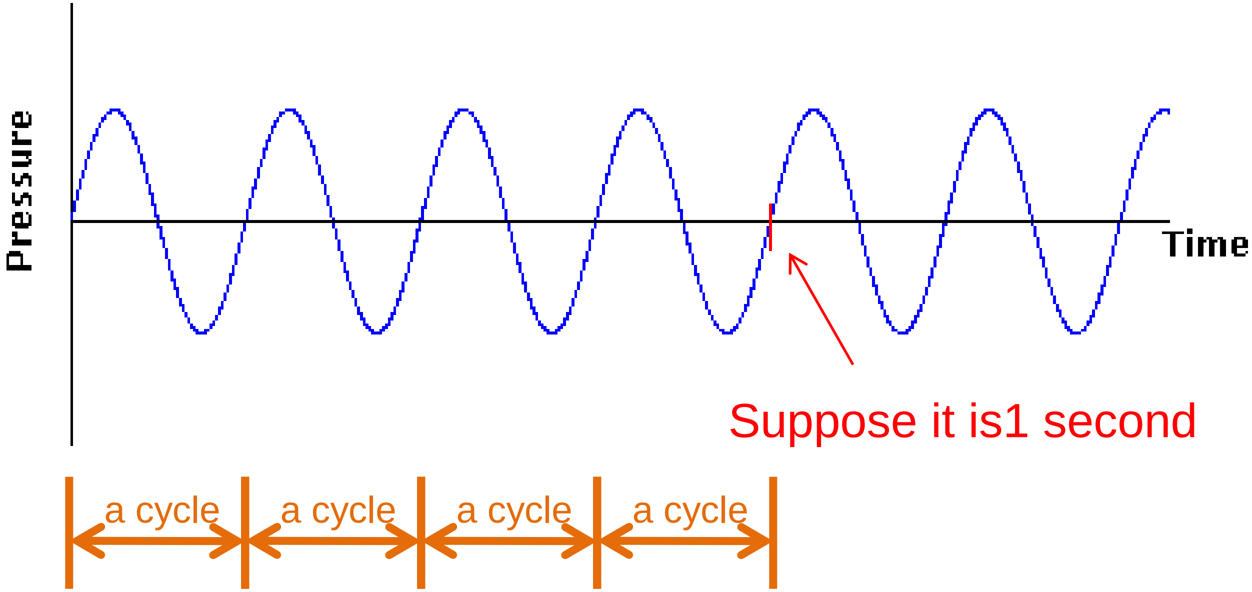
Frequency



Suppose it is1 second

Frequency = 2 Hz (i.e., 2 cycles/second)

Frequency



Frequency = 4 Hz (i.e., 4 cycles/second)
Higher frequency than the previous waveform.

Pitch of Sound

- Sound frequency
- Higher frequency: higher pitch
- human ear can hear sound ranging from 20 Hz to 20,000 Hz

Amplitude measures of sound waves

Amplitude measures the intensity of the sound and is related to its perceived loudness. It can be measured with a variety of units, including voltages, Newtons/m^2, or in the unitless measure called decibels.

- Atmospheric pressure is customarily measured in pascals (newtons/meter^2) – Pa or N/m^2
- The average atmospheric pressure at sea level is approximately 10^5Pa.
- For sound waves, ***air pressure amplitude*** is defined as the average deviation from normal background atmospheric air pressure.
- For example, the threshold of human hearing (for a 1000 Hz sound wave) varies from the normal background atmospheric air pressure by $2*10^{-5}$Pa, so this is its pressure amplitude.

- A decibel is always based upon some agreed-upon reference point, and the reference point varies according to the phenomenon being measured.
- For sound, the reference point is the air pressure amplitude for the threshold of hearing.
- A decibel is based on a ratio of a louder to a softer one.
- A decibel in the context of sound pressure level is called ***decibels-sound-pressure-level*** (***dB_SPL***).

Let E be the pressure amplitude of the sound being measured and E_0 be the sound pressure level of the threshold of hearing. Then ***decibels-sound-pressure-level***, (***dB_SPL***) is defined as

$$dB\_SPL = 20\log_{10}\left(\frac{E}{E_0}\right)$$

What would be the amplitude of the audio threshold of pain, given as 30 Pa?

$$dB\_SPL = 20\log_{10}\left(\frac{30N/m^2}{0.00002N/m^2}\right) = 20\log_{10}(1500000) = 20 * 6.17 \approx 123$$

Note: The threshold of pain varies with frequency and with individual perception.

What would be the pressure amplitude of normal conversation, given as 60 dB?

$$60 = 20\log_{10}\left(\frac{x}{0.00002\,N/m^2}\right)$$

$$60 = 20\log_{10}(50000x\frac{m^2}{N})$$

$$3 = \log_{10}(50000x\frac{m^2}{N})$$

$$10^3 = 50000x\frac{m^2}{N}$$

$$\frac{1000}{50000}\frac{N}{m^2} = x$$

$$x = 0.02\frac{N}{m^2}$$

Decibels-sound-pressure-level are an appropriate unit for measuring sound because the values increase logarithmically rather than linearly.

For example, a voice at normal conversation level could be 100 times the air pressure amplitude of a soft whisper, but to human perception it seems only about 16 times louder.

Application of Decibels

- Many audio-editing programs use decibels for the audio amplitude

- 3 decibels: doubling the sound intensity

- 6 decibels: doubling the electrical voltages corresonding to the sound

- Let's see why 3 and 6 decibels.

Decibels

$$\text{Number of decibels} = 10 \times \log_{10}\left(\frac{I}{I_o}\right)$$

$$= 20 \times \log_{10}\left(\frac{E}{E_o}\right)$$

I and I_o = sound intensity values in comparison
E and E_o= corresponding electrical voltages

Decibels when doubling the sound intensity

$$\begin{aligned}\text{Number of decibels} &= 10 \times \log\left(\frac{I}{I_o}\right) \\ &= 10 \times \log(2) \\ &\cong 10 \times 0.3 \\ &= 3\end{aligned}$$

Decibels when doubling the electrical voltages

$$\begin{aligned}\text{Number of decibels} &= 20\times\log\left(\frac{E}{E_o}\right)\\ &= 20\times\log(2)\\ &\cong 20\times 0.3\\ &= 6\end{aligned}$$

Sound Intensity vs. Loudness

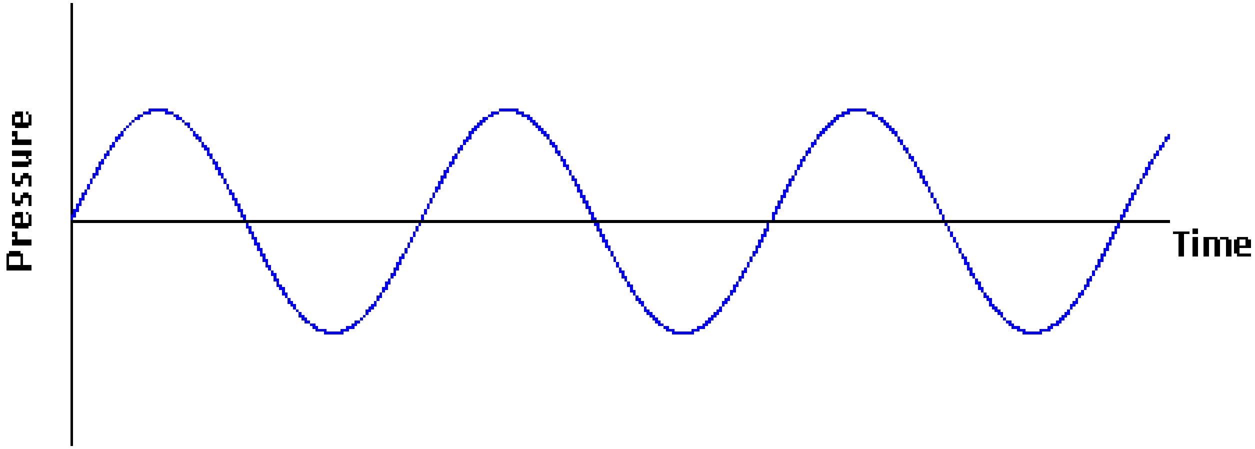
- Sound intensity:
  - an objective measurement
  - can be measured with auditory devices
  - in *decibels* (dB)

- Loudness:
  - a subjective perception
  - measured by human listeners
  - human ears have different sensitivity to different sound frequency
  - in general, higher sound intensity means louder sound

- 0 dB:
  - Threshold of hearing
  - minimum sound pressure level at which humans can hear a sound at a given frequency
  - does NOT mean zero sound intensity
  - does NOT mean absence of sound wave

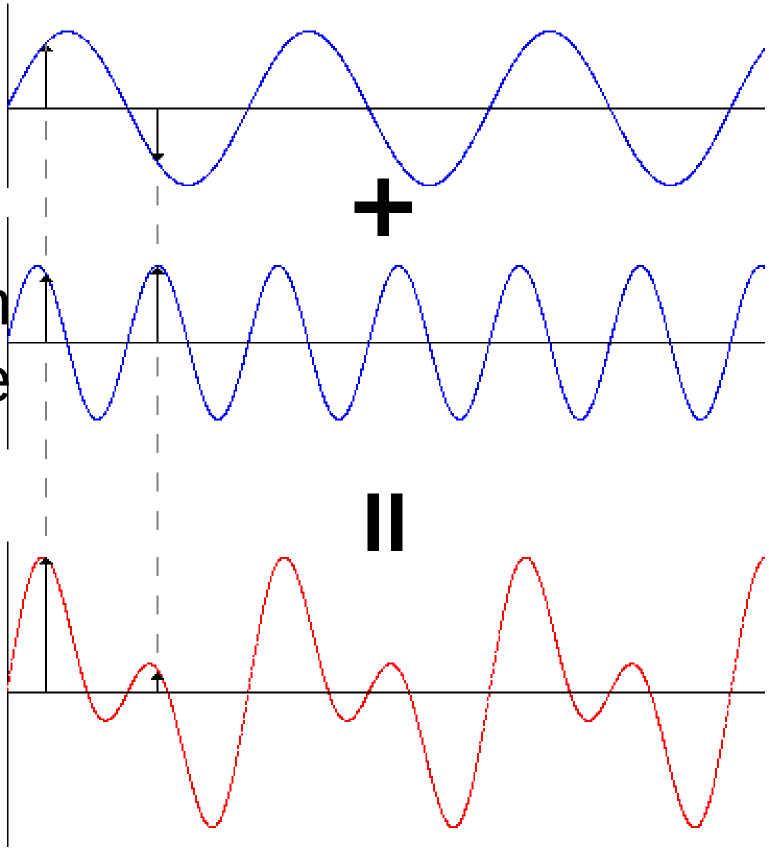
- about 120 dB:
  - threshold of pain
  - sound intensity that is 10^{12} times greater than 0 dB

A Simple Sine Wave Waveform



A sinlge sine wave waveform
A single tone

Adding Sound Waves

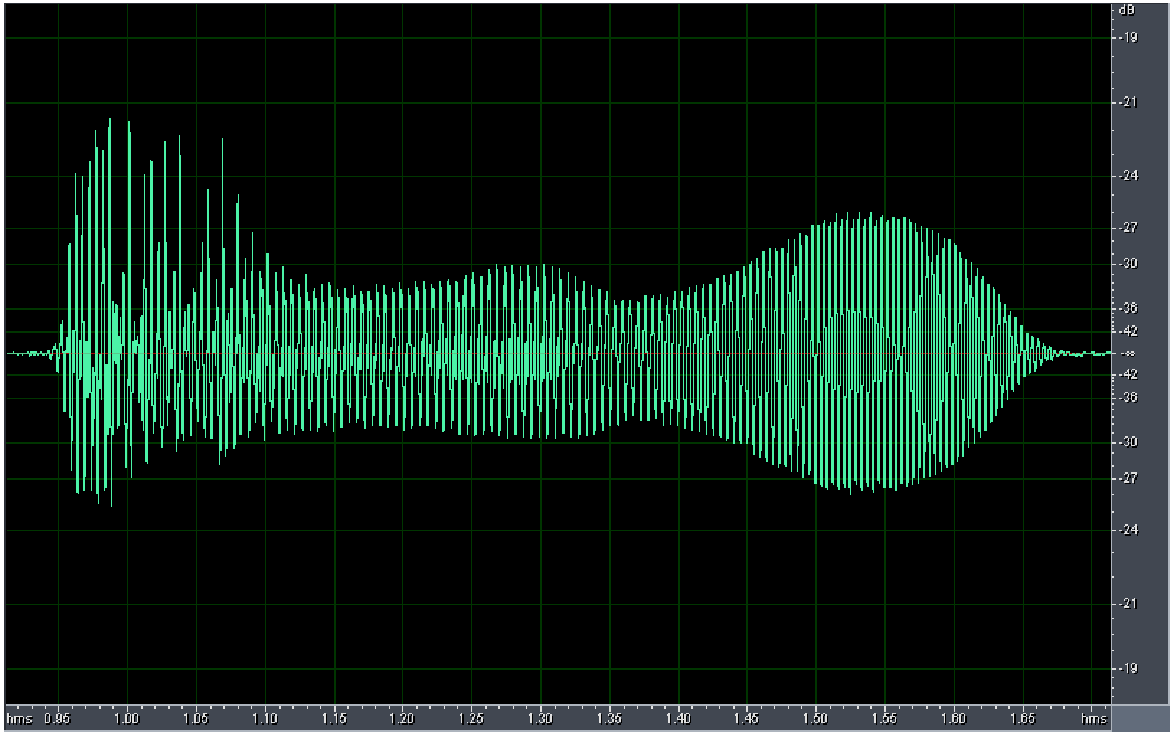


A sinlge sine wave waveform
A single tone

A second sinlge sine wave waveform
A second single tone

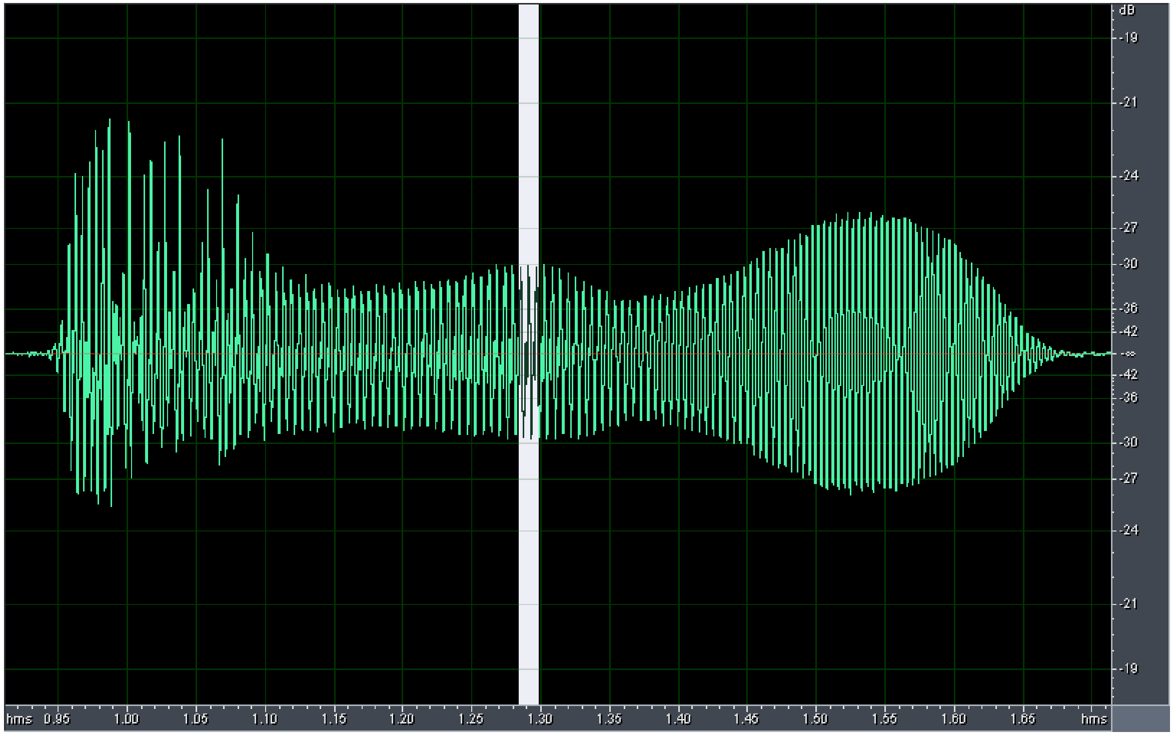
A more complex waveform
A more complex sound

Waveform Example



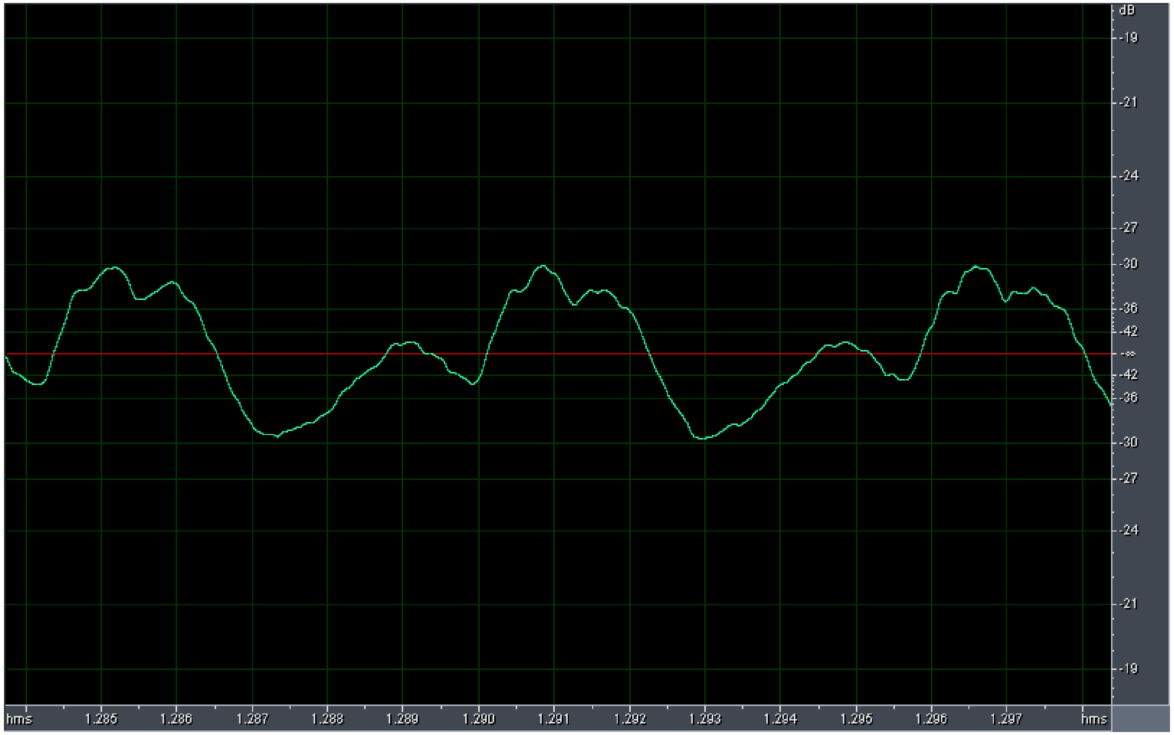
A waveform of the spoken word "one"

Waveform Example



Let's zoom in to take a closer look

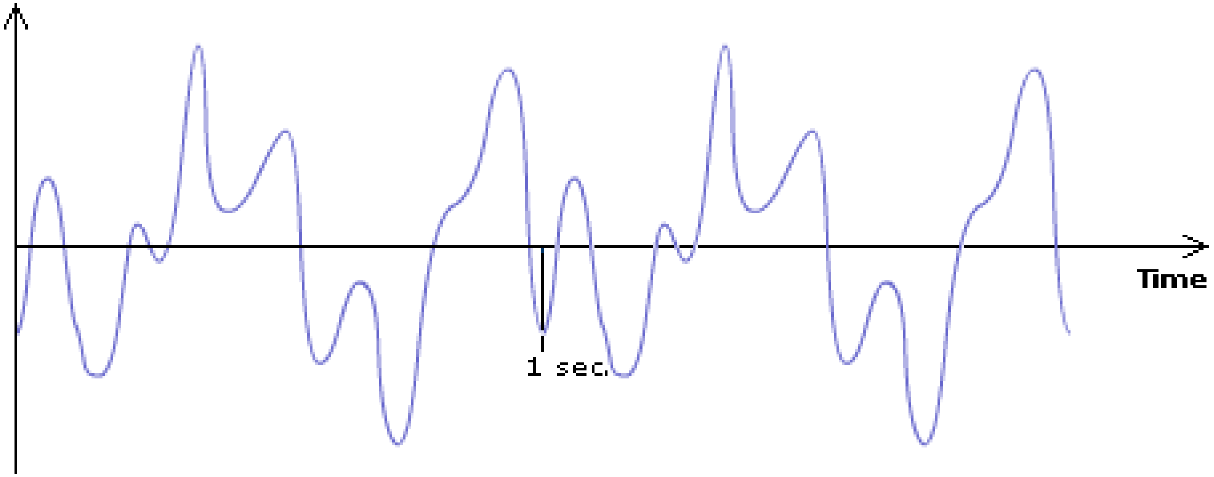
Waveform Example



A closer look

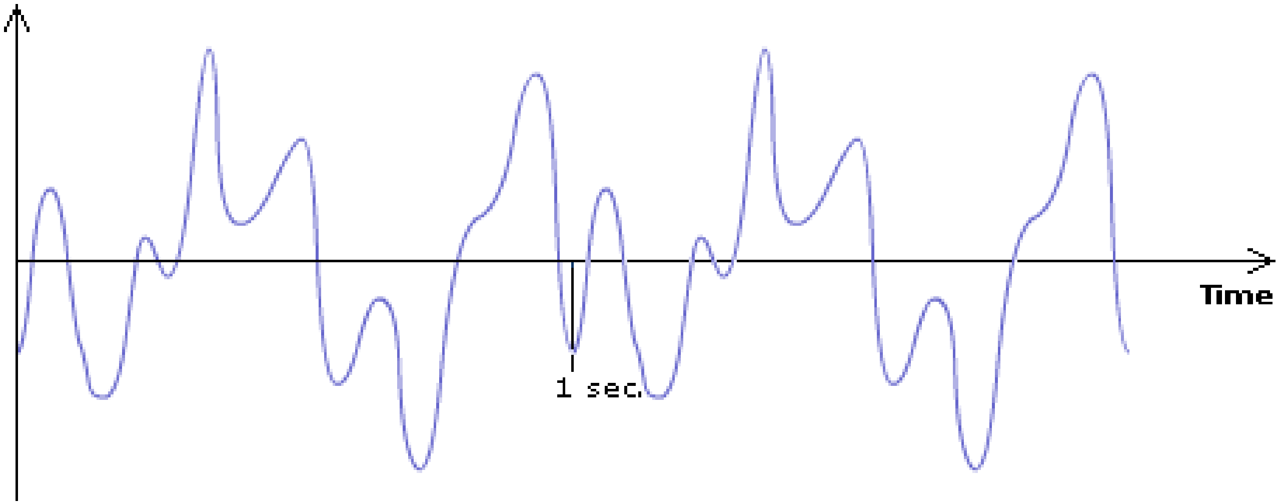
Digitizing Sound

Suppose we want to digitize this sound wave:



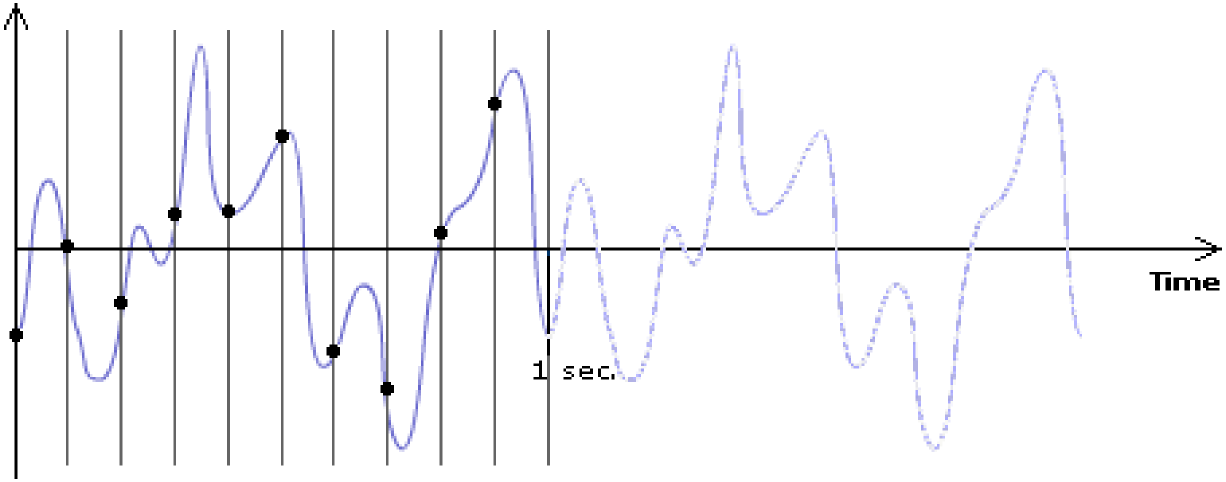
Step 1. Sampling

The sound wave is sampled at a specific rate into discrete samples of amplitude values.



