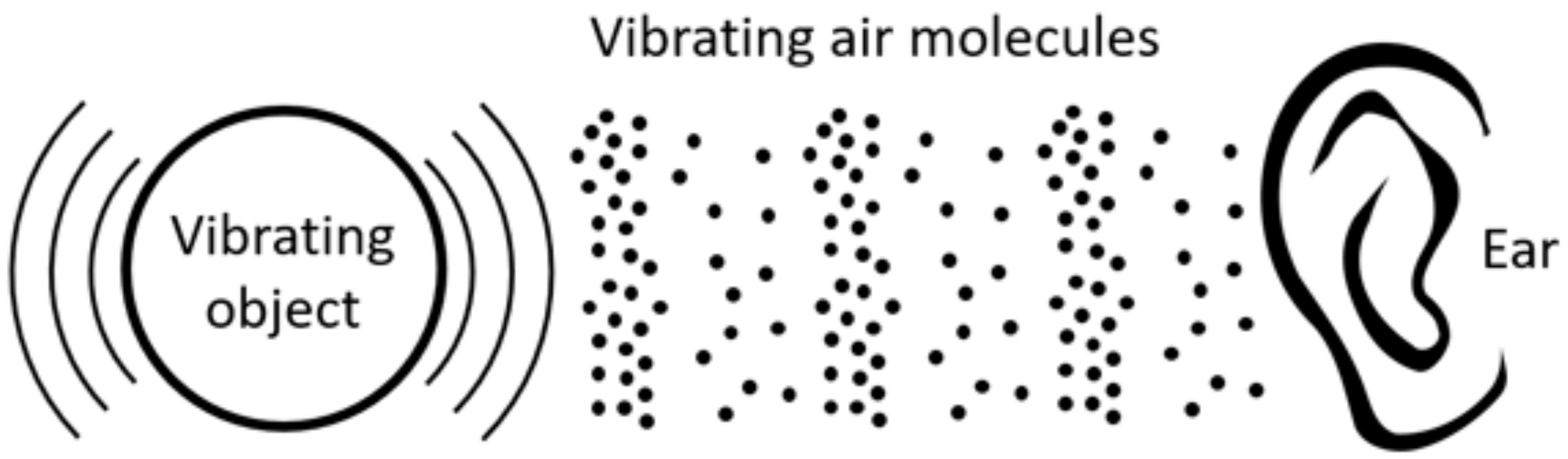
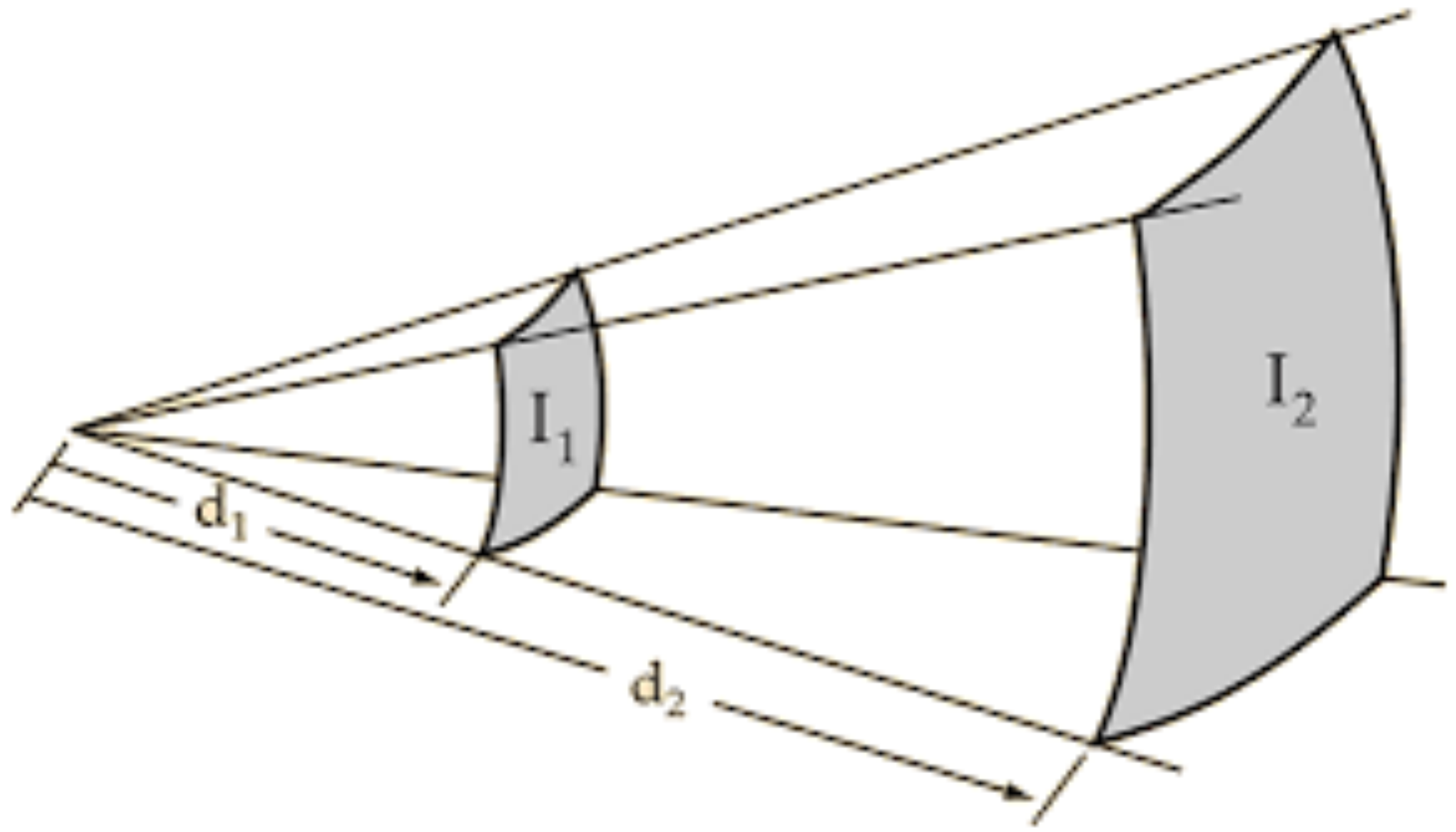
TECH 350: DSP

Class II: Digital Electronic Music Concepts Overview (Part II)

Sound, review (and revisit)





Compression vs. Rarefaction

Wavefront - if sound is omnidirectional, it propagates spherically => inverse-square law

Can measure intensity via SPL or dB (which has a reference for quietness)

Is log.: doubling of sound pressure: +6dB change, for example

Describing Sound Waves, review

f = frequency (measured in Hertz)