Step 1. Sampling

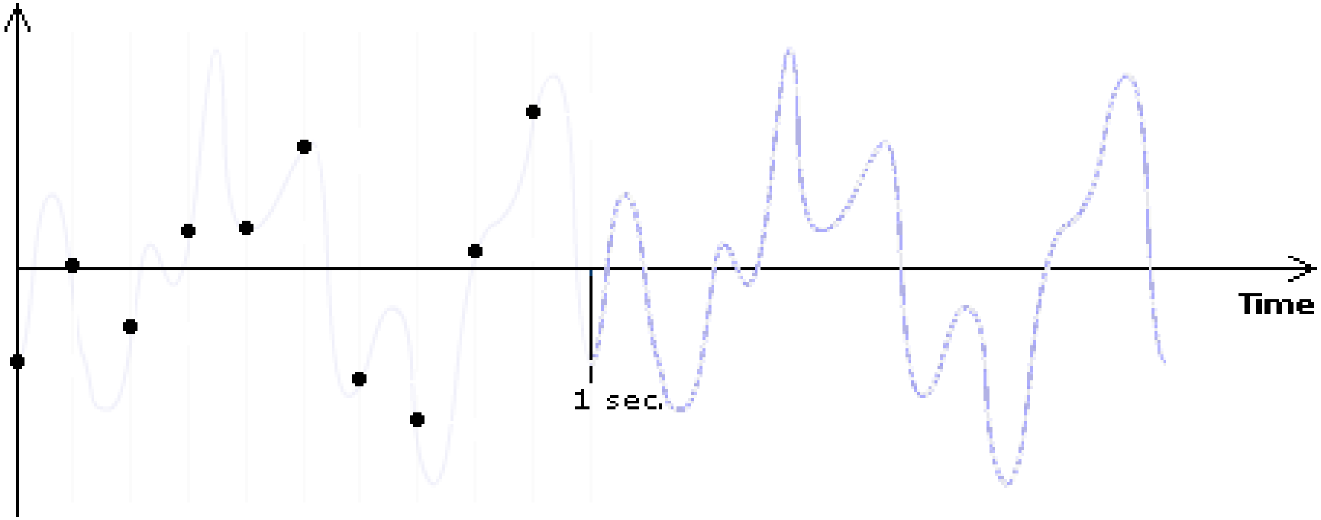
The sound wave is sampled at a specific rate into discrete samples of amplitude values.



Suppose we sample the waveform 10 times a second, i.e., sampleing rate = 10 Hz.

Step 1. Sampling

The sound wave is sampled at a specific rate into discrete samples of amplitude values.

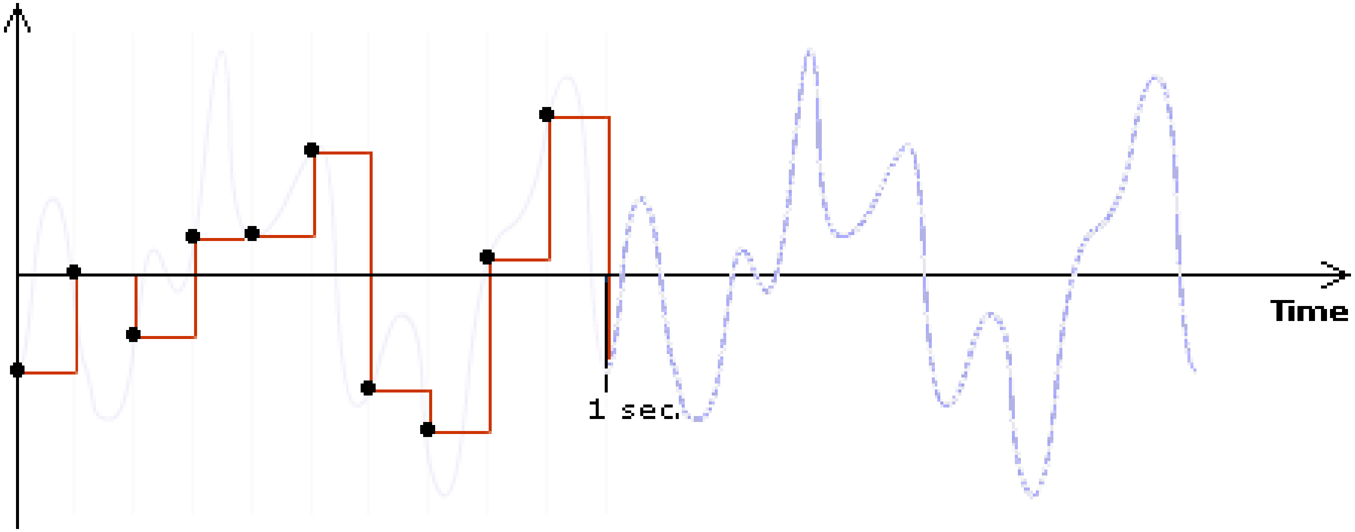


Suppose we sample the waveform 10 times a second, i.e., sampleing rate = 10 Hz.

We get 10 samples per second.

Step 1. Sampling

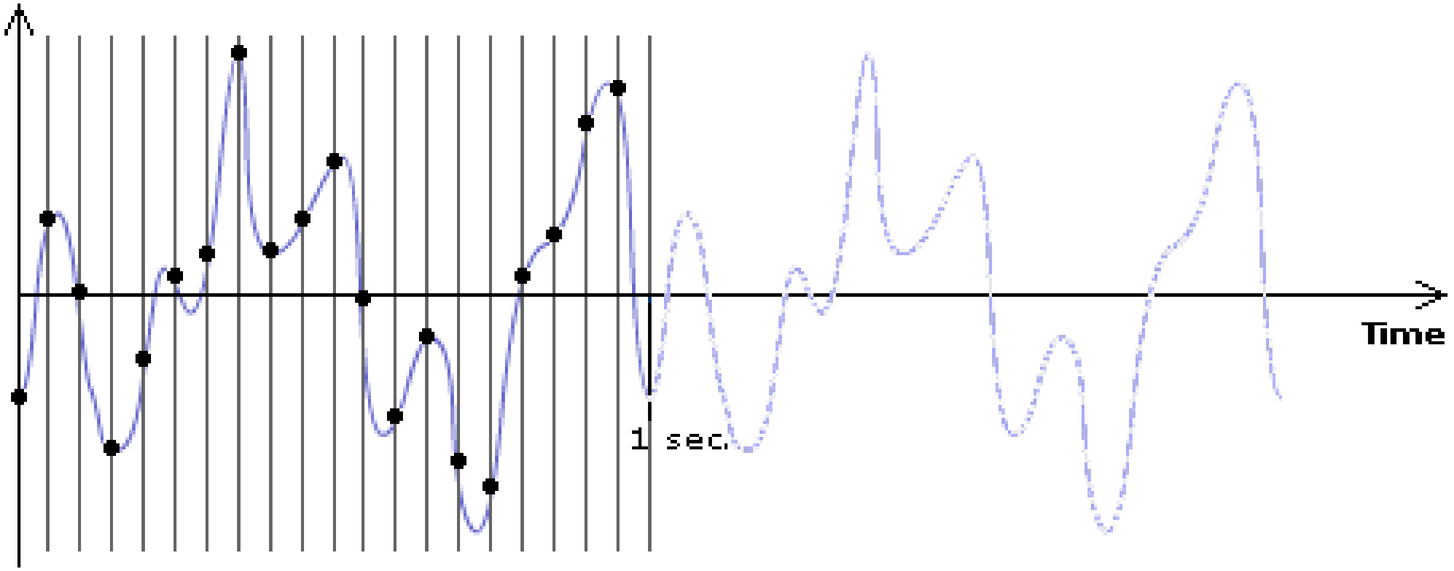
The sound wave is sampled at a specific rate into discrete samples of amplitude values.



Reconstructing the waveform using the discrete sample points.

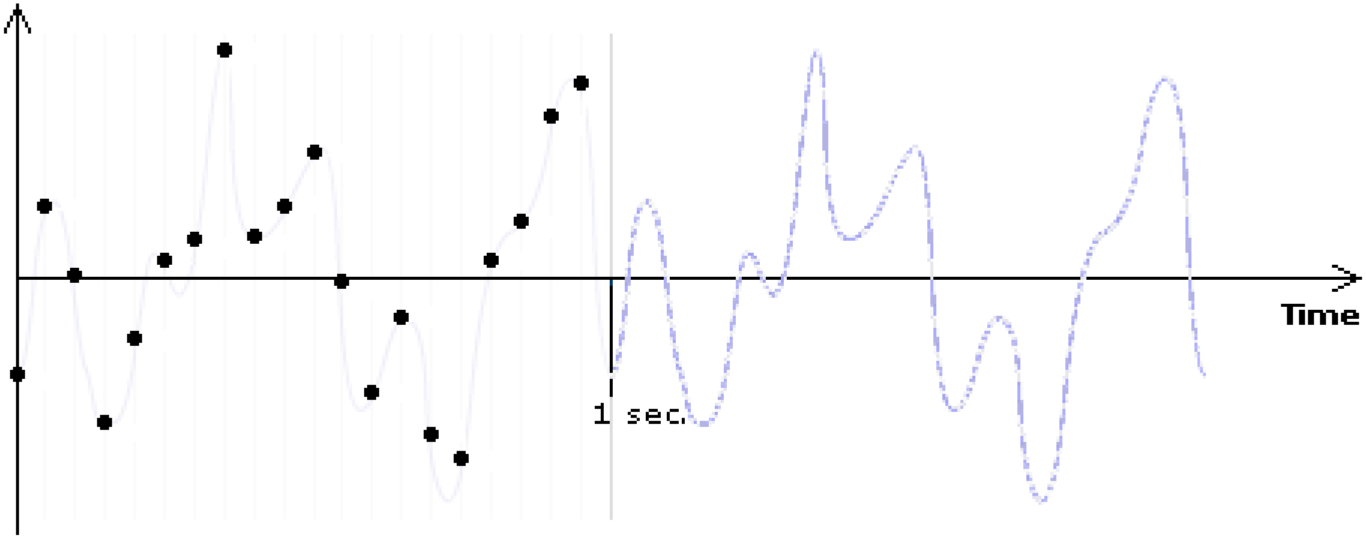
Step 1. Sampling

What if we sample 20 times a second, i.e., sampling rate = 20 Hz?



Step 1. Sampling

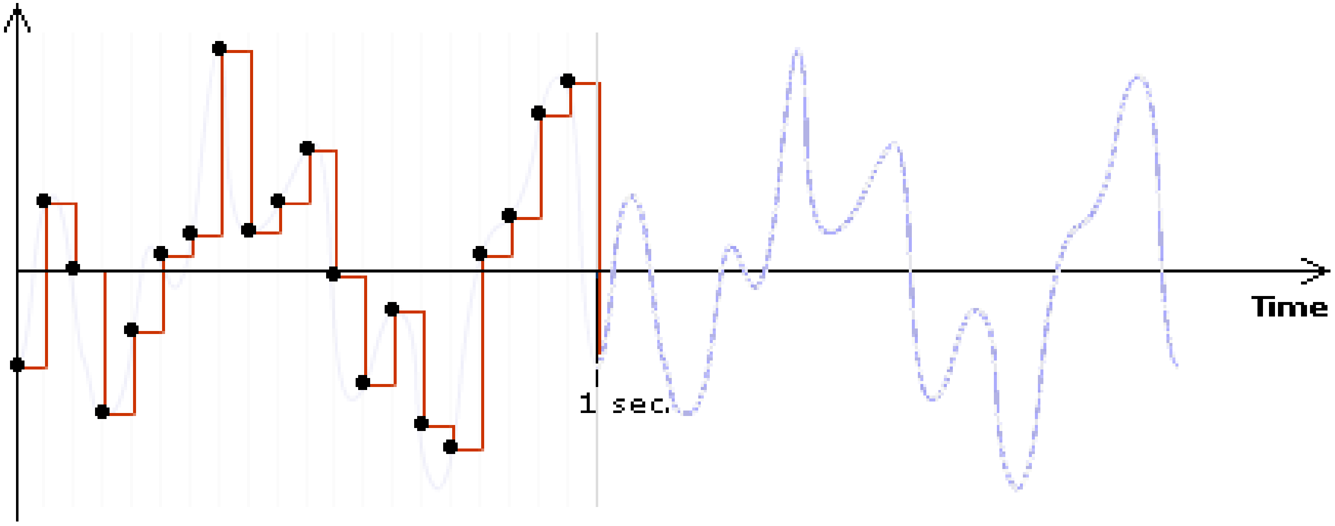
What if we sample 20 times a second, i.e., sampling rate = 20 Hz?



We get 20 samples per second.

Step 1. Sampling

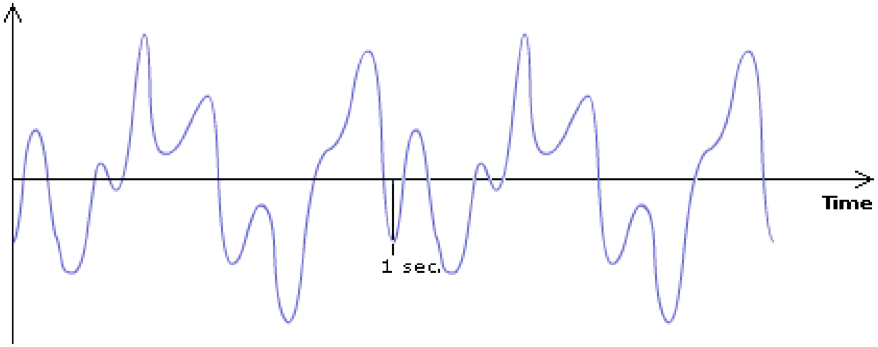
What if we sample 20 times a second, i.e., sampling rate = 20 Hz?



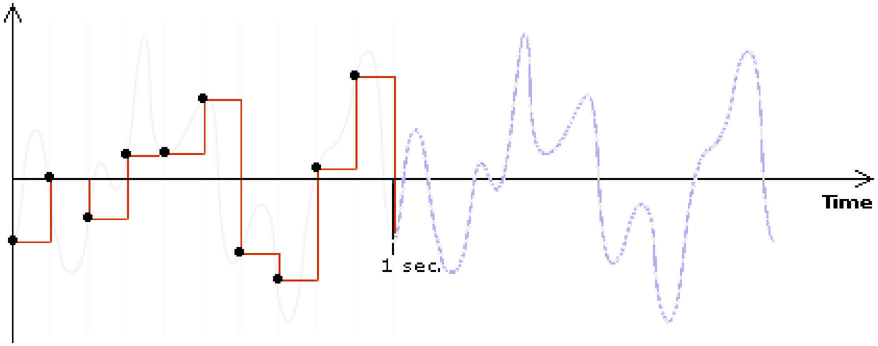
Reconstructing the waveform using the discrete sample points.

Effects of Sampling Rate

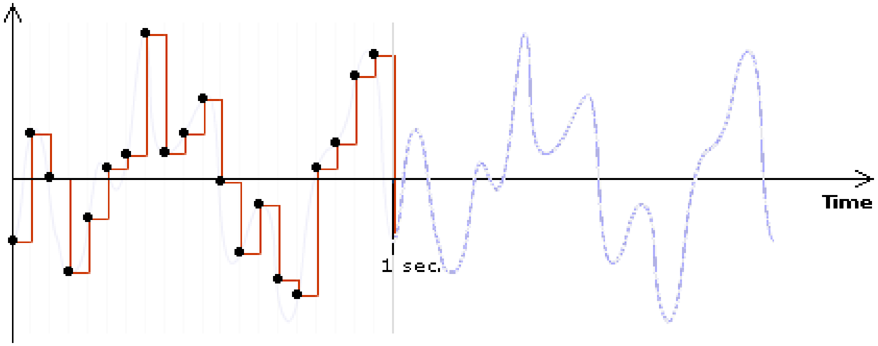
original waveform



sampling rate = 10 Hz



sampling rate = 20 Hz



Effects of Sampling Rate

Higher sampling rate:

- The reconstructed wave looks closer to the original wave

- More sample points, and thus larger file size

Sampling Rate Examples

- 11,025 Hz AM Radio Quality/Speech
- 22,050 Hz Near FM Radio Quality (high-end multimedia)
- 44,100 Hz CD Quality
- 48,000 Hz DAT (digital audio tape) Quality
- 96,000 Hz DVD-Audio Quality
- 192,000 Hz DVD-Audio Quality

Sampling Rate vs. Sound Frequency

- Both uses the unit Hz

BUT:

- sampling rate ≠ sound frequency
- Sample rate: a setting in the digitization process
- Sound frequency: NOT a setting in the digitization process
- Sound frequency: the pitch characteristic of sound
- Higher sampling rate: NOT the pitch characteristic of sound

Step 2. Quantization

- Each of the discrete samples of amplitude values obtained from the sampling step are mapped and rounded to the nearest value on a scale of discrete levels.

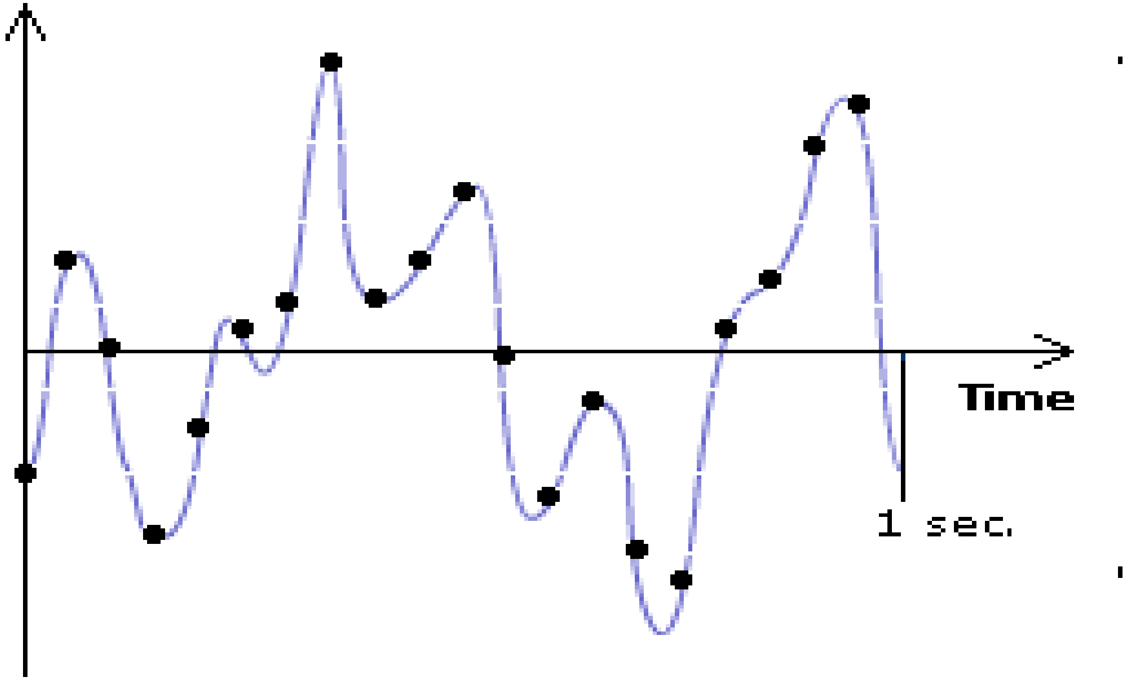
- The number of levels in the scale is expressed in *bit depth*--the power of 2.

- An 8-bit audio allows 2^8 = 256 possible levels in the scale

- CD-quality audio is 16-bit (i.e., 2^{16} = 65,536 possible levels)

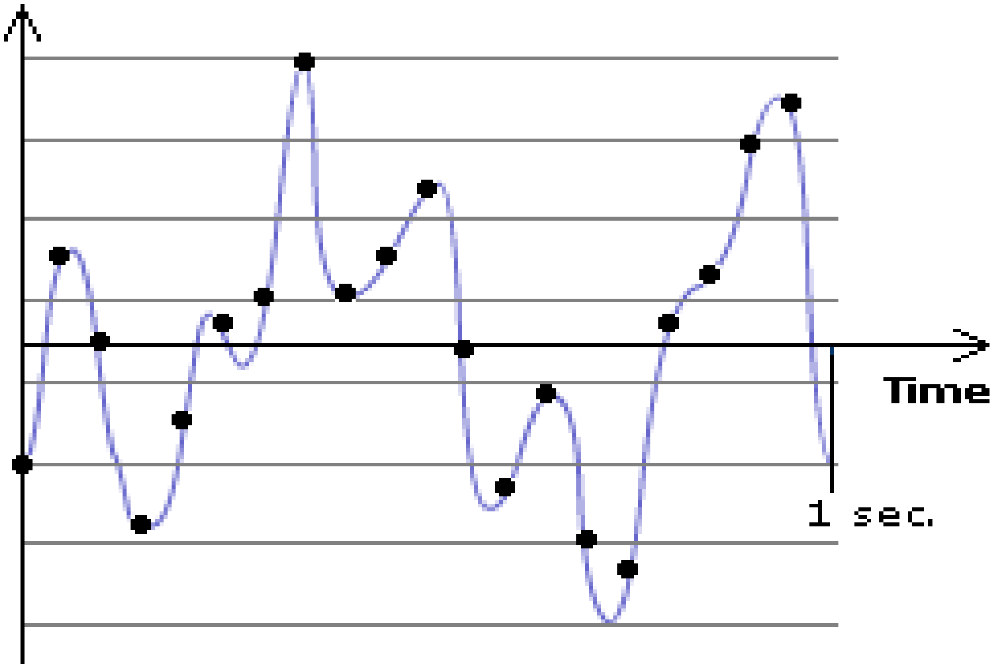
Step 2. Quantization

Suppose we are quantizing the samples using 3 bits (i.e. $2^3 = 8$ levels).



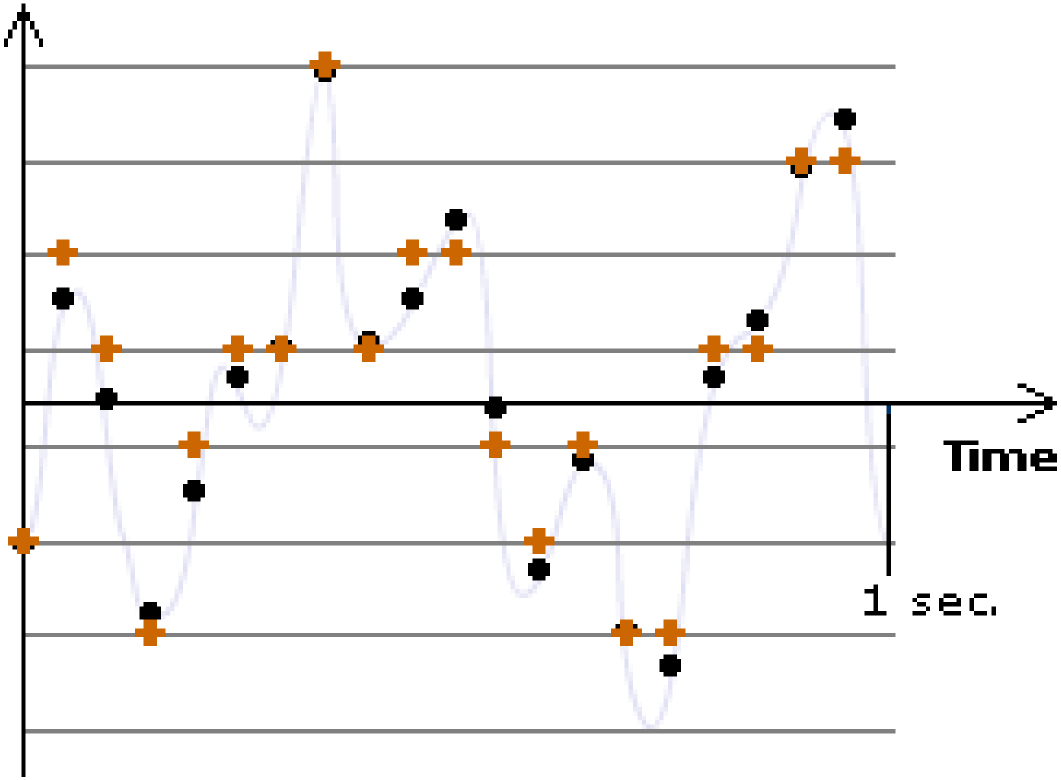
Step 2. Quantization

Suppose we are quantizing the samples using 3 bits (i.e. $2^3 = 8$ levels).



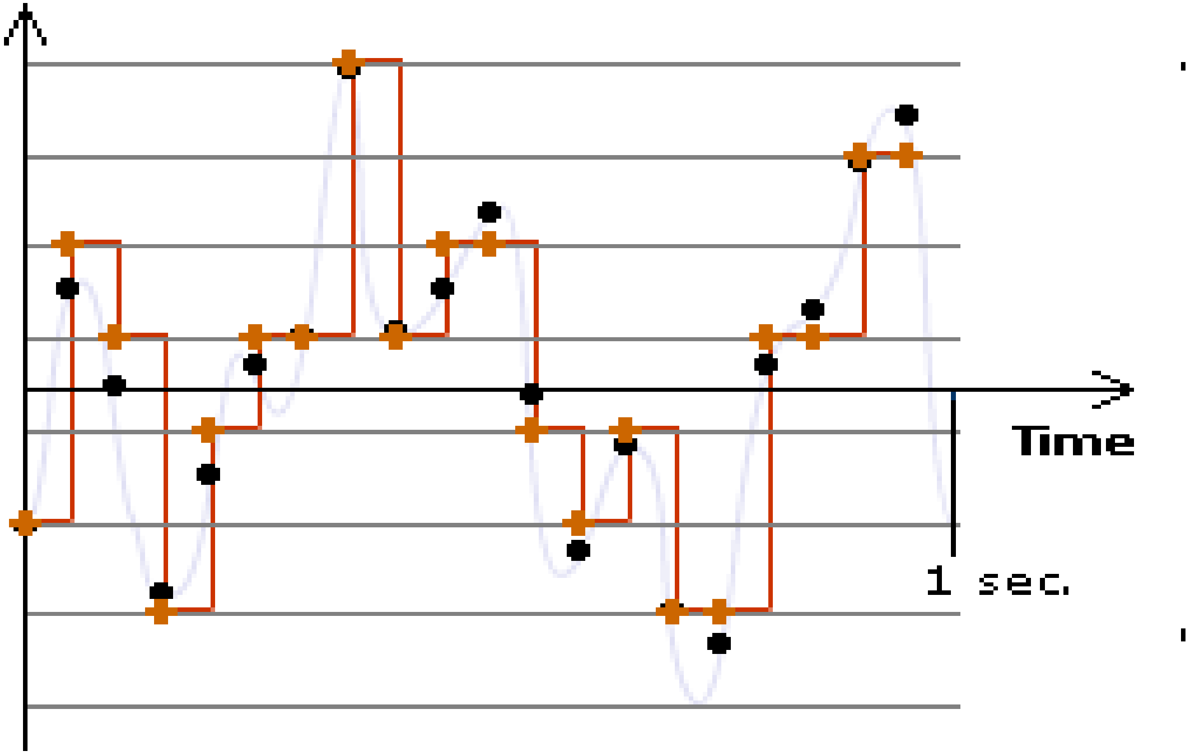
Step 2. Quantization

Now, round each sample to the nearest level.



Step 2. Quantization

Now, reconstruct the waveform using the quantized samples.



Effects of Quantization

- Data with different original amplitudes may be quantized onto the same level $\Rightarrow$ loss of subtle differences of samples

- With lower bit depth, samples with larger differences may also be quantized onto the same level.

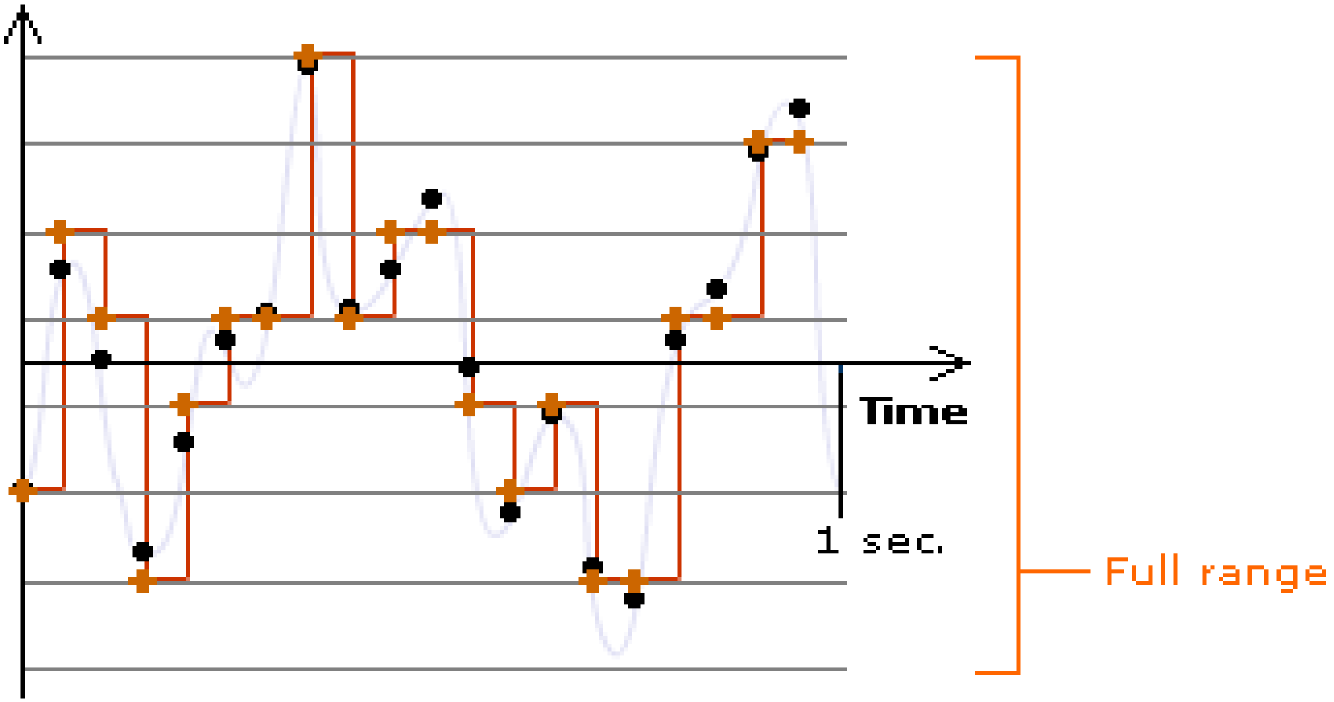
Bit Depth

- Bit depth of a digital audio is also referred to as *resolution*.

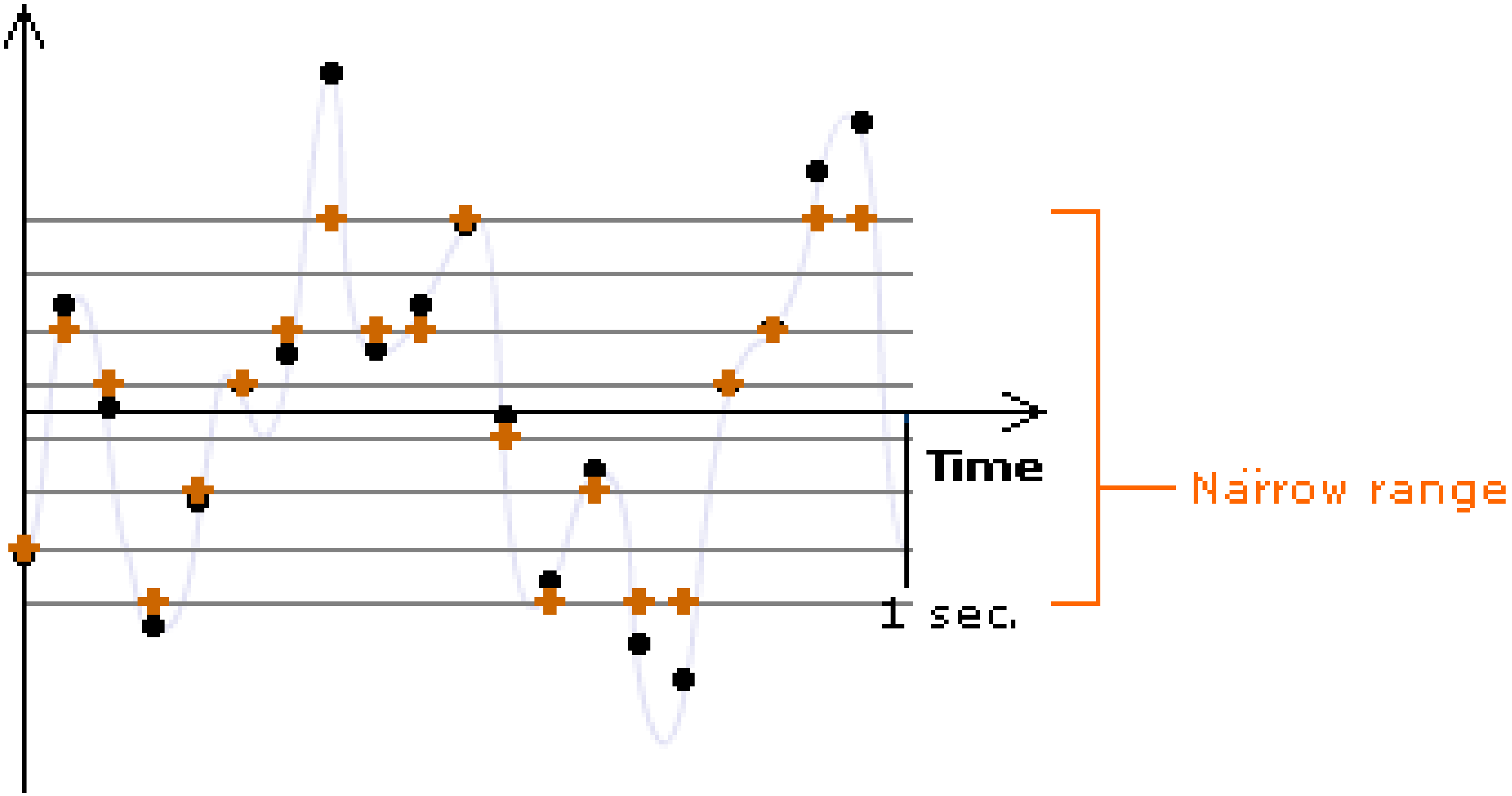
- For digital audio, higher resolution means higher bit depth.

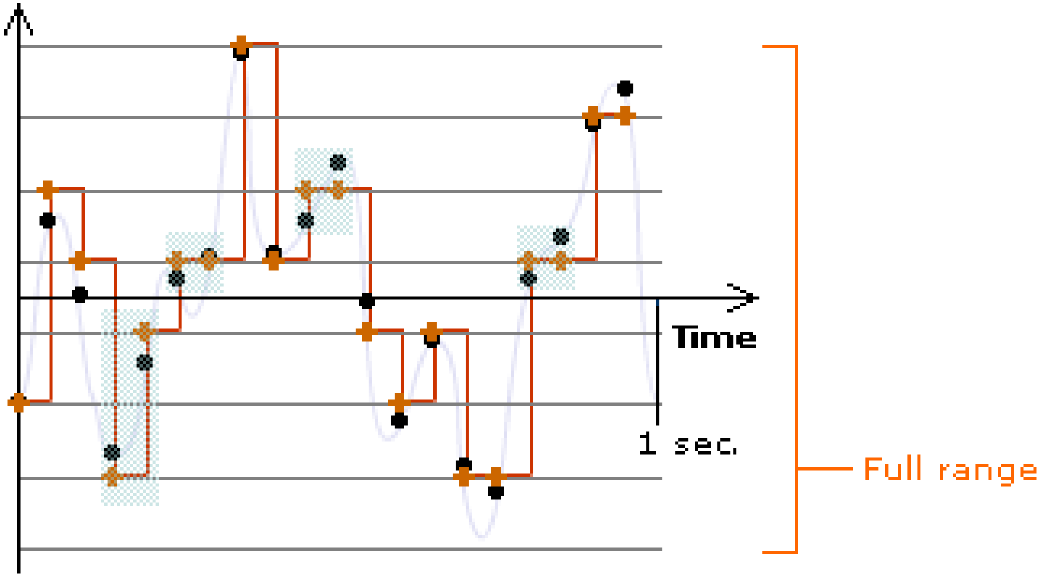
Dynamic Range

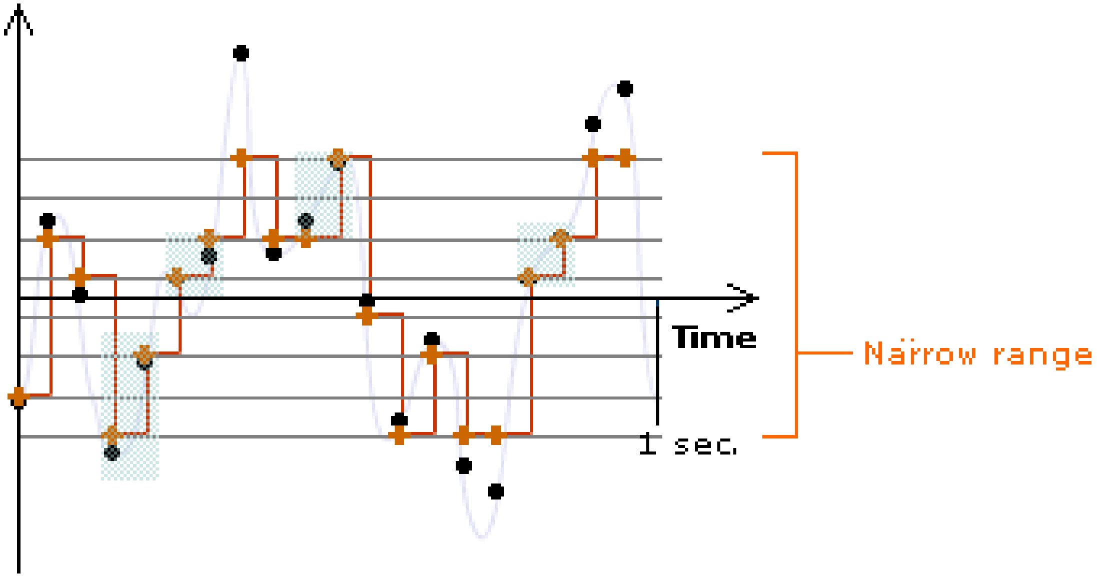
- The range of the scale, from the lowest to highest possible quantization values
- In the previous example:

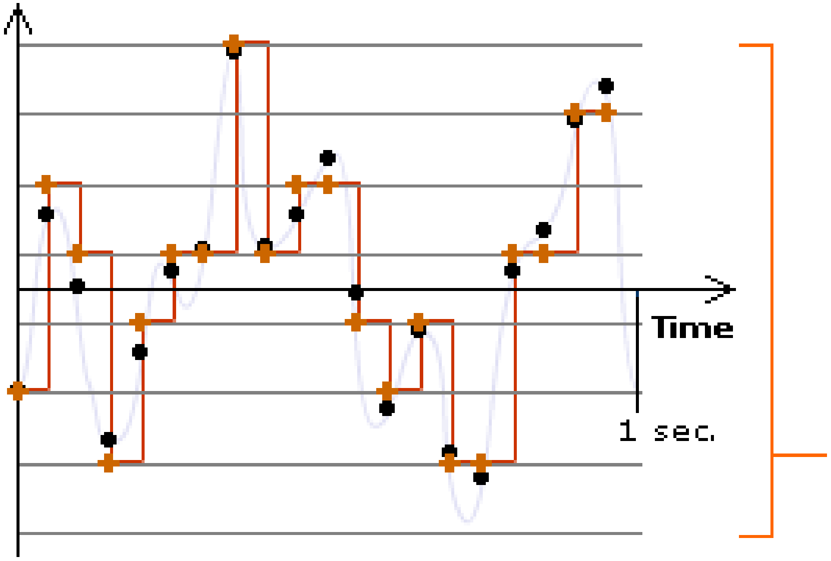


Smaller than the Full Range

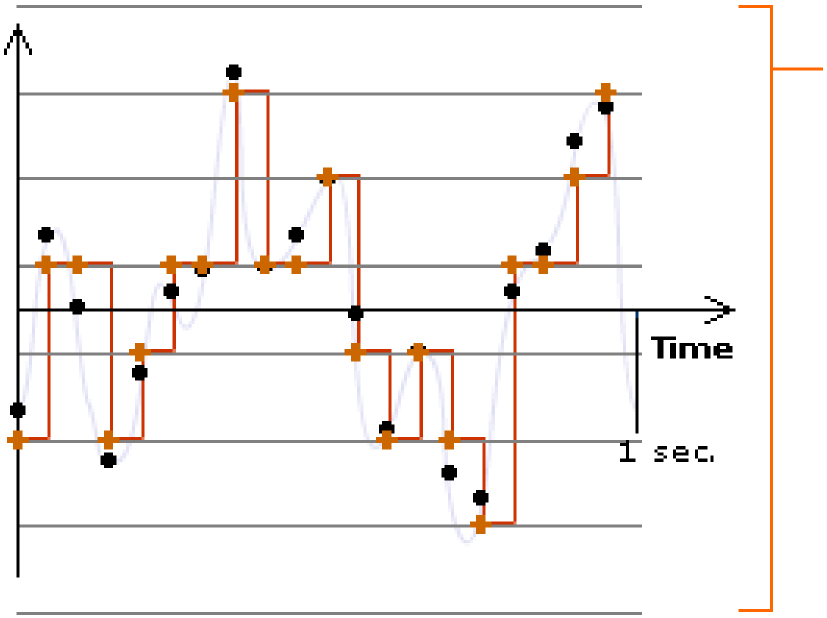



Time
1 sec.
Full range


Time
1 sec.
Narrow range



Full range



Range too large

Choices of Sampling Rate and Bit Depth

Higher sampling rate and bit depth:

- deliver better fidelity of a digitized file
- result in a larger file size (undesirable)

Let's estimate the file size of a 1-minute CD-quality audio file

1-minute CD Qualtiy Audio

- Sampling rate = 44100 Hz
  (i.e., 44,100 samples/second)
- Bit depth = 16
  (i.e., 16 bits/sample)
- Stereo
  (i.e., 2 channels: left and right channels)

File Size of 1-min CD-quality Audio

- 1 minute = 60 seconds

- Total number of samples
  = 60 seconds × 44,100 samples/second
  = 2,646,000 samples

- Total number of bits required for these many samples
  = 2,646,000 samples × 16 bits/sample
  = 42,336,000 bits
  This is for one channel.

- Total bits for two channels
  = 42,336,000 bits/channel × 2 channels
  = 84,672,000 bits

File Size of 1-min CD-quality Audio

84,672,000 bits
= 84,672,000 bits / (8 bits/byte)
= 10,584,000 bytes
= 10,584,000 bytes / (1024 bytes/KB)
≅ 10336 KB
= 10336 KB / (1024 KB/MB)
≅ $\underline{\text{10 MB}}$

Estimate Network Transfer Time

Suppose you are using 1.5Mbps (mega bits per second) broadband to download this 1-minute audio.

The time is no less than

84,672,000 bits / (1.5 Mbps)

= 84,672,000 bits / (1,500,000 bits/seconds)

≅ 56 seconds

File Size of 1-hour CD-quality Audio

$\cong$ 10 MB/minute × 60 minutes/hour
= 600 MB/hour

General Strategies to Reduce Digital Media File Size

- Reduce sampling rate
- Reduce bit depth
- Apply compression

- For digital audio, these can also be options:

  - reducing the number of channels

  - shorten the length of the audio

Reduce Sampling Rate

- Sacrifices the fidelity of the digitized audio
- Need to weigh the quality against the file size
- Need to consider:
  - human perception of the audio (e.g., How perceptibe is the audio with lower sampling rate?)
  - how the audio is used
    - music: may need higher sampling rate
    - short sound clips such as explosion and looping ambient background noise: may work well with lower sampling rate

Sampling Rate Examples

- 11,025 Hz AM Radio Quality/Speech
- 22,050 Hz Near FM Radio Quality (high-end multimedia)
- 44,100 Hz CD Quality
- 48,000 Hz DAT (digital audio tape) Quality
- 96,000 Hz DVD-Audio Quality
- 192,000 Hz DVD-Audio Quality

Estimate Thresholds of Sampling Rate Based on Human Hearing

Let's consider these two factors:

1. Human hearing range
2. A rule called Nyquist's theorem

Human Hearing Range

- Human hearing range: 20 Hz to 20,000 Hz
- Most sensitive to 2,000 Hz to 5,000 Hz

Nyquist Theorem

We must sample at least 2 points in each sound wave cycle to be able to reconstruct the sound wave satisfactorily.

⇒Sampling rate of the audio ≥ twice of the audio frequency (called a *Nyquist rate*)

⇒Sampling rate of the audio is higher for audio with higher pitch

Choosing Sampling Rate

Given the human hearing range (20 Hz to 20,000 Hz) and Nyquist Theorem, why do you think the sampling rate (44,100 Hz) for the CD-quality audio is reasonable?

Choosing Sampling Rate

If we consider human ear's most sensitive range of frequency (2,000 Hz to 5,000 Hz), then what is the lowest sampling rate may be used that still satisfies the Nyquist Theorem?

A. 11,025 Hz AM Radio Quality/Speech
B. 22,050 Hz Near FM Radio Quality (high-end multimedia)
C. 44,100 Hz CD Quality
D. 48,000 Hz DAT (digital audio tape) Quality
E. 96,000 Hz DVD-Audio Quality
F. 192,000 Hz DVD-Audio Quality

Effect of Sampling Rate on File Size

File size = duration × sampling rate × bit depth × number of channels

- File size is reduced in the same proportion as the reduction of the sampling rate
- Example: Reducing the sampling rate from 44,100 Hz to 22,050 Hz will reduce the file size by half.

Effect of Bit Depth on File Size

File size = duration × sampling rate × bit depth × number of channels

- File size is reduced in the same proportion as the reduction of the bit depth
- Example: Reducing the bit depth from 16-bit to 8-bit will reduce the file size by half.

Most Common Choices of Bit Depth

- 8-bit
  - usually sufficient for speech
  - in general, too low for music
- 16-bit
  - minimal bit depth for music
- 24-bit
- 32-bit

Audio File Compression

- Lossless
- Lossy
  - gets rid of some data, but human perception is taken into consideration so that the data removed causes the least noticeable distortion
  - e.g. MP3 (good compression rate while preserving the perceivably high quality of the audio)

Effect of Number of Channels on File Size

File size = duration × sampling rate × bit depth × number of channels

- File size is reduced in the same proportion as the reduction of the number of channels

- Example: Reducing the number of channels from 2 (stereo) to 1 (mono) will reduce the file size by half.

Common Audio File Types

File Type	Acronym For	Originally Created By	File Info & Compression	Platforms
.wav		IBM Microsoft	• Compressed or uncompressed • One of the HTML5 audio formats	• Windows • Plays in Web browsers that support the .wav format of HTML5 audio (Firefox, Safari, Chrome, and Opera)
.mp3	MPEG audio layer 3	Moving Pictures Experts Group	• Good compression rate with perceivably high quality sound • One of the HTML5 audio formats	• Cross-platform •Plays in Web browsers that support the .wav format of HTML5 audio (Safari and IE)
.m4a	MPEG-4 format without the video data	Moving Pictures Experts Group	•AAC compression; same compression as the MPEG-4 H.264 without the video data • One of the HTML5 audio formats	Plays in Web browsers that support the AAC format of HTML5 audio (Safari, IE, and Chrome)

Common Audio File Types

File Type	Acronym For	Originally Created By	File Info & Compression	Platforms
.ogg or .oga		Xiph.Org Foundation	• Usually referred to as Ogg Vorbis format • One of the HTML5 audio formats	Plays in Web browsers that support the Ogg Vorbisformat of HTML5 audio (Firefox, Chrome, and Opera)
.mov	QuickTime movie	Apple	• Not just for video • supports audio track and a MIDI track • a variety of sound compressors • files can be streamed • "Fast Start" technology	Cross-platform; requires QuickTime player

Common Audio File Types

File Type	Acronym For	Originally Created By	File Info & Compression	Platforms
.aiff	Audio Interchange File Format	Apple	compressed, uncompressed	Mac, Windows
.au .snd		Sun	compressed	Sun, Unix, Linux
.ra .rm	Real Audio	Real Systems	compressed; can be streamed with Real Server	Cross-platform; requires Real player
.wma	Window Media Audio	Microsoft		

Choosing an Audio File Type

Determined by the intended use

- File size limitation
- Intended audience
- Whether as a source file

File Size Limitations

- Is your audio used on the Web?
  - file types that offer high compression
  - streaming audio file types

Intended Audience

- What is the equipment that your audience will use to listen to your audio?

- If they are listening on computers, what are their operating systems?
  - cross-platform vs. single platform

Whether as a Source File

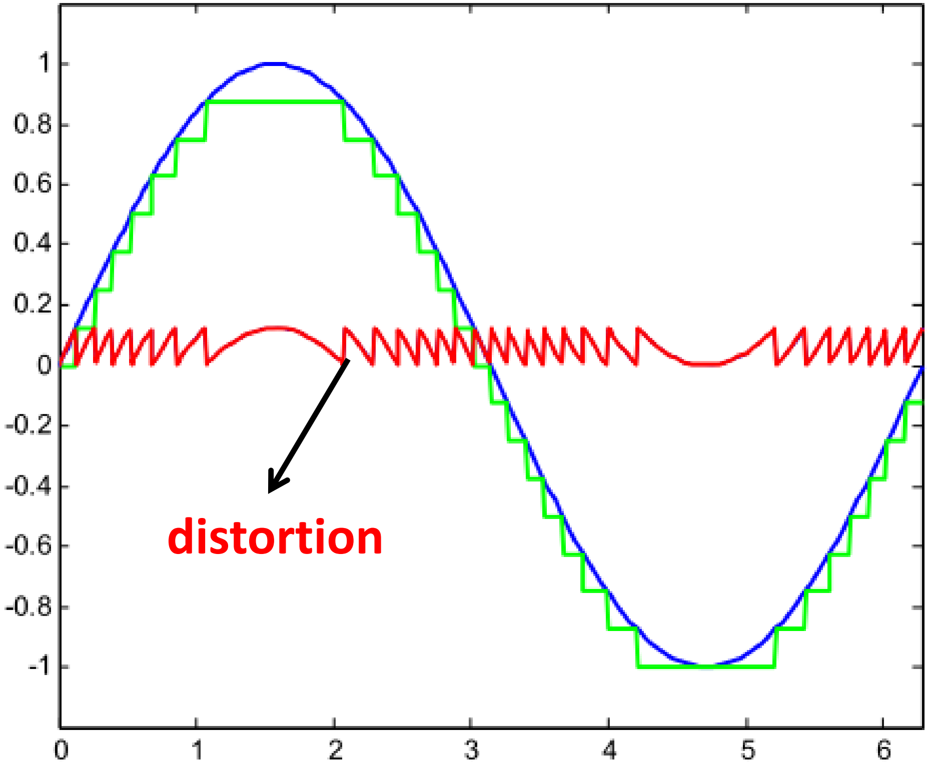
If you are keeping the file for future editing, choose a file type:

- uncompressed
- allows lossless compression

Audio Dithering

Audio dithering is a way to compensate for quantization error. The way to do this is to add small random values to samples in order to mask quantization error.

- The rounding inherent in quantization causes a problem in that at low amplitudes, many values may be rounded down to 0 (noticeable breaks in the sound).
- small random values between 0 and the least significant bit (on the scale of the new bit depth) are added to the signal before it is quantized

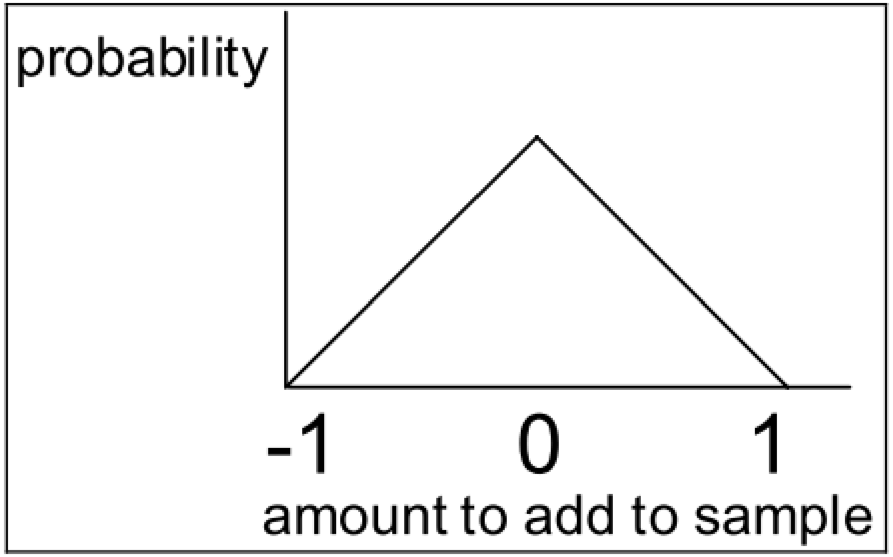


Original wave (the sine wave), quantized sine wave, and quantization error wave.