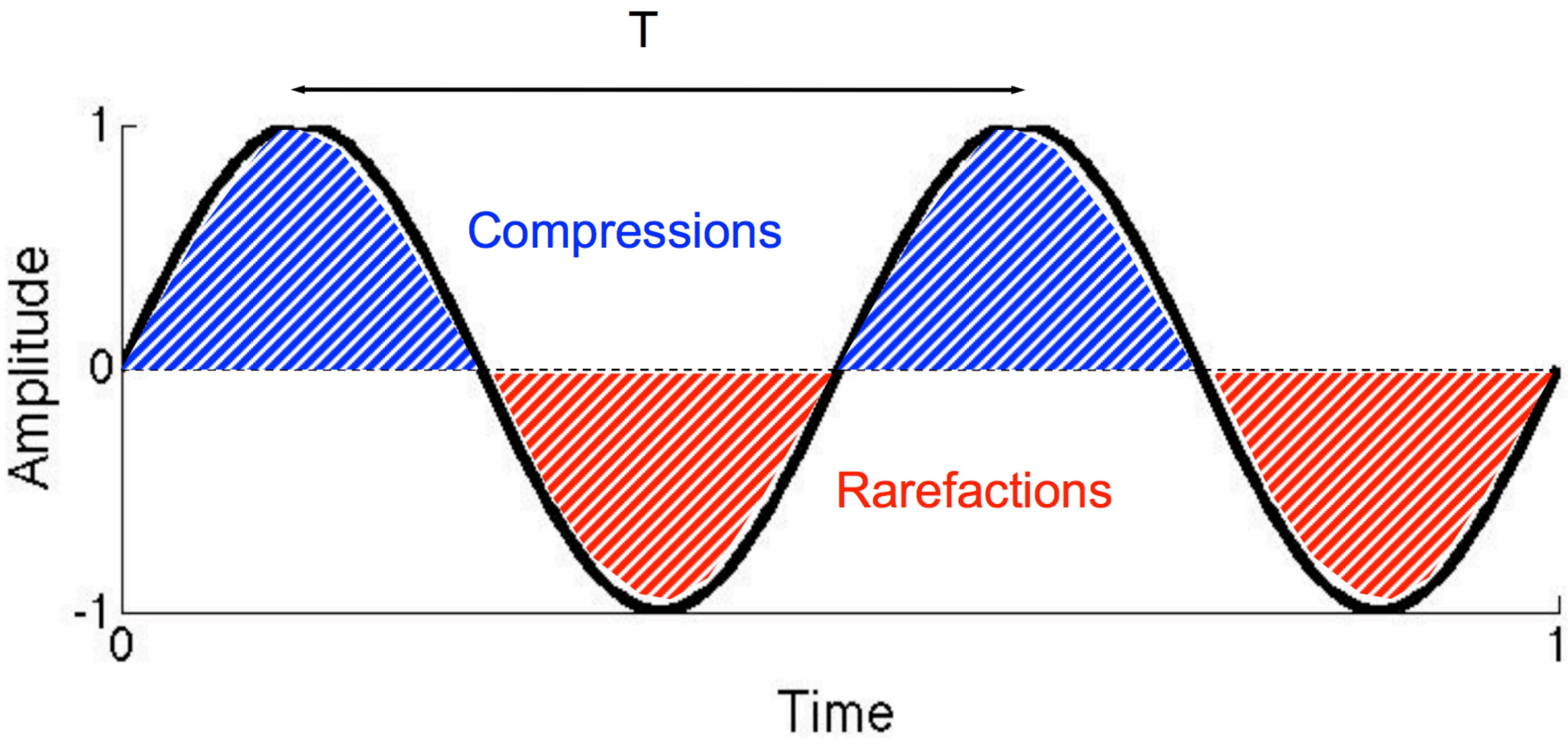
A = amplitude (measured as a raw digital value (0->1. or with SPL or dB)

λ, lambda = wavelength (measured in feet, etc.)