(16 quantization levels)

The error waveform is periodic. Noise is random, while distortion sounds "meaningful" even though it is not. The distortion wave moves in a pattern along with the original wave and thus, to human hearing, it seems to be a meaningful part of the sound.

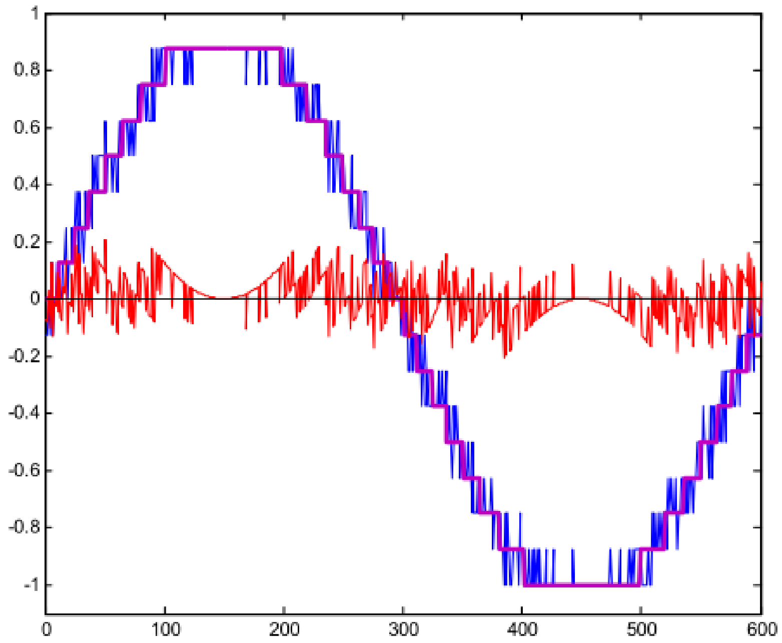
It is easier for the human brain to tune out noise because it seems to have no logical relationship to the dominant patterns of the sound.



Triangular probability function

there is the greatest probability that 0 will be added to a sample

- A simple way to generate values for a triangular probability density function is to take the sum of two random numbers between −0.5 and 0.5 (between -1 and 1).
- Adding this random noise to the original wave eliminates the sharp stairstep effect in the quantized signal.
- there aren't as many neighboring low-amplitude values that are rounded to 0 when they are quantized



Quantized sine wave, dithered quantized wave, and (along horizontal axis) error wave including dithering

Figure shows dithering with the triangular dithering function, which produces random values between −1 and 1. (It's assumed in this figure that the signal is originally at a bit depth of 16 and is being reduced to a bit depth of four, so the amount of dither is scaled by ¼ to between -0.125 and 0.125.

Other dithering functions include the ***rectangular probability density function*** (RPDF), the ***Gaussian PDF***, and ***colored dithering***.

Noise Shaping

Another way to compensate for quantization error. It is an important component in the design of ADCs and DACs.

- The idea behind noise shaping is to **redistribute the quantization error** so that the noise is concentrated in the higher frequencies (less sensitive).
- One can use noise shaping in conjunction with dithering if you reduce the bit depth of an audio file.
- work by computing the error that results from quantizing the *i*th sample and adding this error to the next sample, before that next sample is itself quantized.

Consider the following definition of a first-order feedback loop for noise shaping:

Let $\boldsymbol{F_{in}}$ be an array of *N* digital audio samples that are to be quantized, dithered, and noise shaped, yeilding $\boldsymbol{F_{out}}$.

For $0 \leq i \leq N - 1$, define the following:

$\boldsymbol{F_{in_i}}$ is the *i*th sample vaule, not yet quantized.

$\boldsymbol{D_i}$ is a random dithering value added to the *i*th sample.

The assignment statement $\boldsymbol{F_{in_i} = F_{in_i} + D_i + cE_{i-1}}$ dithers and noise shapes the sample.

Subsequently, $\boldsymbol{F_{out_i}} = \lfloor \boldsymbol{F_{in_i}} \rfloor$ quantizes the sample.

$\boldsymbol{E_i}$ is the error resulting from quantizing the *i*th sample after dithering and noise shaping.

For $i = -1, \boldsymbol{E_i} = 0$. Otherwise, $\boldsymbol{E_i = F_{in_i} - F_{out_i}}$.

An example:

Assume that audio is being recorded in 8 bit samples. On the scale of 8 bits, sound amplitudes can take any value between -128 and 127. (These values do not become integers between -128 and 127 until after quantization.)

$$\boxed{-2^{n-1} \leq x \leq 2^{n-1} - 1}$$

Say that

$F_{in_0} = 68.2, F_{in_1} = 70.4, D_0 = 0.9, D_1 = -0.6, and\ c = 1.$

$$F_{in_0} = F_{in_0} + D_0 + cE_{-1} = 68.2 + 0.9 + 0 = 69.1$$

$$F_{out_0} = \lfloor F_{in_0} \rfloor = 69$$

$$E_0 = F_{in_0} - F_{out_0} = 69.1 - 69 = 0.1$$

$$F_{in_1} = F_{in_1} + D_1 + cE_0 = 70.4 - 0.6 + 0.1 = 69.9$$

$$F_{out_1} = \lfloor F_{in_1} \rfloor = 69$$

$$E_1 = F_{in_i} - F_{out_1} = 69.9 - 69 = 0.9$$

To understand the benefit of noise shaping, think about the frequency spectrum of quantization noise – that is, the range of the frequency components – when noise shaping is not used.

Nyquist theorem?

The idea is that if the error from dithering and quantizing F_i is larger than the previous one, then the error for F_{i+1} ought to be smaller for F_i.

↓

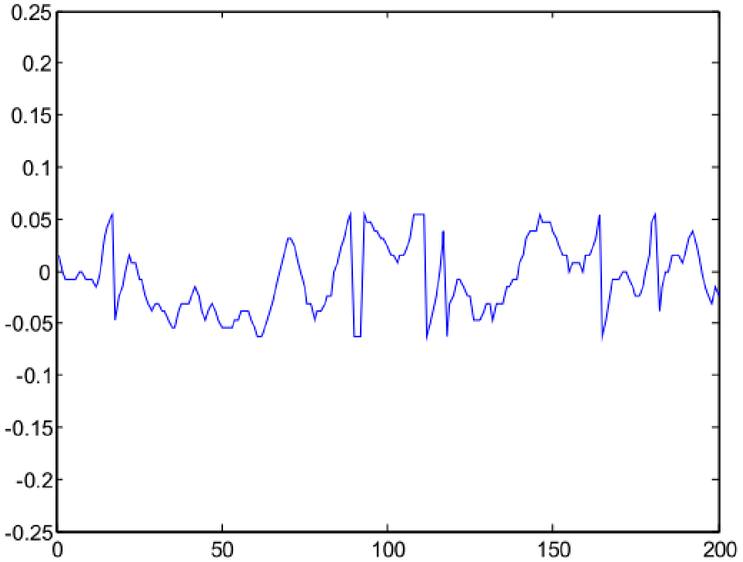
Error wave go up & down rapidly
(move to a higher frequency)

The general statement for an *n*th order noise shaper:

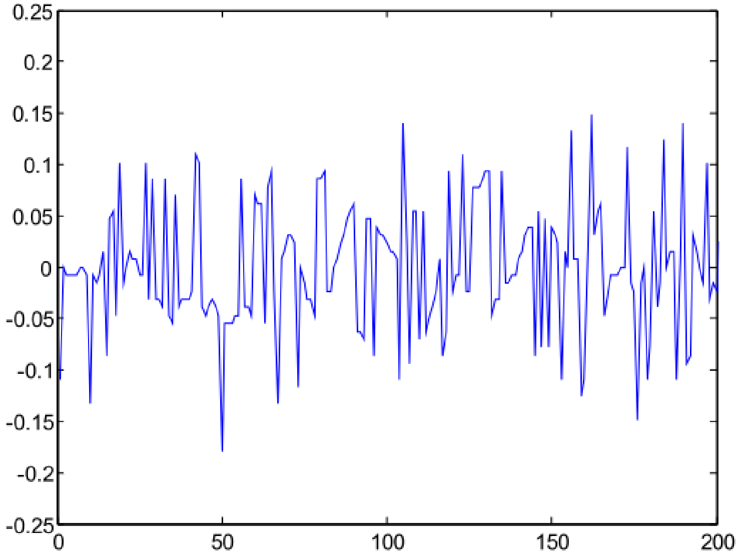
$$\boldsymbol{F_{out_i} = F_{in_i} + D_i + c_{i-1}E_{i-1} + c_{i-2}E_{i-2} + \cdots + c_{i-n}E_{i-n}}$$

recall the distinction between *distortion* & *noise*

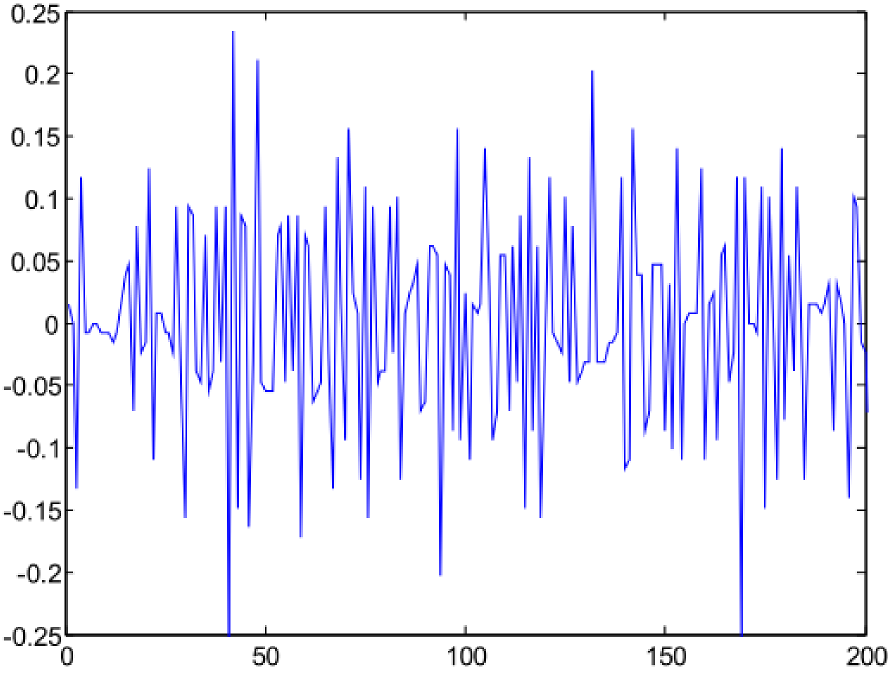
Noise shaping doesn't do anything to dissociate the noise's frequency pattern from the signal, so it's important to use dithering in conjunction with noise shaping.



quantization error wave with no dithering



quantization error wave with dithering



Quantization error wave with dithering & noise shaping

The effect of requantizing an audio file from 16 down to 4 bits.

Noise shaping doesn’t work well if sampling rate < 32kHz.

Non-linear Quantization

Non-linear encoding, or ***companding (compression and then expansion)***, is an encoding method that arose from the need for compression of telephone signals across low bandwidth. It works as follows:

- Take a digital signal with bit depth *n* and requantize it in *m* bits, *m* < *n*, using a non-linear quantization method.
- Transmit the signal.
- Expand the signal to *n* bits at the receiving end. Some information will be lost, since quantization is lossy. However, the non-linear quantization method lessens the error for low amplitude signals as compared to linear quantization.

- Humans can perceive small differences between quiet sounds, but as sounds get louder our ability to perceive a difference in their amplitude diminishes.
- Quantization error generally has more impact on low amplitudes than on high ones.

  -A value of 0.499 rounds down to 0, for 100% error, while value of 126.499 rounds up to 126, for about 0.4% error.

use more quantization levels for low amplitude signals and fewer quantization levels for high amplitudes.

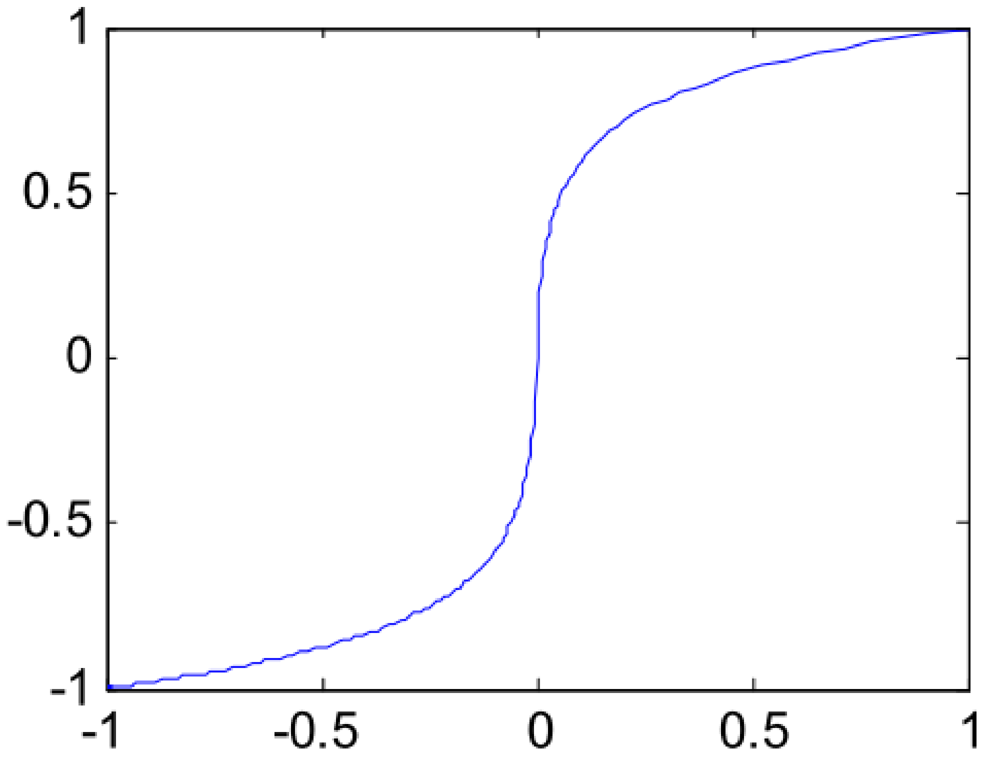
- Non-linear companding schemes are widely used and have been standardized under the CCITT Recommendations for telecommunications (Comité Consulatif Internationale de Télégraphique et Téléphonique).
- ***μ-law*** (also called ***mu-law***) ***encoding*** - the standard for compressing telephone transmissions, using a sampling rate of 8000 Hz and a bit depth of only eight bits, but achieving about 12 bits of dynamic range (the equivalent standard for the rest of the world is called ***A-law*** encoding)

The encoding method is defined by the following function:

Let x be a sample value normalized so that $-1 \leq x < 1$.
Let $sign(x) = -1$ if x is negative and $sign(x) = 1$ otherwise.
Then the ***μ-law function*** (also called ***mu-law***) is defined by

$$m(x) = sign(x)\left(\frac{\ln\left(1+\mu\left|x\right|\right)}{\ln(1+\mu)}\right)$$

$$= sign(x)\left(\frac{\ln(1+255\left|x\right|)}{5.5452}\right) \quad for \;\; \mu = 255$$



Logarithmic function for non-linear audio encoding

Assume you have 16-bit audio samples with values ranging from -32,768 to 32,767. You are going to transmit the signal at a bit depth of 8, and expand it back to 16 bits at the receiving end.

1. normalization

2. apply the function to get m(x)

3. compute [128 m(x)] to scale the value to a bit depth of 8

Example: An initial 16-bit sample of 16

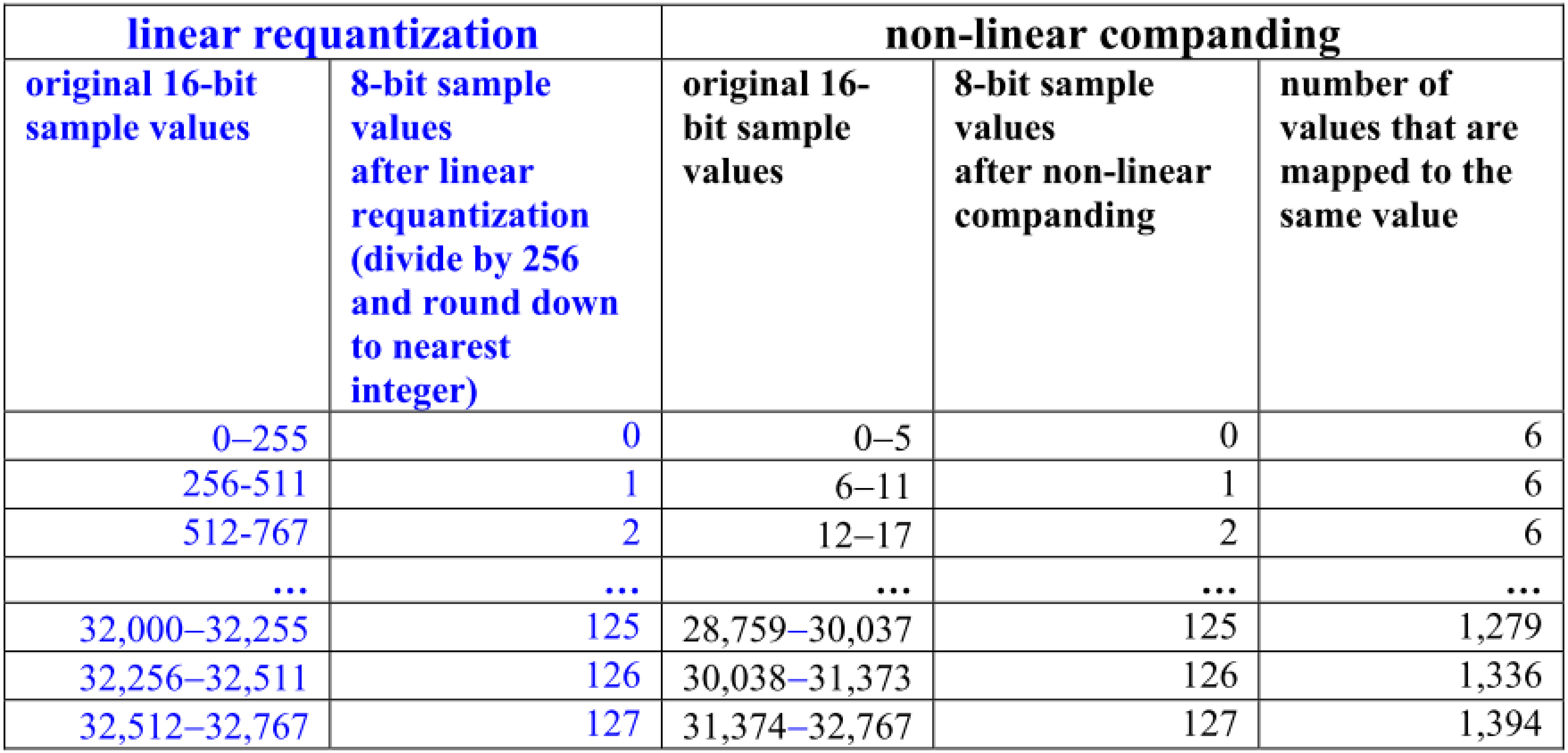
Apply the μ-law function: $m(\frac{16}{32{,}768}) \approx 0.02$

Scale to 8-bit samples: $\lfloor 128 * 0.02 \rfloor = 2$

Example: An initial value of 30,037

Apply the μ-law function: $m(\frac{30{,}037}{32{,}768}) = 0.9844$

Scale to 8-bit samples: $\lfloor 128 * 0.9844 \rfloor = 125$

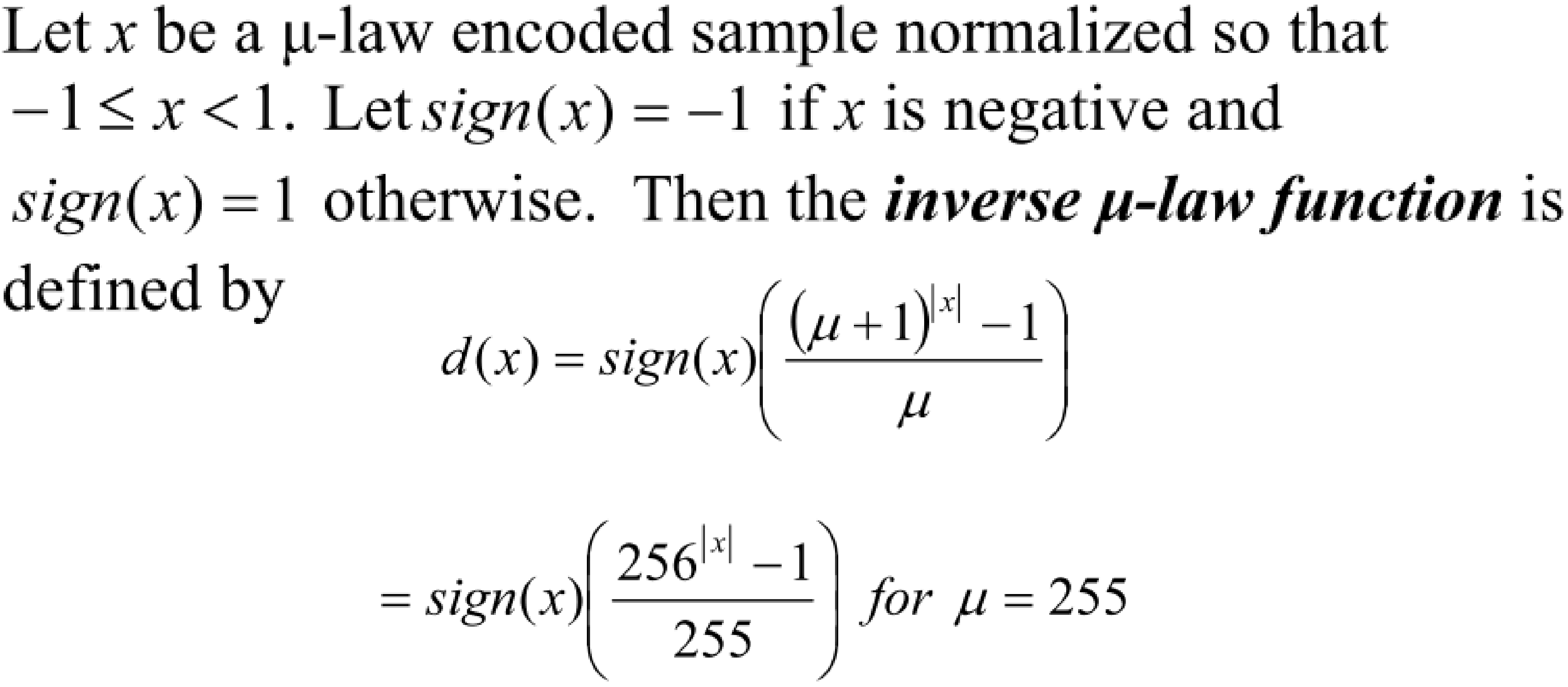


linear requantization		non-linear companding		
original 16-bit sample values	8-bit sample values after linear requantization (divide by 256 and round down to nearest integer)	original 16-bit sample values	8-bit sample values after non-linear companding	number of values that are mapped to the same value
0–255	0	0–5	0	6
256-511	1	6–11	1	6
512-767	2	12–17	2	6
...	...	...	...	...
32,000–32,255	125	28,759–30,037	125	1,279
32,256–32,511	126	30,038–31,373	126	1,336
32,512–32,767	127	31,374–32,767	127	1,394

Comparison of quantization interval size with linear requantization and non-linear companding

Linear quantization using eight bits would create equal-sized quantization intervals – each of them containing 65,536/256 = 256 sample values ranging from -128 and 127.

After compression using the µ-law function, samples can be transmitted in eight bits. At the user-end, they are decompressed to 16 bits with the inverse function.

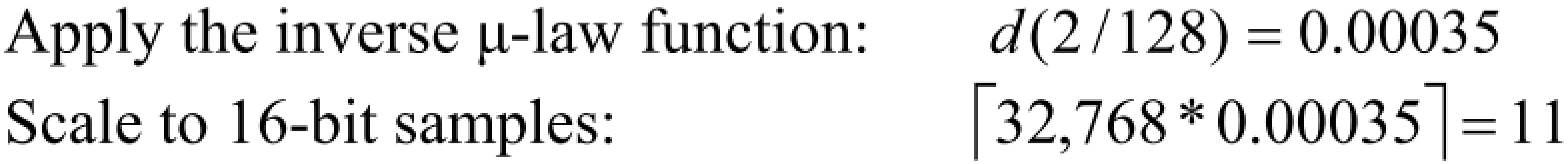


Let x be a µ-law encoded sample normalized so that $-1 \leq x < 1$. Let $sign(x) = -1$ if x is negative and $sign(x) = 1$ otherwise. Then the ***inverse µ-law function*** is defined by

$$d(x) = sign(x)\left(\frac{(\mu+1)^{|x|}-1}{\mu}\right)$$

$$= sign(x)\left(\frac{256^{|x|}-1}{255}\right) \quad for \;\; \mu = 255$$

The μ-law function, scaled to an 8-bit scale, yielded a value of 2. Reversing the process, we do the following:

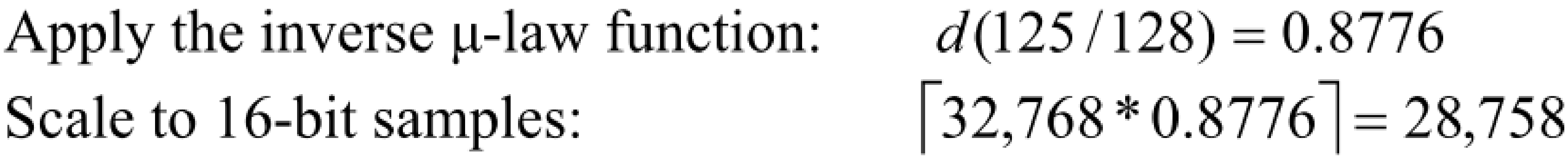


Apply the inverse μ-law function: $d(2/128) = 0.00035$

Scale to 16-bit samples: $\lceil 32{,}768 * 0.00035 \rceil = 11$

Error?