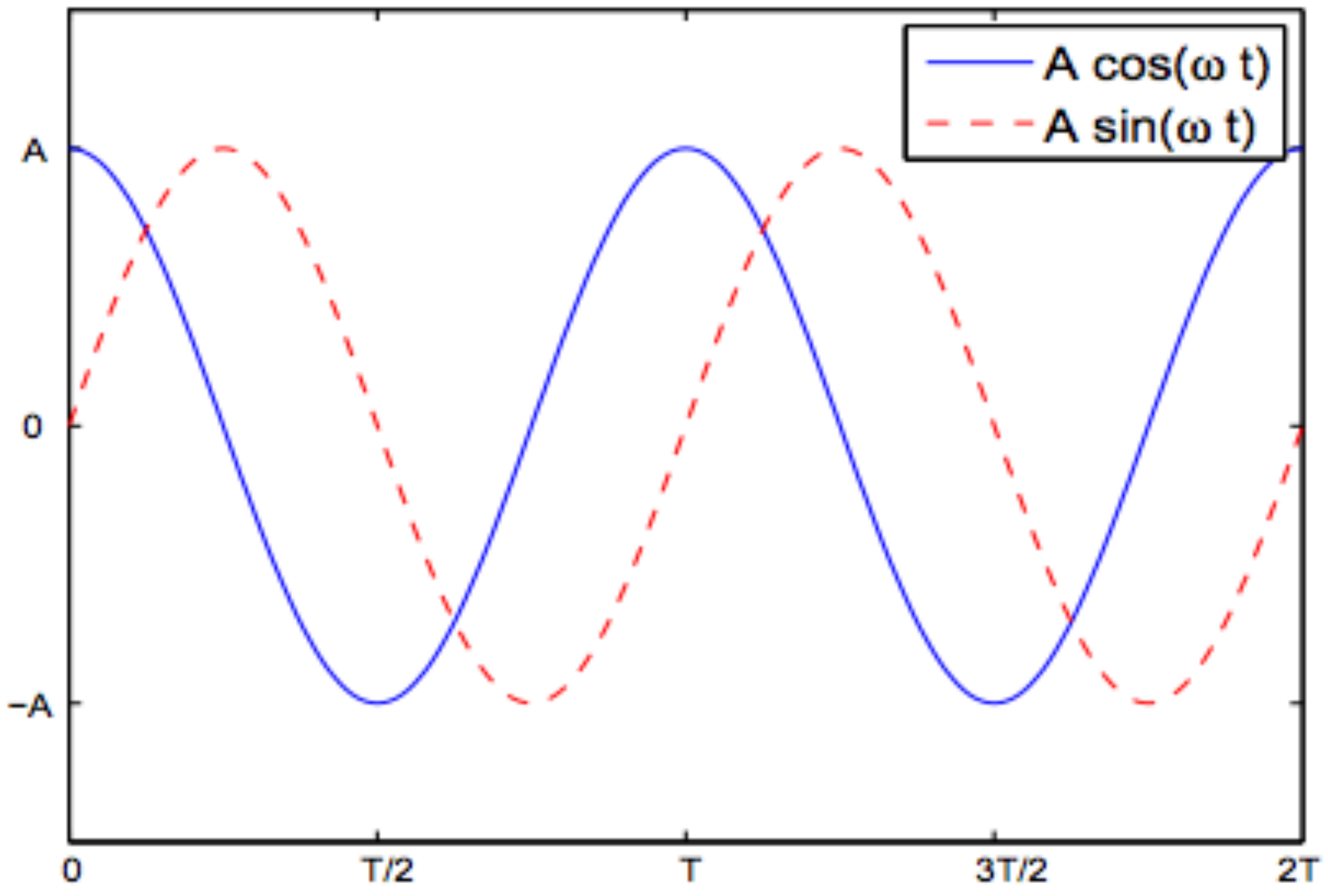
T = period = λ

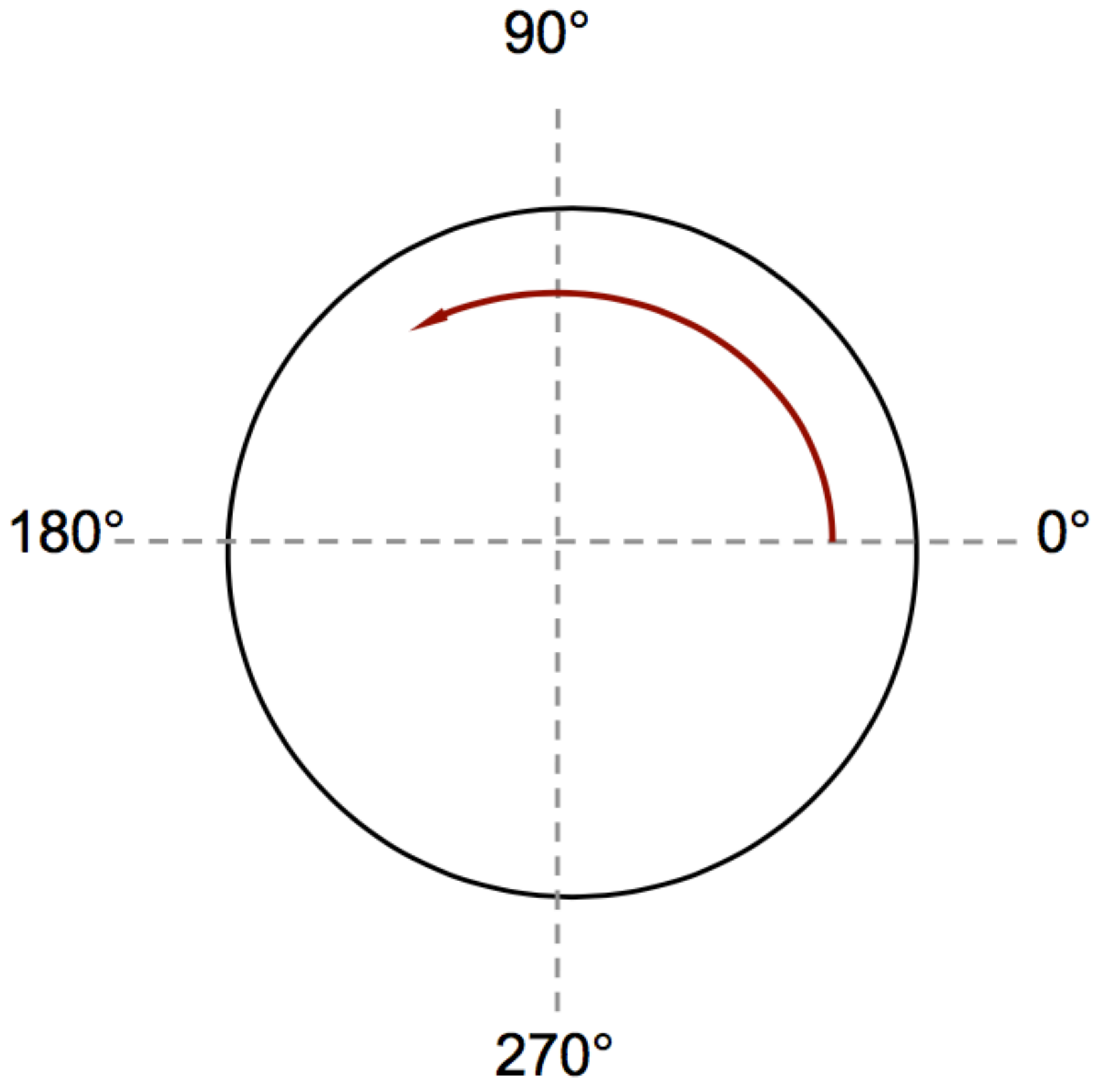
1/T = f

$x(t) = A \cdot \sin(2\pi ft + \theta)$



Describing Sound Waves, review





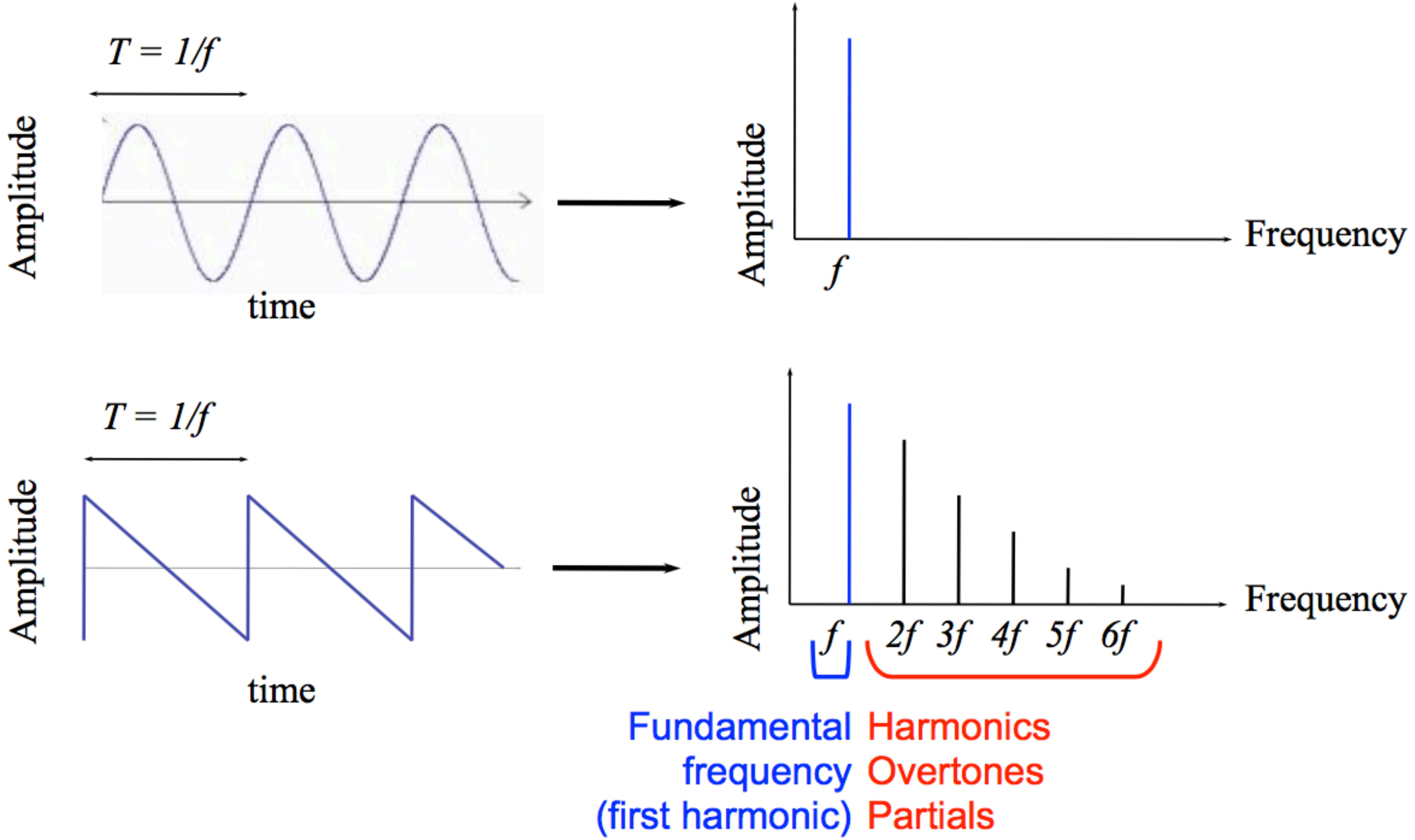
$$x_c(t) = A\cos(2\pi f t)$$

- amplitude A
- period $T = 1/f$
- phase: 0

$$x_s(t) = A\sin(2\pi f t)$$
$$= A\cos(2\pi f t - \pi/2)$$