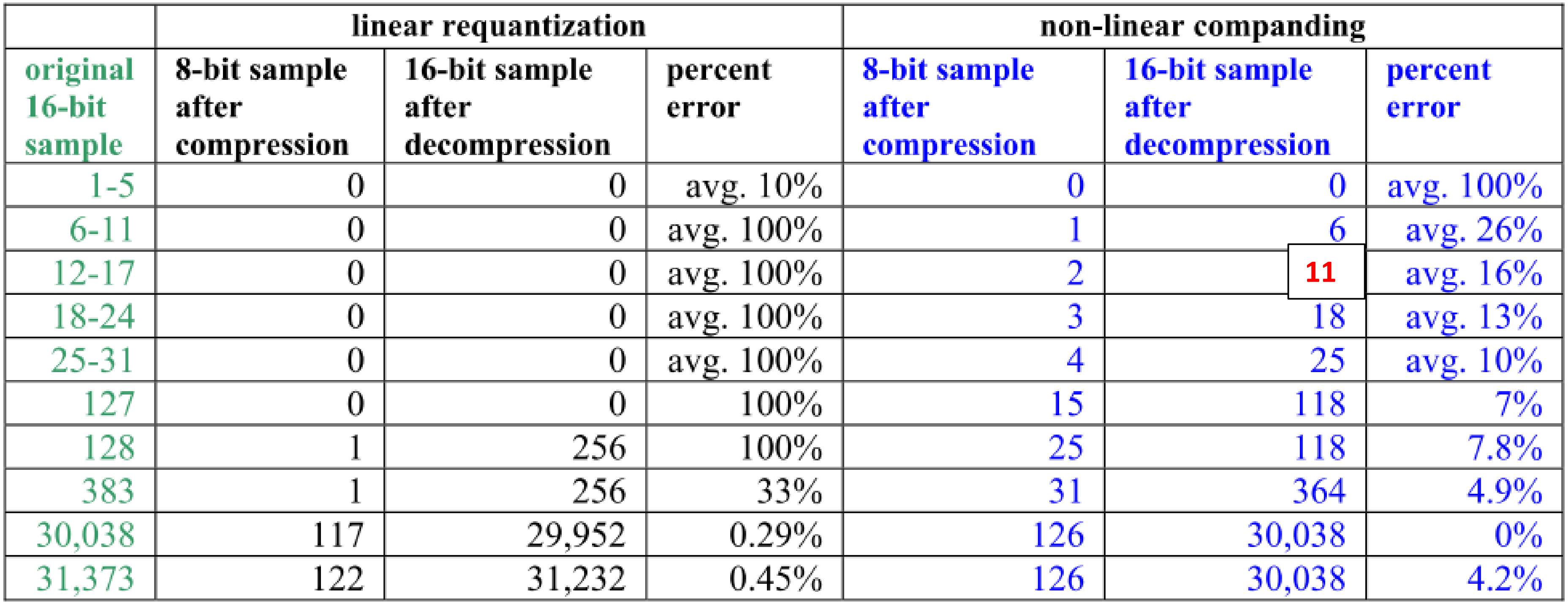
We can do the same computation for the sample that originally was 30,037 and yielded a value of 125 from the μ-law function.



Apply the inverse μ-law function: $d(125/128) = 0.8776$

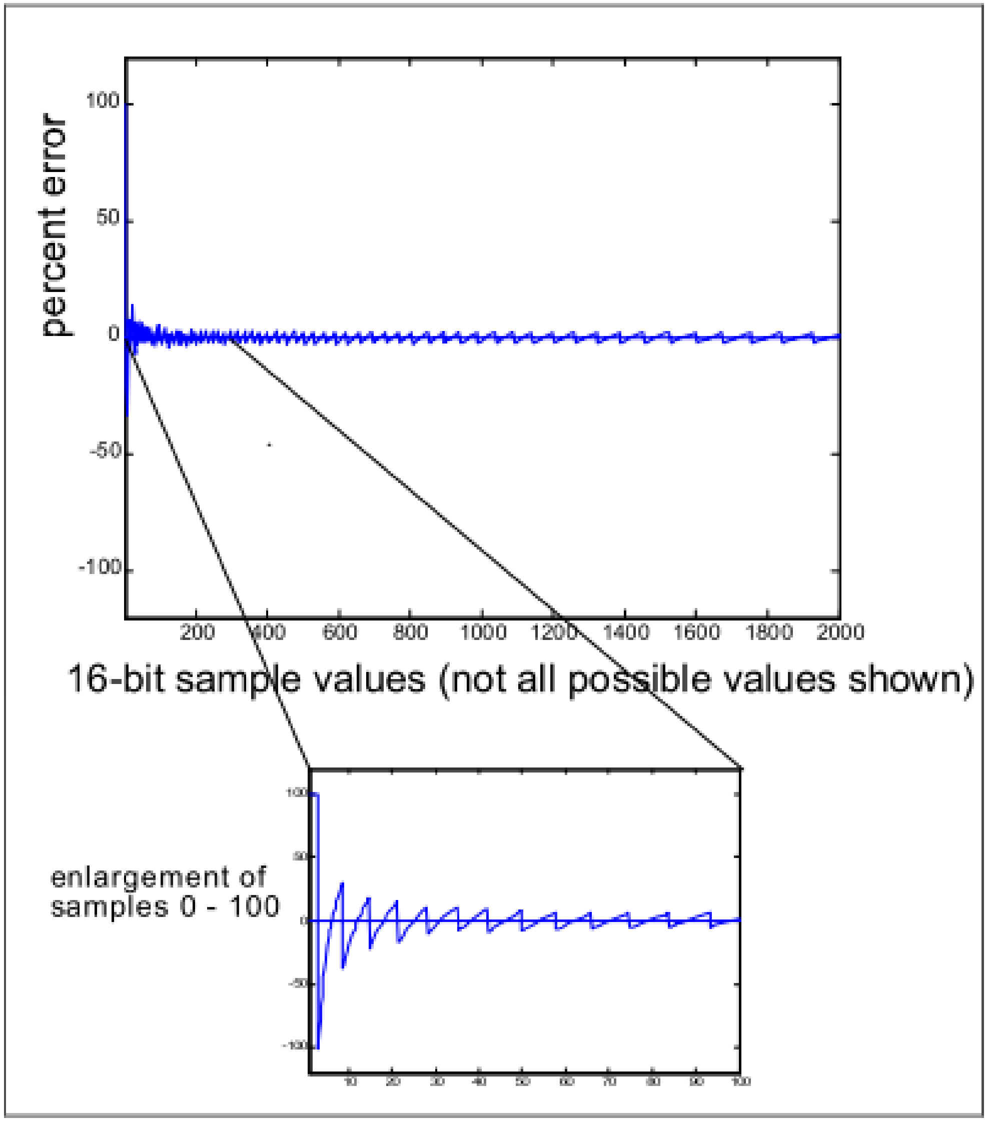
Scale to 16-bit samples: $\lceil 32{,}768 * 0.8776 \rceil = 28{,}758$

Error?

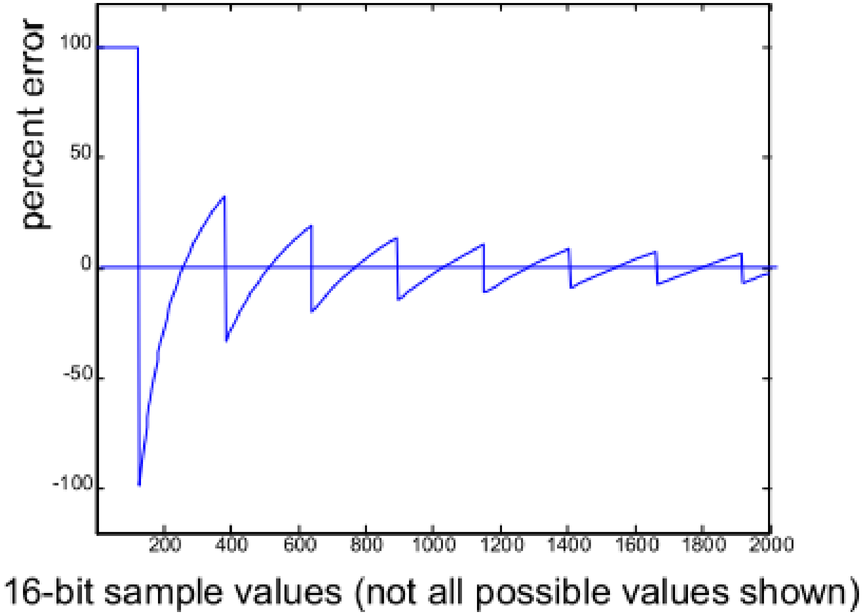


	linear requantization			non-linear companding		
original 16-bit sample	8-bit sample after compression	16-bit sample after decompression	percent error	8-bit sample after compression	16-bit sample after decompression	percent error
1-5	0	0	avg. 10%	0	0	avg. 100%
6-11	0	0	avg. 100%	1	6	avg. 26%
12-17	0	0	avg. 100%	2	11	avg. 16%
18-24	0	0	avg. 100%	3	18	avg. 13%
25-31	0	0	avg. 100%	4	25	avg. 10%
127	0	0	100%	15	118	7%
128	1	256	100%	25	118	7.8%
383	1	256	33%	31	364	4.9%
30,038	117	29,952	0.29%	126	30,038	0%
31,373	122	31,232	0.45%	126	30,038	4.2%

Comparison of error with linear requantization vs. non-linear companding (representative values only)



Percent error with non-linear companding



Percent error with linear quantization

Statistical Analysis of an Audio File

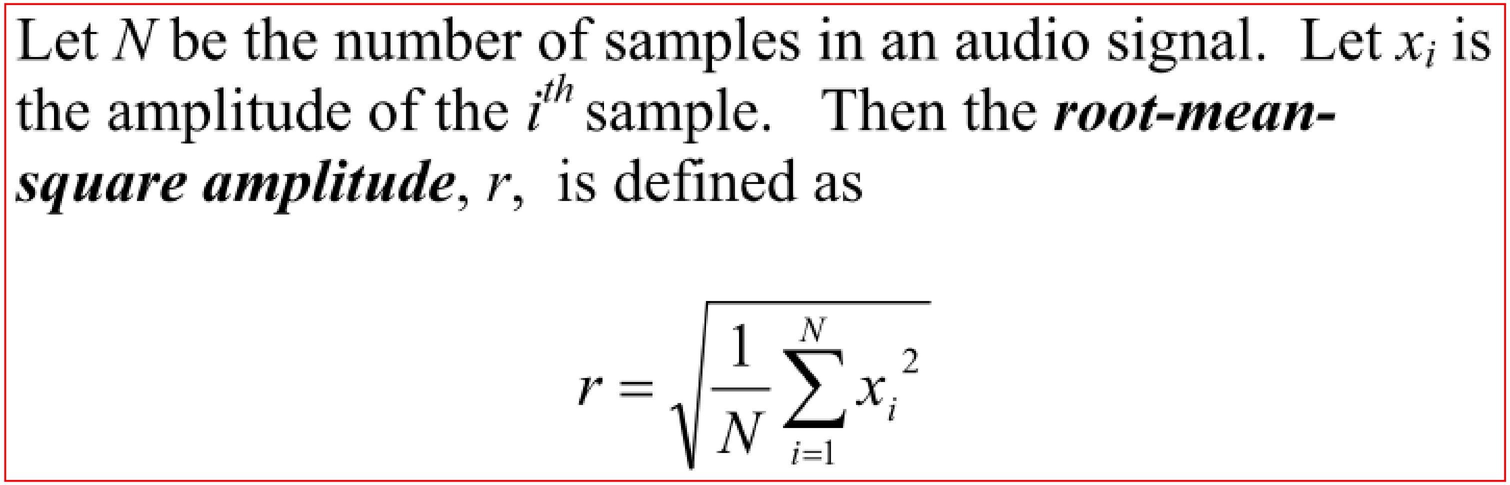
- analyze sample values in the time domain
- The statistical analysis may include the minimum and maximum possible sample values; the peak amplitude in the file; the number of clipped samples; the DC offset; the total, minimum, maximum, and average root-mean-square (RMS) amplitude; and a histogram.

DC offset

- No matter how complex a sound wave is, you can decompose it into frequency components.
- Each frequency component is a pure sinusoidal wave that is centered on the horizontal axis.
- However, analog to digital conversion is not a perfect process, and it can happen that the frequency components of the sampled waveform are not perfectly centered at 0.
- It may affect certain audio processing steps, particularly those based on finding places where the waveform crosses 0, called *zero-crossings.*

RMS amplitude (***RMS power*** or ***RMS level***)

- a measure of the average amplitude in the sound wave over a given period –either over the entire sound wave or over a portion of it

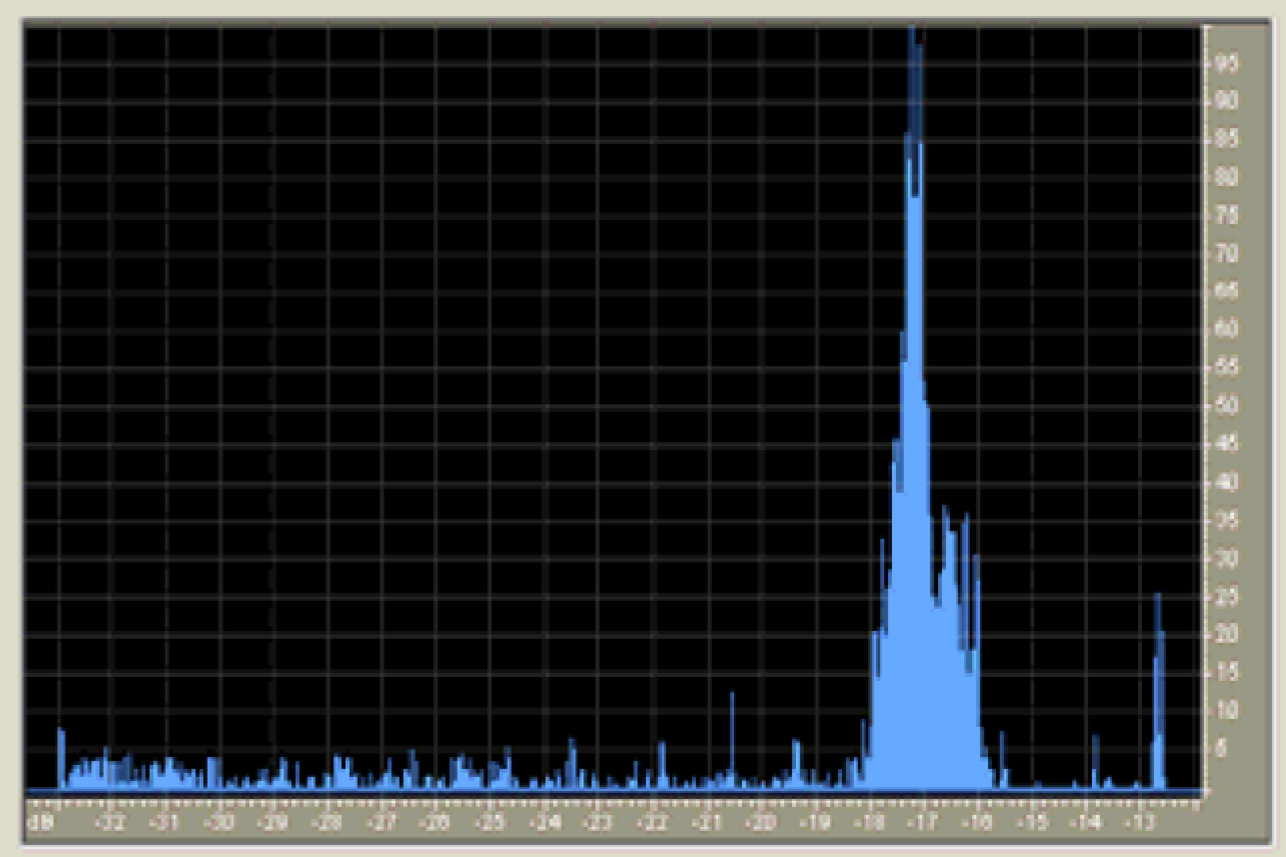


Let N be the number of samples in an audio signal. Let x_i is the amplitude of the i^{th} sample. Then the ***root-mean-square amplitude***, r, is defined as

$$r = \sqrt{\frac{1}{N}\sum_{i=1}^{N} x_i^{\,2}}$$

Average RMS amplitude – average RMS amplitude for all windows of the specified size

An ***audio histogram***– number of samples at each amplitude level in the audio selection.



Audio histogram

Psychoacoustics

- The range of human hearing is about 20 Hz to about 20 kHz

- The frequency range of the voice is typically only from about 500 Hz to 4 kHz

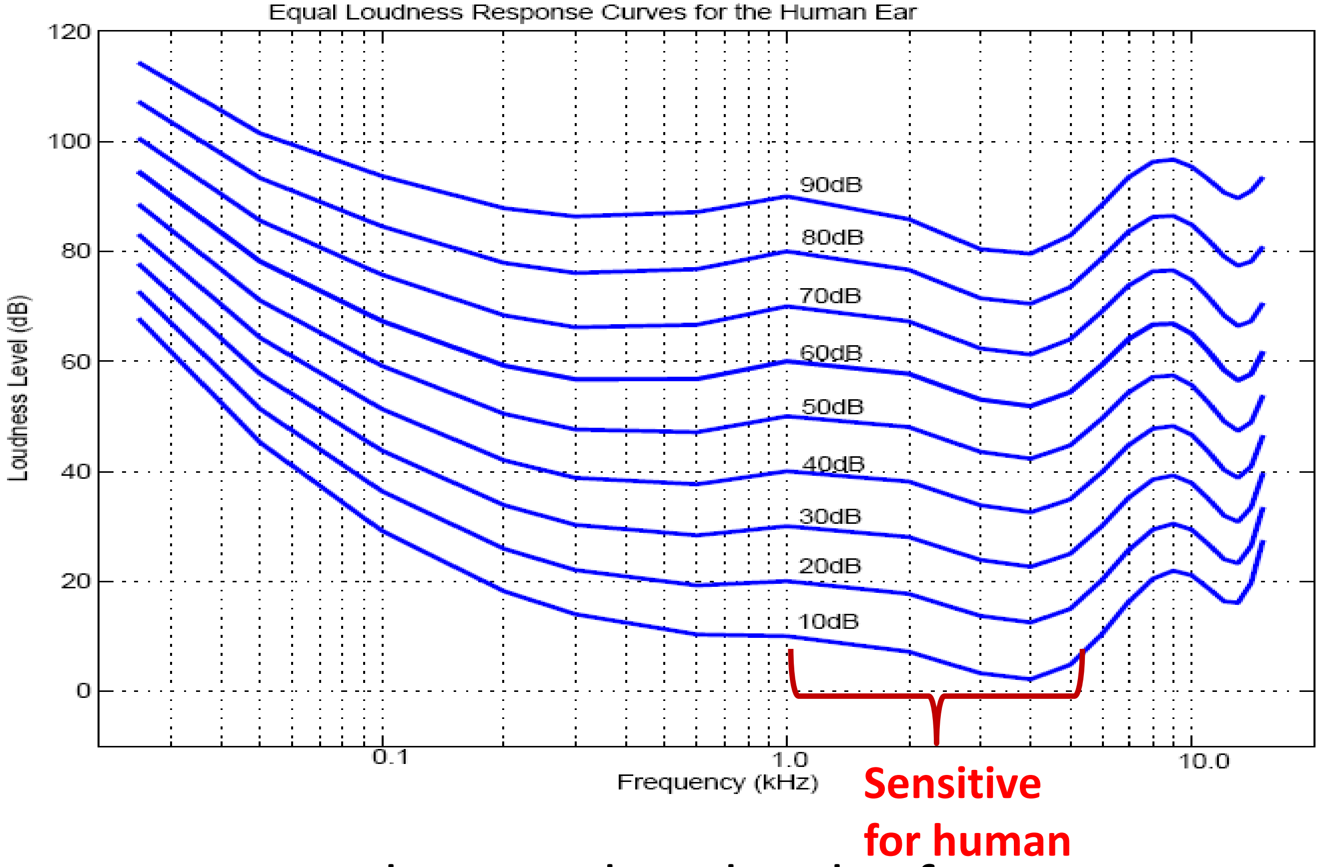
- The dynamic range, the ratio of the maximum sound amplitude to the quietest sound that humans can hear, is on the order of about 120 dB

Equal-Loudness Relations

- **Fletcher-Munson Curves**
  - Equal loudness curves that display the relationship between perceived loudness (“Phons”, in dB) for a given stimulus sound volume (“Sound Pressure Level”, in dB), as a function of frequency

Equal-Loudness Relations

Flaetcher-Munson Curves (re-measured by Robinson and Dadson)



– The bottom curve shows what level of pure tone stimulus is required to produce the perception of a 10 dB sound

– All the curves are arranged so that the perceived loudness level gives the same loudness as for that loudness level of a pure tone at 1 kHz

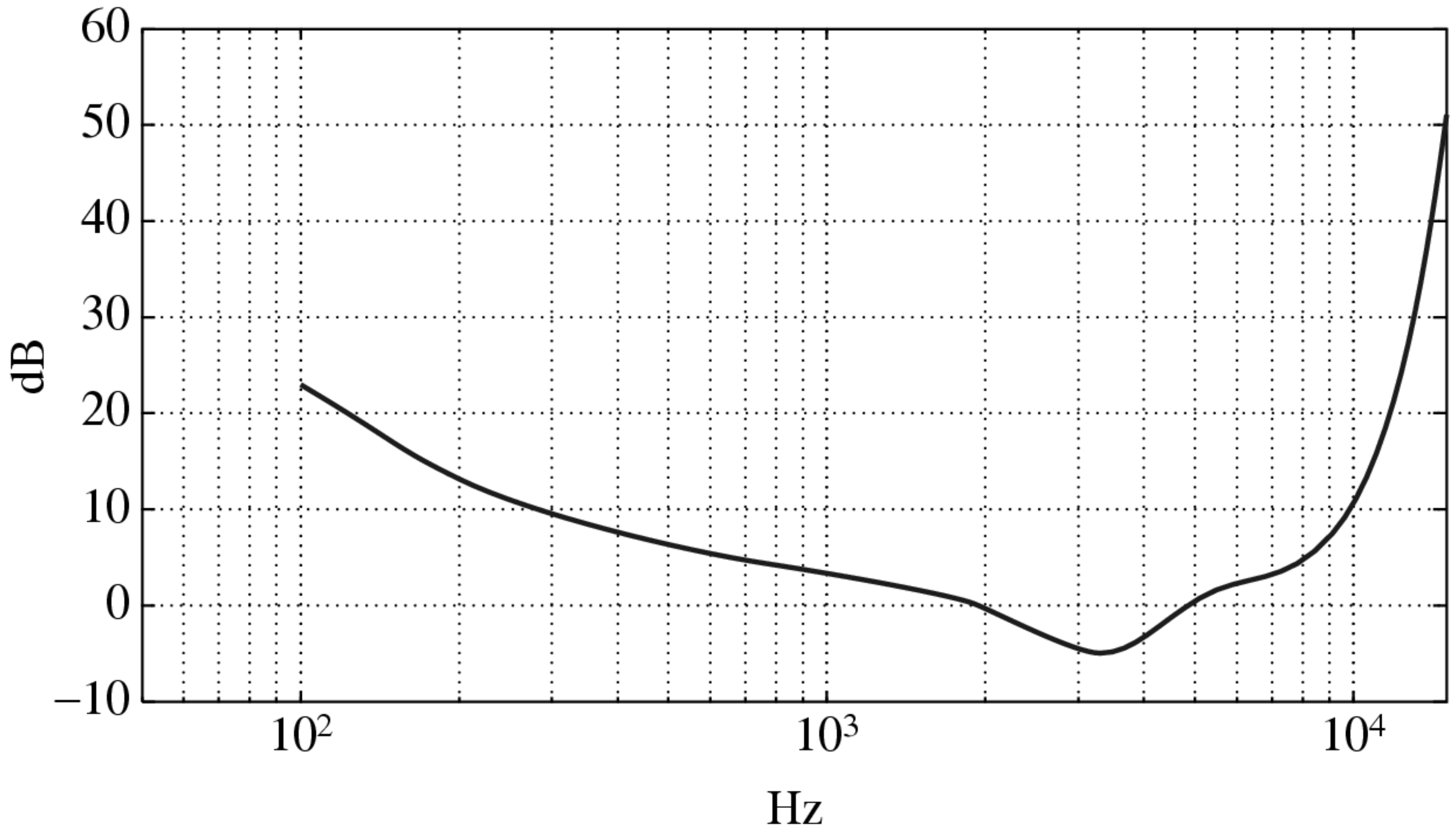
Frequency Masking

- Lossy audio data compression methods, such as MPEG/Audio encoding, remove some sounds which are masked anyway
- The general situation in regard to masking is as follows:

1. A lower tone can effectively mask (make us unable to hear) a higher tone
2. The reverse is not true – a higher tone does not mask a lower tone well
3. The greater the power in the masking tone, the wider is its influence – the broader the range of frequencies it can mask.
4. As a consequence, if two tones are widely separated in frequency then little masking occurs

Threshold of Hearing

A plot of the threshold of human hearing for a pure tone



Threshold of human hearing, for pure tones

Threshold of Hearing (cont'd)