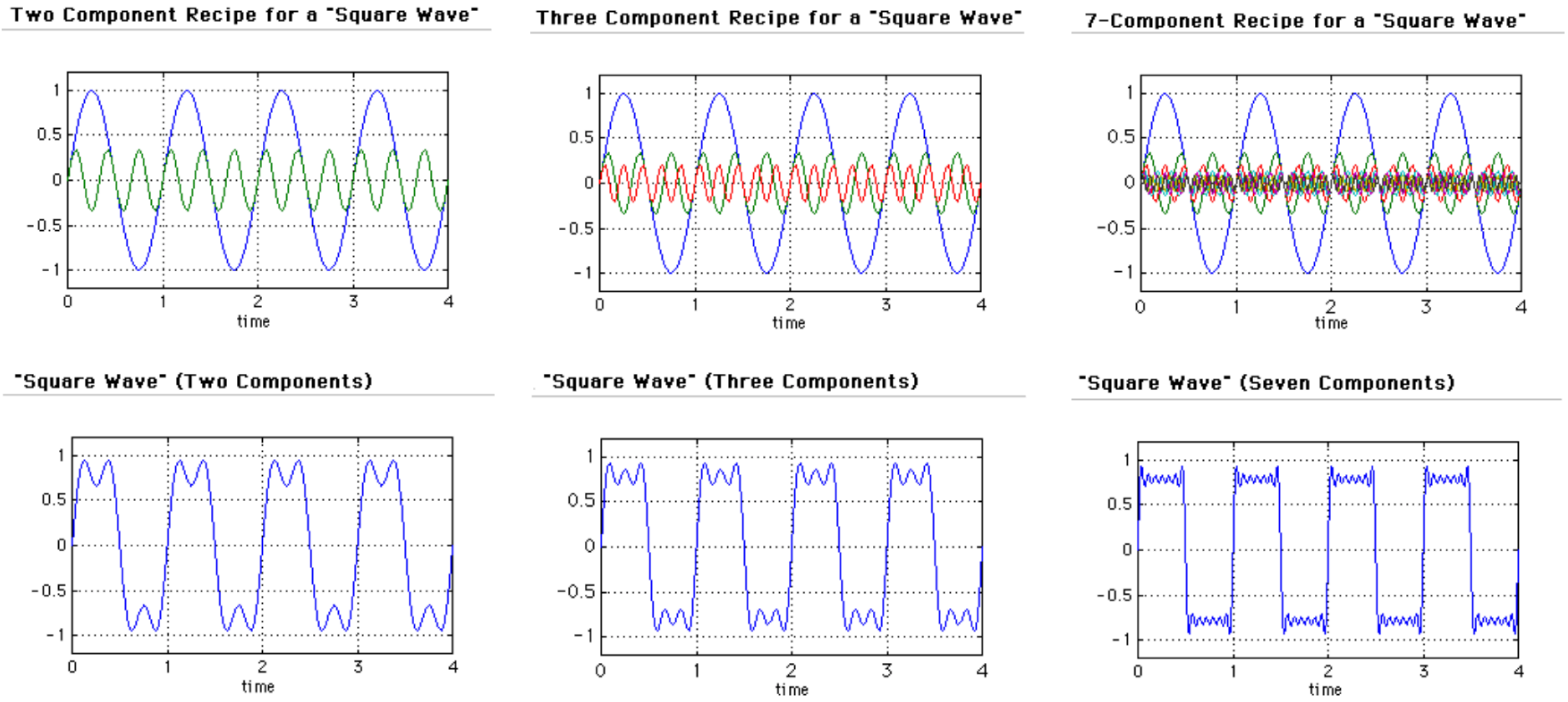
-amplitude A
-period $T = 1/f$
-phase: $-\pi/2$

Fundamental Freq. + Harmonics



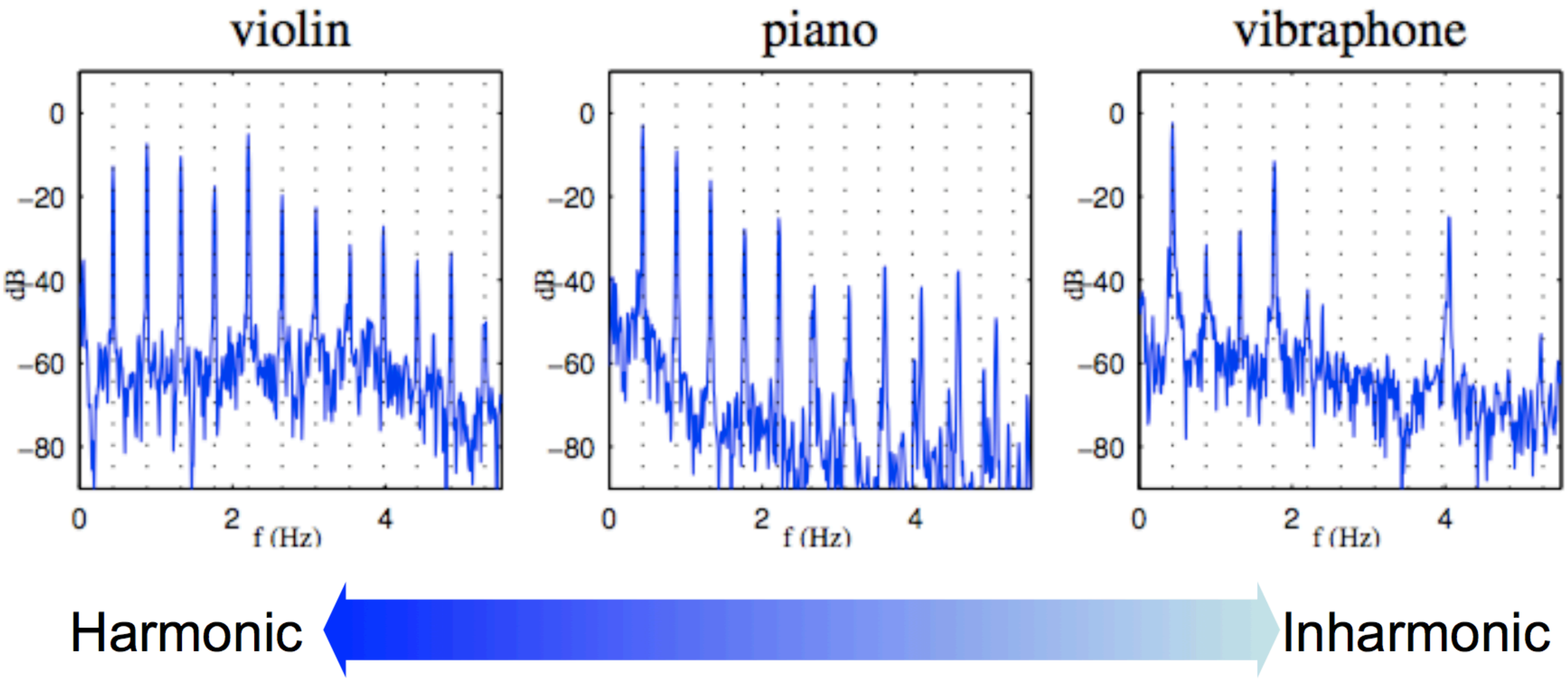
Sound Typology (3)

- Example: Square wave - only odd harmonics (even are missing).
- Amplitude of the nth harmonic = 1/n



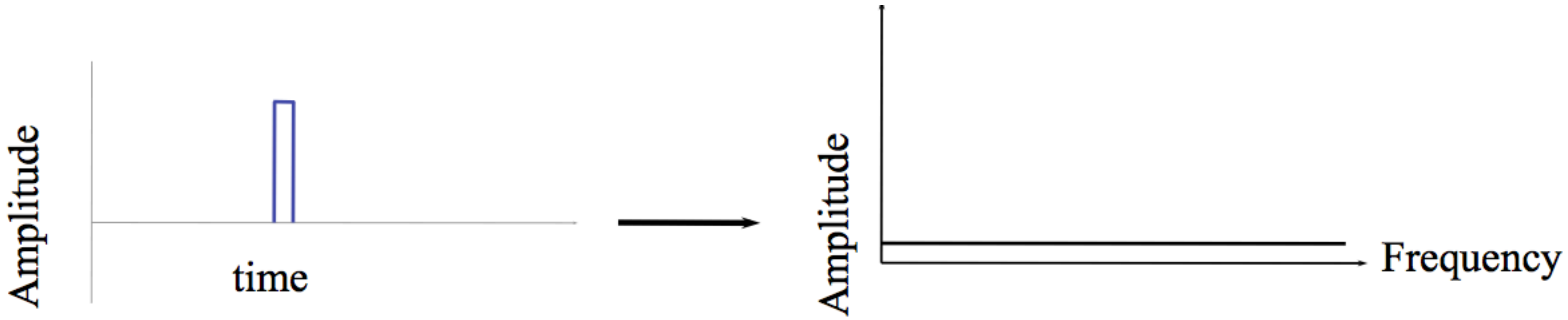
Sound Typology (4)

- Most natural pitched sounds also present overtones which are not integer multiples of the fundamental.
- These are known as inharmonic partials

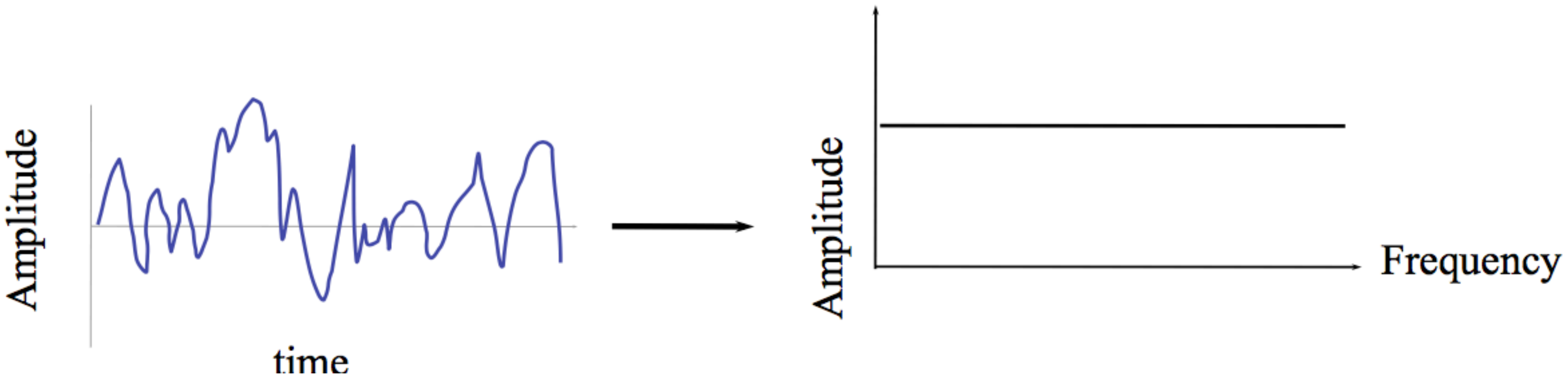


Sound Typology (5)

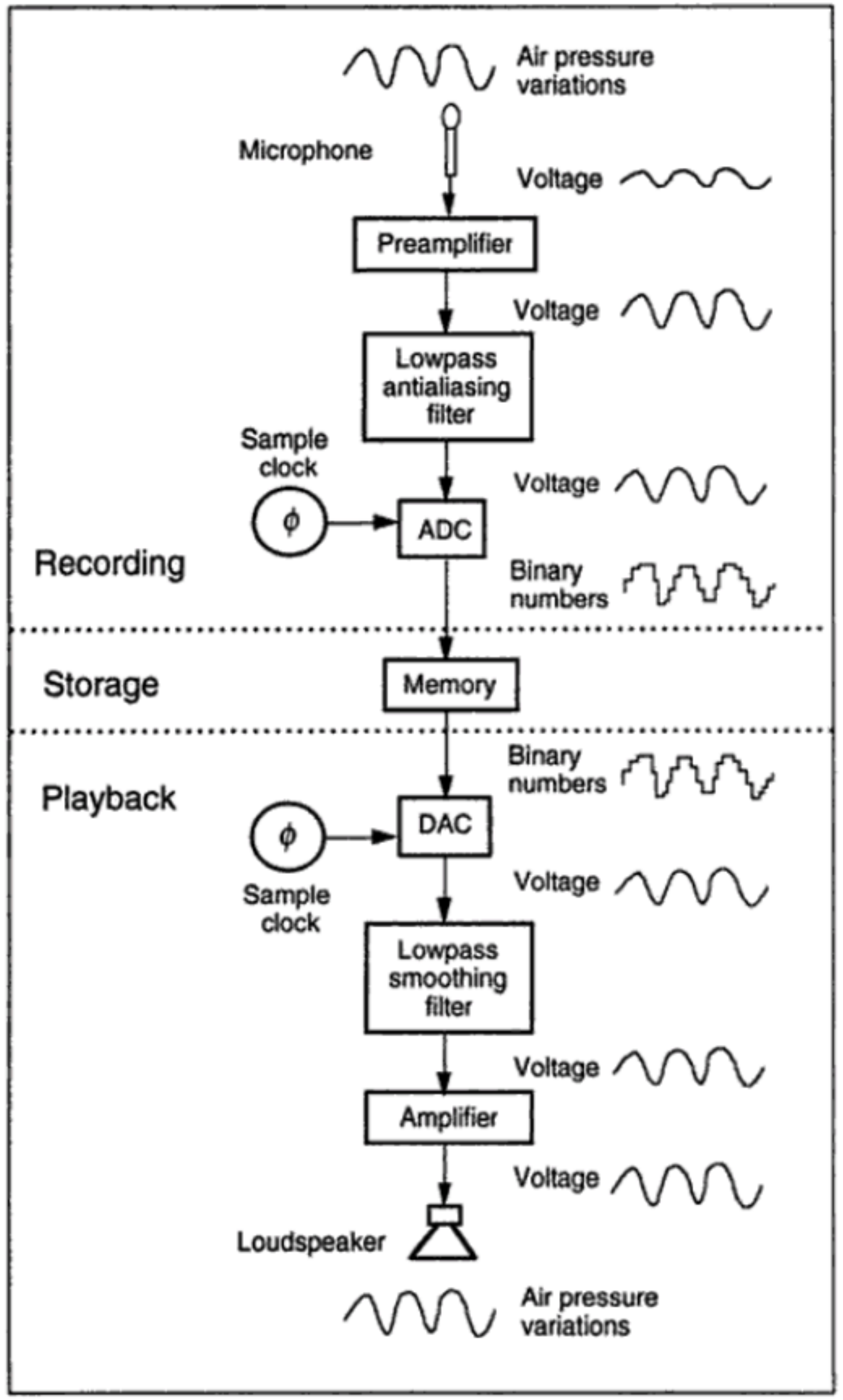
- Non-periodic sounds have no pitch and tend to have continuous spectra, e.g. a short pulse (narrow in time, wide in frequency)



- The most complex sound is white noise (completely random)



ADC and DAC


Air pressure variations
Microphone
Voltage
Preamplifier
Voltage
Lowpass antialiasing filter
Sample clock
Voltage
ADC
Recording
Binary numbers
Storage
Memory
Binary numbers
Playback
DAC
Voltage
Sample clock
Lowpass smoothing filter
Voltage
Amplifier
Voltage
Loudspeaker
Air pressure variations