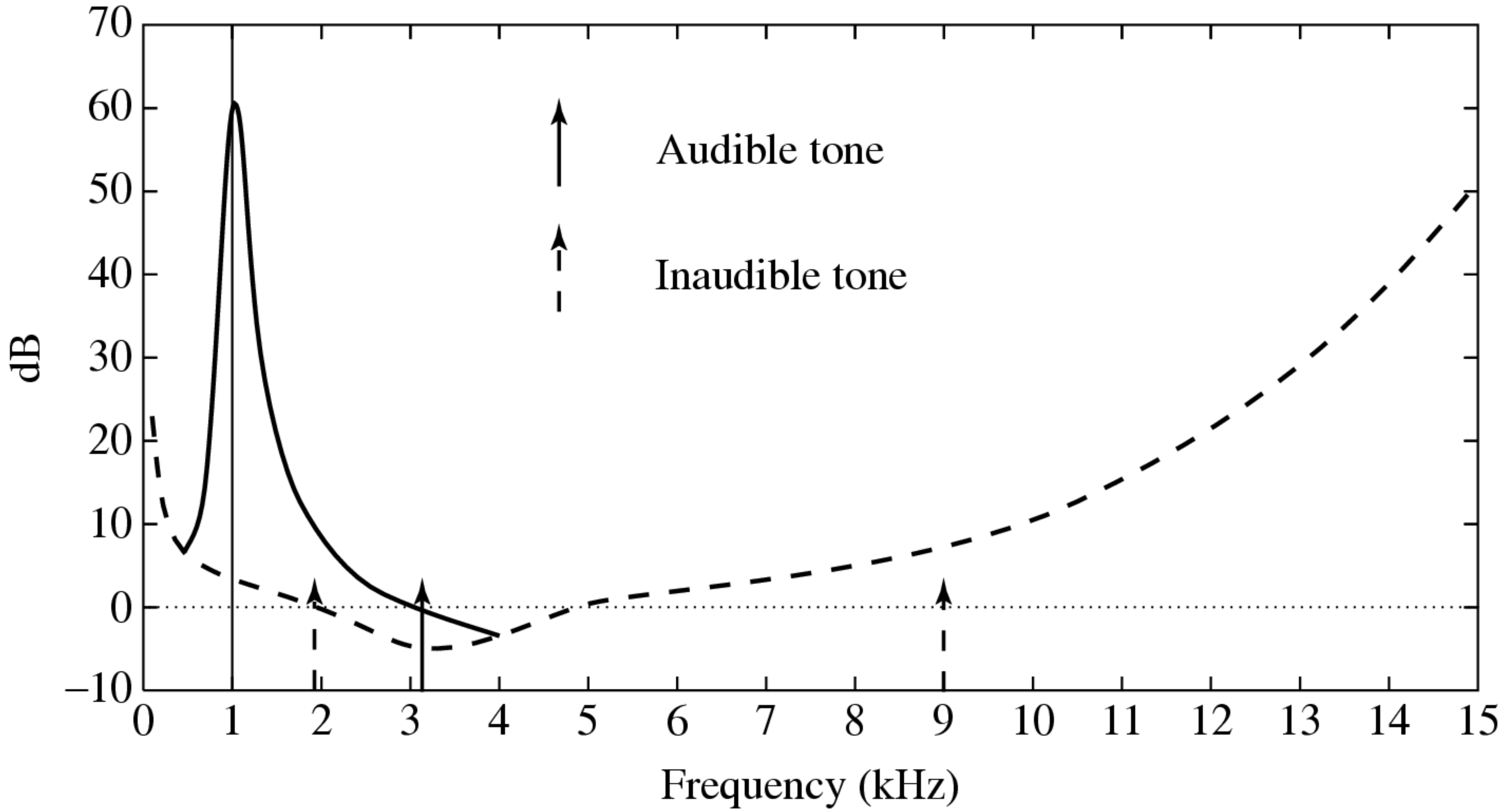
- The threshold of hearing curve: if a sound is above the dB level shown then the sound is audible
- Turning up a tone so that it equals or surpasses the curve means that we can then distinguish the sound
- An approximate formula exists for this curve:

$$\text{Threshold}(f) = 3.64(f/1000)^{-0.8} - 6.5e^{-0.6(f/1000-3.3)^2} + 10^{-3}(f/1000)^4$$

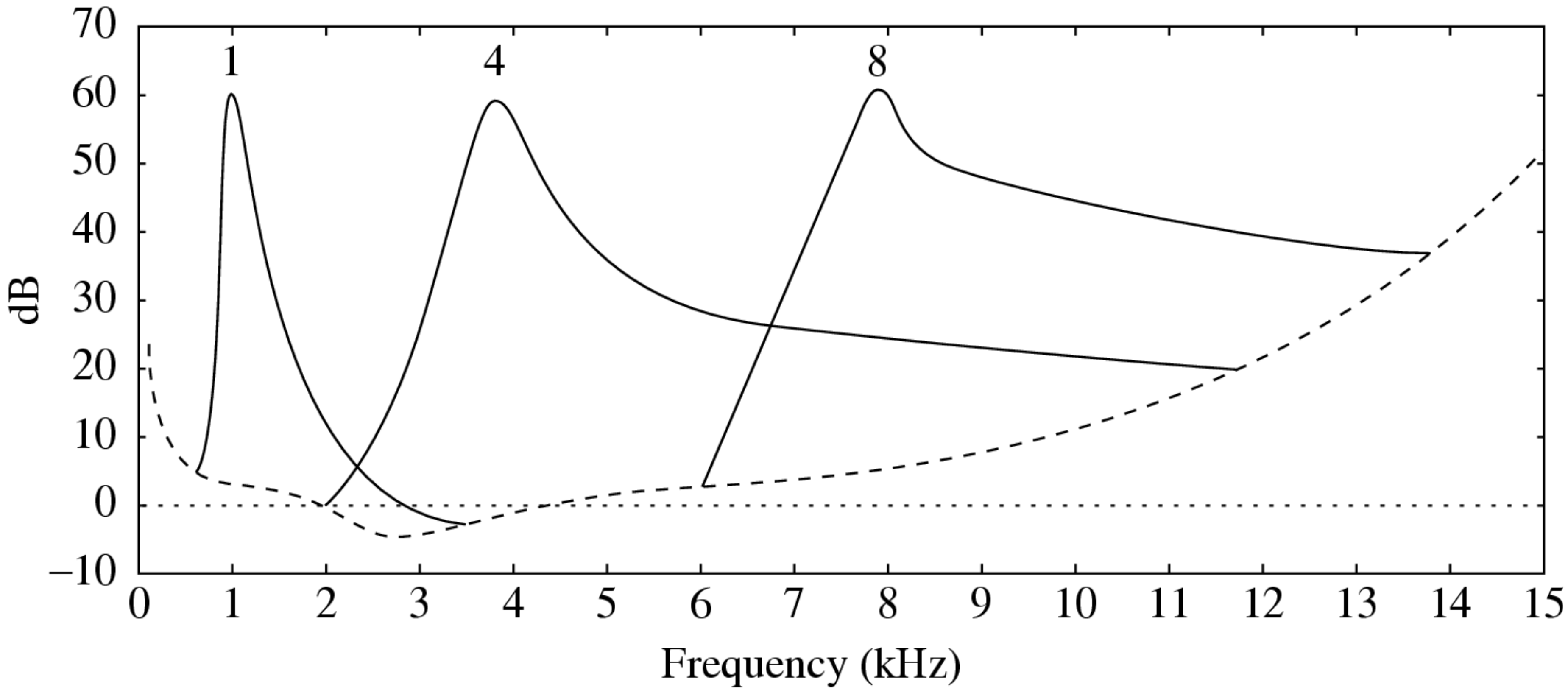
- The threshold units are dB; the frequency for the origin (0,0) in the formula is 2,000 Hz: Threshold(*f*) = 0 at *f* =2 kHz

Frequency Masking Curves

- Frequency masking is studied by playing a particular pure tone, say 1 kHz again, at a loud volume, and determining how this tone affects our ability to hear tones nearby in frequency
  - – one would generate a 1 kHz *masking* tone, at a fixed sound level of 60 dB, and then raise the level of a nearby tone, e.g., 1.1 kHz, until it is just audible



Effect on threshold for 1 kHz masking tone



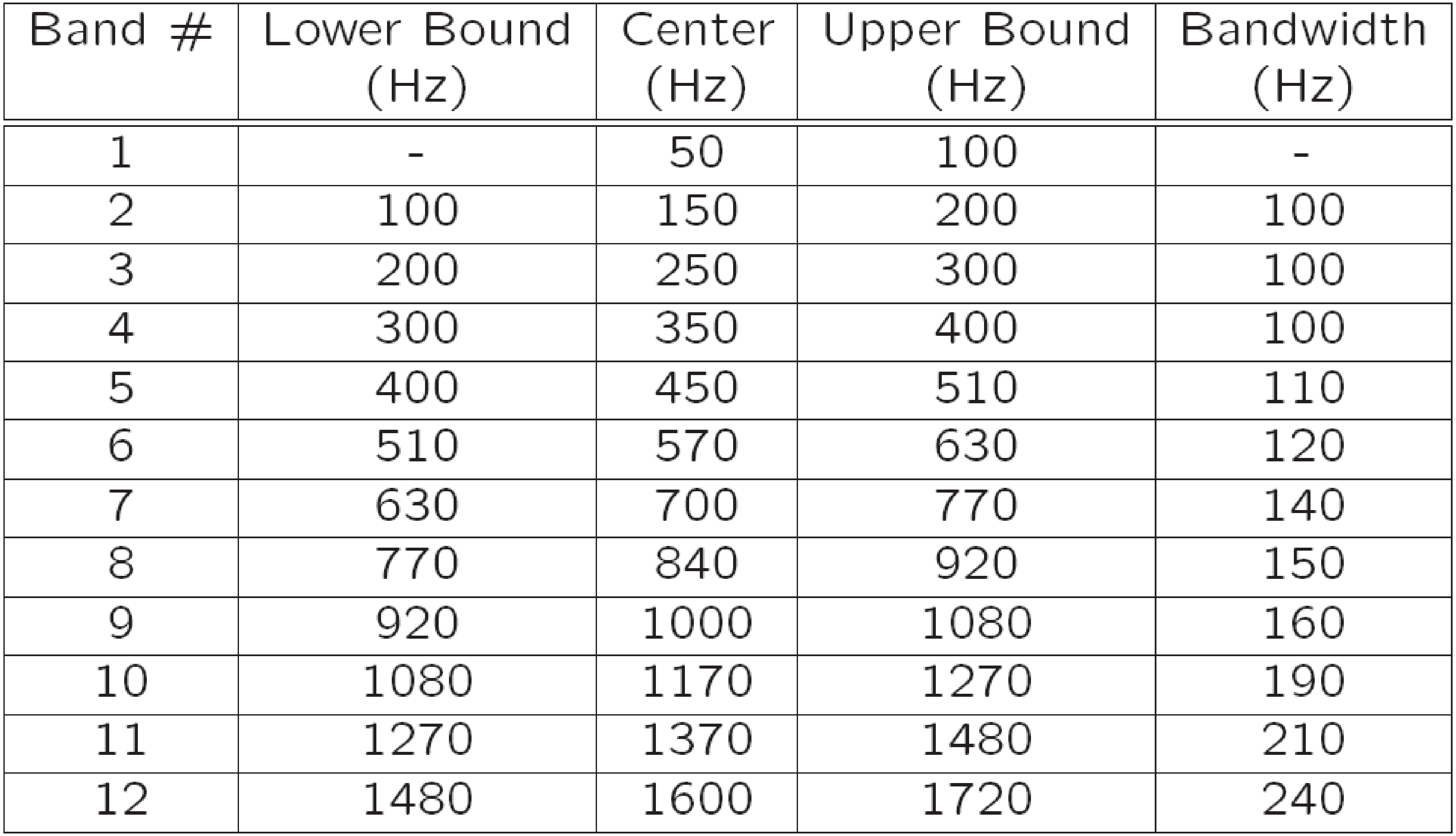
The higher the frequency of the masking tone, the broader a range of influence it has.

Effect of masking tone at three different frequencies

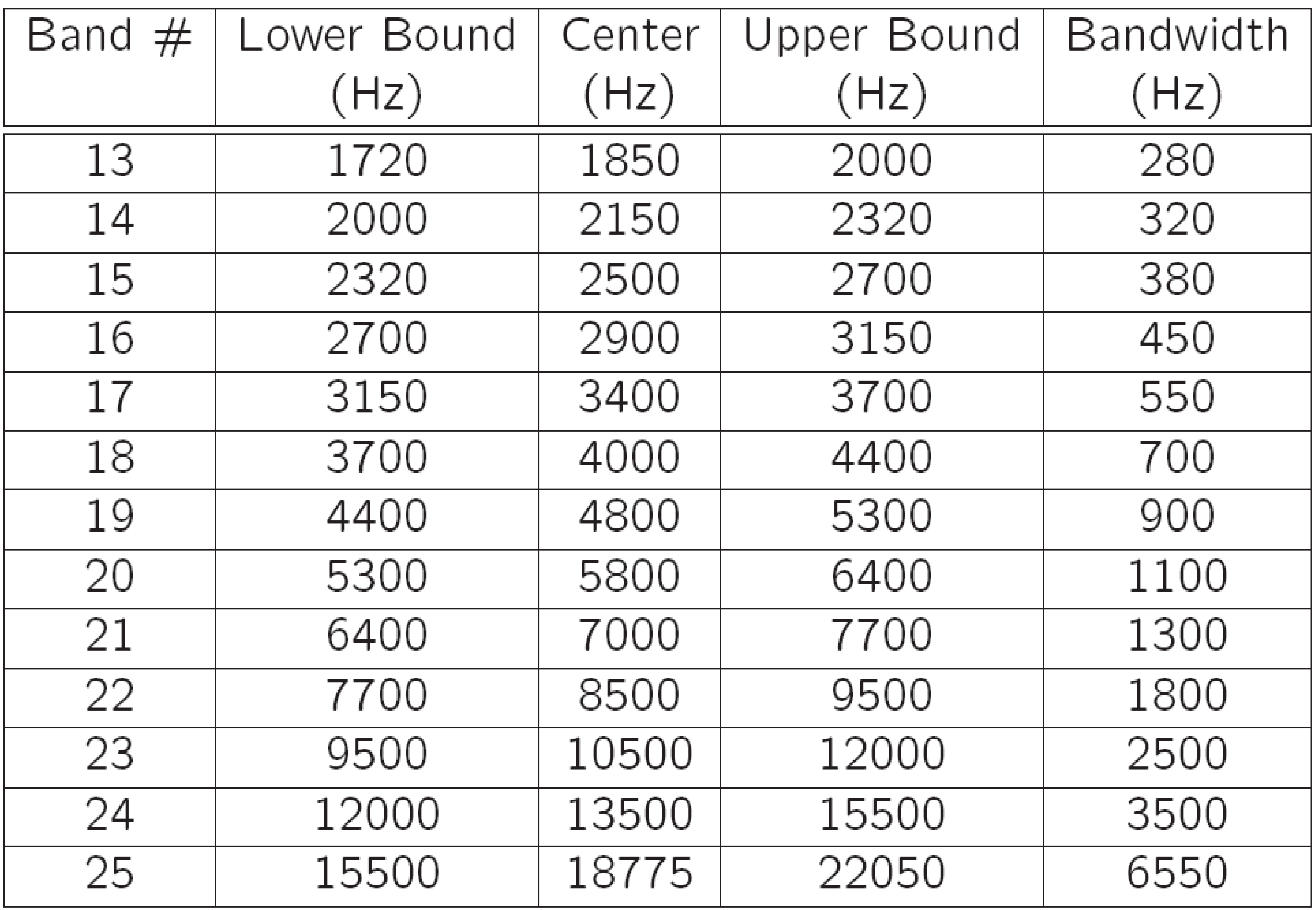
Critical Bands

- human auditory system cannot resolve sounds better than within about one *critical band* when other sounds are present
- **Critical bandwidth** represents the ear's resolving power for simultaneous tones or partials
  - Experiments indicate that At the low-frequency end, a critical band is less than 100 Hz wide, while for high frequencies the width can be greater than 4 kHz (*perceptual nonuniformity*)
- the critical bandwidth:
  - for masking frequencies < 500 Hz: remains approximately constant in width (about 100 Hz)
  - for masking frequencies > 500 Hz: increases approximately linearly with frequency

25-Critical Bands and Bandwidth



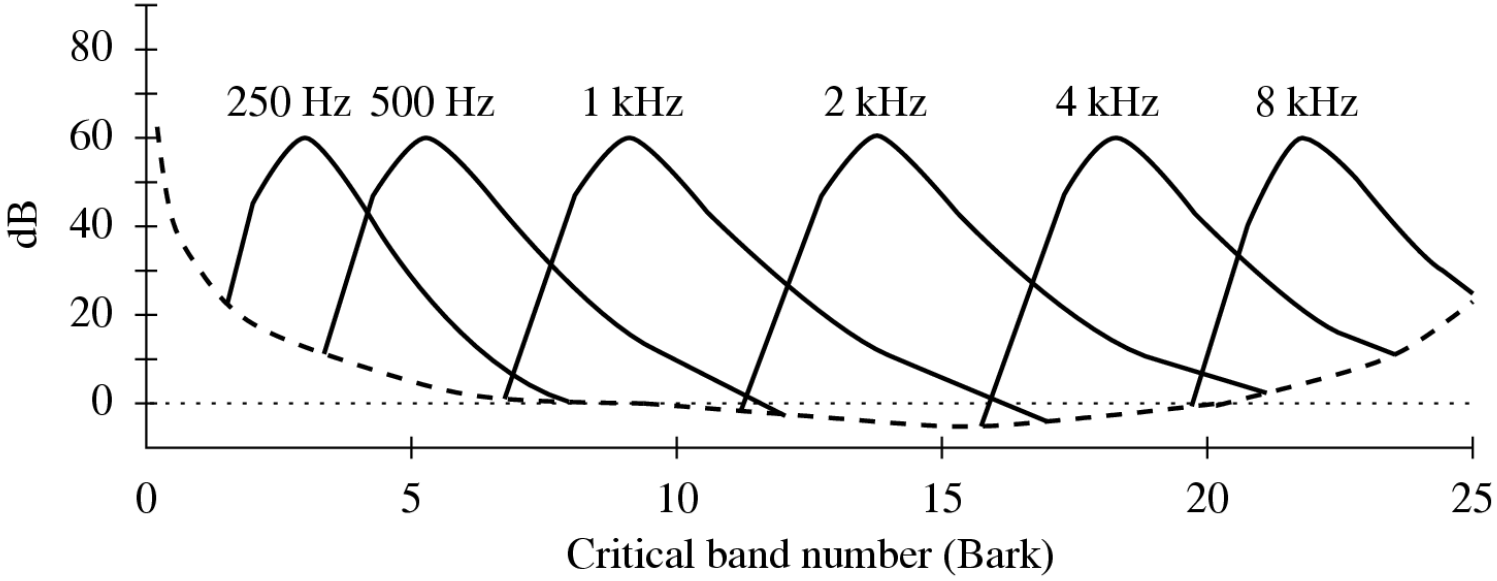
Band #	Lower Bound (Hz)	Center (Hz)	Upper Bound (Hz)	Bandwidth (Hz)
1	-	50	100	-
2	100	150	200	100
3	200	250	300	100
4	300	350	400	100
5	400	450	510	110
6	510	570	630	120
7	630	700	770	140
8	770	840	920	150
9	920	1000	1080	160
10	1080	1170	1270	190
11	1270	1370	1480	210
12	1480	1600	1720	240



Band #	Lower Bound (Hz)	Center (Hz)	Upper Bound (Hz)	Bandwidth (Hz)
13	1720	1850	2000	280
14	2000	2150	2320	320
15	2320	2500	2700	380
16	2700	2900	3150	450
17	3150	3400	3700	550
18	3700	4000	4400	700
19	4400	4800	5300	900
20	5300	5800	6400	1100
21	6400	7000	7700	1300
22	7700	8500	9500	1800
23	9500	10500	12000	2500
24	12000	13500	15500	3500
25	15500	18775	22050	6550

Bark Unit

- **Bark unit** is defined as the width of one critical band, for any masking frequency
- The idea of the Bark unit: every critical band width is roughly equal in terms of Barks



Effect of masking tones, expressed in Bark units

Conversion: Frequency & Critical Band Number

- **Conversion** expressed in the Bark unit:

$$\text{Critical band number (Bark)} = \begin{cases} f/100, & \text{for } f < 500, \\ 9 + 4\log_2(f/1000), & \text{for } f \geq 500. \end{cases}$$

- **Another formula** used for the Bark scale:

$$b = 13.0 \arctan(0.76\, f) + 3.5 \arctan(f^2/56.25)$$

where *f* is in kHz and *b* is in Barks (the same applies to all below)

- **The inverse equation:**

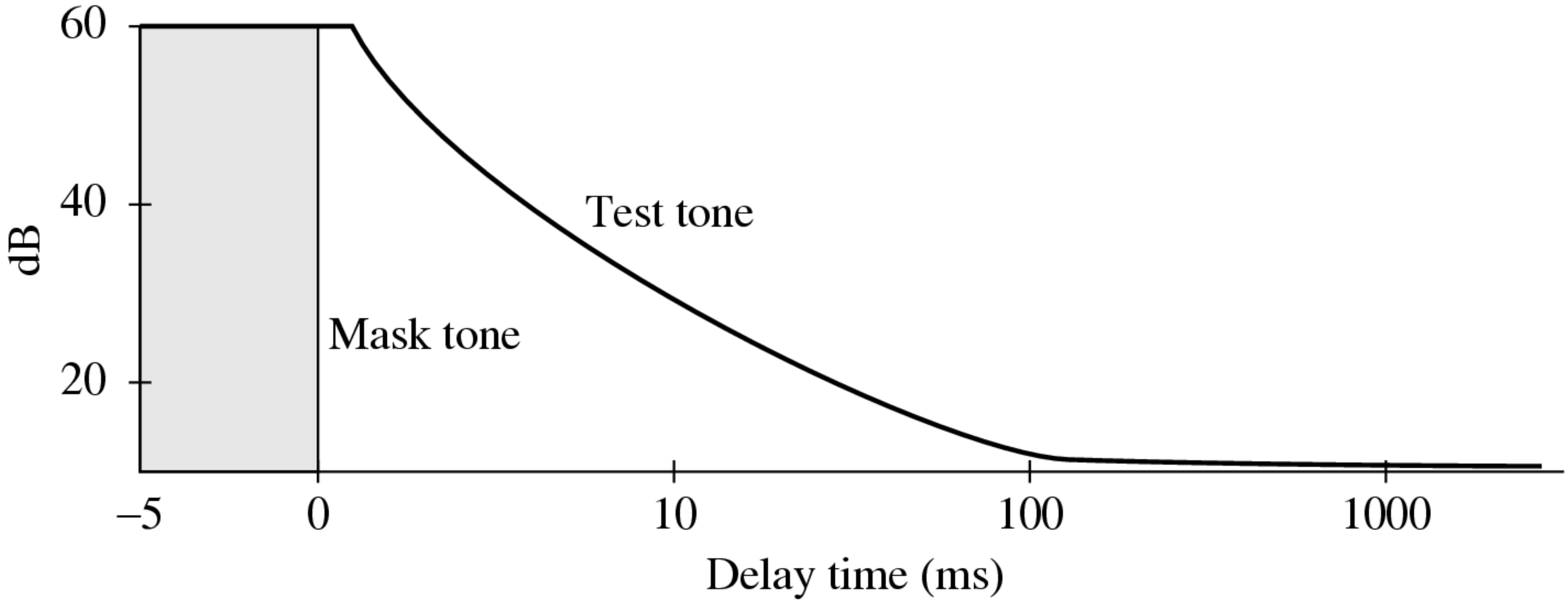
$$f = [(\exp(0.219 * b)/352) + 0.1] * b - 0.032 * \exp[-0.15 * (b-5)^2]$$

The **critical bandwidth (*df*)** for a given center frequency *f* can also be approximated by:

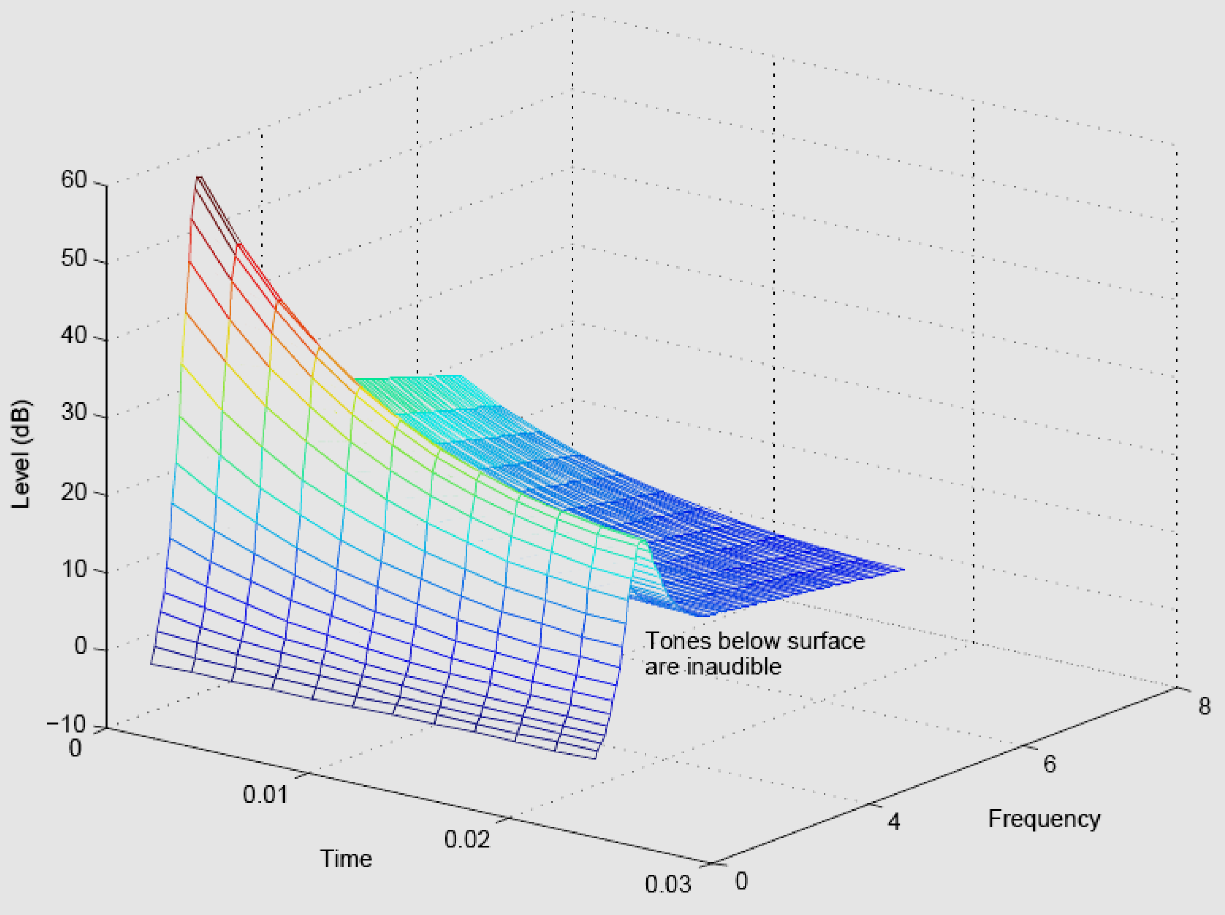
$$df = 25 + 75 \times [1 + 1.4(f^2)]^{0.69}$$

Temporal Masking

Phenomenon: any loud tone will cause the hearing receptors in the inner ear to become *saturated* and require time to recover

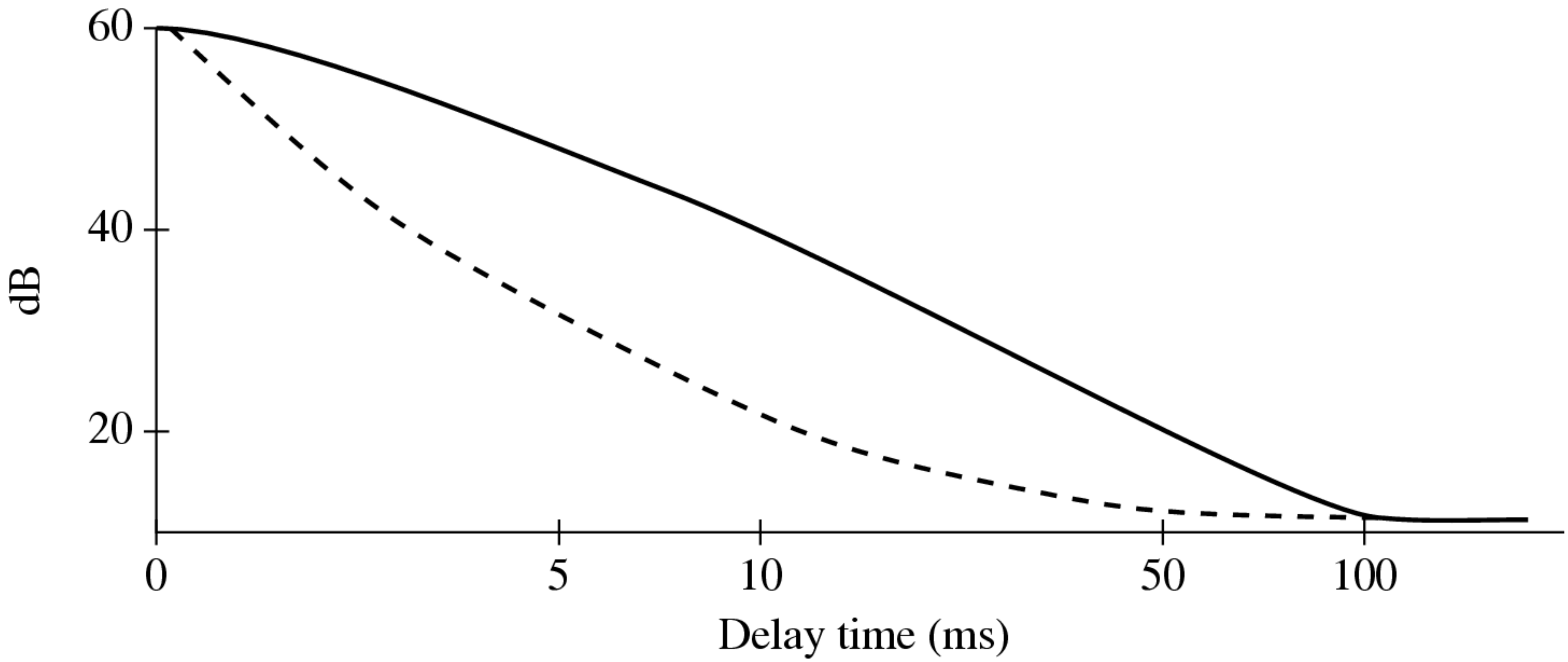


The louder is the test tone, the shorter it takes for our hearing to get over hearing the masking.



The closer the freq. to the masking tone and the closer in time to when the masking tone is stopped, the greater likelihood that a test tone cannot be heard.

Effect of temporal and frequency masking depending on both time and closeness in frequency.



For a masking tone that is ***played for a longer time, it takes longer*** before a test tone can be heard. Solid curve: masking tone played for 200 msec; dashed curve: masking tone played for 100 msec.

MPEG Audio

- **MPEG audio compression** takes advantage of psychoacoustic models, constructing a large multi-dimensional lookup table to transmit masked frequency components using fewer bits
- **MPEG Audio Overview**

  1. Applies a filter bank to the input to break it into its frequency components
  2. In parallel, a psychoacoustic model is applied to the data for bit allocation block
  3. The number of bits allocated are used to quantize the info from the filter bank – providing the compression

MPEG Layers

- MPEG audio offers three compatible layers (1-3):
  - Each succeeding layer able to understand the lower layers

  (same header info)
  - Each succeeding layer offering more complexity in the psychoacoustic model and better compression for a given level of audio quality
  - each succeeding layer, with increased compression effectiveness, accompanied by extra delay

- The objective of MPEG layers: a good tradeoff between quality and bit-rate

MPEG Layers (cont'd)

- Layer 1 quality can be quite good provided a comparatively high bit-rate is available
  - Digital Audio Tape typically uses Layer 1 at around 192 kbps
- Layer 2 has more complexity; was proposed for use in Digital Audio Broadcasting
- Layer 3 (MP3) is most complex, and was originally aimed at audio transmission over ISDN lines
- Most of the **complexity increase is at the encoder**, not the decoder – accounting for the popularity of MP3 players

MPEG Audio Strategy

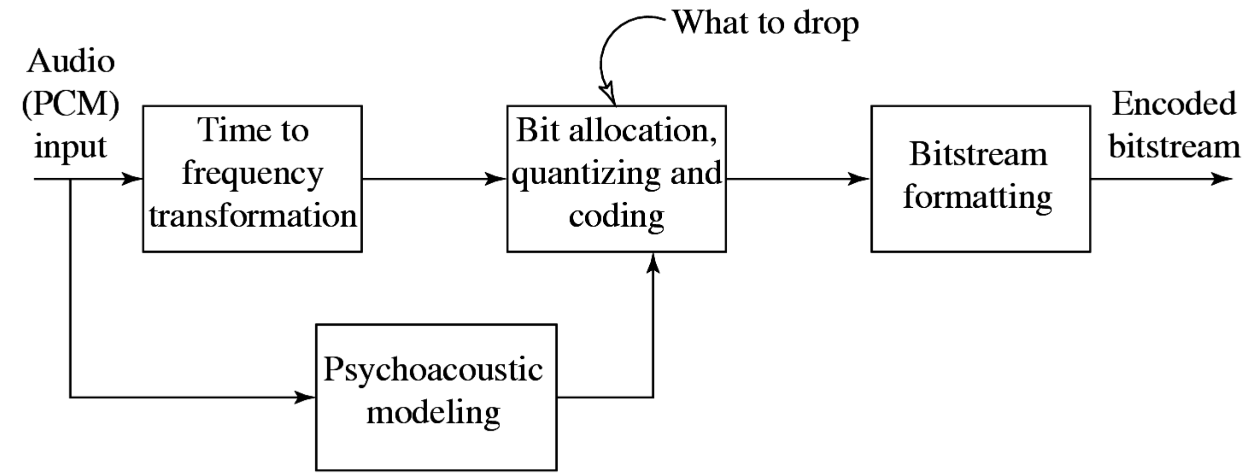
- **MPEG approach to compression** relies on:
  - – Quantization
  - – Human auditory system is not accurate within the width of a critical band (perceived loudness and audibility of a test frequency)
- **MPEG encoder** employs a bank of filters to:
  - – Analyze the frequency ("spectral") components of the audio signal by calculating a frequency transform of a window of signal values
  - – Decompose the signal into subbands (Layer 1 & 2 codec: "***quadrature-mirror filter bank***"; Layer 3 codec: adds a **DCT; psychoacoustic model**: Fourier transform)

MPEG Audio Strategy (cont’d)

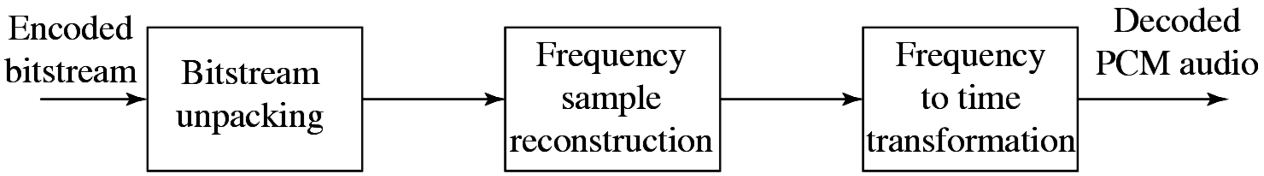
- **Frequency masking**: by using a psychoacoustic model to estimate the just noticeable noise level:
  - Encoder balances the masking behavior and the available number of bits by discarding inaudible frequencies
  - Scaling quantization according to the sound level that is left over, above masking levels
- May take into account the actual width of the critical bands:
  - For practical purposes, audible frequencies are divided into 25 main critical bands
  - To keep simplicity, adopts a ***uniform*** width for all frequency analysis filters, using **32 overlapping** subbands

- Layer 1 – the psychoacoustic model uses only frequency masking. Bitrates: 32 kbps (mono) to 448 kbps (stereo). Near-CD stereo quality is possible with a bitrate of 256-384 kbps.
- Layer 2 – temporal masking by accumulating more samples and examining temporal masking between the current block of samples and the ones just before and just after. Bitrates: 32-192 kbps (mono) and 64-384 kbps (stereo). Stereo CD-audio (192-256 kbps).
- Layer 3 – lower bitrate applications; more sophiscated subband analysis, with nonuniform subband widths. Also nonuniform quantization and entropy coding. Bitrates: 32-320 kbps.

MPEG Audio Compression Algorithm



(a) MPEG Audio Encoder

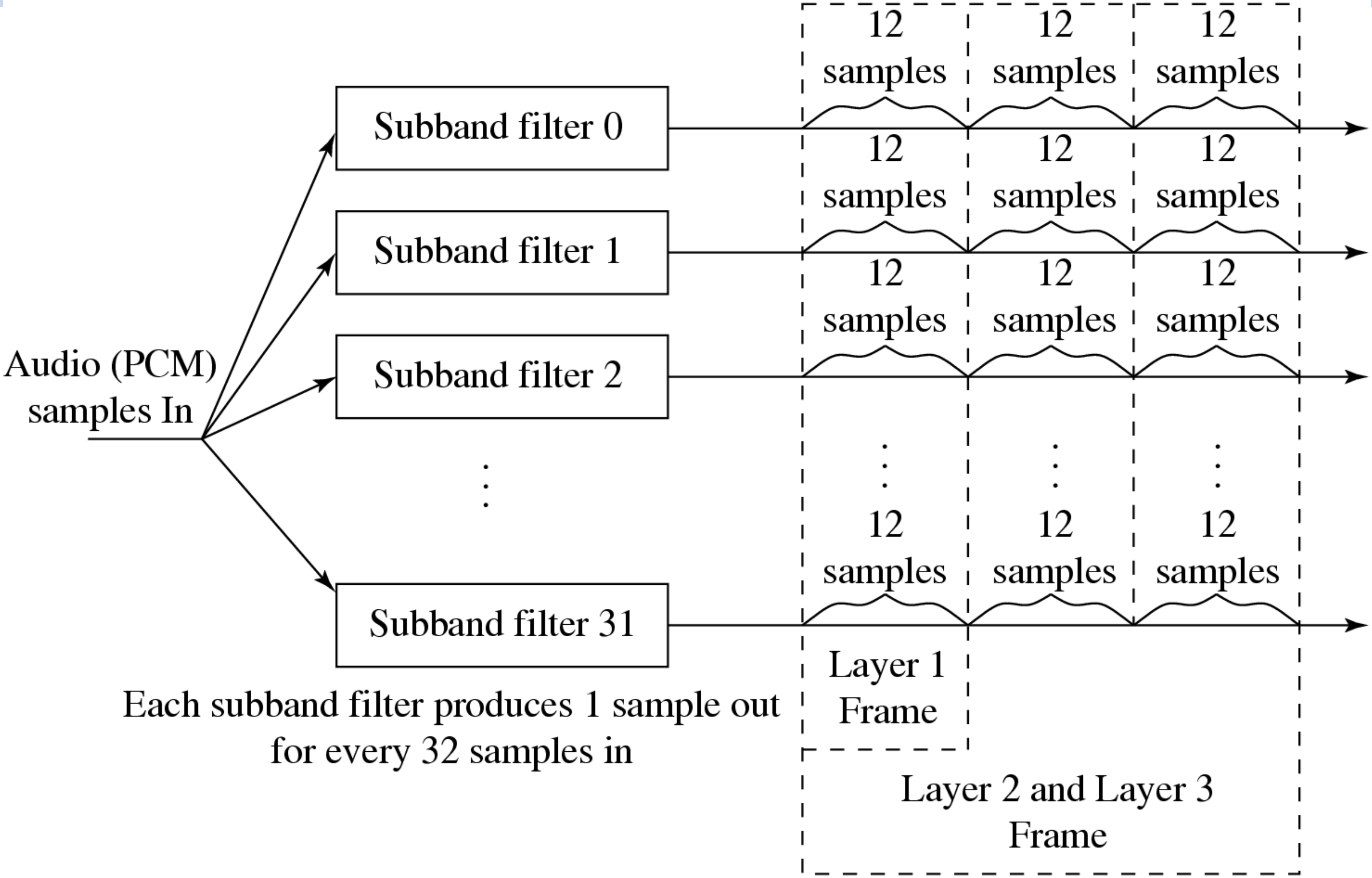

Encoded bitstream
Bitstream unpacking
Frequency sample reconstruction
Frequency to time transformation
Decoded PCM audio

(b) MPEG Audio Decoder

Basic MPEG Audio encoder and decoder.

Basic Algorithm (cont'd)

- The algorithm proceeds by dividing the input into 32 frequency subbands, via a filter bank
  - A linear operation taking 32 PCM samples, sampled in time; output is 32 frequency coefficients
- In the Layer 1 encoder, the sets of 32 PCM values are first assembled into a set of **12 groups of 32s**
  - an inherent time lag in the coder, equal to the time to accumulate 384 (i.e., 12×32) samples
- figure shows how samples are organized
  - A Layer 2 or Layer 3, frame actually accumulates more than 12 samples for each subband: a frame includes 1,152 samples



Each subband filter produces 1 sample out for every 32 samples in

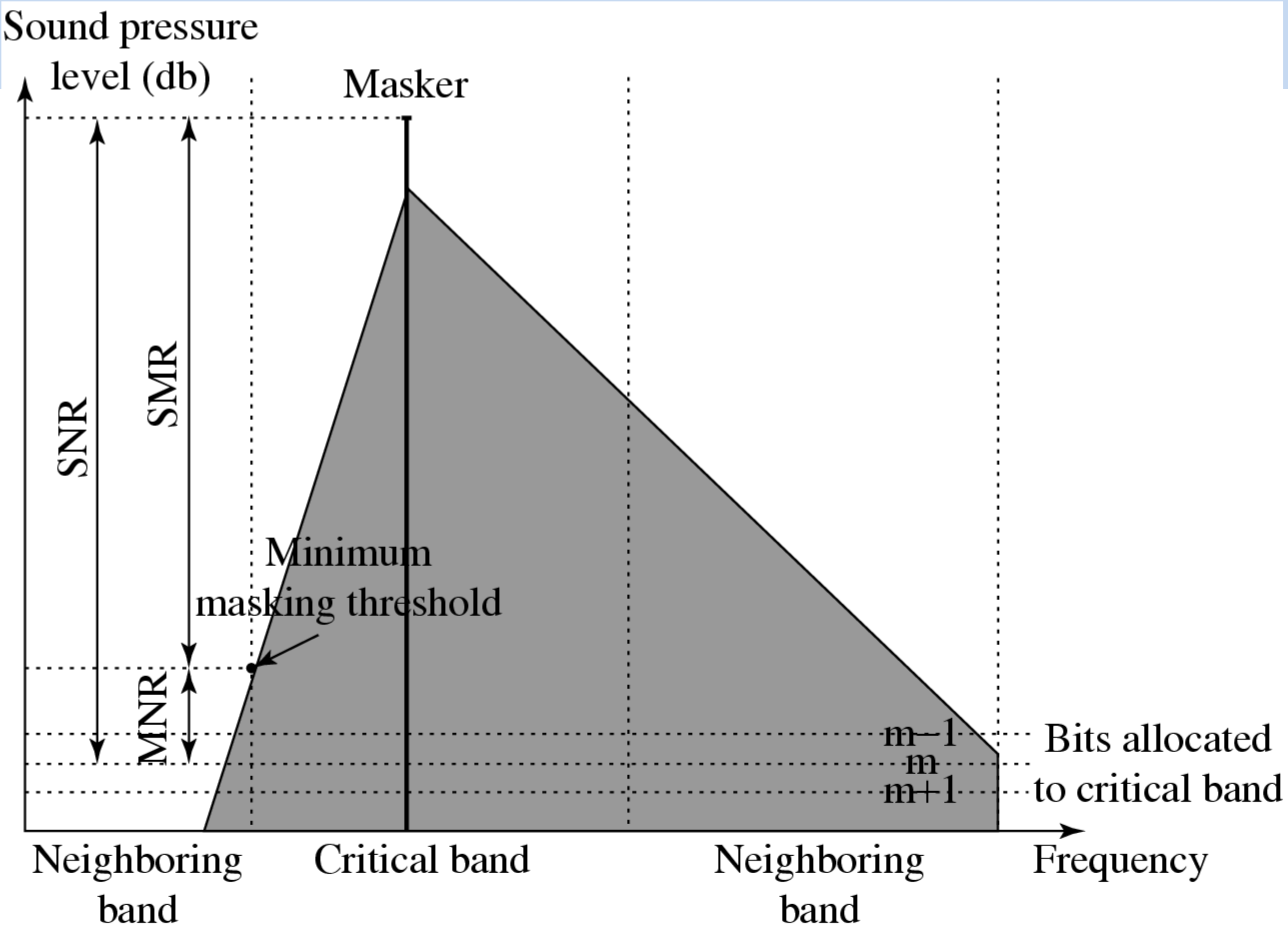
MPEG Audio Frame Sizes

Bit Allocation Algorithm

- **Aim**: ensure that all of the quantization noise is below the masking thresholds
- **One common scheme**:
  - For each subband, the psychoacoustic model calculates the *Signal-to-Mask Ratio* (SMR)in dB
  - Then the “Mask-to-Noise Ratio” (MNR) is defined as the difference:

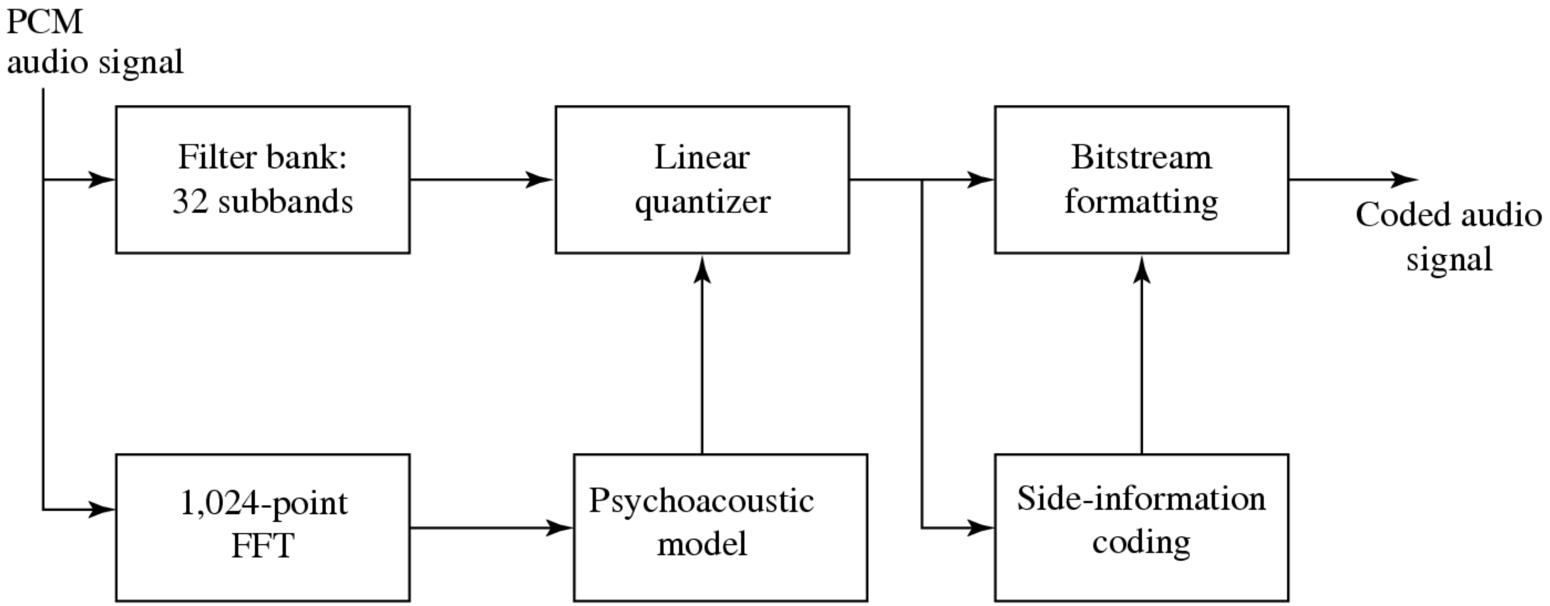
$$\mathrm{MNR}_{\mathrm{dB}} \equiv \mathrm{SNR}_{\mathrm{dB}} - \mathrm{SMR}_{\mathrm{dB}}$$

  - The lowest MNR is determined, and the number of code-bits allocated to this subband is incremented
  - Then a new estimate of the SNR is made, and the process iterates until there are no more bits to allocate


Sound pressure level (db)
Masker
SNR
SMR
Minimum masking threshold
MNR
m−1
m
m+1
Bits allocated to critical band
Neighboring band
Critical band
Neighboring band
Frequency

MNR and SMR. A qualitative view of SNR, SMR and MNR are shown, with one dominate masker and m bits allocated to a particular critical band.

- Mask calculations are performed in parallel with subband filtering.
- The masking curve calculation requires an accurate freq. decomposition of the input signal, using DFT.
- Layer 1 – 16 uniform quantizers are pre-calculated. The index of the quantizer is sent as 4 bits of side information for each subband. The maximum resolution of each quantizer is 15 bits.



MPEG-1 Audio Layers 1 and 2.