Analog-to-Digital Conversion

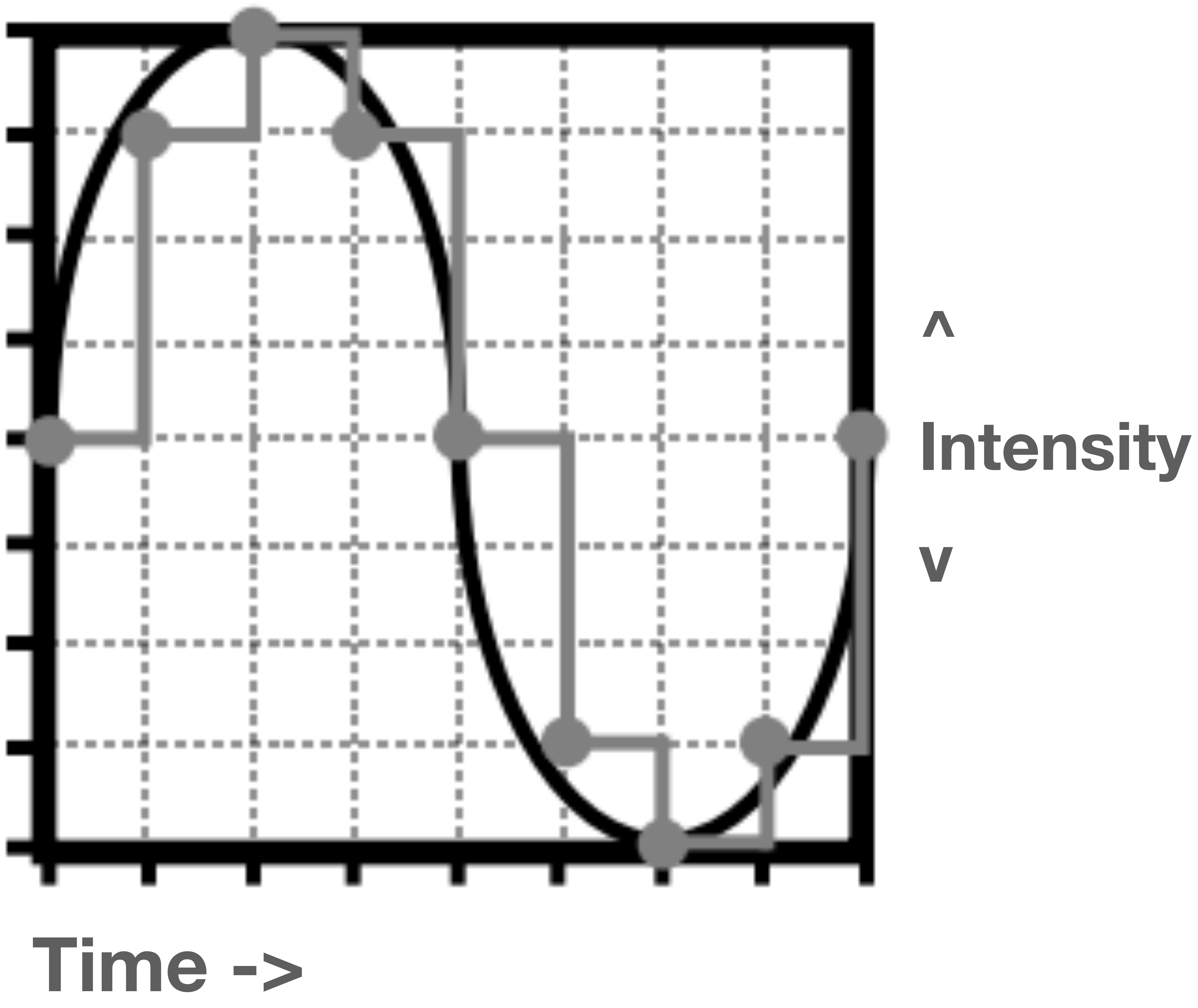
Parameters of ADC:

- **Sampling Rate (f_s) =**

rate at which analog signal is captured (sampling) (in Hertz)

- **Bit Depth =**

number of values for each digital sample (quantization) (in bits)



Binary Digits

Table 1.1 Binary numbers and their decimal equivalents

Binary	Decimal
0	0
1	1
10	2
11	3
100	4
1000	8
10000	16
100000	32
1111111111111111	65535

1. 101 = ?
2. 1011 = ?
3. 111111 = ?

Values of places are 2^x:

… 64 32 16 8 4 2 1

Limitations/Issues with Sampling

Distortion caused by sampling, AKA ALIASING (or foldover)

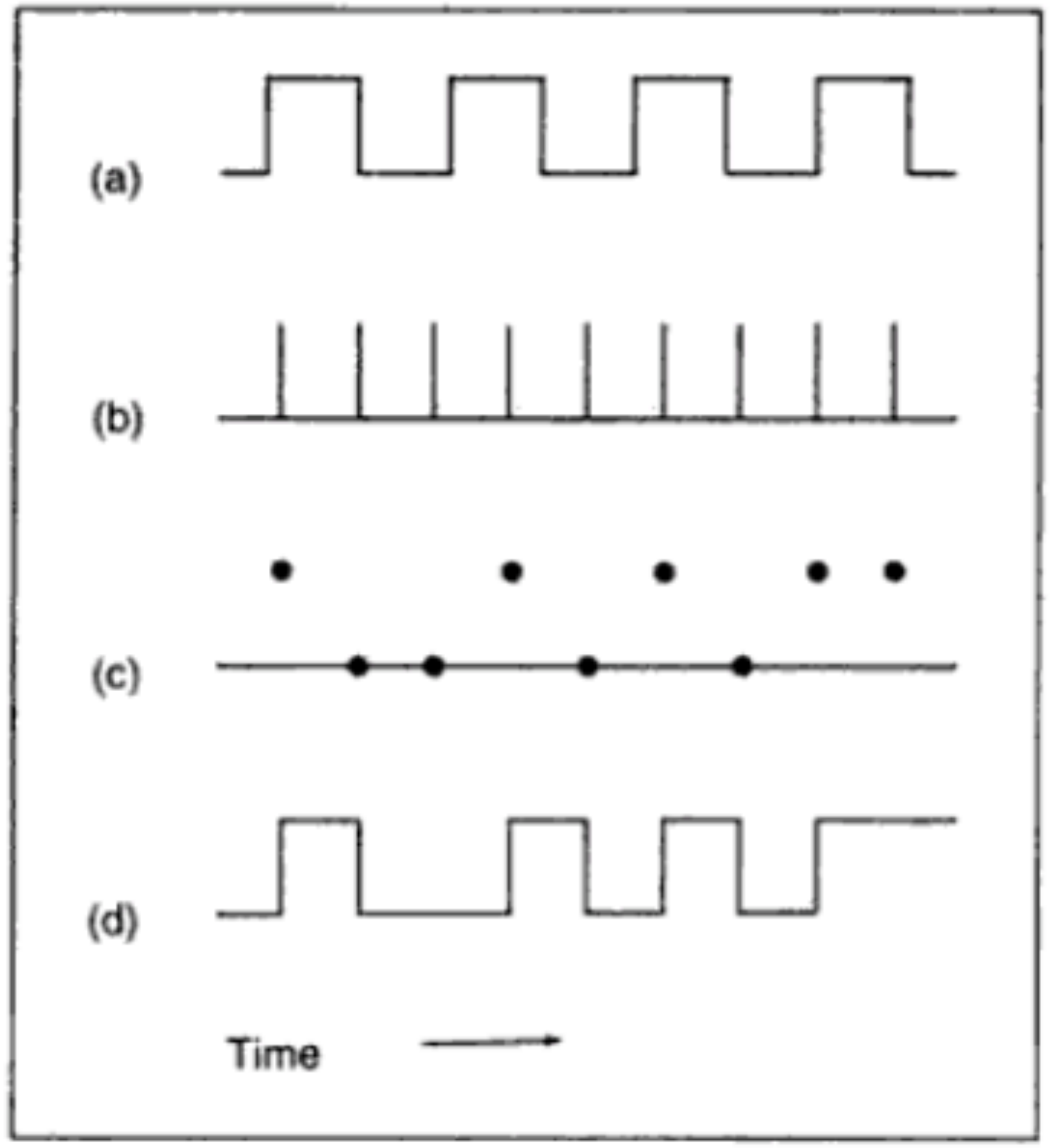


Figure 1.15 Problems in sampling. (*a*) Waveform to be recorded. (*b*) The sampling pulses; whenever a sampling pulse occurs, one sample is taken. (*c*) The waveform as sampled and stored in memory. (*d*) When the waveform from (*c*) is sent to the the DAC, the output might appear as shown here (after Mathews 1969).

How can we rectify (or at least describe) this phenomenon?

Sampling (Nyquist) Theorem

- Can describe the resultant frequency of aliasing via the following (rough) formula,

iff input freq. > half the sampling rate && < sampling rate:

resultant frequency = sampling frequency (f_s) - input frequency

For example, if f_s = 1000Hz and the frequency of our input is at 800Hz:

1000 - 800 = 200, so

resultant frequency is 200Hz (!)

- Nyquist theorem = In order to be able to reconstruct a signal, the sampling frequency must be at least twice the frequency of the signal being sampled
- If you want to represent frequencies up to X Hz, you need f_s = 2X Hz

Ideal Sampling Frequency (for audio)

- What sampling rate should we use for musical applications?
- This is an on-going debate. Benefits of a higher sampling rate? Drawbacks?

- AES Standards:
- Why 44.1kHz? Why 48kHz? Why higher (we can't hear up there, can we?)
- For 44.1kHz and 48kHz answer lies primarily within video standard considerations, actually…
- 44.1kHz = $2^2 \cdot 3^2 \cdot 5^2 \cdot 7^2$, meaning it has a ton of integer factors

- >2 * 20kHz is great, as it allows us to have frequency headroom to work with, and subharmonics (and interactions of phase, etc.) up in that range are within our audible range

Anti-Aliasing Filters + Phase Correction

- How to fix aliasing? Add a low-pass filter set at a special cutoff frequency before we digitize the signal.
- Similarly, before we go from digital data -> analog signal, we add a lowpass filter that smooths the transitions between samples, ideally recreating the original signal as accurately as possible

- These simple filter works great, right? Takes away those pesky higher frequencies before we sample the part of the analog signal that we want or send it out into the world? WRONG

- Early ADC used *brickwall filters* - filters that can cause significant time-delays (phase distortion) in certain frequencies. Produced a harsh sound.
- No analog filter (as we will see) can be both extremely steep and *phase linear*
- *How can we solve this problem?*

A Brief Diversion Into Filters

Equalization/Filtering

Def.: *Changing the amplitudes of particular portions of the frequency range*

High-pass filter – pass frequencies ***above*** a cutoff frequency, attenuate others
(same as **Low-cut filter**)

Low-pass filter – pass frequencies ***below*** a cutoff frequency, attenuate others
(same as **High-cut filter**)

Band-pass filter – pass frequencies ***around*** a center frequency, attenuate others

Notch filter – inverse of Band-pass filter: attenuate frequencies ***around*** a center frequency, pass others

High-shelf filter – boost or attenuate frequencies ***above*** a center frequency

Low-shelf filter – boost or attenuate frequencies ***below*** a center frequency

Peak (also called “Band” or “Bell”) **filter** – boost or attenuate frequencies ***around*** a center frequency

Params:
Slope – the intensity of attenuation across frequencies
Resonance or Q (‘quality factor’) – the sharpness or focus of the filter.

Params.: frequency, gain, slope, ‘Q’/resonance
Ex. Plug-in: *ReaEQ*

Phase Correction (2)

- Can trade off antialiasing properties for less phase distortion (less steep filter = less phase distortion, but can have foldover for high frequency sounds (!))
- Additionally, could use a time correction filter before the ADC to skew the phase relationship in the incoming signal, ultimately preserving (in the digitized version) the original phase relationships

- Current solution: Oversample, then taking advantage of linear phase digital filters!
- When doing ADC, sample at a higher bandwidth, then apply a linear phase filter -> store on disk. When doing DAC, upsample before filtering (smoothing)
- Unfortunately, this does impact the signal-to-noise ratio (SNR) of the system as a whole
- Will talk about this more in a bit…

Back to Quantization

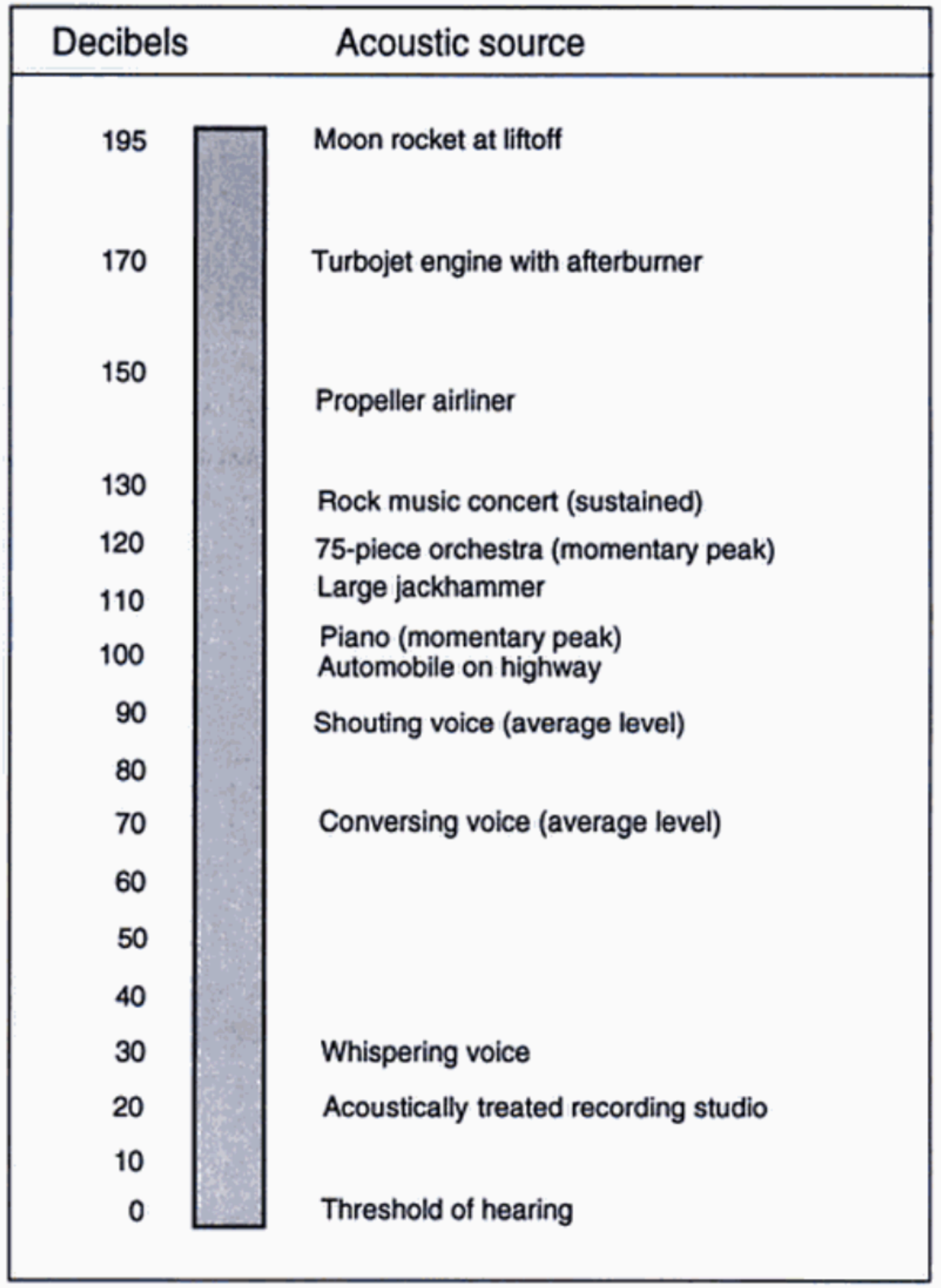


Figure 1.22 Typical acoustic power levels for various acoustic sources. All figures are relative to 0 dB = 10^{-12} watts per square meter.