Layer 2 of MPEG-1 Audio

Main difference:

- Three groups of 12 samples are encoded in each frame and temporal masking is brought into play, as well as frequency masking
- The psychoacoustic model does better at modeling slowly-changing sound if the time window used is longer
- Bit allocation is applied to window lengths of 36 samples instead of 12
- The resolution of the quantizers is increased from 15 bits to 16

Advantage:

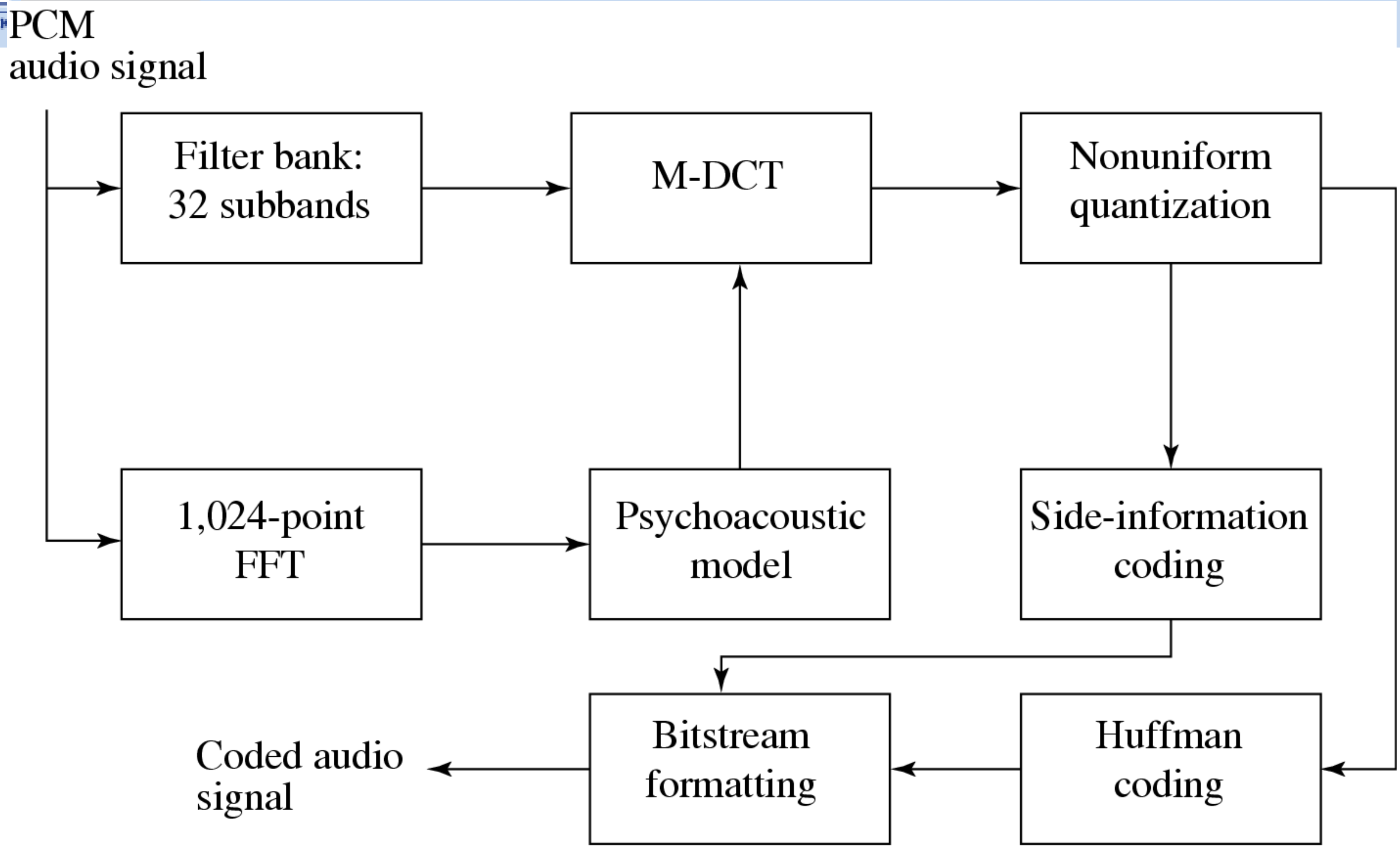
- a single scaling factor can be used for all three groups (before, current, next)

Layer 3 of MPEG-1 Audio

Main difference:

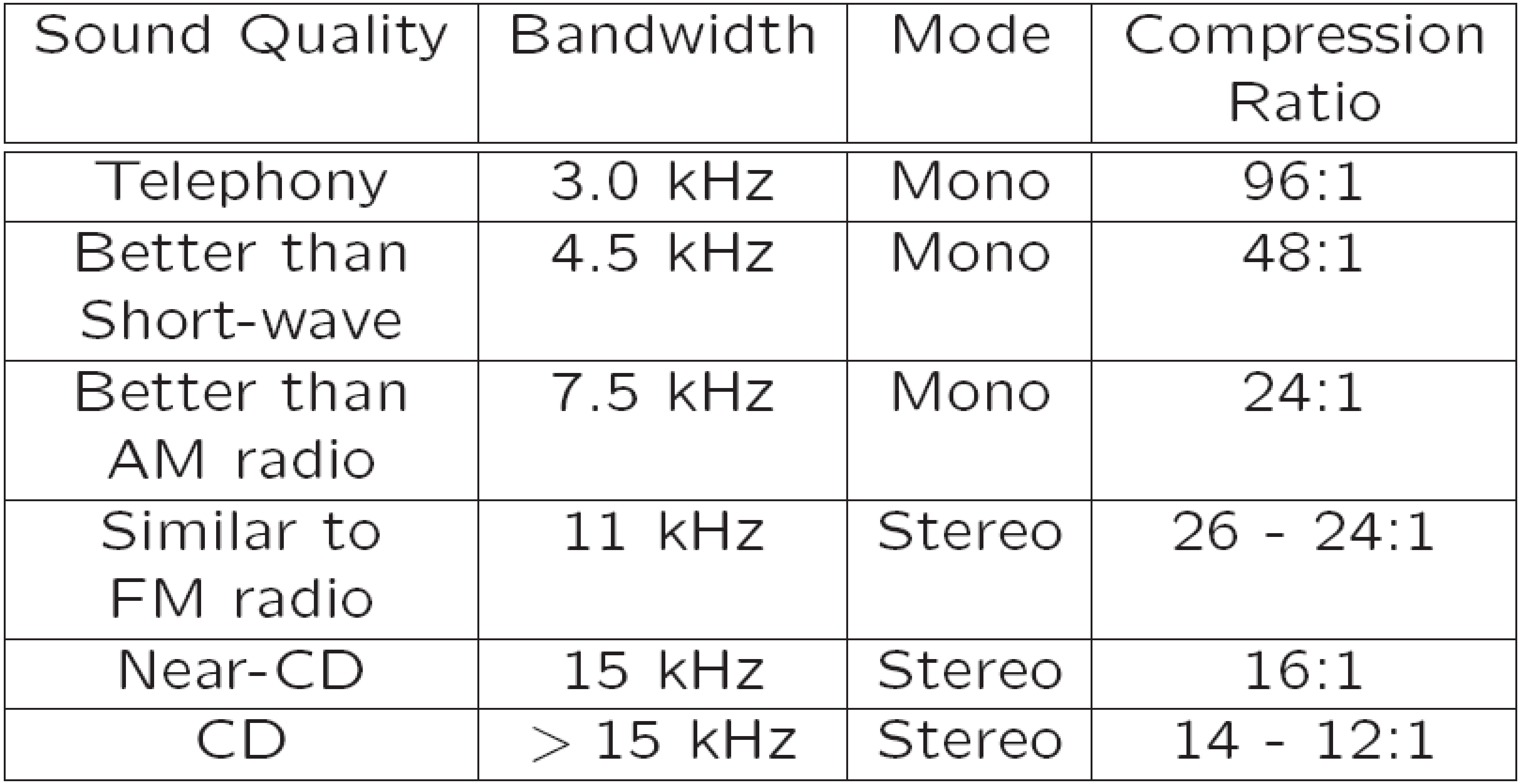
- Employs a similar filter bank to that used in Layer 2, except that now perceptual critical bands are more closely adhere to by using a set of filters with non-equal frequencies
- Takes into account stereo redundancy
- Uses Modified Discrete Cosine Transform (MDCT) — addresses problems that the DCT has at boundaries of the window used by overlapping frames by 50%:

$$F(u) = 2\sum_{i=0}^{N-1} f(i)\cos\left[\frac{2\pi}{N}\left(i + \frac{N/2+1}{2}\right)(u+1/2)\right], u = 0,.., N/2-1$$



MPEG-Audio Layer 3 Coding

MP3 compression performance



Sound Quality	Bandwidth	Mode	Compression Ratio
Telephony	3.0 kHz	Mono	96:1
Better than Short-wave	4.5 kHz	Mono	48:1
Better than AM radio	7.5 kHz	Mono	24:1
Similar to FM radio	11 kHz	Stereo	26 - 24:1
Near-CD	15 kHz	Stereo	16:1
CD	> 15 kHz	Stereo	14 - 12:1

MPEG-4 Audio

- Integrates several different audio components into one standard: speech compression, perceptually based coders, text-to-speech, and MIDI
- *MPEG-4 AAC* (*Advanced Audio Coding*), is similar to the MPEG-2 AAC standard, with some minor changes
- **Perceptual Coders**
  - Incorporate a *Perceptual Noise Substitution* module
  - Include a *Bit-Sliced Arithmetic Coding* (BSAC) module
  - Also include a second perceptual audio coder, a vector-quantization method entitled TwinVQ

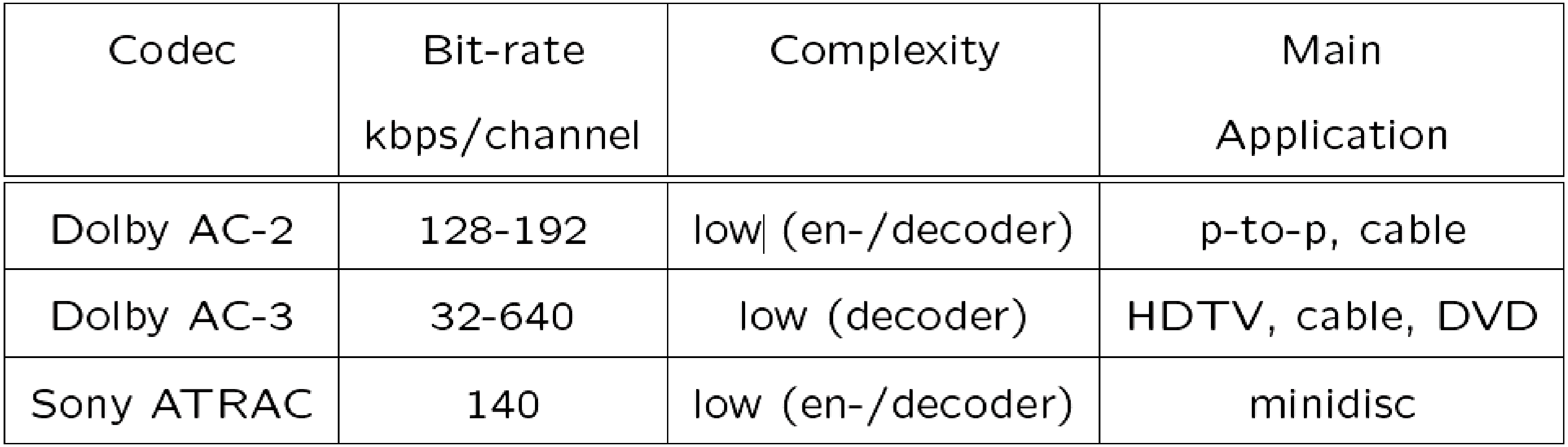
MPEG-4 Audio (Cont’d)

- **Structured Coders**
  - Takes “Synthetic/Natural Hybrid Coding” (SNHC) in order to have very low bit-rate delivery an option
  - **Objective**: integrate both “natural” multimedia sequences, both video and audio, with those arising synthetically – “structured” audio
  - Takes a “toolbox” approach and allows specification of many such models.
  - E.g., *Text-To-Speech* (TTS) is an ultra-low bit-rate method, and actually works, provided one need not care what the speaker actually sounds like

Other Commercial Audio Codecs

The table summarizes the target bit-rate range and main features of other modern general audio codecs

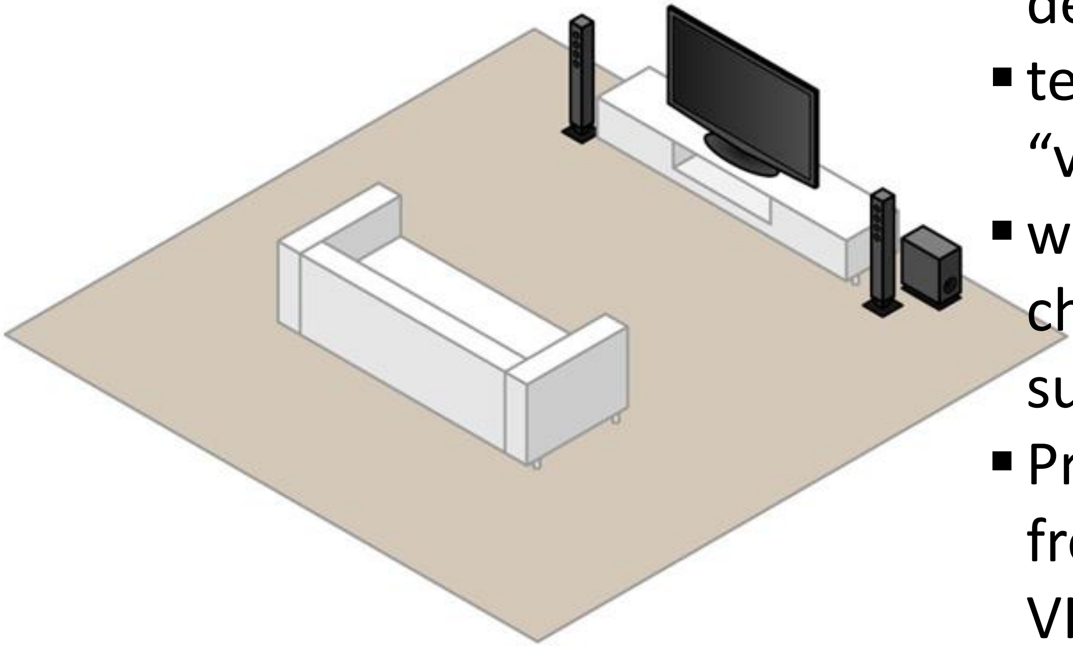
Comparison of audio coding systems



Codec	Bit-rate	Complexity	Main
	kbps/channel		Application
Dolby AC-2	128-192	low (en-/decoder)	p-to-p, cable
Dolby AC-3	32-640	low (decoder)	HDTV, cable, DVD
Sony ATRAC	140	low (en-/decoder)	minidisc

Surround sound standard: Dolby Digital

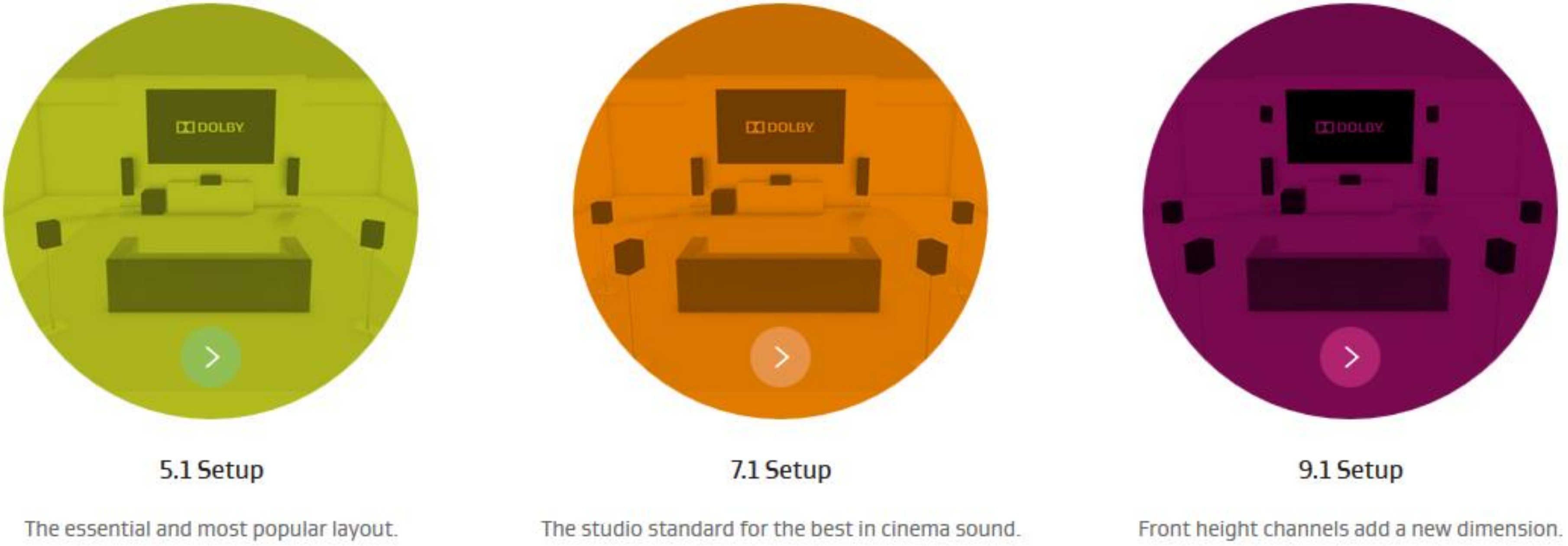
Dolby Pro Logic



- the oldest of the audio decoding technologies
- technically it's classified as "virtual surround"
- was designed to utilize two channels and sometimes a subwoofer
- Pro Logic decoded the audio from a record, cassette tape or VHS movie and played it back over the two channels with the only variation in sound being left and right

Dolby Digital

Dolby Digital is a lossy format. As such, the audio is encoded and compressed by the studio in order to fit on the disc, or to conserve bandwidth when broadcast digitally through cable or satellite.



5.1 Setup

The essential and most popular layout.

7.1 Setup

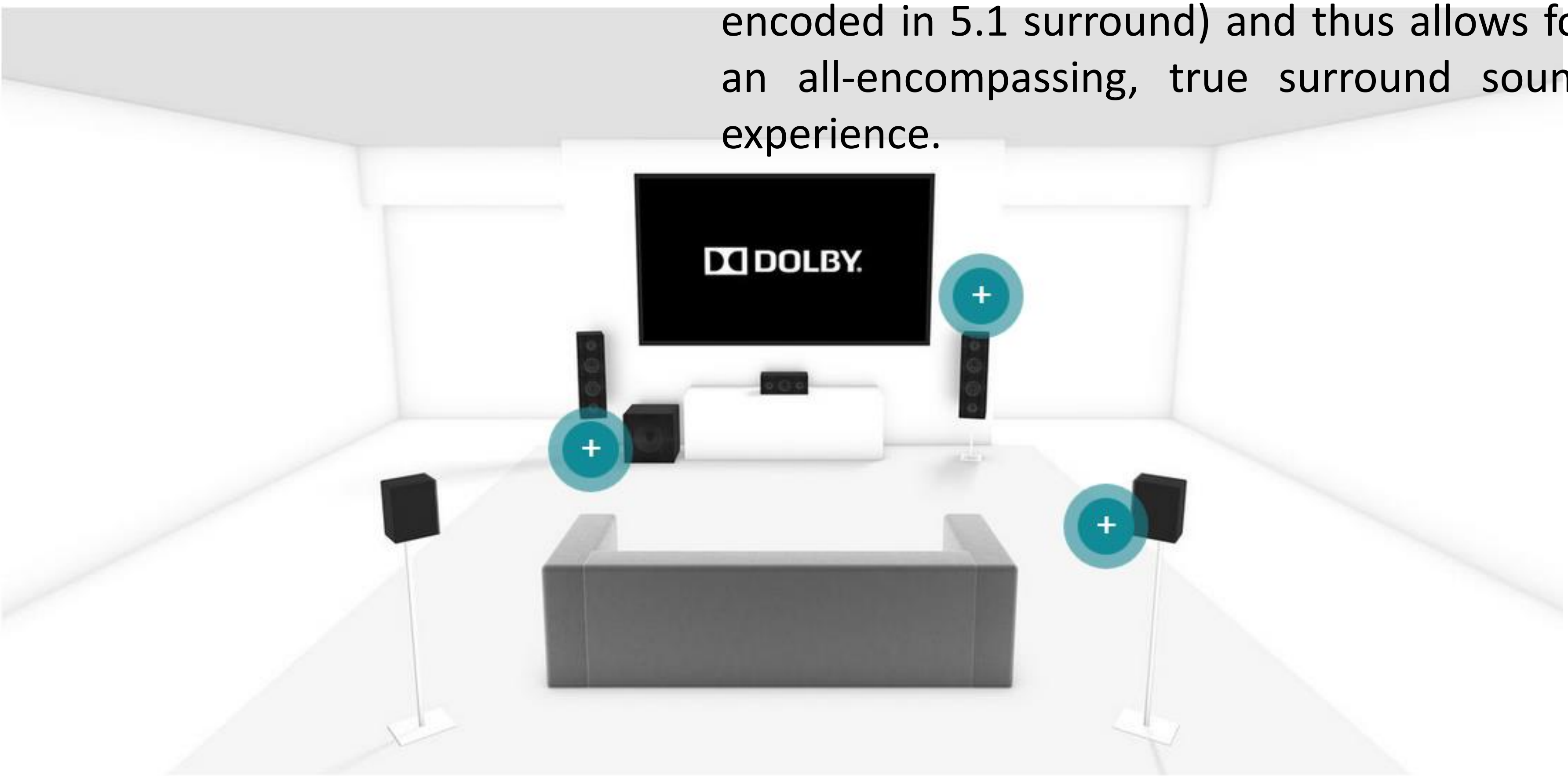
The studio standard for the best in cinema sound.

9.1 Setup

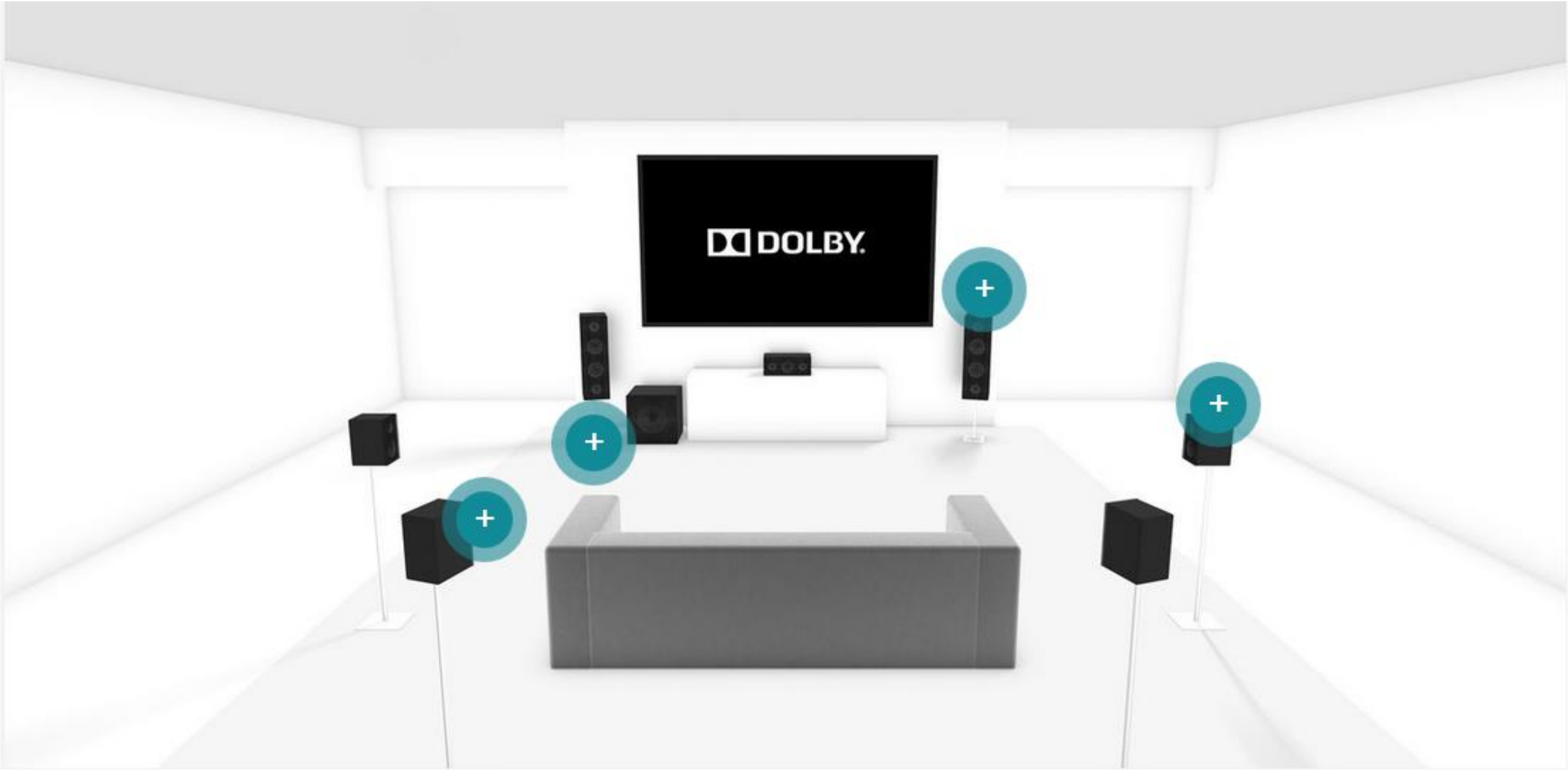
Front height channels add a new dimension.

Dolby Digital 5.1 Setup

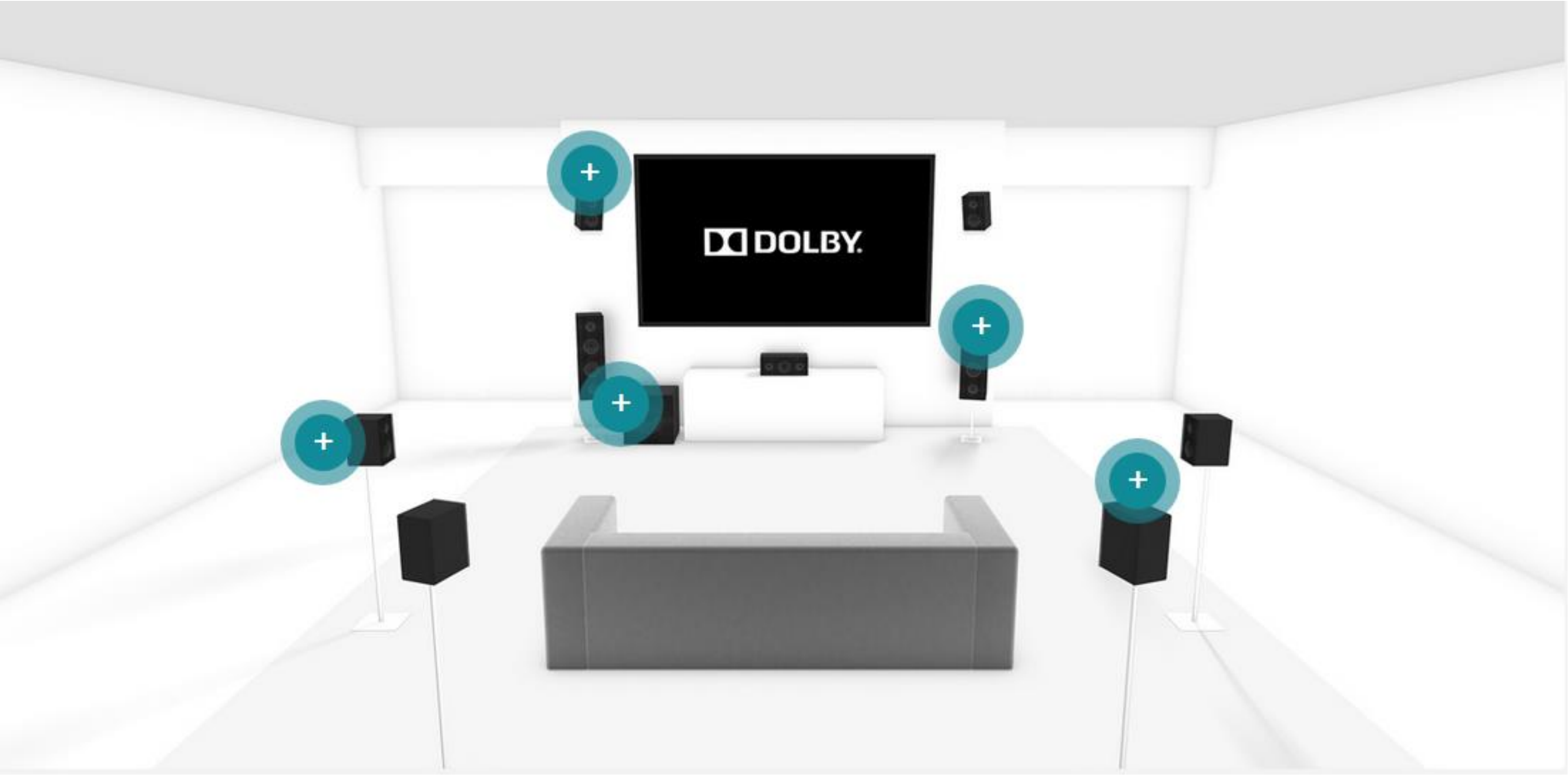
The technology uses a built-in decoder to decode the signal and separate individual sounds in to one of 5 (front left and right, rear left and right, center channel) independent speakers. Each speaker operates independently of the others (if the source is encoded in 5.1 surround) and thus allows for an all-encompassing, true surround sound experience.



Dolby Digital 7.1 Setup



Dolby Digital 9.1 Setup



Dolby Atmos 5.1.2 setup