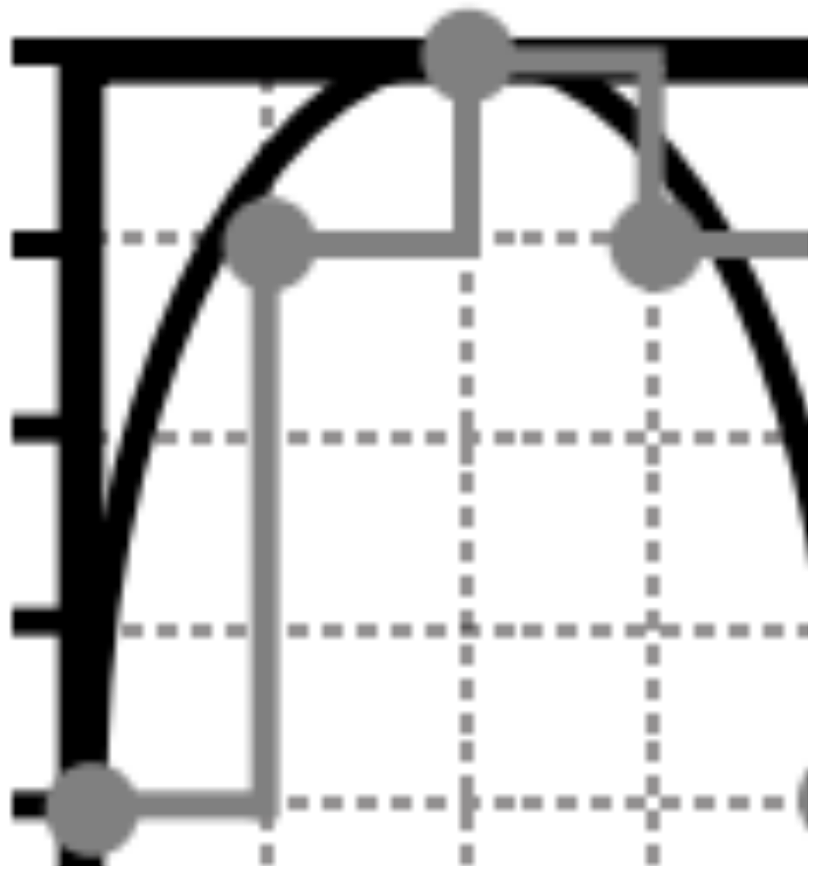
- *dynamic range* = loudest - quietest producible sound
- In a digital system, measured in dBFS (digital, non-referenced decibel level)

- 16-bit (with no dither) = 90dBFS dynamic range
- 16-bit with noise-shaping dither = 120 dBFS or more
- 24-bit - 138 dBFS

- Formula: Level in dBFS = 20 * LOG_{10} (level / max level)

- Compare to early 78s, which were 40 dB

Limitations/Issues with Quantization



- Difference between analog signal and it quantized, digital version = *quantization error*
- Is not random, but rather is deterministic, and can be audible (!), especially when input is at low levels (activating on and off that lowest bit of the quantized value)
- In a linear PCM system (as we most often use), quantization noise is a function of bit depth. The higher the bit depth, the less quantization error, and less resulting quantization noise.
- How can we combat quantization noise? By *dithering* the signal and reducing harmonic distortion

Dither + Noise-Shaping

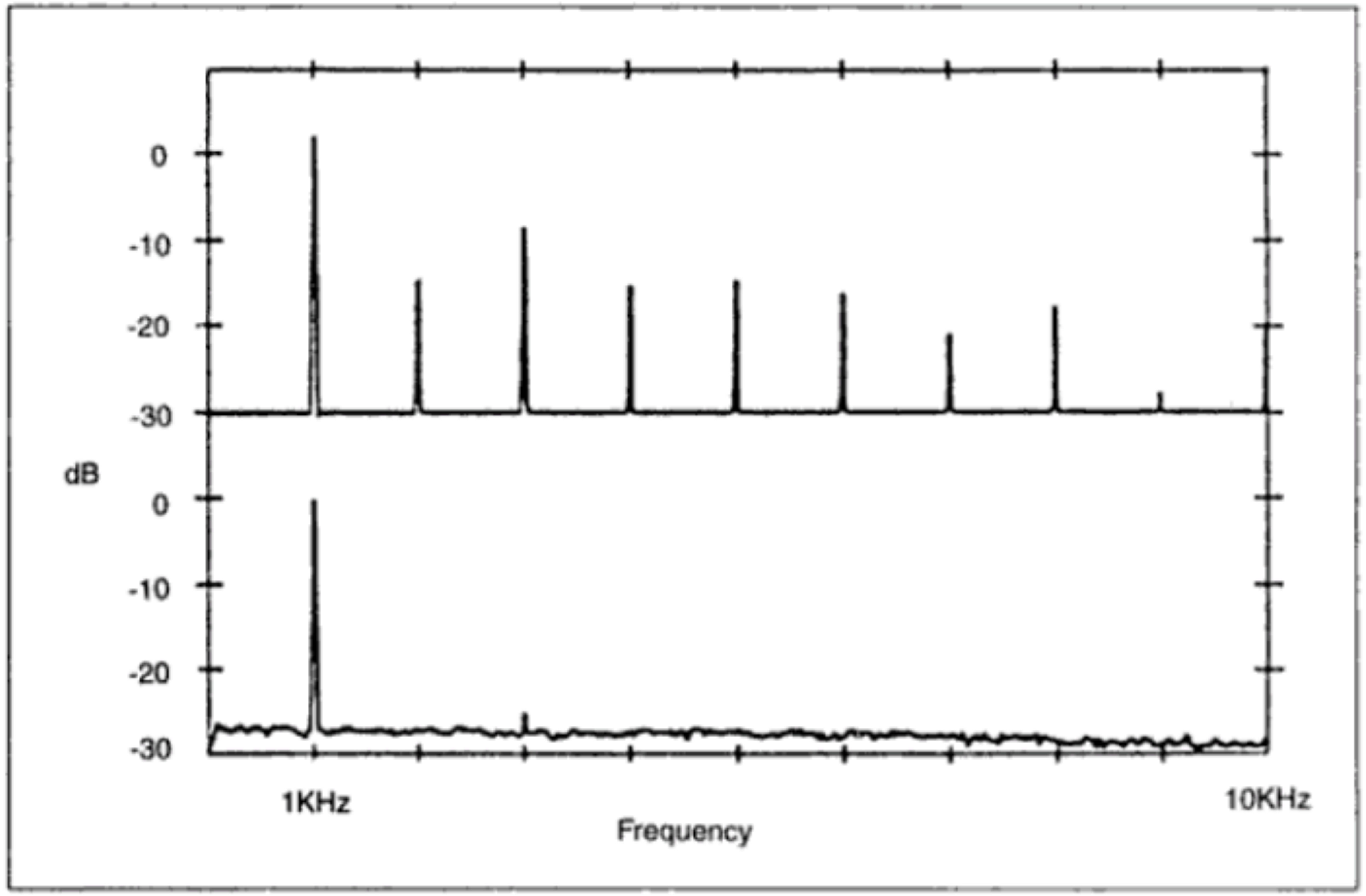


Figure 1.21 Dither reduces harmonic distortion in a digital system. The top part of the figure shows the spectrum of 1 KHz sine wave with an amplitude of 1/2 bit. Note the harmonics produced by the action of the ADC. The lower part shows the spectrum of the same signal after dithering of about 1 bit in amplitude is applied before conversion. Only a small amount of third harmonic noise remains, along with wideband noise. The ear can resolve the sine wave below the noise floor.